

IN-STK5000 Kompendium

Espen H. Kristensen (espenhk)

September 24, 2018

Contents

1	Course notes: “Machine Learning in science and society - From automated science to beneficial artificial intelligence”, Christos Dimitrakakis. (2018)	3
2	“The Practice of Reproducible Research: Case Studies and Lessons from the Data-Intensive Sciences.” Kitzes, J., Turek, D., & Deniz, F. (2018)	131
3	“The Algorithmic Foundations of Differential Privacy”, Dwork, C., Roth, A (2014)	450
4	“Privacy, Big Data, and the Public Good: Frameworks for Engagement”, Lane, J., Stodden, V., Bender, S., Nissenbaum, H. (2014)	731
5	“Decision Making Under Uncertainty and Reinforcement Learning”, Dmimitrakakis, C., Ortner, R. (2018)	1049

Machine learning in science and society

From automated science to beneficial artificial intelligence

Christos Dimitrakakis

September 21, 2018

Contents

1	Introduction	5
1.1	Introduction to machine learning	6
1.1.1	Data analysis, learning and planning	6
1.1.2	Experiment design	10
1.1.3	Bayesian inference.	11
1.1.4	Course overview	14
2	Simple decision problems	17
2.1	Nearest neighbours	18
2.2	Reproducibility	23
2.2.1	The human as an algorithm	26
2.2.2	Algorithmic sensitivity	28
2.3	Beliefs and probabilities	33
2.3.1	Probability and Bayesian inference	36
2.4	Hierarchies of decision making problems	41
2.4.1	Simple decision problems	41
2.4.2	Decision rules	43
2.4.3	Statistical testing	44
2.5	Formalising Classification problems	52
2.6	Classification with stochastic gradient descent	55
2.6.1	Neural network models	56
2.7	Naive Bayes classifiers	60
3	Privacy	63
3.1	Database access models	64
3.2	Privacy in databases	66
3.3	k -anonymity	67
3.4	Differential privacy	68
3.4.1	Other differentially private mechanisms	74
3.4.2	Utility of queries	76
3.4.3	Privacy and reproducibility	77
4	Fairness	81
4.1	Fairness in machine learning	82
4.2	Graphical models	85
4.3	Concepts of fairness	87
4.3.1	Fairness as independence	88
4.3.2	Fairness as meritocracy.	89

4.3.3	Fairness as similarity.	89
4.3.4	Bayesian fairness	90
4.4	Project: Credit risk for mortgages	91
4.4.1	Deadline 1: September 14	91
4.4.2	Deadline 2: September 28	91
5	Recommendation systems	93
5.1	Recommendation systems	94
5.2	Clustering	98
5.3	Social networks	100
5.4	Sequential structures	101
6	Bandit problems	103
6.1	Introduction	105
6.2	Bandit problems	105
6.2.1	An example: Bernoulli bandits	107
6.2.2	Decision-theoretic bandit process	108
6.3	Experiment design	110
7	Markov decision processes	111
7.1	Markov decision processes and reinforcement learning	112
7.1.1	Value functions	114
7.2	Finite horizon, undiscounted problems	115
7.2.1	Policy evaluation	115
7.2.2	Monte-Carlo policy evaluation	116
7.2.3	Backwards induction policy evaluation	117
7.2.4	Backwards induction policy optimisation	118
7.3	Infinite-horizon	119
7.3.1	Examples	119
7.3.2	MDP Algorithms	122
8	Safety	125

Chapter 1

Introduction

1.1 Introduction to machine learning

What are the central problems in machine learning?

Problems in machine learning are similar to problems in science. Scientists must plan experiments intelligently and collect data. They must be able to use the data to verify a different hypothesis. More generally, they must be able to make decisions under uncertainty (Without uncertainty, there would be no need to gather more data). Similar problems appear in more mundane tasks, like learning to drive a car.

For that reason, science is a very natural application area for machine learning. We can model the effects of climate change and how to mitigate it; discover structure in social networks; map the existence of dark matter in the universe by intelligently shifting through weak gravitational lens data, and not only study the mechanisms of protein folding, but discover methods to synthesize new drugs.

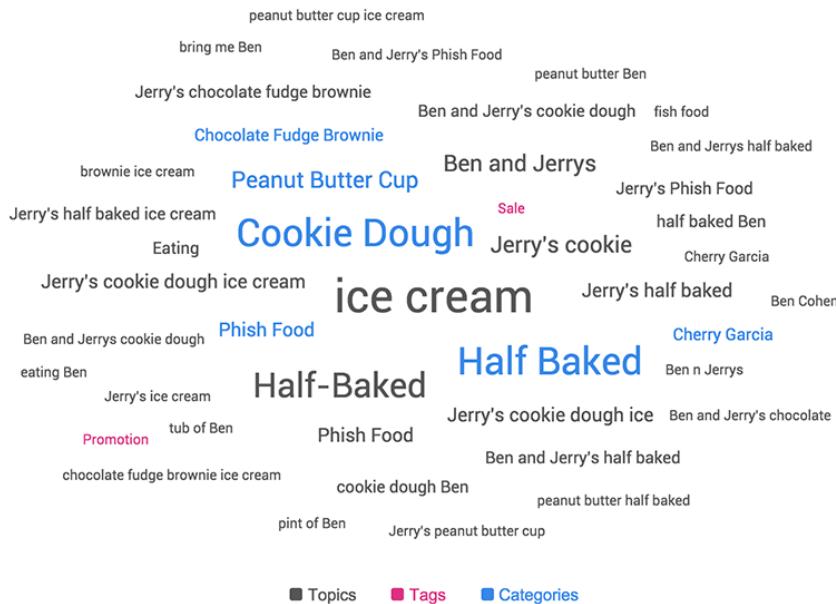
We must be careful, however. In many cases we need to be able to interpret what our model tells us. We also must make sure that the any results we obtain are reproducible. This is something that we shall emphasize in this course.

While machine learning models in science are typically carefully handcrafted by scientists and experts in machine learning and statistics, this is not typically the case in everyday applications. Nevertheless, well-known or home-grown machine learning models are being deployed across the application spectrum. This involve home assistants that try and get you want, web advertising, which tries to find new things for you to want, lending, which tries to optimally lend you money so that you buy what you didn't need before. We also have autonomous vehicles, which take you where you want to go, and ridesharing services, which do the same thing, but use humans instead. Finally, there are many applications in public policy, such as crime prevention, justice, and disease control which use machine learning. In all those cases, we have to worry about a great many things that are outside the scope of the machine learning problems itself. These are (a) privacy: you don't want your data used in ways that you have not consented to (b) fairness: you don't want minorities to be disadvantaged and (c) safety: you don't want your car to crash.

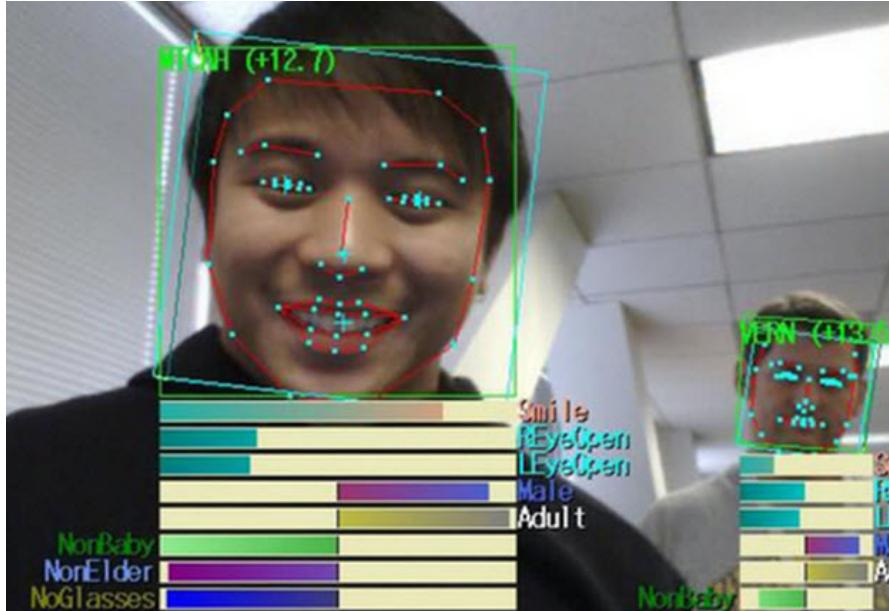
1.1.1 Data analysis, learning and planning

To make the above more concrete, let's have a look at a number of problems in machine learning. These involve learning from and analysing data, including inferring decision rules, and constructing complex plans using the evidence gleaned from the data. Machine learning problems are commonly separated in three different types: supervised, unsupervised and reinforcement learning. Typical supervised learning problems include classification and regression, while unsupervised problems include compression, clustering and topic modelling. Reinforcement learning, on the other hand, is concerned with artificially intelligent agents more generally, with examples including game playing and adaptive control. Their main differences are two. Firstly, the *type* of feedback we have about learning performance. Secondly, and perhaps more importantly, whether or not the problem involves *active data collection*. In this course, we will try and take a global view of these problems in the context of decision theory.

Can machines learn from data?



An unsupervised learning problem: topic modelling



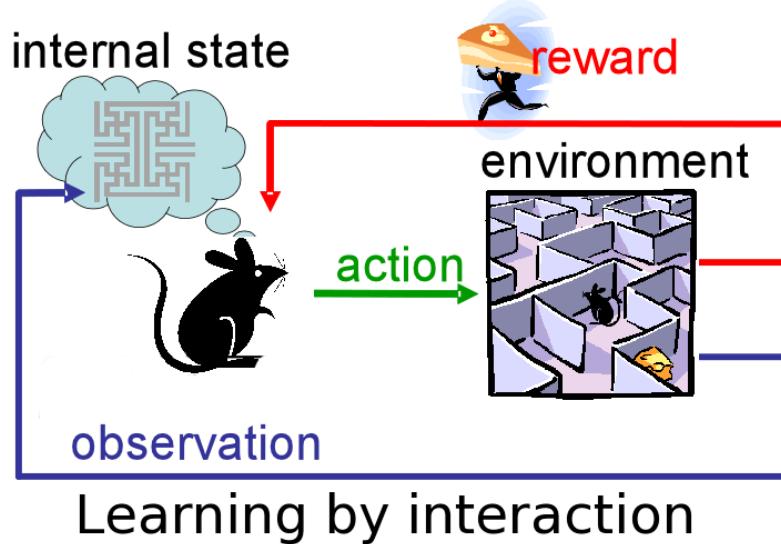
A supervised learning problem: object recognition

You can use machine learning just to analyse, or find structure in the data. This is generally called unsupervised learning. One such example is topic modelling, where you let the algorithm find topics from a corpus of text. These days machines are used to learn from in many applications. These include speech recognition, facial authentication, weather prediction, etc. In general, in these problems we are given a *labelled* dataset with, say, example images from each class. Unfortunately this does not scale very well, because obtaining labels is expensive.

This is partially how science works, because what we need to do is to find a general rule

of nature from data. Starting from some hypothesis and some data, we reach a conclusion. However, many times we may need to actively experiment to obtain more data, perhaps because we found that our model is wrong.

Can machines learn from their mistakes?



Reinforcement learning

Take actions a_1, \dots, a_t , so as to maximise utility $U = \sum_{t=1}^T r_t$

So, what happens when we make a mistake? Can we somehow recognise it? Humans and other animals can actually learn from their mistakes. Consider the proverbial rat in the maze. At some intervals, the experimenter places some cheese in there, and the rat must do a series of actions to obtain it, such as navigating the maze and pulling some levers. It doesn't know how to get to the cheese easily, but it slowly learns the layout of the maze through observation, and in the end, through trial-and-error it is able to get to the cheese very efficiently.

We can formalise this as a reinforcement learning problem, where the rat takes a series of actions; at each step it also obtains a reward, let's say equal to 0 when it has no cheese, and 1 when it eats cheese. Then we can declare that the rat's utility is the sum of all rewards over time, i.e. the total amount of cheese it can eat before it dies. The rat needs to explore the environment in order to be able to get to the cheese.

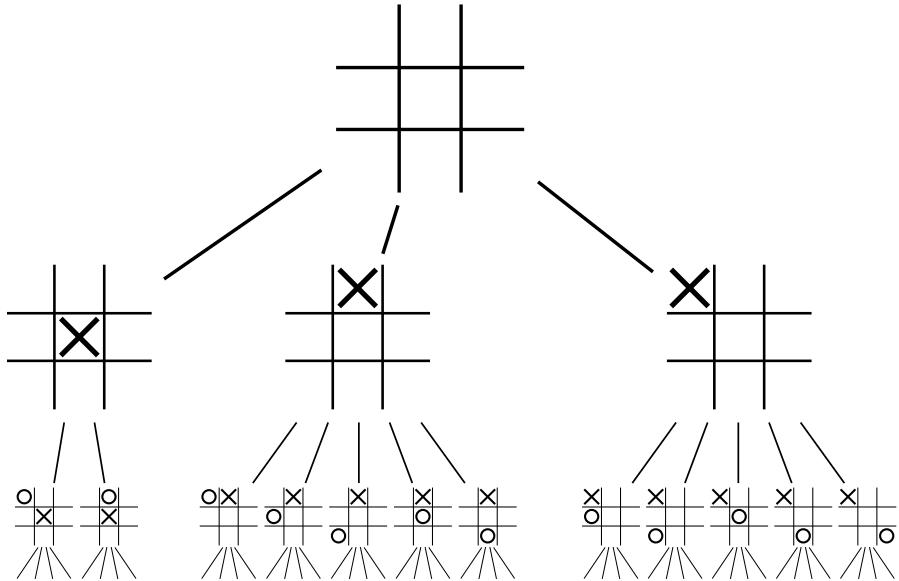
An example in robotics is trying to teach a robot to flip pancakes. One easy thing we can try is to show the robot how to do it, and then let it just copy the demonstrated movement. However, this doesn't work! The robot needs to explore variations of the movement, until it manages to successfully flip pancakes. Again, we can formulate this as a reinforcement learning problem, with a reward that is high whenever the pancake's position is flipped, and on the pan; and low everywhere else. Then the robot can learn to perform this behaviour through trial and error. It's important to note that in this example, merely demonstration is not enough. Neither is reinforcement learning enough. The same thing is true for the recent success of AlphaGo in beating a master human: apart from planning, they used both demonstration data and self-play, so that it could learn through trial and error.

Can machines make complex plans?



I suppose the first question is whether machines can plan ahead. Indeed, even for large problems, such as Go, machines can now perform at least as well as top-rated humans. How is this achieved?

Machines can make complex plans!



The basic construction is the planning tree. This is an enumeration of all possible future events. If a complete enumeration is impossible, a partial tree is constructed. However this requires evaluating non-terminal game positions. In the old times, this was done with heuristics, but now this is data-driven, both through the use of expert databases, and through self-play and reinforcement learning.

1.1.2 Experiment design

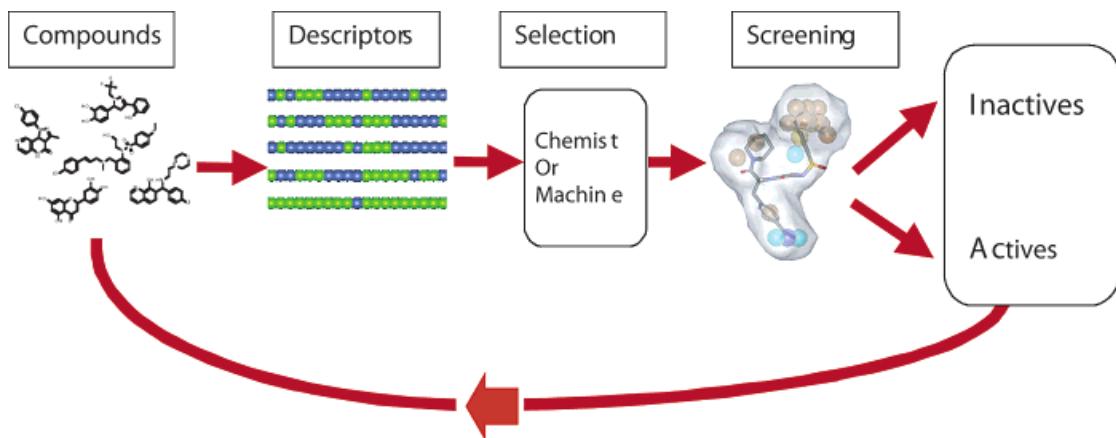
An example that typifies trial and error learning are bandit problems. Imagine that you are in a Casino and you wish to maximise the amount of money you make during the night. There are a lot of machines to play. If you knew which one was the best, then you'd just play it all night long. However, you must also spend time trying out different machines, in order to get an estimate of how much money each one gives out. The trade off between trying out different machines and playing the one you currently think is best is called the exploration-exploitation trade-off and it appears in many problems of experiment design for science.

Adam, the robot scientist



Let's say we want to build a robot scientist and tell it to discover a cure for cancer. What does the scientist do and how can the robot replicate it??

Drug discovery



Simplifying the problem a bit, consider that you have a large number of drug candidates for cancer and you wish to discover those that are active against it. The idea is that you select some of them, then screen them, to sort them into active and inactive. However, there are too many drugs to screen, so the process is interactive. At each cycle, we select some drugs to screen, classify them, and then use this information to select more drugs to screen. This cycle, consequently has two parts: 1. Selecting some drugs given our current knowledge. 2. Updating our knowledge given new evidence.

Drawing conclusions from results

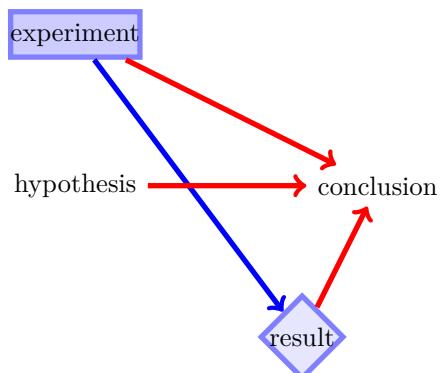


Figure 1.1: Dependence diagram between selection of an experiment, formulation of a hypothesis, and drawing of a conclusion. The result depends only on the experiment. However, the conclusion depends on the experiment, hypothesis and the obtained result. The red lines indicate computational dependencies, while the blue lines indicate physical dependencies.

In general, we would like to have some method which can draw conclusions from results. This involves starting with a hypothesis, performing an experiment to verify or refute it, obtain some experimental result; and then concluding for or against the hypothesis. Here the arrows show dependencies between these variables. So what do we mean by "hypothesis" in this case?

1.1.3 Bayesian inference.

Tycho Brahe's minute eye measurements

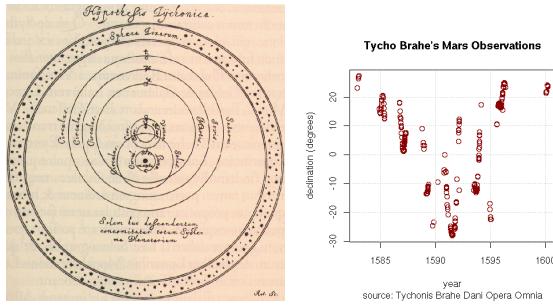
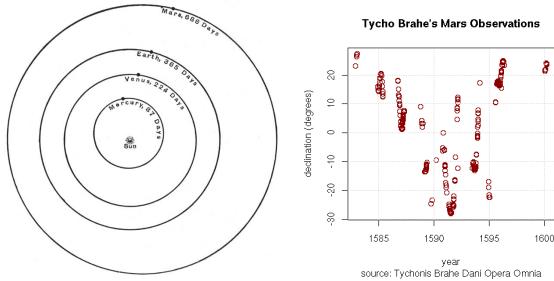


Figure 1.2: Tycho's measurements of the orbit of Mars and the conclusion about the actual orbits, under the assumption of an earth-centric universe with circular orbits.

- Hypothesis: Earth-centric, Circular orbits
- Conclusion: *Specific* circular orbits

Let's take the example of planetary orbits. Here Tycho famously spent 20 years experimentally measuring the location of Mars. He had a hypothesis: that planetary orbits were circular, but he didn't know which were the right orbits. When he tried to fit his data to this hypothesis, he concluded a specific circular orbit for Mars ...around Earth.

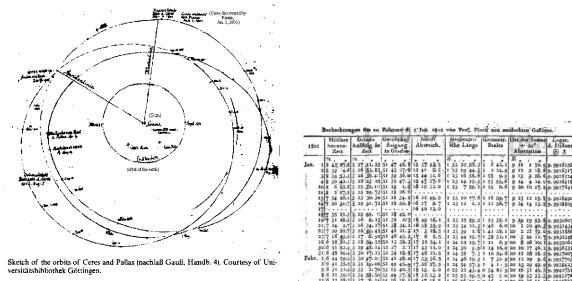
Johannes Kepler's alternative hypothesis



- Hypothesis: Circular *or* elliptic orbits
- Conclusion: Specific *elliptic* orbits

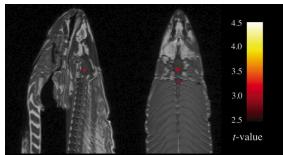
Kepler had a more general hypothesis: that orbits could be circular or elliptic, and he actually accepted that the planets orbited the sun. This led him to the broadly correct model of all planets being in elliptical orbits around the sun. However, the actual verification that all things do not revolve around earth, requires different experiments.

200 years later, Gauss formalised this statistically



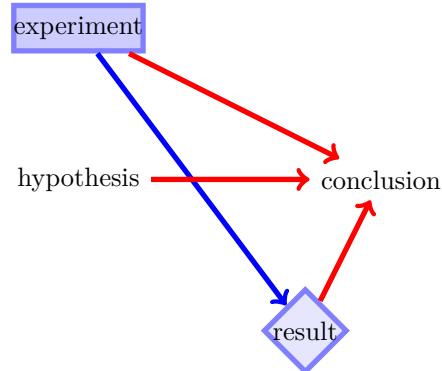
Later on, Gauss collected even more experimental data to calculate the orbit of Ceres. He did this using one of the first formal statistical methods; this allowed him to avoid cheating (like Kepler did, to accentuate his finding that orbits were elliptical).

A warning: The dead salmon mirage



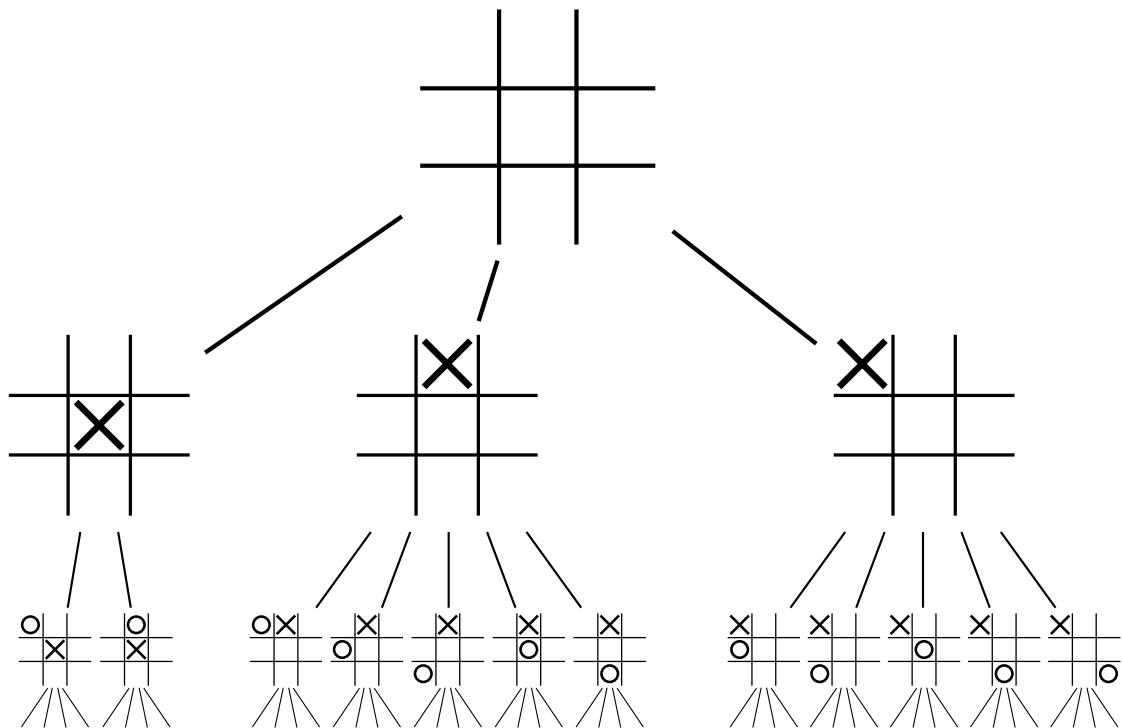
It is quite easy to draw the wrong conclusions from applying machine learning / statistics to your data. For example, it was fashionable to perform fMRI studies in humans to see whether some neurons have a particular functional role. There were even articles saying that "we found the neurons encoding for Angelina Jolie". So some scientists tried to replicate those results. They took a dead salmon, and put it in an fMRI scanner. They checked its brain activity when it was shown images of happy or sad people. Perhaps surprisingly, they found an area of the brain that was correlated with the pictures - so it seemed, as though the dead salmon could distinguish photos of happy people from sad ones. However, this was all due to a misapplication of statistics. In this course, we will try and teach you to avoid such mistakes.

Planning future experiments



I mentioned before that we must decide what experiment to do. This is indeed difficult, especially in setting such as drug discovery where the number of experiments is huge. However, conceptually, there is a simple and elegant solution to this problem.

Planning experiments is like Tic-Tac-Toe



The basic idea is to think of experiment design as a game between the scientist and Nature. At every step, the scientist plays an X to denote an experiment. Then Nature responds with an Observation. The main difference from a game is that Nature is (probably) not adversarial. We can also generalise this idea to problems in robotics, etc.

These kinds of techniques, coming from the reinforcement learning literature have been successfully used at the university of Manchester to create a robot, called Eve, that recently (re)-discovered a malaria drug.

1.1.4 Course overview

Machine learning in practice

Avoiding pitfalls

- Choosing hypotheses.
- Correctly interpreting conclusions.
- Using a good testing methodology.

Machine learning in society

- Privacy — Credit risk.
- Fairness — Job market.
- Safety — Medicine.

One of the things we want to do in this course is teach you to avoid common pitfalls.

Now I want to get into a different track. So far everything has been about pure research, but now machine learning is pervasive: Our phones, cars, watches, bathrooms, kettles are connected to the internet and send a continuous stream of data to companies. In addition, many companies and government actors use machine learning algorithms to make or support decisions. This creates a number of problems in privacy, fairness and safety.

Technical topics

Machine learning problems

- Unsupervised learning. Loosely speaking, this is simply the problem of estimating some structure from data. In statistical terms, it is usually the problem of estimating some joint distribution of random variables under some model assumptions. Problems in unsupervised learning include clustering, anomaly detection, compression.
- Supervised learning. In this setting data can be split in two groups of variables. One group that is always available, and another group that must be predicted. A special case of the problem is when we wish to estimate some function $f : \mathcal{X} \rightarrow \mathcal{Y}$ from data. Classical problems in this setting are classification and regression.
- Reinforcement learning. This is a very general sequential decision problem, where an agent must learn how to behave optimally in an unknown environment only by limited feedback and reinforcement. The standard setting involves the agent trying to maximise its (expected) cumulative reward over time.

Algorithms and models

- Bayesian inference and graphical models.
- Stochastic optimisation and neural networks.
- Backwards induction and Markov decision processes.

Course structure

Module structure

- *Activity-based*, hands-on.
- Mini-lectures with short exercises in each class.
- Technical tutorials and labs in alternate week.

Modules

Three mini-projects.

- Simple decision problems: Credit risk.
- Structured problems: Fake news.
- Sequential problems: Medical diagnostics and treatment.

Chapter 2

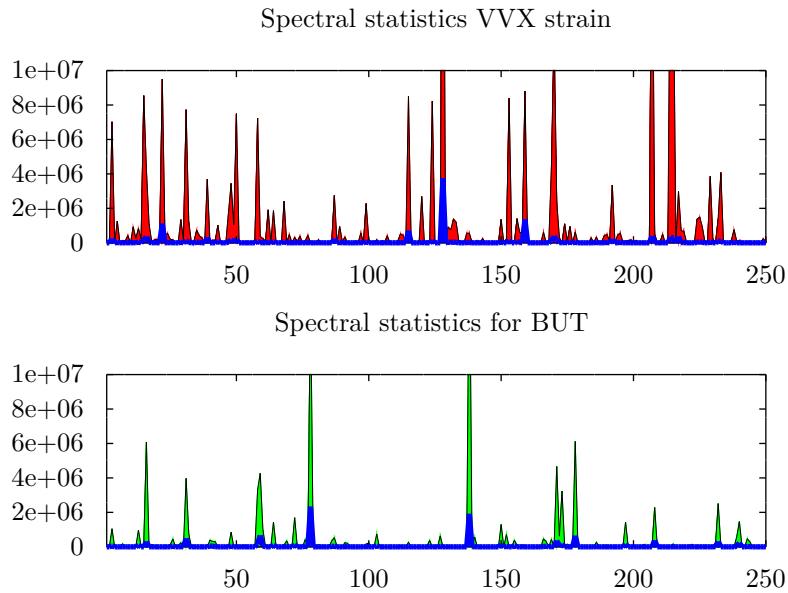
Simple decision problems

This chapter deals with simple decision problems, whereby a decision maker (DM) makes a simple choice among many. In some of these problems the DM has to make a decision after first observing some side-information. Then the DM uses a *decision rule* to assign a probability to each possible decision for each possible side-information. However, designing the decision rule is not trivial, as it relies on previously collected data. A higher-level decision includes choosing the decision rule itself. The problems of classification and regression fall within this framework. While most steps in the process can be automated and formalised, a lot of decisions are actual design choices made by humans. This creates scope for errors and misinterpretation of results.

In this chapter, we shall formalise all these simple decision problems from the point of view of statistical decision theory. The first question is, given a real world application, what type of decision problem does it map to? Then, what kind of machine learning algorithms can we use to solve it? What are the underlying assumptions and how valid are our conclusions?

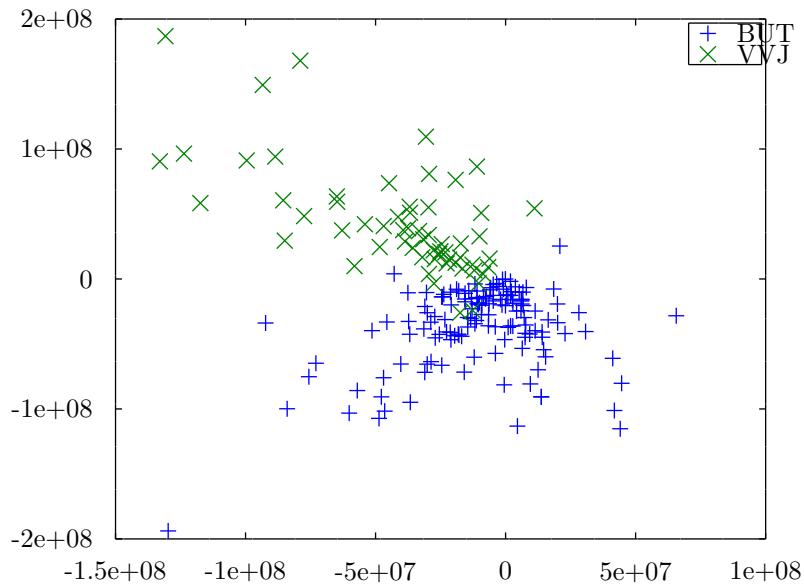
2.1 Nearest neighbours

Discriminating between diseases



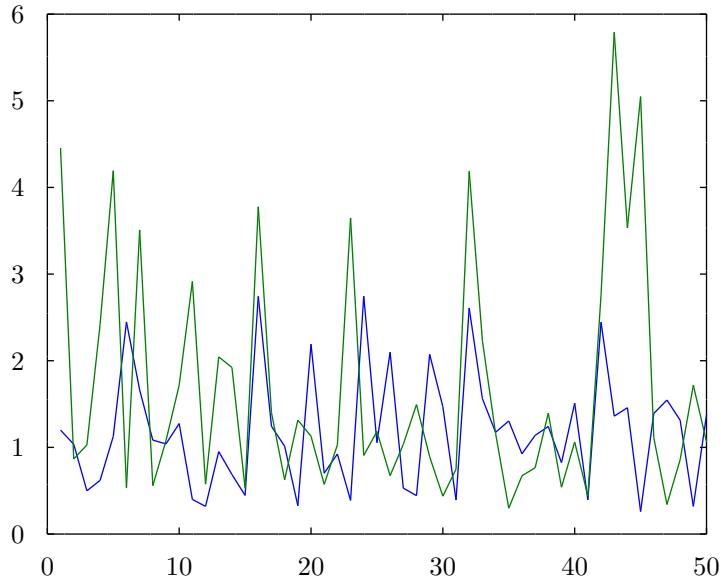
Let's tackle the problem of discriminating between different disease vectors. Ideally, we'd like to have a simple test that tells us what ails us. One kind of test is mass spectrometry. This graph shows spectrometry results for two types of bacteria. There is plenty of variation within each type, both due to measurement error and due to changes in the bacterial strains. Here, we plot the average and maximum energies measured for about 100 different examples from each strain.

Nearest neighbour: the hidden secret of machine learning



Now, is it possible to identify an unknown strain based on this data? Actually, this is possible. Sometimes, very simple algorithms work very well. One of the simplest one involves just measuring the distance between the description of a new unknown strain and known ones. In this visualisation, I projected the 1300-dimensional data into a 2-dimensional space. Here you can clearly see that it is possible to separate the two strains. We can use the distance to examples VVT and BUT in order to decide the type of an unknown strain.

Comparing spectral data



The choice of distance in this kind of algorithm is important, particularly for very high dimensions. For something like a spectrogram, one idea is look at the total area of the difference between two spectral lines.

The nearest neighbour algorithm

The nearest neighbour algorithm for classification (Alg. 1) does not include any complicated learning. Given a training dataset D , it returns a classification decision for any new point x by simply comparing it to its closest k neighbours in the dataset. It then estimates the probability p_y of each class y by calculating the average number of times the neighbours take the class y .

Algorithm 1 k-NN Classify

- 1: **Input** Data $D = \{(x_1, y_1), \dots, (x_T, y_T)\}$, $k \geq 1$, $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+$, new point $x \in \mathcal{X}$
 - 2: $D = \text{Sort}(D, d)$ % Sort D so that $d(x, x_i) \leq d(x, x_{i+1})$.
 - 3: $p_y = \sum_{i=1}^k \mathbb{I}\{y_i = y\} / k$ for $y \in \mathcal{Y}$.
 - 4: **Return** $\mathbf{p} \triangleq (p_1, \dots, p_k)$
-

Algorithm parameters

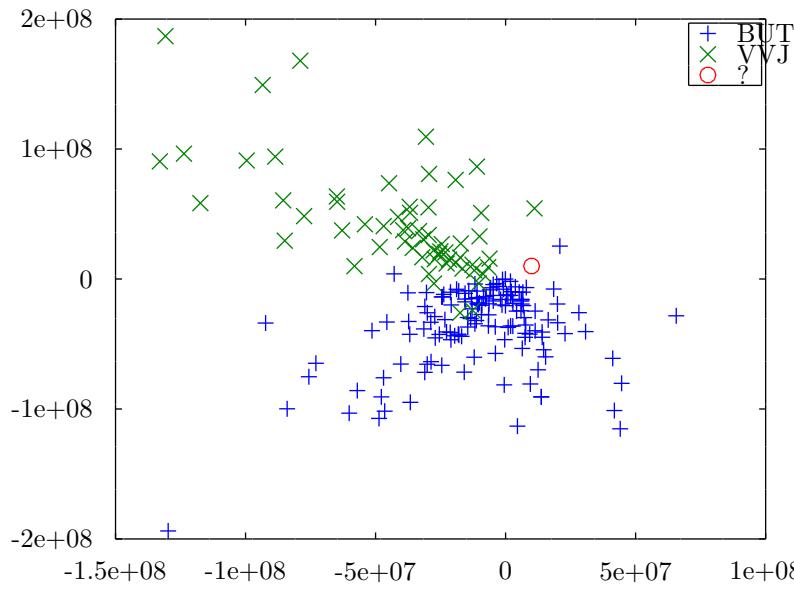
In order to use the algorithm, we must specify some parameters, namely.

- Neighbourhood $k \geq 1$. The number of neighbours to consider.
- Distance $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+$. The function we use to determine what is a neighbour.



Figure 2.1: The nearest neighbours algorithm was introduced by Fix and Hodges Jr³, who also proved consistency properties.

Nearest neighbour: What type is the new bacterium?



Given that the + points represent the BUT type, and the \times points the VVJ type, what type of bacterium could the circle point be?

Separating the model from the classification policy

- The k -NN algorithm returns a model giving class probabilities for new data points.

- It is up to us to decide how to use this model to decide upon a given class. A typical decision making rule can be in the form of a policy π that depends on what the model says. However, the simplest decision rule is to take the most likely class:

$$\pi(a | x) = \mathbb{I}\{p_a \geq p_y \forall y\}, \quad p = \text{k-NN}(D, k, d, x)$$

Discussion: Shortcomings of k -nearest neighbour

- Choice of k The larger k is, the more data you take into account when making your decision. This means that you expect your classes to be more spread out.
- Choice of metric d . The metric d encodes prior information you have about the structure of the data.
- Representation of uncertainty. The predictions of kNN models are simply based on distances and counting. This might not be a very good way to represent uncertainty about class label. In particular, label probabilities should be more uncertain when we have less data.
- Scaling with large amounts of data. A naive implementation of kNN requires the algorithm to shift through all the training data to find the k nearest neighbours, suggesting a super-linear computation time. However, advanced data structures such as Cover Trees (or even KD-trees in low dimensional spaces) can be used to find the neighbours in polylog time.
- Meaning of label probabilities. It is best to think of k-NN as a *model* for predicting the class of a new example from a finite set of existing classes. The model itself might be incorrect, but this should nevertheless be OK for our purposes. In particular, we might later use the model in order to derive classification rules.

Learning outcomes

Understanding

- How kNN works
- The effect of hyperparameters k, d for nearest neighbour.
- The use of kNN to classify new data.

Skills

- Use a standard kNN class in python
- Optimise kNN hyperparameters in an unbiased manner.
- Calculate probabilities of class labels using kNN.

Reflection

- When is kNN a good model?
- How can we deal with large amounts of data?
- How can we best represent uncertainty?

2.2 Reproducibility

One of the main problems in science is reproducibility: when we are trying to draw conclusions from one specific data set, it is easy to make a mistake. For that reason, the scientific process requires us to use our conclusions to make testable predictions, and then test those predictions with new experiments. These new experiments should bear out the results of the previous experiments. In more detail, reproducibility can be thought of as two different aspects of answering the question “can this research be replicated?”

Computational reproducibility: Can the study be repeated?

Can we, from the available information and data, exactly reproduce the reported methods and results?

This is something that is useful to be able to even to the original authors of a study. The standard method for achieving this is using version control tools so that the exact version of algorithms, text and data used to write up the study is appropriately labelled. Ideally, any other researcher should be able to run a single script to reproduce all of the study and its computations. The following tools are going to be used in this course:

- `jupyter` notebooks for interactive note taking.
- `svn`, `git` or `mercurial` version control systems for tracking versions, changes and collaborating with others.

Scientific reproducibility: Is the conclusion correct?

Can we, from the available information and a *new* set of data, reproduce the conclusions of the original study?

Here followup research may involve using exactly the same methods. In AI research would mean for example testing whether an algorithm is really performing as well as it is claimed, by testing it in new problems. This can involve a re-implementation. In more general scientific research, it might be the case that the methodology proposed by an original study is flawed, and so a better method should be used to draw better conclusions. Or it might simply be that more data is needed.

When publishing results about a *new method*, computational reproducibility is essential for

scientific reproducibility.



simple example is the 2016 election. While we can make models about people's opinions regarding candidates in order to predict voting totals, the test of these models comes in the actual election. Unfortunately the only way we have of tuning our models is on previous elections, which are not that frequent, and on the results of previous polls. In addition, predicting the winner of an election is slightly different from predicting how many people are prepared to vote for them across the country. This, together with other factors such as shifting opinions, motivation and how close the sampling distribution is to the voting distribution have a significant effect on accuracy.

The same thing can be done in when dealing purely with data, by making sure we use some

of the data as input to the algorithm, and other data to measure the quality of the algorithm itself. In the following, we assume we have some algorithm $\lambda : \mathcal{D} \rightarrow \Pi$, where \mathcal{D} is the universe of possible input data and Π the possible outputs, e.g. all possible classification policies. We also assume the existence of some quality measure U . How should we measure the quality of our algorithmic choices?

Take classification as an example. For a given training set, simply memorising all the labels of each example gives us perfect performance on the training set. Intuitively, this is not a good measure of performance, as we'd probably get poor results on a freshly sampled set. We can think of the training data as input to an algorithm, and the resulting classifier as the algorithm output. The evaluation function also requires some data in order to measure the performance of the policy. This can be expressed into the following principle.

The principle of independent evaluation

Data used for estimation cannot be used for evaluation.

This applies both to computer-implemented and human-implemented algorithms.

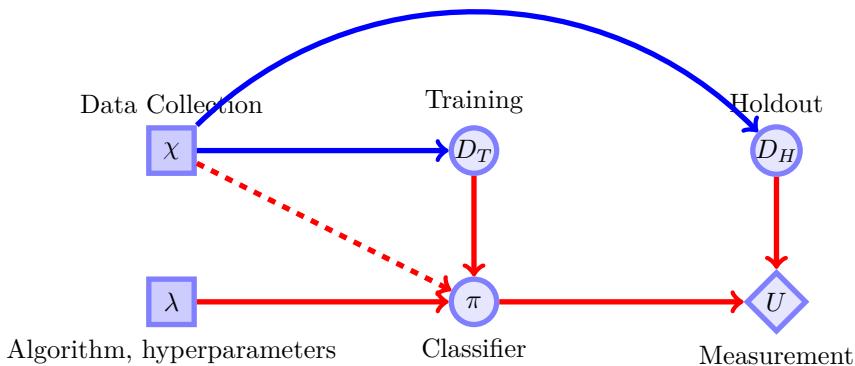


Figure 2.2: The decision process in classification.

One can think of the decision process in classification as follows. First, we decide to collect some data according to some experimental protocol χ . We also decide to use some algorithm (with associated hyperparameters) λ together with data D_T we will obtain from our data collection in order to obtain a classification policy π . Typically, we need to measure the quality of a policy according to how well it classifies on unknown data. This is because our policy has been generated using D_T , and so any measurement of its quality is going to be biased.

For classification problems, there is a natural metric U to measure. The classification accuracy of the classifier. If the classification decisions are stochastic, then the classifier assigns probability $\pi(a | x)$ to each possible label a , and our utility is simply the identity function $U(a, y) \triangleq \mathbb{I}\{a = y\}$.

Classification accuracy

$$\mathbb{E}_\chi[U(\pi)] = \sum_{x,y} \underbrace{\mathbb{P}_\chi(x,y)}_{\text{Data probability}} \underbrace{\pi(a=y|x)}_{\text{Decision probability}}$$

The classification accuracy of policy π under χ is the expected number of times the policy decides π chooses the correct class. However, when approximating χ with a sample D_H , we instead obtain the empirical estimate:

$$\mathbb{E}_{D_H} U(\pi) = \sum_{(x,y) \in D_H} \pi(a=y|x)/|D_H|.$$

Of course, there is no reason to limit ourselves to the identity function. The utility could very well be such that some errors are penalised more than other errors. Consider for example an intrusion detection scenario: it is probably more important to correctly classify intrusions.

2.2.1 The human as an algorithm

The human as an algorithm.

The same way with which an algorithm creates a model from some prior assumptions and data, so can a human select an algorithm and associated hyperparameters by executing an algorithm herself. This involves trying different algorithms and hyperparameters on the same training data D_T and then measuring their performance in the holdout set D_H .

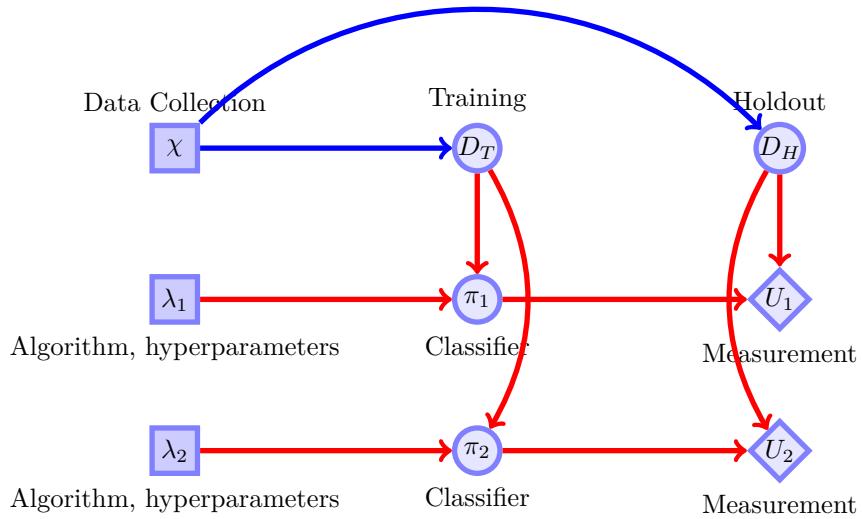


Figure 2.3: Selecting algorithms and hyperparameters through holdouts

Holdout sets

To summarise, holdout sets are used in order to be able to evaluate the performance of specific algorithms, or hyperparameter selection.

- Original data D , e.g. $D = (x_1, \dots, x_T)$.
- Training data $D_T \subset D$, e.g. $D_T = x_1, \dots, x_n$, $n < T$.
- Holdout data $D_H = D \setminus D_T$, used to measure the quality of the result.
- Algorithm λ with hyperparameters ϕ .
- Get algorithm output $\pi = \lambda(D_T, \phi)$.
- Calculate quality of output $U(\pi, D_H)$

We start with some original data D , e.g. $D = (x_1, \dots, x_T)$. We then split this into a training data set $D_T \subset D$, e.g. $D_T = x_1, \dots, x_n$, $n < T$ and holdout dataset $D_H = D \setminus D_T$. This is used to measure the quality of selected algorithms λ and hyperparameters ϕ . We run an algorithm/hyperparameter combination on the training data and obtain a result $\pi = \lambda(D_T, \phi)$.

¹ We then calculate the quality of the output $U(\pi, D_H)$ on the holdout set. Unfortunately, the combination that appears the best due to the holdout result may look inferior in a fresh sample. Following the principle of “data used for evaluation cannot be used for estimation”, we must measure performance on another sample. This ensures that we are not biased in our decision about what is the best algorithm.

Holdout and test sets for unbiased algorithm comparison

Consider the problem of comparing a number of different algorithms in Λ . Each algorithm λ has a different set of hyperparameters Φ_λ . The problem is to choose the best parameters for each algorithm, and then to test them independently. A simple meta-algorithm for doing this is based on the use of a *holdout* set for choosing hyperparameters for each algorithm, and a *test* set to measure algorithmic performance.

Algorithm 2 Unbiased adaptive evaluation through data partitioning

```

Partition data into  $D_T, D_H, D^*$ .
for  $\lambda \in \Lambda$  do
    for  $\phi \in \Phi_\lambda$  do
         $\pi_{\phi,\lambda} = \lambda(D_T, \phi)$ .
    end for
    Get  $\pi_\lambda^*$  maximising  $U(\pi_{\phi,\lambda}, D_H)$ .
     $u_\lambda = U(\pi_\lambda^*, D^*)$ .
end for
 $\lambda^* = \arg \max_\lambda u_\lambda$ .

```

¹ As typically algorithms are maximising the quality metric on the training data,

$$\lambda(D_T) = \arg \max_y U(y, D_T)$$

we typically obtain a biased estimate, which depends both on the algorithm itself and the training data. For k -NN in particular, when we measure accuracy on the training data, we can nearly always obtain near-perfect accuracy, but not always perfect. Can you explain why?

Final performance measurement

When comparing many algorithms, where we must select a hyperparameter for each one, then we can use one dataset as input to the algorithms, and another for selecting hyperparameters. That means that we must use another dataset to measure performance. This is called the testing set. Figure 2.4 illustrates this.

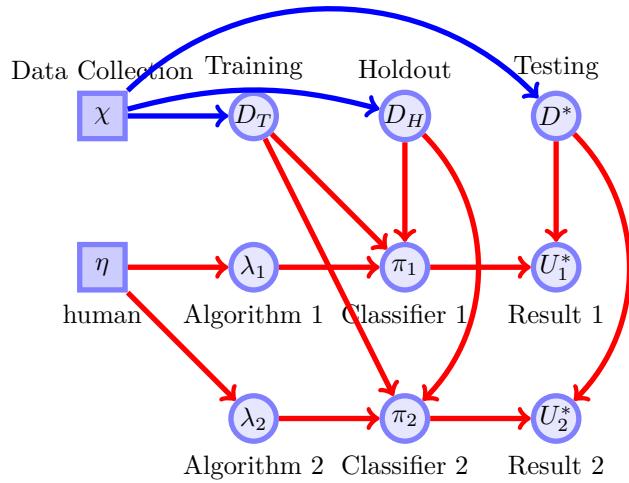


Figure 2.4: Simplified dependency graph for selecting hyperparameters for different algorithms, and comparing them on an independent test set. For the i -th algorithm, the classifier model is

2.2.2 Algorithmic sensitivity

The algorithm's output does have a dependence on its input, obviously. So, how sensitive is the algorithm to the input?

Independent data sets

One simple idea is to just collect independent datasets and see how the output of the algorithm changes when the data changes. However, this is quite expensive, as it might not be easy to collect data in the first place.

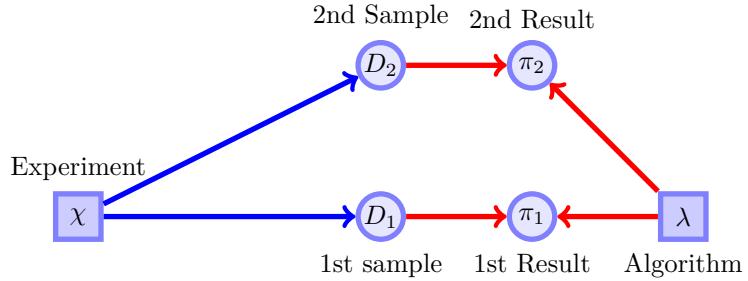


Figure 2.5: Multiple samples

Bootstrap samples

A more efficient idea is to only collect one dataset, but then use it to generate more datasets. The simplest way to do that is by sampling with replacement from the original dataset, new datasets of the same size as the original. Then the original dataset is sufficiently large, this is approximately the same as sampling independent datasets. As usual, we can evaluate our algorithm on an independent data set.

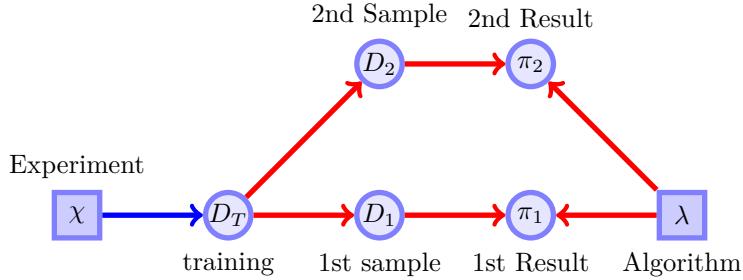


Figure 2.6: Bootstrap replicates of a single sample

Bootstrapping

Bootstrapping is a general technique that can be used to:

- Estimate the sensitivity of λ to the data x .
- Obtain a distribution of estimates π from λ and the data x .
- When estimating the performance of an algorithm on a small dataset D^* , use bootstrap samples of D^* . This allows us to take into account the inherent uncertainty in measured performance. It is very useful to use bootstrapping with pairwise comparisons.

Bootstrapping

1. **Input** Training data D , number of samples k .
2. **For** $i = 1, \dots, k$
3. $D^{(i)} = \text{Bootstrap}(D)$
4. **return** $\{D^{(i)} \mid i = 1, \dots, k\}$.

where $\text{Bootstrap}(D)$ samples with replacement $|D|$ points from D_T .

In more detail, remember that even though the test score is an *independent* measurement of an algorithm's performance, it is *not* the actual expected performance. At best, it's an unbiased estimate of performance. Hence, we'd like to have some way to calculate a likely performance range from the test data. Bootstrapping can help: by taking multiple samples of the test set and calculating performance on each one, we obtain an empirical distribution of scores.

Secondly, we can use it to tell us something about the sensitivity of our algorithm. In particular, by taking multiple samples from the training data, we can end up with multiple

models. If the models are only slightly different, then the algorithm is more stable and we can be more confident in its predictions.

Finally, bagging also allows us to generate probabilistic predictions from deterministic classification algorithms, by simply averaging predictions from multiple bootstrapped predictors. This is called *bagging predictors*¹.

Cross-validation

While we typically use a single training, hold-out and test set, it might be useful to do this multiple times in order to obtain more robust performance estimates. In the simplest case, cross-validation can be used to obtain multiple training and hold-out sets from a single dataset. This works by simply partitioning the data in *k folds* and then using one of the folds as a holdout and the remaining $k - 1$ as training data. This is repeated k times. When k is the same size as the original training data, then the method is called *leave-one-out cross-validation*.

***k*-fold Cross-Validation**

1. **Input** Training data D_T , number of folds k , algorithm λ , measurement function U
2. Create the partition $D^{(1)}, \dots, D^{(k)}$ so that $\bigcup_{i=1}^k D^{(k)} = D$.
3. Define $D_T^{(i)} = D \setminus D^{(i)}$
4. $\pi_i = \lambda(D_T^{(i)})$
5. **For** $i = 1, \dots, k$:
6. $\pi_i = \lambda(D^{(i)})$
7. $u_i = U(\pi_i)$
8. **return** $\{y_1, \dots, y_k\}$.

Independent replication

The gold standard for reproducibility is independent replication. Simply have another team try and reproduce the results you obtained, using completely new data. If the replication is successful, then you can be pretty sure there was no flaw in your original analysis.

Replication study

1. Reinterpret the original hypothesis and experiment.
2. Collect data according to the original protocol, *unless flawed*. It is possible that the original experimental protocol had flaws. Then the new study should try and address this through an improved data collection process. For example, the original study might not have been double-blind. The new study can replicate the results in a double-blind regime.

3. Run the analysis again, *unless flawed*. It is possible that the original analysis had flaws. For example, possible correlations may not have been taken into account.
4. See if the conclusions are in agreement.

Learning outcomes

Understanding

- What is a hold-out set, cross-validation and bootstrapping.
- The idea of not reusing data input to an algorithm to evaluate it.
- The fact that algorithms can be implemented by both humans and machines.

Skills

- Use git and notebooks to document your work.
- Use hold-out sets or cross-validation to compare parameters/algorithms in Python.
- Use bootstrapping to get estimates of uncertainty in Python.

Reflection

- What is a good use case for cross-validation over hold-out sets?
- When is it a good idea to use bootstrapping?
- How can we use the above techniques to avoid the false discovery problem?
- Can these techniques fully replace independent replication?

EXERCISE 1. Work in teams of 2-3 students.

Select an arbitrary classification dataset from <https://archive.ics.uci.edu/ml/datasets.html?task=cla>.

Select any arbitrary machine learning algorithm for classification from `scikitlearn` that can be used with this dataset, and identify its main hyperparameters.

Varying at least one hyperparameter, use bootstrapping and/or cross-validation to find the optimal value for that hyperparameter, and report its performance. How close to the reported accuracy do you expect its performance to be in reality? What are the factors that might cause it to deviate?

Write a short report summarising both your methodology and your results. Exchange this report with another group of students. See whether you can reproduce exactly what they have done.

2.3 Beliefs and probabilities

Probability can be used to describe purely chance events, as in for example quantum physics. However, it is mostly used to describe uncertain events, such as the outcome of a dice roll or a coin flip, which only appear random. In fact, one can take it even further than that, and use it to model subjective uncertainty about any arbitrary event. Although probabilities are not the only way in which we can quantify uncertainty, it is a simple enough model, and with a rich enough history in mathematics, statistics, computer science and engineering that it is the most useful.

Uncertainty

- We cannot perfectly predict the future.
- We cannot know for sure what happened in the past.
- How can we quantify this uncertainty?
- Probabilities!

Axioms of probability

For any probability measure² P on (Ω, Σ) ,

1. The probability of the certain event is $P(\Omega) = 1$
2. The probability of the impossible event is $P(\emptyset) = 0$
3. The probability of any event $A \in \Sigma$ is $0 \leq P(A) \leq 1$.
4. If A, B are disjoint, i.e. $A \cap B = \emptyset$, meaning that they cannot happen at the same time, then

$$P(A \cup B) = P(A) + P(B)$$

Sometimes we would like to calculate the probability of some event A happening given that we know that some other event B has happened. For this we need to first define the idea of conditional probability.

Definition 2.3.1 (Conditional probability). The probability of A happening if we know that B has happened is defined to be:

$$P(A | B) \triangleq \frac{P(A \cap B)}{P(B)}.$$

Conditional probabilities obey the same rules as probabilities. Here, the probability measure of any event A given B is defined to be the probability of the intersection of the events divided by the second event. We can rewrite this definition as follows, by using the definition for $P(B | A)$

Bayes's theorem

For $P(A_1 \cup A_2) = 1$, $A_1 \cap A_2 = \emptyset$,

$$P(A_i | B) = \frac{P(B | A_i)P(A_i)}{P(B)} = \frac{P(B | A_i)P(A_i)}{P(B | A_1)P(A_1) + P(B | A_2)P(A_2)}$$

EXAMPLE 1 (probability of rain). What is the probability of rain given a forecast x_1 or x_2 ?

$$\begin{array}{l|l} \omega_1: \text{rain} & P(\omega_1) = 80\% \\ \omega_2: \text{dry} & P(\omega_2) = 20\% \end{array}$$

Table 2.1: Prior probability of rain tomorrow

$$\begin{array}{l|l} x_1: \text{rain} & P(x_1 | \omega_1) = 90\% \\ x_2: \text{dry} & P(x_2 | \omega_2) = 50\% \end{array}$$

Table 2.2: Probability the forecast is correct

$$\begin{aligned} P(\omega_1 | x_1) &= 87.8\% \\ P(\omega_1 | x_2) &= 44.4\% \end{aligned}$$

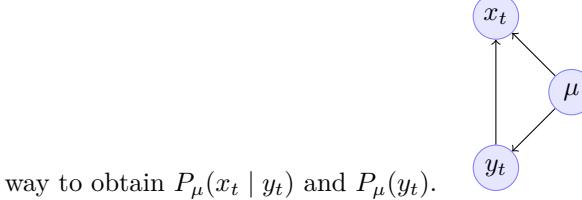
Table 2.3: Probability that it will rain given the forecast

Classification in terms of conditional probabilities

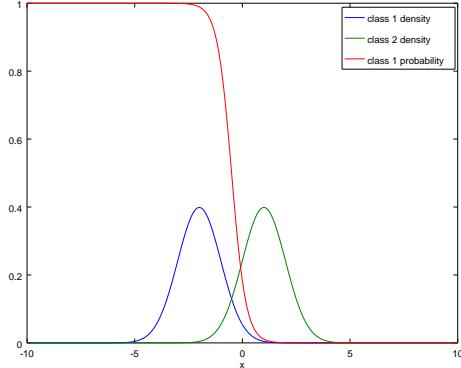
Conditional probability naturally appears in classification problems. Given a new example vector of data $x_t \in \mathcal{X}$, we would like to calculate the probability of different classes $c \in \mathcal{Y}$ given the data, $P_\mu(y_t = c | x_t)$. If we somehow obtained the distribution of data $P_\mu(x_t | y_t)$ for each possible class, as well as the prior class probability $P_\mu(y_t = c)$, from Bayes's theorem, we see that we can obtain the probability of the class:

$$P_\mu(y_t = c | x_t) = \frac{P_\mu(x_t | y_t = c)P_\mu(y_t = c)}{\sum_{c' \in \mathcal{Y}} P_\mu(x_t | y_t = c')P_\mu(y_t = c')}$$

for any class c . This directly gives us a method for classifying new data, as long as we have a

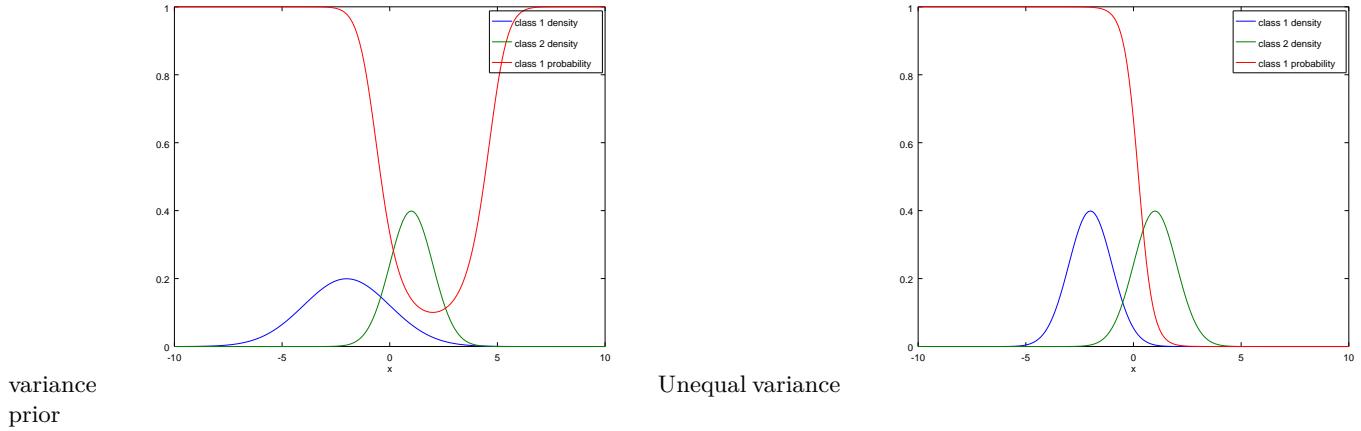


way to obtain $P_\mu(x_t | y_t)$ and $P_\mu(y_t)$.



EXAMPLE 2 (Normal distribution).

Equal prior and



But how can we get a probability model in the first place?

Subjective probability

While probabilities apply to truly random events, they are also useful for representing subjective uncertainty. In this course, we will use a special symbol for subjective probability, ξ .

Subjective probability measure ξ

- If we think event A is more likely than B , then $\xi(A) > \xi(B)$.
- Usual rules of probability apply:
 1. $\xi(A) \in [0, 1]$.
 2. $\xi(\emptyset) = 0$.
 3. If $A \cap B = \emptyset$, then $\xi(A \cup B) = \xi(A) + \xi(B)$.

Bayesian inference illustration

Use a subjective belief $\xi(\mu)$ on \mathcal{M}

- *Prior* belief $\xi(\mu)$ represents our initial uncertainty.
- We *observe history* h .
- Each possible μ assigns a *probability* $P_\mu(h)$ to h .
- We can use this to *update* our belief via Bayes' theorem to obtain the *posterior* belief:

$$\xi(\mu | h) \propto P_\mu(h)\xi(\mu) \quad (\text{conclusion} = \text{evidence} \times \text{prior})$$



2.3.1 Probability and Bayesian inference

One of the most important methods in machine learning and statistics is that of Bayesian inference. This is the most fundamental method of drawing conclusions from data and explicit prior assumptions. In Bayesian inference, prior assumptions are represented as probabilities on a space of hypotheses. Each hypothesis is seen as a probabilistic model of all possible data that we can see.

Frequently, we want to draw conclusions from data. However, the conclusions are never solely inferred from data, but also depend on prior assumptions about reality.

Some examples

EXAMPLE 3. John claims to be a medium. He throws a coin n times and predicts its value always correctly. Should we believe that he is a medium?

- μ_1 : John is a medium.
- μ_0 : John is not a medium.

The answer depends on what we *expect* a medium to be able to do, and how likely we thought he'd be a medium in the first place.

EXAMPLE 4. Traces of DNA are found at a murder scene. We perform a DNA test against a database of 10^4 citizens registered to be living in the area. We know that the probability of a false positive (that is, the test finding a match by mistake) is 10^{-6} . If there is a match in the database, does that mean that the citizen was at the scene of the crime?

Bayesian inference

Now let us apply this idea to our specific problem. We already have the probability of the observation for each model, but we just need to define a *prior probability* for each model. Since this is usually completely subjective, we give it another symbol.

Prior probability

The prior probability ξ on a set of models \mathcal{M} specifies our subjective belief $\xi(\mu)$ that each model is true.³

This allows us to calculate the probability of John being a medium, given the data:

$$\xi(\mu_1 | \mathbf{x}) = \frac{\mathbb{P}(\mathbf{x} | \mu_1)\xi(\mu_1)}{\mathbb{P}_\xi(\mathbf{x})},$$

where

$$\mathbb{P}_\xi(\mathbf{x}) \triangleq \mathbb{P}(\mathbf{x} | \mu_1)\xi(\mu_1) + \mathbb{P}(\mathbf{x} | \mu_0)\xi(\mu_0).$$

The only thing left to specify is $\xi(\mu_1)$, the probability that John is a medium before seeing the data. This is our subjective prior belief that mediums exist and that John is one of them. More generally, we can think of Bayesian inference as follows:

- We start with a set of mutually exclusive models $\mathcal{M} = \{\mu_1, \dots, \mu_k\}$.
- Each model μ is represented by a specific probabilistic model for any possible data x , that is $P_\mu(x) \equiv \mathbb{P}(x | \mu)$.
- For each model, we have a prior probability $\xi(\mu)$ that it is correct.
- After observing the data, we can calculate a posterior probability that the model is correct:

$$\xi(\mu | x) = \frac{\mathbb{P}(x | \mu)\xi(\mu)}{\sum_{\mu' \in \mathcal{M}} \mathbb{P}(x | \mu')\xi(\mu')} = \frac{P_\mu(x)\xi(\mu)}{\sum_{\mu' \in \mathcal{M}} P_{\mu'}(x)\xi(\mu')}.$$

Interpretation

- \mathcal{M} : Set of all possible models that could describe the data.
- $P_\mu(x)$: Probability of x under model μ .
- Alternative notation $\mathbb{P}(x | \mu)$: Probability of x given that model μ is correct.
- $\xi(\mu)$: Our belief, before seeing the data, that μ is correct.
- $\xi(\mu | x)$: Our belief, after seeing the data, that μ is correct.

It must be emphasized that $P_\mu(x) = \mathbb{P}(x | \mu)$ as they are simply two different notations for the same thing. In words the first can be seen as the probability that model μ assigns to data x , while the second as the probability of x if μ is the true model. Combining the prior belief with evidence is key in this procedure. Our posterior belief can then be used as a new prior belief when we get more evidence.

EXERCISE 2 (Continued example for medium). Now let us apply this idea to our specific problem. We first make an independence assumption. In particular, we can assume that success and failure comes from a Bernoulli distribution with a parameter depending on the model.

$$P_\mu(x) = \prod_{t=1}^n P_\mu(x_t). \quad (\text{independence property})$$

We first need to specify how well a medium could predict. Let's assume that a true medium would be able to predict perfectly, and that a non-medium would only predict randomly. This leads to the following models:

$$\begin{array}{lll} P_{\mu_1}(x_t = 1) = 1, & P_{\mu_1}(x_t = 0) = 0. & (\text{true medium model}) \\ P_{\mu_0}(x_t = 1) = 1/2, & P_{\mu_0}(x_t = 0) = 1/2. & (\text{non-medium model}) \end{array}$$

The only thing left to specify is $\xi(\mu_1)$, the probability that John is a medium before seeing the data. This is our subjective prior belief that mediums exist and that John is one of them.

$$\xi(\mu_0) = 1/2, \quad \xi(\mu_1) = 1/2. \quad (\text{prior belief})$$

Combining the prior belief with evidence is key in this procedure. Our posterior belief can then be used as a new prior belief when we get more evidence.

$$\begin{aligned} \xi(\mu_1 | x) &= \frac{P_{\mu_1}(x)\xi(\mu_1)}{\mathbb{P}_\xi(x)} && (\text{posterior belief}) \\ \mathbb{P}_\xi(x) &\triangleq P_{\mu_1}(x)\xi(\mu_1) + P_{\mu_0}(x)\xi(\mu_0). && (\text{marginal distribution}) \end{aligned}$$

Throw a coin 4 times, and have a classmate make a prediction. What your belief that your classmate is a medium? Is the prior you used reasonable?

Sequential update of beliefs

Assume you have n meteorologists. At each day t , each meteorologist i gives a probability $p_{t,\mu_i} \triangleq P_{\mu_i}(x_t = \text{rain})$ for rain. Consider the case of there being three meteorologists, and each one making the following prediction for the coming week. Start with a uniform prior $\xi(\mu) = 1/3$ for each model.

	M	T	W	T	F	S	S
CNN	0.5	0.6	0.7	0.9	0.5	0.3	0.1
SMHI	0.3	0.7	0.8	0.9	0.5	0.2	0.1
YR	0.6	0.9	0.8	0.5	0.4	0.1	0.1
Rain?	Y	Y	Y	N	Y	N	N

Table 2.4: Predictions by three different entities for the probability of rain on a particular day, along with whether or not it actually rained.

EXERCISE 3.

- n meteorological stations $\{\mu_i \mid i = 1, \dots, n\}$
- The i -th station predicts rain $P_{\mu_i}(x_t \mid x_1, \dots, x_{t-1})$.
- Let $\xi_t(\mu)$ be our belief at time t . Derive the next-step belief $\xi_{t+1}(\mu) \triangleq \xi_t(\mu|y_t)$ in terms of the current belief ξ_t .
- Write a python function that computes this posterior

$$\xi_{t+1}(\mu) \triangleq \xi_t(\mu|x_t) = \frac{P_\mu(x_t \mid x_1, \dots, x_{t-1})\xi_t(\mu)}{\sum_{\mu'} P_{\mu'}(x_t \mid x_1, \dots, x_{t-1})\xi_t(\mu')}$$

Bayesian inference for Bernoulli distributions

Estimating a coin's bias

A fair coin comes heads 50% of the time. We want to test an unknown coin, which we think may not be completely fair.

For a sequence of throws $x_t \in \{0, 1\}$,

$$P_\theta(x) \propto \prod_t \theta^{x_t} (1 - \theta)^{1-x_t} = \theta^{\#\text{Heads}} (1 - \theta)^{\#\text{Tails}}$$

Say we throw the coin 100 times and obtain 70 heads. Then we plot the *likelihood* $P_\theta(x)$ of different models. From these, we calculate a *posterior* distribution over the correct models. This represents our conclusion given our prior and the data. If the prior distribution is described by the so-called Beta density

$$f(\theta \mid \alpha, \beta) \propto \theta^{\alpha-1} (1 - \theta)^{\beta-1}$$

where α, β describe the shape of the Beta distribution.

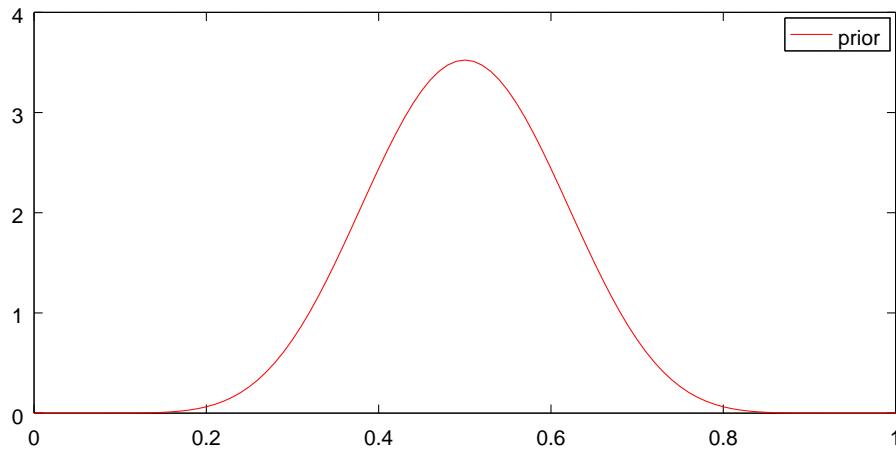


Figure 2.7: Prior belief ξ about the coin bias θ .

Learning outcomes

Understanding

- The axioms of probability, marginals and conditional distributions.
- The philosophical underpinnings of Bayesianism.
- The simple conjugate model for Bernoulli distributions.

Skills

- Be able to calculate with probabilities using the marginal and conditional definitions and Bayes rule.
- Being able to implement a simple Bayesian inference algorithm in Python.

Reflection

- How useful is the Bayesian representation of uncertainty?
- How restrictive is the need to select a prior distribution?
- Can you think of another way to explicitly represent uncertainty in a way that can incorporate new evidence?

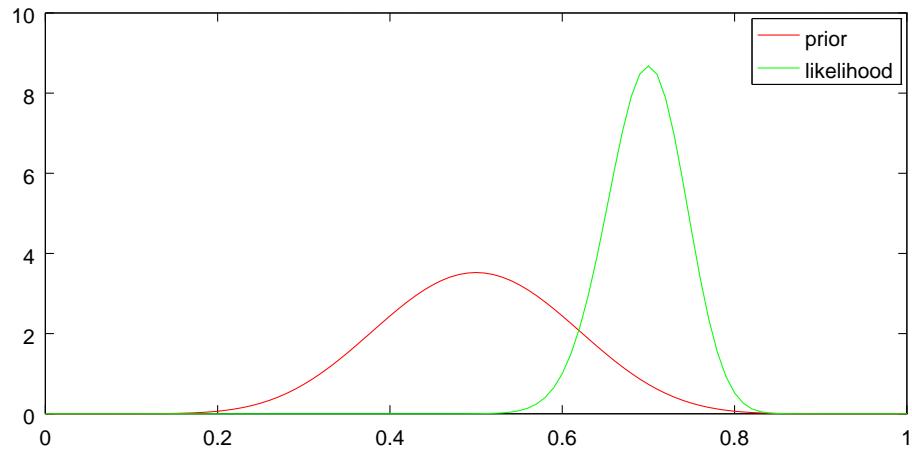


Figure 2.8: Prior belief ξ about the coin bias θ and likelihood of θ for the data.

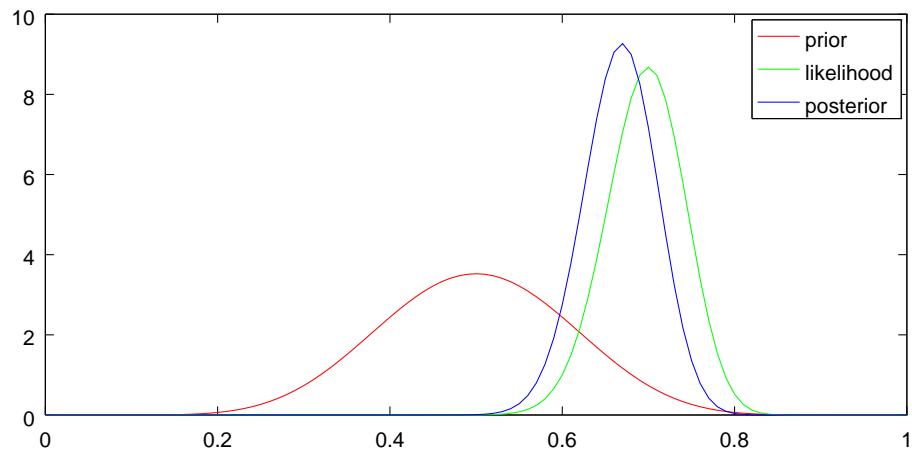


Figure 2.9: Prior belief $\xi(\theta)$ about the coin bias θ , likelihood of θ for the data, and posterior belief $\xi(\theta | x)$

2.4 Hierarchies of decision making problems

All machine learning problems are essentially decision problems. This essentially means replacing some human decisions with machine decisions. One of the simplest decision problems is classification, where you want an algorithm to decide the correct class of some data, but even within this simple framework there is a multitude of decisions to be made. The first is how to frame the classification problem the first place. The second is how to collect, process and annotate the data. The third is choosing the type of classification model to use. The fourth is how to use the collected data to find an optimal classifier within the selected type. After all this has been done, there is the problem of classifying new data. In this course, we will take a holistic view of the problem, and consider each problem in turn, starting from the lowest level and working our way up.

2.4.1 Simple decision problems

Preferences

The simplest decision problem involves selecting one item from a set of choices, such as in the following examples

EXAMPLE 5. Food

- A McDonald's cheeseburger
- B Surstromming
- C Oatmeal

Money

- A 10,000,000 SEK
- B 10,000,000 USD
- C 10,000,000 BTC

Entertainment

- A Ticket to Liseberg
- B Ticket to Rebstar
- C Ticket to Nutcracker

Rewards and utilities

In the decision theoretic framework, the things we receive are called rewards, and we assign a utility value to each one of them, showing which one we prefer.

- Each choice is called a *reward* $r \in \mathcal{R}$.
- There is a *utility function* $U : \mathcal{R} \rightarrow \mathbb{R}$, assigning values to reward.

- We (weakly) prefer A to B iff $U(A) \geq U(B)$.

In each case, given U the choice between each reward is trivial. We just select the reward:

$$r^* \in \arg \max_r U(r)$$

The main difficult is actually selecting the appropriate utility function. In a behavioural context, we simply assume that humans act with respect to a specific utility function. However, figuring out this function from behavioural data is non trivial. Even when this assumption is correct, individuals do not have a common utility function.

EXERCISE 4. From your individual preferences, derive a *common utility function* that reflects everybody's preferences in the class for each of the three examples. Is there a simple algorithm for deciding this? Would you consider the outcome fair?

Preferences among random outcomes

EXAMPLE 6. Would you rather ...

A Have 100 EUR now?

B Flip a coin, and get 200 EUR if it comes heads?

The expected utility hypothesis

Rational decision makers prefer choice A to B if

$$\mathbb{E}(U|A) \geq \mathbb{E}(U|B),$$

where the expected utility is

$$\mathbb{E}(U|A) = \sum_r U(r) \mathbb{P}(r|A).$$

In the above example, $r \in \{0, 100, 200\}$ and $U(r)$ is increasing, and the coin is fair.

- If U is convex, we prefer B.
- If U is concave, we prefer A.
- If U is linear, we don't care.

Uncertain rewards

However, in real life, there are many cases where we can only choose between uncertain outcomes. The simplest example are lottery tickets, where rewards are essentially random. However, in many cases the rewards are not really random, but simply uncertain. In those cases it is useful to represent our uncertainty with probabilities as well, even though there is nothing really random.

- Decisions $a \in \mathcal{A}$
- Each choice is called a *reward* $r \in \mathcal{R}$.

$\rho(\omega, a)$	a_1	a_2
ω_1	dry, carrying umbrella	wet
ω_2	dry, carrying umbrella	dry
$U[\rho(\omega, a)]$	a_1	a_2
ω_1	0	-10
ω_2	0	1

Table 2.5: Rewards and utilities.

$\rho(\omega, a)$	a_1	a_2
ω_1	dry, carrying umbrella	wet
ω_2	dry, carrying umbrella	dry
$U[\rho(\omega, a)]$	a_1	a_2
ω_1	0	-10
ω_2	0	1
$\mathbb{E}_P(U a)$	0	-1.2

Table 2.6: Rewards, utilities, expected utility for 20% probability of rain.

- There is a *utility function* $U : \mathcal{R} \rightarrow \mathbb{R}$, assigning values to reward.
- We (weakly) prefer A to B iff $U(A) \geq U(B)$.

EXAMPLE 7. You are going to work, and it might rain. What do you do?

- a_1 : Take the umbrella.
- a_2 : Risk it!
- ω_1 : rain
- ω_2 : dry
- $\max_a \min_\omega U = 0$
- $\min_\omega \max_a U = 0$

Expected utility

$$\mathbb{E}(U | a) = \sum_r U[\rho(\omega, a)] \mathbb{P}(\omega | a)$$

EXAMPLE 8. You are going to work, and it might rain. The forecast said that the probability of rain (ω_1) was 20%. What do you do?

- a_1 : Take the umbrella.
- a_2 : Risk it!

2.4.2 Decision rules

We now move from simple decisions to decisions that depend on some observation. We shall start with a simple problem in applied meteorology. Then we will discuss hypothesis testing as a decision making problem. Finally, we will go through an exercise in Bayesian methods for classification.

Bayes decision rules

Consider the case where outcomes are independent of decisions:

$$U(\xi, a) \triangleq \sum_{\mu} U(\mu, a)\xi(\mu)$$

This corresponds e.g. to the case where $\xi(\mu)$ is the belief about an unknown world.

Definition 2.4.1 (Bayes utility). The maximising decision for ξ has an expected utility equal to:

$$U^*(P) \triangleq \max_{a \in \mathcal{A}} U(\xi, a). \quad (2.4.1)$$

The n -meteorologists problem

Of course, we may not always just be interested in classification performance in terms of predicting the most likely class. It strongly depends on the problem we are actually wanting to solve. In biometric authentication, for example, we want to guard against the unlikely event that an impostor will successfully be authenticated. Even if the decision rule that always says 'OK' has the lowest classification error in practice, the expected cost of impostors means that the optimal decision rule must sometimes say 'Failed' even if this leads to false rejections sometimes.

EXERCISE 5. Assume you have n meteorologists. At each day t , each meteorologist i gives a probability $p_{t,\mu_i} \triangleq P_{\mu_i}(x_t = \text{rain})$ for rain. Consider the case of there being three meteorologists, and each one making the following prediction for the coming week. Start with a uniform prior $\xi(\mu) = 1/3$ for each model.

	M	T	W	T	F	S	S
CNN	0.5	0.6	0.7	0.9	0.5	0.3	0.1
SMHI	0.3	0.7	0.8	0.9	0.5	0.2	0.1
YR	0.6	0.9	0.8	0.5	0.4	0.1	0.1
Rain?	Y	Y	Y	N	Y	N	N

Table 2.7: Predictions by three different entities for the probability of rain on a particular day, along with whether or not it actually rained.

1. What is your belief about the quality of each meteorologist after each day?
2. What is your belief about the probability of rain each day?

$$P_\xi(x_t = \text{rain} \mid x_1, x_2, \dots, x_{t-1}) = \sum_{\mu \in \mathcal{M}} P_\mu(x_t = \text{rain} \mid x_1, x_2, \dots, x_{t-1})\xi(\mu \mid x_1, x_2, \dots, x_{t-1})$$

3. Assume you can decide whether or not to go running each day. If you go running and it does not rain, your utility is 1. If it rains, it's -10. If you don't go running, your utility is 0. What is the decision maximising utility in expectation (with respect to the posterior) each day?

2.4.3 Statistical testing

A common type of decision problem is a statistical test. This arises when we have a set of possible candidate models \mathcal{M} and we need to be able to decide which model to select after we see the evidence. Many times, there is only one model under consideration, μ_0 , the so-called *null hypothesis*. Then, our only decision is whether or not to accept or reject this hypothesis.

Simple hypothesis testing

Let us start with the simple case of needing to compare two models.

The simple hypothesis test as a decision problem

- $\mathcal{M} = \{\mu_0, \mu_1\}$
- a_0 : Accept model μ_0
- a_1 : Accept model μ_1

U	μ_0	μ_1
a_0	1	0
a_1	0	1

Table 2.8: Example utility function for simple hypothesis tests.

There is no reason for us to be restricted to this utility function. As it is diagonal, it effectively treats both types of errors in the same way.

EXAMPLE 9 (Continuation of the medium example).

- μ_1 : that John is a medium.

- μ_0 : that John is not a medium.

Let x_t be 0 if John makes an incorrect prediction at time t and $x_t = 1$ if he makes a correct prediction. Let us once more assume a Bernoulli model, so that John's claim that he can predict our tosses perfectly means that for a sequence of tosses $\mathbf{x} = x_1, \dots, x_n$,

$$P_{\mu_1}(\mathbf{x}) = \begin{cases} 1, & x_t = 1 \forall t \in [n] \\ 0, & \exists t \in [n] : x_t = 0. \end{cases}$$

That is, the probability of perfectly correct predictions is 1, and that of one or more incorrect prediction is 0. For the other model, we can assume that all draws are independently and identically distributed from a fair coin. Consequently, no matter what John's predictions are, we have that:

$$P_{\mu_0}(\mathbf{x} = 1 \dots 1) = 2^{-n}.$$

So, for the given example, as stated, we have the following facts:

- If John makes one or more mistakes, then $\mathbb{P}(\mathbf{x} | \mu_1) = 0$ and $\mathbb{P}(\mathbf{x} | \mu_0) = 2^{-n}$. Thus, we should perhaps say that then John is not a medium
- If John makes no mistakes at all, then

$$\mathbb{P}(\mathbf{x} = 1, \dots, 1 | \mu_1) = 1, \quad \mathbb{P}(\mathbf{x} = 1, \dots, 1 | \mu_0) = 2^{-n}. \quad (2.4.2)$$

Now we can calculate the posterior distribution, which is

$$\xi(\mu_1 | \mathbf{x} = 1, \dots, 1) = \frac{1 \times \xi(\mu_1)}{1 \times \xi(\text{model}_1) + 2^{-n}(1 - \xi(\mu_1))}.$$

Our expected utility for taking action a_0 is actually

$$\mathbb{E}_{\xi}(U | a_0) = 1 \times \xi(\mu_0 | \mathbf{x}) + 0 \times \xi(\mu_1 | \mathbf{x}), \quad \mathbb{E}_{\xi}(U | a_1) = 0 \times \xi(\mu_0 | \mathbf{x}) + 1 \times \xi(\mu_1 | \mathbf{x})$$

Null hypothesis test

Many times, there is only one model under consideration, μ_0 , the so-called *null hypothesis*. This happens when, for example, we have no simple way of defining an appropriate alternative. Consider the example of the medium: How should we expect a medium to predict? Then, our only decision is whether or not to accept or reject this hypothesis.

The null hypothesis test as a decision problem

- a_0 : Accept model μ_0
- a_1 : Reject model μ_0

EXAMPLE 10. Construction of the test for the medium

- μ_0 is simply the $Bernoulli(1/2)$ model: responses are by chance.
- We need to design a policy $\pi(a | \mathbf{x})$ that accepts or rejects depending on the data.
- Since there is no alternative model, we can only construct this policy according to its properties when μ_0 is true.
- In particular, we can fix a policy that only chooses a_1 when μ_0 is true a proportion δ of the time.
- This can be done by construcing a threshold test from the inverse-CDF.

Using *p*-values to construct statistical tests

Definition 2.4.2 (Null statistical test). A statistical test π is a decision rule for accepting or rejecting a hypothesis on the basis of evidence. A *p*-value test rejects a hypothesis whenever the value of the statistic $f(x)$ is smaller than a threshold. The statistic $f : \mathcal{X} \rightarrow [0, 1]$ is designed to have the property:

$$P_{\mu_0}(\{x \mid f(x) \leq \delta\}) = \delta.$$

If our decision rule is:

$$\pi(a | x) = \begin{cases} a_0, & f(x) \leq \delta \\ a_1, & f(x) > \delta, \end{cases}$$

the probability of rejecting the null hypothesis when it is true is exactly δ .

This is because, by definition, $f(x)$ has a uniform distribution under μ_0 . Hence the value of $f(x)$ itself is uninformative: high and low values are equally likely. In theory we should simply choose δ before seeing the data and just accept or reject based on whether $f(x) \leq \delta$. However nobody does that in practice, meaning that *p*-values are used incorrectly. Better not to use them at all, if uncertain about their meaning.

Issues with *p*-values

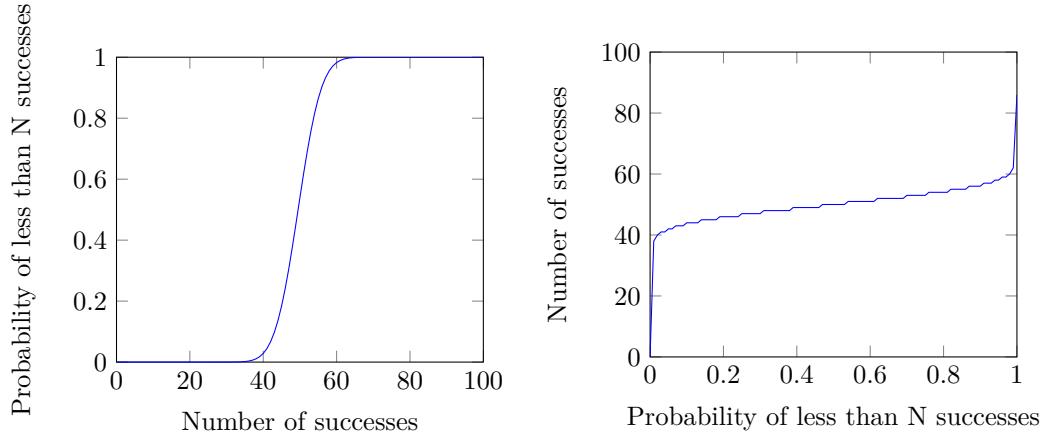
- They only measure quality of fit *on the data*.
- Not robust to model misspecification. For example, zero-mean testing using the χ^2 -test has a normality assumption.

- They ignore effect sizes. For example, a linear analysis may determine that there is a significant deviation from zero-mean, but with only a small effect size of 0.01. Thus, reporting only the p -value is misleading
- They do not consider prior information.
- They do not represent the probability of having made an error. In particular, a p -value of δ does not mean that the probability that the null hypothesis is false given the data x , is δ , i.e. $\delta \neq \mathbb{P}(\neg \mu_0 | x)$.
- The null-rejection error probability is the same irrespective of the amount of data (by design).

p-values for the medium example

Let us consider the example of the medium.

- μ_0 is simply the $Bernoulli(1/2)$ model: responses are by chance.
- CDF: $P_{\mu_0}(N \leq n | K = 100)$ is the probability of at most N successes if we throw the coin 100 times. This is in fact the cumulative probability function of the binomial distribution. Recall that the binomial represents the distribution for the number of successes of independent experiments, each following a Bernoulli distribution.
- ICDF: the number of successes that will happen with probability at least δ
- e.g. we'll get at most 50 successes a proportion $\delta = 1/2$ of the time.
- Using the (inverse) CDF we can construct a policy π that selects a_1 when μ_0 is true only a δ portion of the time, for any choice of δ .



Building a test

The test statistic

We want the test to reflect that we don't have a significant number of failures.

$$f(x) = 1 - \text{binocdf}\left(\sum_{t=1}^n x_t, n, 0.5\right)$$

What $f(x)$ is and is not

- It is a **statistic** which is $\leq \delta$ a δ portion of the time when μ_0 is true.
- It is **not** the probability of observing x under μ_0 .
- It is **not** the probability of μ_0 given x .

EXERCISE 6. • Let us throw a coin 8 times, and try and predict the outcome.

- Select a p -value threshold so that $\delta = 0.05$. For 8 throws, this corresponds to > 6 successes or $\geq 87.5\%$ success rate.
- Let's calculate the p -value for each one of you
- What is the rejection performance of the test?

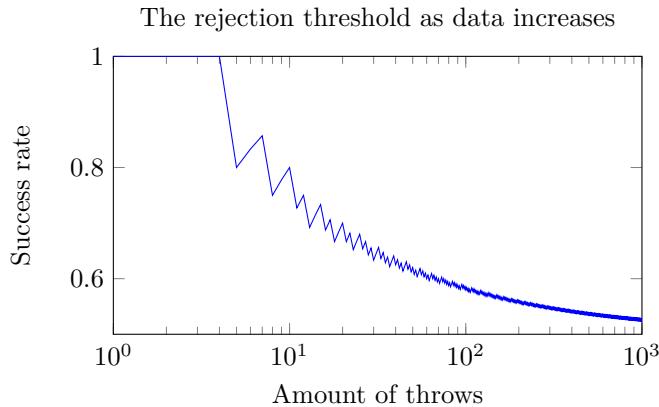


Figure 2.10: Here we see how the rejection threshold, in terms of the success rate, changes with the number of throws to achieve an error rate of $\delta = 0.05$.

As the amount of throws goes to infinity, the threshold converges to 0.5. This means that a statistically significant difference from the null hypothesis can be obtained, even when the actual model from which the data is drawn is only slightly different from 0.5.

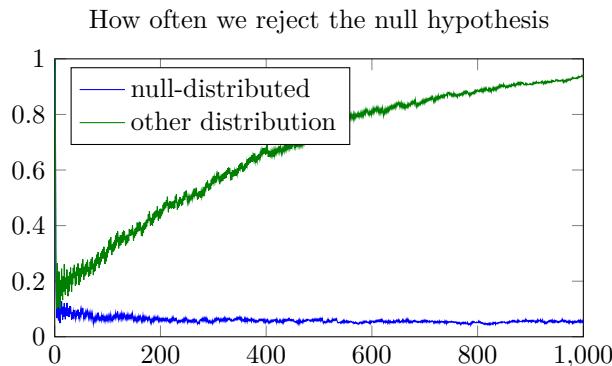


Figure 2.11: Here we see the rejection rate of the null hypothesis (μ_0) for two cases. Firstly, for the case when μ_0 is true. Secondly, when the data is generated from $Bernoulli(0.55)$.

As we see, this method keeps its promise: the null is only rejected 0.05 of the time when it's true. We can also examine how often the null is rejected when it is false... but what should we compare against? Here we are generating data from a $Bernoulli(0.55)$ model, and we can see the rejection of the null increases with the amount of data. This is called the *power* of the test with respect to the $Bernoulli(0.55)$ distribution.

Statistical power and false discovery.

Beyond not rejecting the null when it's true, we also want:

- High power: Rejecting the null when it is false.
- Low false discovery rate: Accepting the null when it is true.

Power

The power depends on what hypothesis we use as an alternative. This implies that we cannot simply consider a plain null hypothesis test, but must formulate a specific alternative hypothesis.

False discovery rate

False discovery depends on how likely it is *a priori* that the null is false. This implies that we need to consider a prior probability for the null hypothesis being true.

Both of these problems suggest that a Bayesian approach might be more suitable. Firstly, it allows us to consider an infinite number of possible alternative models as the alternative hypothesis, through Bayesian model averaging. Secondly, it allows us to specify prior probabilities for each alternative. This is especially important when we consider some effects unlikely.

The Bayesian version of the test

1. Set $U(a_i, \mu_j) = \mathbb{I}\{i = j\}$. This choice makes sense if we care equally about either type of error.
2. Set $\xi(\mu_i) = 1/2$. Here we place an equal probability in both models.
3. μ_0 : $Bernoulli(1/2)$. This is the same as the null hypothesis test.
4. μ_1 : $Bernoulli(\theta)$, $\theta \sim Unif([0, 1])$. This is an extension of the simple hypothesis test, with an alternative hypothesis that says “the data comes from an arbitrary Bernoulli model”.
5. Calculate $\xi(\mu | x)$.
6. Choose a_i , where $i = \arg \max_j \xi(\mu_j | x)$.

Bayesian model averaging for the alternative model μ_1

In this scenario, μ_0 is a simple point model, e.g. corresponding to a $Bernoulli(1/2)$. However μ_1 is a marginal distribution integrated over many models, e.g. a *Beta* distribution over Bernoulli parameters.

$$P_{\mu_1}(x) = \int_{\Theta} B_{\theta}(x) d\beta(\theta) \quad (2.4.3)$$

$$\xi(\mu_0 | x) = \frac{P_{\mu_0}(x)\xi(\mu_0)}{P_{\mu_0}(x)\xi(\mu_0) + P_{\mu_1}(x)\xi(\mu_1)} \quad (2.4.4)$$

Posterior probability of null hypothesis

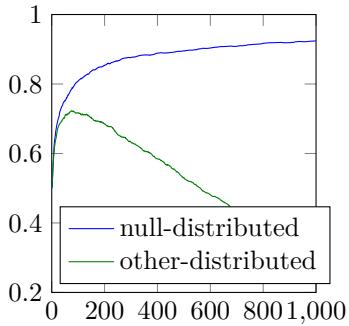
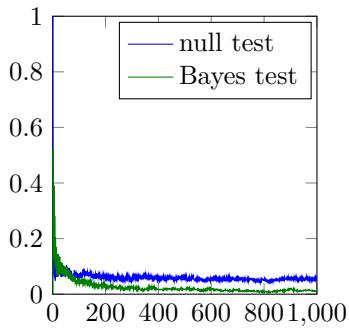


Figure 2.12: Here we see the convergence of the posterior probability.

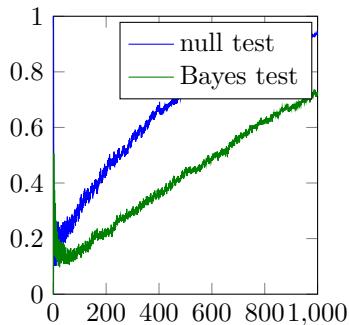
As can be seen in the figure above, in both cases, the posterior converges to the correct value, so it can be used to indicate our confidence that the null is true.

Rejection of null hypothesis for Bernoulli(0.5)

Figure 2.13: Comparison of the rejection probability for the null and the Bayesian test when μ_0 is true.

Now we can use this Bayesian test, with uniform prior, to see how well it performs. While the plain null hypothesis test has a fixed rejection rate of 0.05, the Bayesian test's rejection rate converges to 0 as we collect more data.

Rejection of null hypothesis for Bernoulli(0.55)

Figure 2.14: Comparison of the rejection probability for the null and the Bayesian test when μ_1 is true.

However, both methods are able to reject the null hypothesis more often when it is false, as long as we have more data.

Further reading

Points of significance (Nature Methods)

- Importance of being uncertain <https://www.nature.com/articles/nmeth.2613>
- Error bars <https://www.nature.com/articles/nmeth.2659>
- P values and the search for significance <https://www.nature.com/articles/nmeth.4120>

- Bayes' theorem <https://www.nature.com/articles/nmeth.3335>
- Sampling distributions and the bootstrap <https://www.nature.com/articles/nmeth.3414>

2.5 Formalising Classification problems

One of the simplest decision problems is classification. At the simplest level, this is the problem of observing some data point $x_t \in \mathcal{X}$ and making a decision about what class \mathcal{Y} it belongs to. Typically, a fixed classifier is defined as a decision rule $\pi(a|x)$ making decisions $a \in \mathcal{A}$, where the decision space includes the class labels, so that if we observe some point x_t and choose $a_t = 1$, we essentially declare that $y_t = 1$.

Typically, we wish to have a classification policy that minimises classification error.

Deciding a class given a model

In the simplest classification problem, we observe some features x_t and want to make a guess a_t about the true class label y_t . Assuming we have some probabilistic model $P_\mu(y_t | x_t)$, we want to define a decision rule $\pi(a_t | x_t)$ that is optimal, in the sense that it maximises expected utility for P_μ .

- Features $x_t \in \mathcal{X}$.
- Label $y_t \in \mathcal{Y}$.
- Decisions $a_t \in \mathcal{A}$.
- Decision rule $\pi(a_t | x_t)$ assigns probabilities to actions.

Standard classification problem

In the simplest case, the set of decisions we make are the same as the set of classes

$$\mathcal{A} = \mathcal{Y}, \quad U(a, y) = \mathbb{I}\{a = y\}$$

EXERCISE 7. If we have a model $P_\mu(y_t | x_t)$, and a suitable U , what is the optimal decision to make?

Deciding the class given a model family

- Training data $D_T = \{(x_i, y_i) \mid i = 1, \dots, T\}$
- Models $\{P_\mu \mid \mu \in \mathcal{M}\}$.
- Prior ξ on \mathcal{M} .

Similarly to our example with the meteorological stations, we can define a posterior distribution over models.

Posterior over classification models

$$\xi(\mu | D_T) = \frac{P_\mu(y_1, \dots, y_T | x_1, \dots, x_T)\xi(\mu)}{\sum_{\mu' \in \mathcal{M}} P_{\mu'}(y_1, \dots, y_T | x_1, \dots, x_T)\xi(\mu')}$$

This posterior form can be seen as weighing each model according to how well they can predict the class labels. It is a correct form as long as, for every pair of models μ, μ' we have that $P_\mu(x_1, \dots, x_T) = P_{\mu'}(x_1, \dots, x_T)$. This assumption can be easily satisfied without specifying a particular model for the x . If not dealing with time-series data, we assume independence between x_t :

$$P_\mu(y_1, \dots, y_T | x_1, \dots, x_T) = \prod_{i=1}^T P_\mu(y_i | x_i)$$

The *Bayes rule* for maximising $\mathbb{E}_\xi(U | a, x_t, D_T)$

The decision rule simply chooses the action:

$$a_t \in \arg \max_{a \in \mathcal{A}} \sum_y \sum_{\mu \in \mathcal{M}} P_\mu(y_t = y | x_t)\xi(\mu | D_T)U(a, y) \quad (2.5.1)$$

$$= \arg \max_{a \in \mathcal{A}} \sum_y \mathbb{P}_{\xi|D_T}(y_t | x_t)U(a, y) \quad (2.5.2)$$

We can rewrite this by calculating the posterior marginal label probability

$$\mathbb{P}_{\xi|D_T}(y_t | x_t) \triangleq \mathbb{P}_\xi(y_t | x_t, D_T) = \sum_{\mu \in \mathcal{M}} P_\mu(y_t | x_t)\xi(\mu | D_T).$$

Approximating the model

Full Bayesian approach for infinite \mathcal{M}

Here ξ can be a probability density function and

$$\xi(\mu | D_T) = P_\mu(D_T)\xi(\mu) / \mathbb{P}_\xi(D_T), \quad \mathbb{P}_\xi(D_T) = \int_{\mathcal{M}} P_\mu(D_T)\xi(\mu) d,$$

can be hard to calculate.

Maximum a posteriori model

We only choose a single model through the following optimisation:

$$\mu_{MAP}(\xi, D_T) = \arg \max_{\mu \in \mathcal{M}} P_\mu(D_T)\xi(\mu) = \arg \max_{\mu \in \mathcal{M}} \overbrace{\ln P_\mu(D_T)}^{\text{goodness of fit}} + \underbrace{\ln \xi(\mu)}_{\text{regulariser}}.$$

You can think of the goodness of fit as how well the model fits the training data, while the regulariser term simply weighs models according to some criterion. Typically, lower weights are used for more complex models.

Learning outcomes

Understanding

- Preferences, utilities and the expected utility principle.
- Hypothesis testing and classification as decision problems.
- How to interpret p -values Bayesian tests.
- The MAP approximation to full Bayesian inference.

Skills

- Being able to implement an optimal decision rule for a given utility and probability.
- Being able to construct a simple null hypothesis test.

Reflection

- When would expected utility maximisation not be a good idea?
- What does a p value represent when you see it in a paper?
- Can we prevent high false discovery rates when using p values?
- When is the MAP approximation good?

2.6 Classification with stochastic gradient descent

Classification as an optimisation problem.

Finding the optimal policy for our belief ξ is not normally very difficult. However, it requires that we maintain the complete distribution ξ and that we also under some probability distribution P . In simple decision problems, e.g. where the set of actions \mathcal{A} is finite, it is possible to do this calculation on-the-fly. However, in some cases we might not have a model.

Recall that we wish to maximise expected utility for some policy under some distribution. In general, this has the form

$$\max_{\pi} \mathbb{E}_{\mu}^{\pi}(U).$$

We also know that any expectation can be approximated by sampling. Let $P_{\mu}(\omega)$ be the distribution on outcomes defined by our model. Then

$$\mathbb{E}_{\mu}^{\pi}(U) = \sum_{\omega} U(a, \omega) P_{\mu}(\omega) \approx T^{-1} \sum_{t=1}^T U(a, \omega_t), \quad \omega_t \sim P_{\mu}(\omega),$$

i.e. when we can replace the explicit summation over all possible outcomes, weighed by their probability through averaging over T outcomes sampled from the correct distribution. In fact this approximation is *unbiased*, as its expectation is equal to the expected utility.

The μ -optimal classifier

Since the performance measure is simply an expectation, it is intuitive to directly optimise the decision rule with respect to an approximation of the expectation

$$\max_{\theta \in \Theta} f(\pi_{\theta}, \mu, U), \quad f(\pi_{\theta}, \mu, U) \triangleq \mathbb{E}_{\mu}^{\pi_{\theta}}(U) \quad (2.6.1)$$

$$f(\pi_{\theta}, \mu, U) = \sum_{x, y, a} U(a, y) \pi_{\theta}(a | x) P_{\mu}(y | x) P_{\mu}(x) \quad (2.6.2)$$

$$\approx \sum_{t=1}^T \sum_{a_t} U(a_t, y_t) \pi_{\theta}(a_t | x_t), \quad (x_t, y_t)_{t=1}^T \sim P_{\mu}. \quad (2.6.3)$$

In practice, this is the empirical expectation on the training set $\{(x_t, y_t) \mid t = 1, \dots, T\}$. However, when the amount of data is insufficient, this expectation may be far from reality, and so our classification rule might be far from optimal.

The Bayes-optimal classifier

An alternative idea is to use our uncertainty to create a distribution over models, and then use this distribution to obtain a single classifier that does take the uncertainty into account.

$$\max_{\theta} f(\pi_{\theta}, \xi) \approx \max_{\theta} N^{-1} \sum_{n=1}^N \pi(a_t = y_n \mid x_t = x_n), \quad (x_n, y_n) \sim P_{\mu_n}, \mu_n \sim \xi.$$

In this case, the integrals are replaced by sampling models μ_n from the belief, and then sampling (x_n, y_n) pairs from P_{μ_n} .

Stochastic gradient methods

To find the maximum of a differentiable function g , we can use gradient descent

Gradient ascent

$$\theta_{i+1} = \theta_i + \alpha \nabla_\theta g(\theta_i).$$

When f is an expectation, we don't need to calculate the full gradient. In fact, we only need to take one sample from the related distribution.

Stochastic gradient ascent

$$g(\theta) = \int_{\mathcal{M}} f(\theta, \mu) d\xi(\mu)$$

$$\theta_{i+1} = \theta_i + \alpha \nabla_\theta f(\theta_i, \mu_i), \quad \mu_i \sim \xi.$$

Stochastic gradient methods are commonly employed in neural networks.

2.6.1 Neural network models

Two views of neural networks

In the simplest sense a neural network is simply as parametrised functions f_θ . In classification, neural networks can be used as probabilistic models, so they describes the probability $P_\theta(y|\mathbf{x})$, or as classification policies so that $f_\theta(x, a)$ describes the probability $\pi_\theta(a | x)$ of selecting class label a . Let us begin by describing the simplest type of neural network model, the perceptron.

Neural network classification model $P_\theta(\mathbf{y} | \mathbf{x}_t)$



Objective: Find the best model for D_T .

Neural network classification policy $\pi(a_t | \mathbf{x}_t)$



Objective: Find the best policy for $U(a, \mathbf{x})$.

Difference between the two views

- We can use standard probabilistic methods for P .

- Finding the optimal π is an optimisation problem. However, estimating P can also be formulated as an optimisation.

Linear networks and the perceptron algorithm

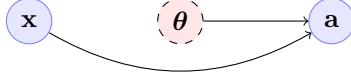


Figure 2.15: Abstract graphical model for a neural network

A neural network as used for modelling classification or regression problems, is simply a parametrised mapping $\mathcal{X} \rightarrow \mathcal{Y}$. If we include the network parameters, then it is instead a mapping $\mathcal{X} \times \Theta \rightarrow \mathcal{Y}$, as seen in Figure 2.17.

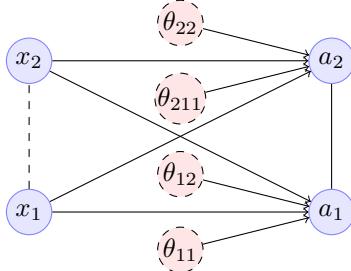


Figure 2.16: Graphical model for a linear neural network.

If we see each possible output as a different random variable, this creates a dependence. After all, we are really splitting one variable into many. In particular, if the network's output is the probability of each action, then we must make sure these sum to 1.

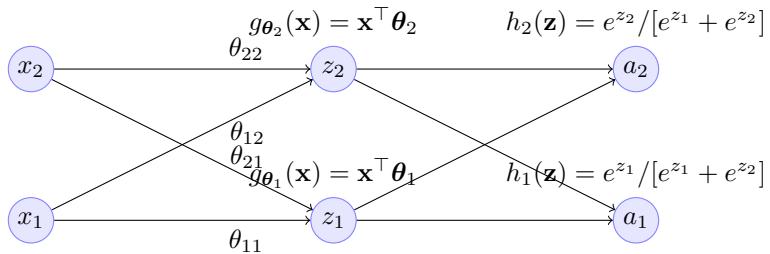


Figure 2.17: Architectural view of a linear neural network.

Definition 2.6.1 (Linear classifier). A linear classifier with N inputs and C outputs is parametrised by

$$\boldsymbol{\Theta} = [\boldsymbol{\theta}_1 \quad \dots \quad \boldsymbol{\theta}_C] = \begin{bmatrix} \theta_{1,1} & \dots & \theta_{1,C} \\ \vdots & \ddots & \vdots \\ \theta_{N,1} & \dots & \theta_{N,C} \end{bmatrix}$$

$$\pi_{\theta}(a | \mathbf{x}) = \exp(\boldsymbol{\theta}_a^\top \mathbf{x}) / \sum_{a'} \exp(\boldsymbol{\theta}_{a'}^\top \mathbf{x})$$

Even though the classifier has a linear structure, the final non-linearity at the end is there to ensure that it defines a proper probability distribution over decisions. For classification problems, the observations \mathbf{x}_t are features $\mathbf{x}_t = (x_{t,1}, \dots, x_{t,n})$ so that $\mathcal{X} \subset \mathbb{R}^N$. It is convenient to consider the network output as a vector on the simplex $\mathbf{y} \in \Delta^A$, i.e. $\sum_{i=1}^C y_{t,i} = 1$, $y_{t,i} \geq 0$. In the neural network model for classification, we typically ignore dependencies between the $x_{t,i}$ features, as we are not very interested in the distribution of \mathbf{x} itself.

Gradient ascent for a matrix U

$$\begin{aligned} & \max_{\theta} \sum_{t=1}^T \sum_{a_t} U(a_t, y_t) \pi_{\theta}(a_t | x_t) && \text{(objective)} \\ & \nabla_{\theta} \sum_{t=1}^T \sum_{a_t} U(a_t, y_t) \pi_{\theta}(a_t | x_t) && \text{(gradient)} \\ & = \sum_{t=1}^T \sum_{a_t} U(a_t, y_t) \nabla_{\theta} \pi_{\theta}(a_t | x_t) && (2.6.4) \end{aligned}$$

We now need to calculate the gradient of the policy.

Chain Rule of Differentiation

$$\begin{aligned} f(z), z = g(x), \quad & \frac{df}{dx} = \frac{df}{dg} \frac{dg}{dx} && \text{(scalar version)} \\ \nabla_{\theta} \pi = \nabla_g \pi \nabla_{\theta} g & && \text{(vector version)} \end{aligned}$$

Learning outcomes

Understanding

- Classification as an optimisation problem.
- (Stochastic) gradient methods and the chain rule.
- Neural networks as probability models or classification policies.
- Linear neural networks.
- Nonlinear network architectures.

Skills

- Using a standard NN class in python.

Reflection

- How useful is the ability to have multiple non-linear layers in a neural network.
- How rich is the representational power of neural networks?
- Is there anything special about neural networks other than their allusions to biology?

2.7 Naive Bayes classifiers

One special case of this idea is in classification, when each hypothesis corresponds to a specific class. Then, given a new example vector of data \mathbf{x} , we would like to calculate the probability of different classes C given the data, $\mathbb{P}(C | \mathbf{x})$. So here, the class is the hypothesis.

From Bayes's theorem, we see that we can write this as

$$\mathbb{P}(C | \mathbf{x}) = \frac{\mathbb{P}(\mathbf{x} | C) \mathbb{P}(C)}{\sum_i \mathbb{P}(\mathbf{x} | C_i) \mathbb{P}(C_i)}$$

for any class C . This directly gives us a method for classifying new data, as long as we have a way to obtain $\mathbb{P}(\mathbf{x} | C)$ and $\mathbb{P}(C)$.

But should we use for the probability model \mathbb{P} ?

Naive Bayes classifier

Naive Bayes classifiers are one of the simplest classification methods. They can have a full Bayesian interpretation under some assumptions, but otherwise they are too simplistic to be useful.

Calculating the prior probability of classes

A simple method is to simply count the number of times each class appears in the training data $D_T = ((x_t, y_t))_{t=1}^T$. Then we can set

$$\mathbb{P}(C) = 1/T \sum_{t=1}^T \mathbb{I}\{y_t = C\}$$

The Naive Bayes classifier uses the following model for observations, where observations are independent of each other given the class. Thus, for example the result of three different tests for lung cancer (stethoscope, radiography and biopsy) only depend on whether you have cancer, and not on each other.

Probability model for observations

$$\mathbb{P}(\mathbf{x} | C) = \mathbb{P}(x(1), \dots, x(n) | C) = \prod_{k=1}^n \mathbb{P}(x(k) | C).$$

There are two different types of models we can have, one of which is mostly useful for continuous attributes and the other for discrete attributes. In the first, we just need to count the number of times each feature takes different values in different classes.

Discrete attribute model.

Here we simply count the average number of times that the attribute k had the value i when the label was C . This is in fact analogous to the conditional probability definition.

$$\mathbb{P}(x(k) = i | C) = \frac{\sum_{t=1}^T \mathbb{I}\{x_t(k) = i \wedge y_t = C\}}{\sum_{t=1}^T \mathbb{I}\{y_t = C\}} = \frac{N_k(i, C)}{N(C)},$$

where $N_k(i, C)$ is the number of examples in class C whose k -th attribute has the value i , and $N(C)$ is the number of examples in class C .

Full Bayesian approach versus maximum likelihood

This estimation is simple maximum likelihood, as it does not maintain a distribution over the parameters.

Sometimes we need to be able to deal with cases where there are no examples at all of one class. In that case, that class would have probability zero. To get around this problem, we add “fake observations” to our data. This is called *Laplace smoothing*.

Remark 2.7.1. In Laplace smoothing with constant λ , our probability model is

$$\mathbb{P}(x(k) = i \mid C) = \frac{\sum_{t=1}^T \mathbb{I}\{x_t(k) = i \wedge y_t = C\} + \lambda}{\sum_{t=1}^T \mathbb{I}\{y_t = C\} + n_k \lambda} = \frac{N_k(i, C) + \lambda}{N(C) + n_k \lambda}.$$

where n_k is the number of values that the k -th attribute can take. This is necessary, because we want $\sum_{i=1}^{n_k} \mathbb{P}(x(k) = i \mid C) = 1$. (You can check that this is indeed the case as a simple exercise).

Remark 2.7.2. In fact, the Laplace smoothing model corresponds to a so-called Dirichlet prior over polynomial parameters with a marginal probability of observation equal to the Laplace smoothing. This is an extension of Beta-Bernoulli example from binary outcomes to multiple outcomes.

Continuous attribute model.

Here we can use a Gaussian model for each continuous dimension.

$$\mathbb{P}(x(k) = v \mid C) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(v-\mu)^2}{\sigma^2}},$$

where μ and σ are the mean and variance of the Gaussian, typically calculated from the training data as:

$$\mu = \frac{\sum_{t=1}^T x_t(k) \mathbb{I}\{y_t = C\}}{\sum_{t=1}^T \mathbb{I}\{y_t = C\}},$$

i.e. μ is the mean of the k -th attribute when the label is C and

$$\sigma = \sqrt{\frac{\sum_{t=1}^T [x_t(k) - \mu]^2 \mathbb{I}\{y_t = C\}}{\sum_{t=1}^T \mathbb{I}\{y_t = C\}}},$$

i.e. σ is the variance of the k -th attribute when the label is C . Sometimes we can just fix σ to a constant value, i.e. $\sigma = 1$.

Full Bayesian approach

This estimation is simple maximum likelihood, as it selects a single parameter pair $\boldsymbol{\mu} = (\mu_1, \dots, \mu_n)$ and $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_n)$ for every class and does not maintain a distribution over the parameters. It also assumes independence between the features. The full Bayesian approach considers an arbitrary covariance matrix $\boldsymbol{\Sigma}$ and maintains a distribution $\xi(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Chapter 3

Privacy

Participating in a study always carries a risk for individuals, namely that of data disclosure. In this chapter, we first explain how simple database query methods, and show even a small number of queries to a database they can compromise the privacy of individuals. We then introduce to formal concepts of privacy protection: k -anonymity and differential privacy. The first is relatively simple to apply and provides some limited resistance to identification of individuals through record linkage attacks. The latter is a more general concept, and can be simple apply in some settings, while it offers information-theoretic protection to individuals. A major problem with any privacy definition and method, however is correct interpretation of the privacy concept used, and correct implementation of the algorithm used.

3.1 Database access models

Databases

ID	Name	Salary	Deposits	Age	Postcode	Profession
1959060783	Mike Pence	150,000	1e6	60	1001	Politician
1946061408	Donald Trump	300,000	-1e9	72	1001	Rentier
2100010101	A. B. Student	10,000	100,000	40	1001	Time Traveller

EXAMPLE 11 (Typical relational database in a tax office).

Database access

- When owning the database: Direct look-up.
- When accessing a server etc: Query model.

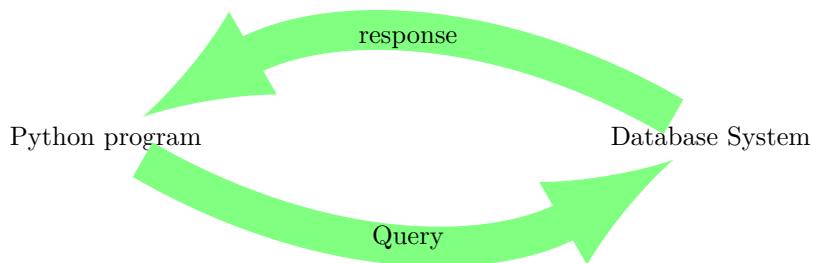


Figure 3.1: Database access model

Queries in SQL

The SELECT statement

- `SELECT column1, column2 FROM table;` This selects only some columns from the table
- `SELECT * FROM table;` This selects all the columns from the table

Selecting rows

```
SELECT * FROM table WHERE column = value;
```

Arithmetic queries

Here are some example SQL statements

- `SELECT COUNT(column) FROM table WHERE condition;` This allows you to count the number of rows matching `condition`
- `SELECT AVG(column) FROM table WHERE condition;` This lets you to count the number of rows matching `condition`
- `SELECT SUM(column) FROM table WHERE condition;` This is used to sum up the values in a column.

3.2 Privacy in databases

Anonymisation

If we wish to publish a database, frequently we need to protect identities of people involved. The simplest method for doing that is simply erasing directly identifying information. However, this does not really work most of the time, especially since attackers can have side-information that can reveal the identities of individuals in the original data.

Birthday	Name	Height	Weight	Age	Postcode	Profession
06/07	Li Pu	190	80	60-70	1001	Politician
06/14	Sara Lee	185	110	70+	1001	Rentier
01/01	A. B. Student	170	70	40-60	6732	Time Traveller

EXAMPLE 12 (Typical relational database in Tinder).

The simple act of hiding or using random identifiers is called anonymisation. However this is generally insufficient as other identifying information may be used to re-identify individuals in the data.

Record linkage

In particular, anonymisation is not enough as record linkage can allow you to still extract information using data from another database through *quasi-identifiers*.

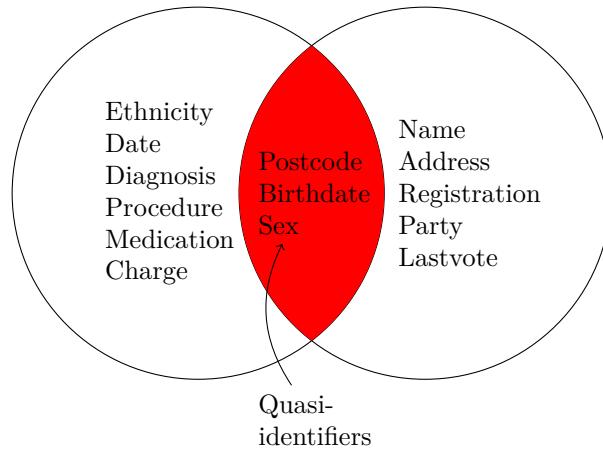


Figure 3.2: An example of two datasets, one containing sensitive and the other public information. The two datasets can be linked and individuals identified through the use of quasi-identifiers.

ID	Name	Salary	Deposits	Age	Postcode	Profession
1959060783	Li Pu	150,000	1e6	60	1001	Politician
1946061408	Sara Lee	300,000	-1e9	72	1001	Rentier
2100010101	A. B. Student	10,000	100,000	40	6732	Time Traveller

EXAMPLE 13 (Typical relational database in a tax office).

Birthday	Name	Height	Weight	Age	Postcode	Profession
06/07		190	80	60-70	1001	Politician
06/14		185	110	70+	1001	Rentier
01/01		170	70	40-60	6732	Time Traveller

EXAMPLE 14 (Typical relational database in Tinder).

3.3 k -anonymity

k -anonymity



(a) Samarati

(b) Sweeney

The concept of k -anonymity was introduced by Samarati and Sweeney⁴ and provides good guarantees when accessing a single database

Definition 3.3.1 (k -anonymity). A database provides k -anonymity if for every person in the database is indistinguishable from $k - 1$ persons with respect to *quasi-identifiers*.

It's the analyst's job to define quasi-identifiers

Birthday	Name	Height	Weight	Age	Postcode	Profession
06/07	Li Pu	190	80	60+	1001	Politician
06/14	Sara Lee	185	110	60+	1001	Rentier
06/12	Nikos Papadopoulos	170	82	60+	1243	Politician
01/01	A. B. Student	170	70	40-60	6732	Time Traveller
05/08	Li Yang	175	72	30-40	6910	Time Traveller

Table 3.1: 1-anonymity.

Birthday	Name	Height	Weight	Age	Postcode	Profession
		180-190	80+	60+	1*	
		180-190	80+	60+	1*	
		170-180	60-80	69+	1*	
		170-180	60-80	20-60	6*	
		170-180	60-80	20-60	6*	

Table 3.2: 2-anonymity: the database can be partitioned in sets of at least 2 records

However, with enough information, somebody may still be able to infer something about the individuals

3.4 Differential privacy

While k -anonymity can protect against specific re-identification attacks when used with care, it says little about what to do when the adversary has a lot of power. For example, if the adversary knows the data of everybody that has participated in the database, it is trivial for them to infer what our own data is. Differential privacy offers protection against adversaries with unlimited side-information or computational power. Informally, an algorithmic computation is differentially-private if an adversary cannot distinguish two similar database based on the result of the computation. While the notion of similarity is for the analyst to define, it is common to say that two databases are similar when they are identical apart from the data of one person.

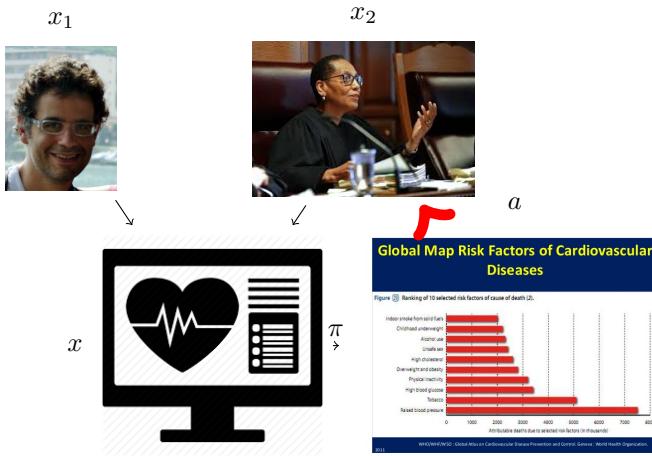


Figure 3.3: If two people contribute their data $x = (x_1, x_2)$ to a medical database, and an algorithm π computes some public output a from x , then it should be hard to infer anything about the data from the public output.

Privacy desiderata

Consider a scenario where n persons give their data x_1, \dots, x_n to an analyst. This analyst then performs some calculation $f(x)$ on the data and published the result. The following properties are desirable from a general standpoint.

Anonymity. Individual participation in the study remains a secret. From the release of the calculations results, nobody can significantly increase their probability of identifying an individual in the database.

Secrecy. The data of individuals is not revealed. The release does not significantly increase the probability of inferring individual's information x_i .

Side-information. Even if an adversary has arbitrary side-information, he cannot use that to amplify the amount of knowledge he would have obtained from the release.

Utility. The released result has, with high probability, only a small error relative to a calculation that does not attempt to safeguard privacy.

Example: The prevalence of drug use in sport

Let's say you need to perform a statistical analysis of the drug-use habits of athletes. Obviously, even if you promise the athlete not to reveal their information, you still might not convince them. Yet, you'd like them to be truthful. The trick is to allow them to randomly change their answers, so that you can't be *sure* if they take drugs, no matter what they answer.

Algorithm for randomising responses about drug use

1. Flip a coin.
2. If it comes heads, respond truthfully.
3. Otherwise, flip another coin and respond **yes** if it comes heads and **no** otherwise.

EXERCISE 8. Assume that the observed rate of positive responses in a sample is p , that everybody follows the protocol, and the coin is fair. Then, what is the true rate q of drug use in the population?

Solution. Since the responses are random, we will deal with expectations first

$$\begin{aligned}\mathbb{E} p &= \frac{1}{2} \times \frac{1}{2} + q \times \frac{1}{2} = \frac{1}{4} + \frac{q}{2} \\ q &= 2\mathbb{E} p - \frac{1}{2}.\end{aligned}$$

□

The problem with this approach, of course, is that we are effectively throwing away half of our data. In particular, if we repeated the experiment with a coin that came heads at a rate ϵ , then our error bounds would scale as $O(1/\sqrt{\epsilon n})$ for n data points.

The randomised response mechanism

The above idea can be generalised. Consider we have data x_1, \dots, x_n from n users and we transform it randomly to y_1, \dots, y_n using the following mapping.

Definition 3.4.1 (Randomised response). The i -th user, whose data is $x_i \in \{0, 1\}$, responds with $a_i \in \{0, 1\}$ with probability

$$\pi(a_i = j \mid x_i = k) = p, \quad \pi(a_i = k \mid x_i = k) = 1 - p,$$

where $j \neq k$.

Given the complete data x , the mechanism's output is $a = (a_1, \dots, a_n)$. Since the algorithm independently calculates a new value for each data entry, the output is

$$\pi(a \mid x) = \prod_i \pi(a_i \mid x_i)$$

This mechanism satisfies so-called ϵ -differential privacy, which we will define later.

EXERCISE 9. Let the adversary have a prior $\xi(x = 0) = 1 - \xi(x = 1)$ over the values of the true response of an individual. we use the randomised response mechanism with p and the adversary observes the randomised data $a = 1$ for that individual, then what is $\xi(x = 1 \mid a = 1)$?

The local privacy model

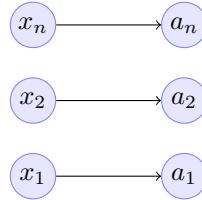


Figure 3.4: The local privacy model

In the local privacy model, the i -th individual's data x_i is used to generate a private response a_i . This means that no individual will provide their true data with certainty. This model allows us to publish a complete dataset of private responses.

Differential privacy.



Now let us take a look at a way to characterise the inherent privacy properties of algorithms. This is called differential privacy, and it can be seen as a bound on the information an adversary with arbitrary power or side-information could extract from the result of a computation π on the data. For reasons that will be made clear later, this computation has to be stochastic.

Definition 3.4.2 (ϵ -Differential Privacy). A stochastic algorithm $\pi : \mathcal{X} \rightarrow \mathcal{A}$, where \mathcal{X} is endowed with a neighbourhood relation N , is said to be ϵ -differentially private if

$$\left| \ln \frac{\pi(a | x)}{\pi(a | x')} \right| \leq \epsilon, \quad \forall x N x'. \quad (3.4.1)$$

Typically, algorithms are applied to datasets $x = (x_1, \dots, x_n)$ composed of the data of n individuals. Thus, all privacy guarantees relate to the data contributed by these individuals.

In this context, two datasets are usually called neighbouring if $x = (x_1, \dots, x_{i-1}, x_i, x_{i+1} \dots, x_n)$ and $x' = (x_1, \dots, x_{i-1}, x_i, x_{i+1} \dots, x_n)$, i.e. if one dataset is missing an element.

A slightly weaker definition of neighbourhood is to say that $x N x'$ if $x' = (x_1, \dots, x_{i-1}, x'_i, x_{i+1} \dots, x_n)$, i.e. if one dataset has an altered element. We will usually employ this latter definition, especially for the local privacy model.

The definition of differential privacy

- First rigorous mathematical definition of privacy.

- Relaxations and generalisations possible.
- Connection to learning theory and reproducibility.

Current uses

- Apple. DP is used internally in the company to “protect user privacy”. It is not clear exactly what they are doing but their efforts seem to be going in the right direction.
- Google. The company has a DP API available based on randomised response, RAPPOR.
- Uber. Elastic sensitivity for SQL queries, which is available as open source. This is a good thing, because it is easy to get things wrong with privacy.
- US 2020 Census. It uses differential privacy to protect the confidentiality of responders’ information while maintaining data that are suitable for their intended uses.

Open problems

- Complexity of differential privacy.
- Verification of implementations and queries.

Remark 3.4.1. Any differentially private algorithm must be stochastic.

To prove that this is necessary, consider the example of counting how many people take drugs in a competition. If the adversary only doesn’t know whether you in particular take drugs, but knows whether everybody else takes drugs, it’s trivial to discover your own drug habits by looking at the total. This is because in this case, $f(x) = \sum_i x_i$ and the adversary knows x_i for all $i \neq j$. Then, by observing $f(x)$, he can recover $x_j = f(x) - \sum_{i \neq j} x_i$. Consequently, it is not possible to protect against adversaries with arbitrary side information without stochasticity.

Remark 3.4.2. The randomised response mechanism with $p \leq 1/2$ is $(\ln \frac{1-p}{p})$ -DP.

Proof. Consider $x = (x_1, \dots, x_j, \dots, x_n)$, $x' = (x_1, \dots, x'_j, \dots, x_n)$. Then

$$\begin{aligned}\pi(a | x) &= \prod_i \pi(a_i | x_i) \\ &= \pi(a_j | x_j) \prod_{i \neq j} \pi(a_i | x_i) \\ &\leq \frac{p}{1-p} \pi(a_j | x'_j) \prod_{i \neq j} \pi(a_i | x_i) \\ &= \frac{1-p}{p} \pi(a | x')\end{aligned}$$

$\pi(a_j = k \mid x_j = k) = 1 - p$ so the ratio is $\max\{(1 - p)/p, p/(1 - p)\} \leq (1 - p)/p$ for $p \leq 1/2$. \square

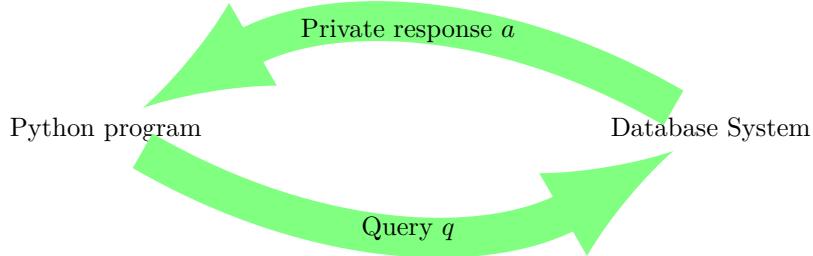


Figure 3.5: Private database access model

Response policy

The policy defines a distribution over responses a given the data x and the query q .

$$\pi(a \mid x, q)$$

Differentially private queries

There is no actual DP-SELECT statement, but we can imagine it.

The DP-SELECT statement

- DP-SELECT ϵ column1, column2 FROM table; This selects only some columns from the table
- DP-SELECT ϵ * FROM table; This selects all the columns from the table

Selecting rows

DP-SELECT ϵ * FROM table WHERE column = value;

Arithmetic queries

Here are some example SQL statements

- DP-SELECT ϵ COUNT(column) FROM table WHERE condition; This allows you to count the number of rows matching condition
- DP-SELECT ϵ AVG(column) FROM table WHERE condition; This lets you to count the number of rows matching condition

- DP-SELECT ϵ SUM(column) FROM table WHERE condition; This is used to sum up the values in a column.

Depending on the DP scheme, each query answered may leak privacy. In particular, if we always respond with an ϵ -DP mechanism, after T queries our privacy guarantee is $T\epsilon$. There exist mechanisms that do not respond to each query independently, which can bound the total privacy loss.

Definition 3.4.3 (T -fold adaptive composition). In this privacy model, an adversary is allowed to compose T queries. The composition is *adaptive*, in the sense that the next query is allowed to depend on the previous queries and their results.

Theorem 3.4.1. *For any $\epsilon > 0$, the class of ϵ -differentially private mechanism satisfy $T\epsilon$ -differential privacy under T -fold adaptive composition.*

EXERCISE 10. Adversary knowledge Assume that the adversary knows that the data is either \mathbf{x} or \mathbf{x}' . For concreteness, assume the data is either

$$\mathbf{x} = (x_1, \dots, x_j = 0, \dots, x_n)$$

where x_i indicates whether or not the i -th person takes drugs, or

$$\mathbf{x}' = (x_1, \dots, x_j = 1, \dots, x_n).$$

In other words, the adversary knows the data of all people apart from one, the j -th person. We can assume that the adversary has some prior belief

$$\xi(\mathbf{x}) = 1 - \xi(\mathbf{x}')$$

for the two cases. Assume the adversary knows the output a of a mechanism π . What can we say about the posterior distribution of the adversary $\xi(\mathbf{x} | a, \pi)$ after having seen the output, if π is ϵ -DP?

Solution

We can write the adversary posterior as follows.

$$\xi(\mathbf{x} | a, \pi) = \frac{\pi(a | \mathbf{x})\xi(\mathbf{x})}{\pi(a | \mathbf{x})\xi(\mathbf{x}) + \pi(a | \mathbf{x}')\xi(\mathbf{x}')} \quad (3.4.2)$$

$$\geq \frac{\pi(a | \mathbf{x})\xi(\mathbf{x})}{\pi(a | \mathbf{x})\xi(\mathbf{x}) + \pi(a | \mathbf{x})e^\epsilon\xi(\mathbf{x}')} \quad (\text{from DP definition})$$

$$= \frac{\xi(\mathbf{x})}{\xi(\mathbf{x}) + e^\epsilon\xi(\mathbf{x}')} \quad (3.4.3)$$

But this is not very informative. We can also write

$$\frac{\xi(\mathbf{x} | a, \pi)}{\xi(\mathbf{x}' | a, \pi)} = \frac{\pi(a | \mathbf{x})\xi(\mathbf{x})}{\pi(a | \mathbf{x}')\xi(\mathbf{x}')} \geq \frac{\pi(a | \mathbf{x})\xi(\mathbf{x})}{\pi(a | \mathbf{x})e^{-\epsilon}\xi(\mathbf{x}')} = \frac{\xi(\mathbf{x})}{\xi(\mathbf{x}')}e^\epsilon \quad (3.4.4)$$

Dealing with multiple attributes.

Up to now we have been discussing the case where each individual only has one attribute. However, in general each individual t contributes multiple data $x_{t,i}$, which can be considered as a row \mathbf{x}_t in a database. Then the mechanism can release each $a_{t,i}$ independently.

Independent release of multiple attributes.

For n users and k attributes, if the release of each attribute i is ϵ -DP then the data release is $k\epsilon$ -DP. Thus to get ϵ -DP overall, we need ϵ/k -DP per attribute.

The result follows immediately from the composition theorem. We can see each attribute release as the result of an individual query.

3.4.1 Other differentially private mechanisms

The Laplace mechanism.

A simple method to obtain a differentially private algorithm from a deterministic function $f : \mathcal{X} \rightarrow \mathbb{R}$, is to use additive noise, so that the output of the algorithm is simply

$$a = f(x) + \omega, \quad \omega \sim \text{Laplace}.$$

The amount of noise added, together with the smoothness of the function f , determine the amount of privacy we have.

Definition 3.4.4 (The Laplace mechanism). For any function $f : \mathcal{X} \rightarrow \mathbb{R}$,

$$\pi(a | x) = \text{Laplace}(f(x), \lambda), \quad (3.4.5)$$

where the Laplace density is defined as

$$p(\omega | \mu, \lambda) = \frac{1}{2\lambda} \exp\left(-\frac{|\omega - \mu|}{\lambda}\right).$$

and has mean μ and variance $2\lambda^2$.

Here, $\text{Laplace}(\mu, \lambda)$ is the density $f(x) = \frac{\lambda}{2} \exp(-\lambda|x - \mu|)$.

EXAMPLE 15 (Calculating the average salary). • The i -th person receives salary x_i

- We wish to calculate the average salary in a private manner.

Local privacy model

- Obtain $y_i = x_i + \omega$, where $\omega \sim \text{Laplace}(\lambda)$.
- Return $a = n^{-1} \sum_{i=1}^n y_i$.

Centralised privacy model

Return $a = n^{-1} \sum_{i=1}^n x_i + \omega$, where $\omega \sim \text{Laplace}(\lambda')$.

How should we add noise in order to guarantee privacy?

The centralised privacy model

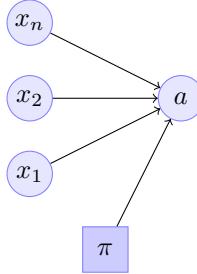


Figure 3.6: The centralised privacy model

Assumption 3.4.1. *The data x is collected and the result a is published by a trusted curator*

DP properties of the Laplace mechanism

Definition 3.4.5 (Sensitivity). The sensitivity of a function f is

$$\mathbb{L}(f) \triangleq \sup_{xNx'} |f(x) - f(x')|$$

If we define a metric d , so that $d(x, x') = 1$ for xNx' , then:

$$|f(x) - f(x')| \leq \mathbb{L}(f) d(x, x'),$$

i.e. f is $\mathbb{L}(f)$ -Lipschitz with respect to d .

EXAMPLE 16. If $f : \mathcal{X} \rightarrow [0, B]$, e.g. $\mathcal{X} = \mathbb{R}$ and $f(x) = \min\{B, \max\{0, x\}\}$, then $\mathbb{L}(f) = B$.

EXAMPLE 17. If $f : [0, B]^n \rightarrow [0, B]$ is $f = \frac{1}{n} \sum_{t=1}^n x_t$, then $\mathbb{L}(f) = B/n$.

Proof. Consider two neighbouring datasets x, x' differing in example j . Then

$$f(x) - f(x') = \frac{1}{n} [f(x_j) - f(x'_j)] \leq \frac{1}{n} [B - 0]$$

□

Theorem 3.4.2. *The Laplace mechanism on a function f with sensitivity $\mathbb{L}(f)$, ran with $\text{Laplace}(\lambda)$ is $\mathbb{L}(f)/\lambda$ -DP.*

Proof.

$$\frac{\pi(a | x)}{\pi(a | x')} = \frac{e^{|a-f(x')|/\lambda}}{e^{|a-f(x)|/\lambda}} \leq \frac{e^{|a-f(x)|/\lambda + \mathbb{L}(f)/\lambda}}{e^{|a-f(x)|/\lambda}} = e^{\mathbb{L}(f)/\lambda}$$

□

So we need to use $\lambda = \mathbb{L}(f)/\epsilon$ for ϵ -DP. What is the effect of applying the Laplace mechanism in the local versus centralised model? Here let us assume $x_i \in [0, B]$ for all i and consider the problem of calculating the average.

Laplace in the local privacy model

The sensitivity of the individual data is B , so to obtain ϵ -DP we need to use $\lambda = B/\epsilon$. The variance of each component is $2(M/\epsilon)^2$, so the total variance is $2M^2/\epsilon^2n$.

Laplace in the centralised privacy model

The sensitivity of f is M/n , so we only need to use $\lambda = \frac{M}{ne}$. The variance of a is $2(M/en)^2$.

Thus the two models have a significant difference in the variance of the estimates obtained, for the same amount of privacy. While the central mechanism has variance $O(n^{-2})$, the local one is $O(n^{-1})$ and so our estimates will need much more data to be accurate under this mechanism. In particular, we need square the amount of data in the local model as we need in the central model. Nevertheless, the local model may be the only possible route if we have no specific use for the data.

3.4.2 Utility of queries

Rather than saying that we wish to calculate a private version of some specific function f , sometimes it is more useful to consider the problem from the perspective of the utility of different answers to queries. More precisely, imagine the interaction between a database system and a user:

Interactive queries

- System has data x .
- User asks query q .
- System responds with a .
- There is a common utility function $U : \mathcal{X}, \mathcal{A}, \mathcal{Q} \rightarrow \mathbb{R}$.

We wish to maximise U with our answers, but are constrained by the fact that we also want to preserve privacy.

The utility $U(x, a, q)$ describes how appropriate each response a given by the system for a query r is given the data x . It can be seen as how useful the response is¹ It allows us to quantify exactly how much we would gain by replying correctly. The exponential mechanism, described below is a simple differentially private mechanism for responding to queries while trying to maximise utility for *any possible* utility function.

The Exponential Mechanism.

Here we assume that we can answer queries q , whereby each possible answer a to the query has a different utility to the DM: $U(q, a, x)$. Let $\mathbb{L}(U(q)) \triangleq \sup_{x \in \mathcal{X}} |U(q, a, x) - U(q, a, x')|$ denote the sensitivity of a query. Then the following mechanism is ϵ -differentially private.

¹This is essentially the utility to the user that asks the query, but it could be the utility to the person that answers. In either case, the motivation does not matter the action should maximise it, but is constrained by privacy.

Definition 3.4.6 (The Exponential mechanism). For any utility function $U : \mathcal{Q} \times \mathcal{A} \times \mathcal{X} \rightarrow \mathbb{R}$, define the policy

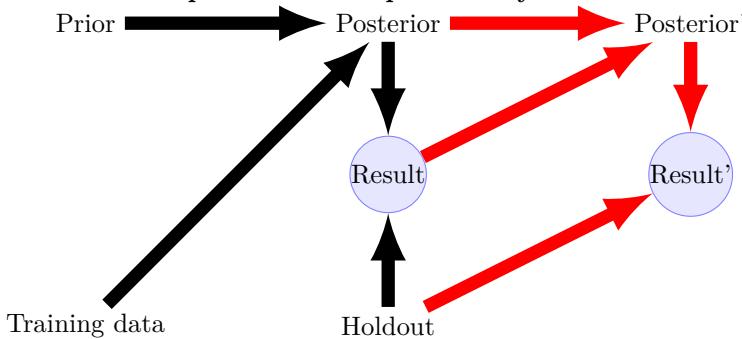
$$\pi(a | x) \triangleq \frac{e^{\epsilon U(q, a, x) / \mathbb{L}(U(q))}}{\sum_{a'} e^{\epsilon U(q, a', x) / \mathbb{L}(U(q))}} \quad (3.4.6)$$

Clearly, when $\epsilon \rightarrow 0$, this mechanism is uniformly random. When $\epsilon \rightarrow \infty$ the action maximising $U(q, a, x)$ is always chosen.

Although the exponential mechanism can be used to describe most known DP mechanisms, its best use is in settings where there is a natural utility function.

3.4.3 Privacy and reproducibility

The unfortunate practice of adaptive analysis



In the ideal data analysis,

we start from some prior hypothesis, then obtain some data, which we split into training and holdout. We then examine the training data and obtain a posterior that corresponds to our conclusions. We can then measure the quality of these conclusions in the independent holdout set.

However, this is not what happens in general. Analysts typically use the same holdout repeatedly, in order to improve the performance of their algorithms. This can be seen as indirectly using the holdout data to obtain a new posterior, and so it is possible that you can overfit on the holdout data, even if you never directly see it. It turns out we can solve this problem if we use differential privacy, so that the analyst only sees a differentially private version of queries.

The reusable holdout²²

One idea to solve this problem is to only allow the analyst to see a private version of the result. In particular, the analyst will only see whether or not the holdout result is τ -close to the training result.

Algorithm parameters

- Performance measure f .
- Threshold τ . How close do we want f to be on the training versus holdout set?
- Noise σ . How much noise should we add?
- Budget B . How much are we allowed to learn about the holdout set?

²²Also see <https://ai.googleblog.com/2015/08/the-reusable-holdout-preserving.html>

Algorithm idea

Run algorithm λ on data D_T and get e.g. classifier parameters θ .
 Run a DP version of the function $f(\theta, D_H) = \mathbb{I}\{U(\theta, D_T) \geq \tau U(\theta, D_H)\}$.

So instead of reporting the holdout performance at all, you just see if you are much worse than the training performance, i.e. if you're overfitting. The fact that the mechanism is DP also makes it difficult to learn the holdout set. See the thresholdout link for more details.

Available privacy toolboxes

k-anonymity

- <https://github.com/qiyuangong/Mondrian> Mondrian *k*-anonymity

Differential privacy

- <https://github.com/bmcmenamin/thresholdOut-explorations> Threshold out
- <https://github.com/steven7woo/Accuracy-First-Differential-Privacy> Accuracy-constrained DP
- <https://github.com/menisadi/pydp> Various DP algorithms
- <https://github.com/haiphanNJIT/PrivateDeepLearning> Deep learning and DP

Learning outcomes

Understanding

- Linkage attacks and *k*-anonymity.
- Inferring data from summary statistics.
- The local versus global differential privacy model.
- False discovery rates.

Skills

- Make a dataset satisfy k -anonymity with respect to identifying attributes.
- Apply the randomised response and Laplace mechanism to data.
- Apply the exponential mechanism to simple decision problems.
- Use differential privacy to improve reproducibility.

Reflection

- How can potentially identifying attributes be chosen to achieve k -anonymity?
- How should the parameters of the two ideas, ϵ -DP and k -anonymity be chosen?
- Does having more data available make it easier to achieve privacy?

Chapter 4

Fairness

When machine learning algorithms are applied at scale, it can be difficult to imagine



Economist.com

Figure 4.1: In some cases, it appears as though automating this procedure might lead to better outcomes. But is that generally true?

4.1 Fairness in machine learning

The problem of fairness in machine learning and artificial intelligence has only recently been widely recognised. When any algorithm is implemented at scale, no matter the original objective and whether it is satisfied, it has significant societal effects. In particular, even when considering the narrow objective of the algorithm, even if it improves it overall, it may increase inequality.

In this course we will look at two aspects of fairness. The first has to do with disadvantaged populations that form distinct social classes due to a shared income stratum, race or gender. The second has to do with meritocratic notions of fairness.

Bail decisions

For our example regarding disadvantaged populations, consider the example of bail decisions in the US court system. When a defendant is charged, the judge has the option to either place them in jail pending trial, or set them free, under the condition that the defendant pays some amount of bail. The amount of bail (if any) is set to an amount that would be expected to deter flight or a relapse.

Whites get lower scores than blacks¹

In a different study, it was shown that a commonly used software tool for determining 'risk scores' in the US was biased towards white defendants, who seemed to be always getting lower scores than blacks.

¹Pro-publica, 2016

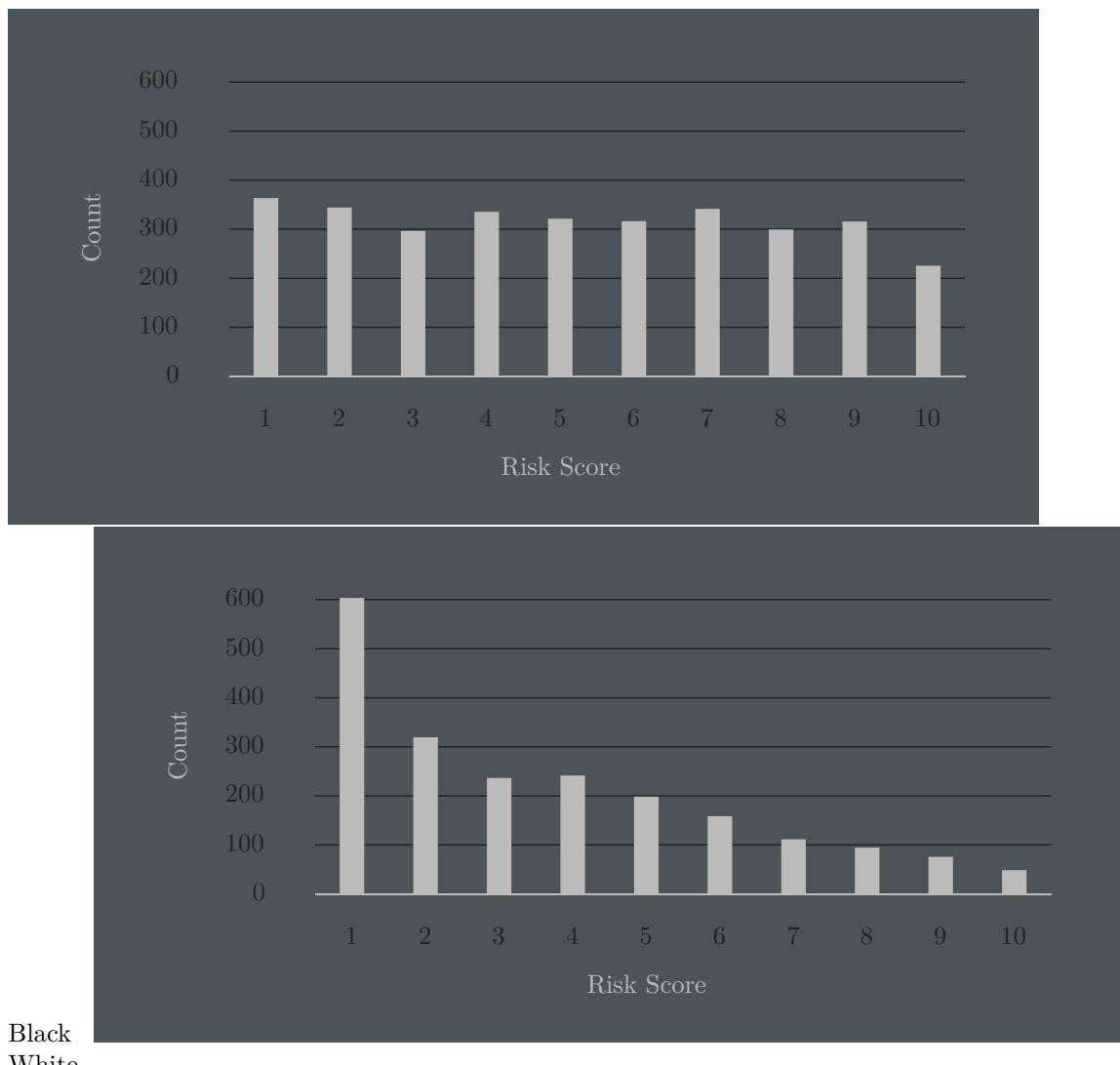


Figure 4.2: Apparent bias in risk scores towards black versus white defendants.

But scores equally accurately predict recidivism²

On the other hand, the scores generated by the software seemed to be very predictive on whether or not defendants would re-offend, independently of their race.

²Washington Post, 2016

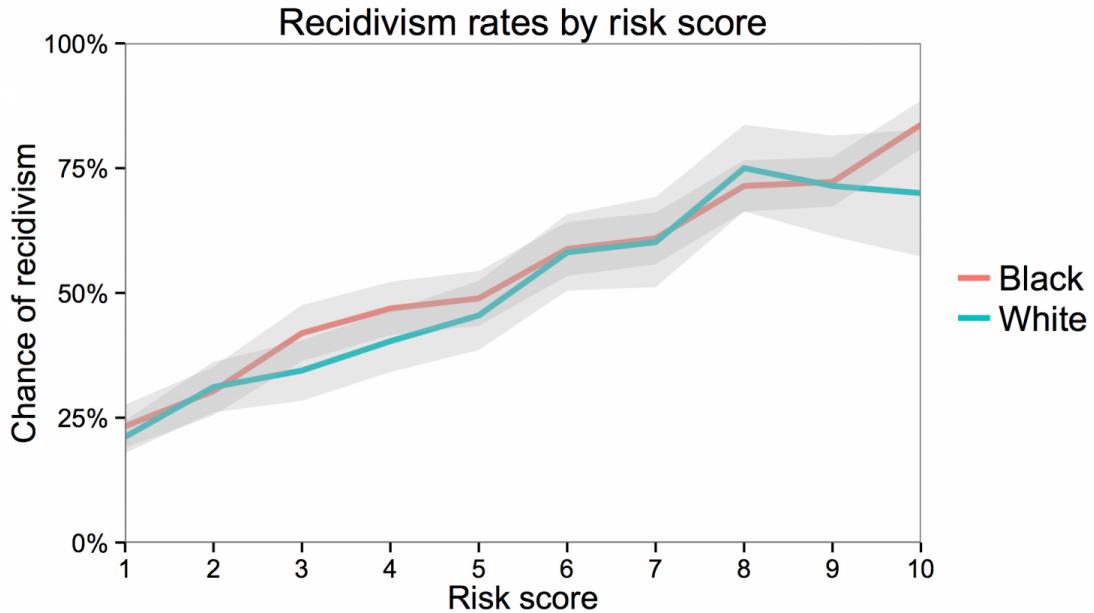


Figure 4.3: Recidivism rates by risk score.

But non-offending blacks get higher scores

On the third hand, we see that the system seemed to give higher risk scores to non-offending blacks. So, is there a way to fix that or not?

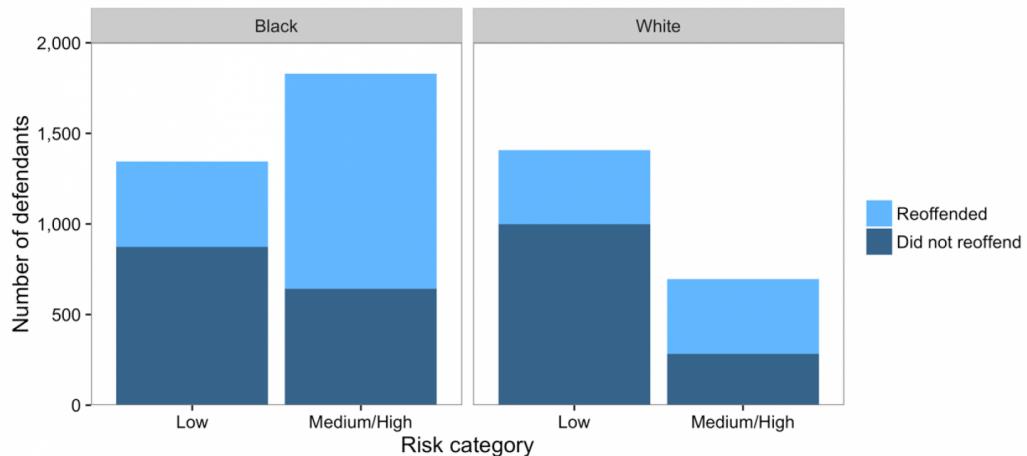


Figure 4.4: Score breakdown based on recidivism rates.

How can we explain this discrepancy? We can show that in fact, each one of these different measures of bias in our decision rules can be seen as a notion of conditional independence.

4.2 Graphical models

Graphical models are a very useful tool for modelling the relationship between multiple variables. The simplest such models, probabilistic graphical models (otherwise known as Bayesian networks) involve directed acyclic graphs between random variables.

Graphical models

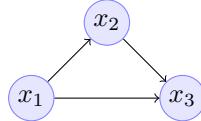


Figure 4.5: Graphical model for three variables.

Consider for example the model in Figure 117. It involves three variables, x_1, x_2, x_3 and there are three arrows, which show how one variable depends on another. Simply put, if you think of each x_k as a stochastic function, then x_k 's value only depends on the values of its parents, i.e. the nodes that are point to it. In this example, x_1 does not depend on any other variable, but the value of x_2 depends on the value of x_1 . Such models are useful when we want to describe the joint probability distribution of all the variables in the collection.

Joint probability

Let $\mathbf{x} = (x_1, \dots, x_n)$. Then $\mathbf{x} : \Omega \rightarrow X$, $X = \prod_i X_i$ and:

$$\mathbb{P}(\mathbf{x} \in A) = P(\{\omega \in \Omega \mid \mathbf{x}(\omega) \in A\}).$$

When X_i are finite, we can typically write

$$\mathbb{P}(\mathbf{x} = \mathbf{a}) = P(\{\omega \in \Omega \mid \mathbf{x}(\omega) = \mathbf{a}\}),$$

for the probability that $x_i = a_i$ for all $i \in [n]$.

Factorisation

For any subsets $B \subset [n]$ and its complement C so that $\mathbf{x}_B = (x_i)_{i \in B}$, $\mathbf{x}_C = (x_i)_{i \notin B}$

$$\mathbb{P}(\mathbf{x}) = \mathbb{P}(\mathbf{x}_B \mid \mathbf{x}_C) \mathbb{P}(\mathbf{x}_C)$$

So we can write any joint distribution as

$$\mathbb{P}(x_1) \mathbb{P}(x_2 \mid x_1) \mathbb{P}(x_3 \mid x_1, x_2) \cdots \mathbb{P}(x_n \mid x_1, \dots, x_{n-1}).$$

Although the above factorisation is always possible to do, sometimes our graphical model has a structure that makes the factors much simpler. In fact, the main reason for introducing graphical models is to represent dependencies between variables. For a given model, we can infer whether some variables are in fact dependent, independent, or conditionally independent.

Directed graphical models and conditional independence

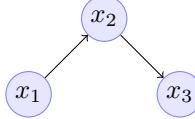


Figure 4.6: Graphical model for the factorisation $\mathbb{P}(x_1 | x_2) \mathbb{P}(x_2 | x_3) \mathbb{P}(x_3)$.

Conditional independence

We say x_i is conditionally independent of \mathbf{x}_B given \mathbf{x}_D and write $x_i | \mathbf{x}_D \perp\!\!\!\perp \mathbf{x}_B$ iff

$$\mathbb{P}(x_i, \mathbf{x}_B | \mathbf{x}_D) = \mathbb{P}(x_i | \mathbf{x}_D) \mathbb{P}(\mathbf{x}_D | \mathbf{x}_B).$$

Directed graphical models

A graphical model is a convenient way to represent conditional independence between variables. There are many variants of graphical models, whose name is context dependent. Other names used in the literature are probabilistic graphical models, Bayesian networks, causal graphs, or decision diagrams. In this set of notes we focus on directed graphical models that depict dependencies between random variables.

Definition 4.2.1 (Directed graphical model). A collection of n random variables $x_i : \Omega \rightarrow X_i$, and let $X \triangleq \prod_i X_i$, with underlying probability measure P on Ω . Let $\mathbf{x} = (x_i)_{i=1}^n$ and for any subset $B \subset [n]$ let

$$\mathbf{x}_B \triangleq (x_i)_{i \in B} \tag{4.2.1}$$

$$\mathbf{x}_{-j} \triangleq (x_i)_{i \neq j} \tag{4.2.2}$$

In a Graphical model, conditional independence is represented through directed edges.

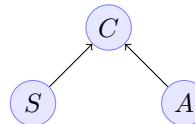


Figure 4.7: Smoking and lung cancer graphical model, where S : Smoking, C : cancer, A : asbestos exposure.

EXAMPLE 18 (Smoking and lung cancer). It has been found by ? that lung incidence not only increases with both asbestos exposure and smoking. This is in agreement with the graphical model shown. The study actually found that there is an amplification effect, whereby smoking and asbestos exposure increases cancer risk by 28 times compared to non-smokers. This implies that the risk is not simply additive. The graphical model only tells us that there is a dependency, and does not describe the nature of this dependency precisely.

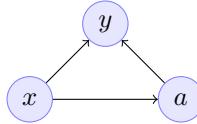


Figure 4.8: Kidney treatment model, where x : severity, y : result, a : treatment applied

EXAMPLE 19 (Treatment effects).

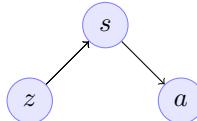


Figure 4.9: Simplified school admission graphical model, where z : gender, s : school applied to, a : whether you were admitted.

EXAMPLE 20 (School admission).

Deciding conditional independence

There is an algorithm for deciding conditional independence of any two variables in a graphical model. However, this is beyond the scope of these notes. Here, we shall just use these models as a way to encode dependencies that we assume exist.

Measuring independence

The simplest way to measure independence is by looking at whether or not the distribution of the possibly dependent variable changes when we change the value of the other variables.

Theorem 4.2.1. *If $x_i \mid \mathbf{x}_B \perp\!\!\!\perp \mathbf{x}_D$ then*

$$\mathbb{P}(x_i \mid \mathbf{x}_B, \mathbf{x}_D) = \mathbb{P}(x_i \mid \mathbf{x}_D)$$

This implies

$$\mathbb{P}(x_i \mid \mathbf{x}_B, \mathbf{x}_D) = \mathbb{P}(x_i \mid \mathbf{x}'_B, \mathbf{x}_D)$$

so we can measure independence by seeing how the distribution of x_i changes when we vary \mathbf{x}'_B , keeping \mathbf{x}_D fixed.

4.3 Concepts of fairness

Bail decisions, revisited

Let us think of this problem in terms of bail decisions made by a judge using some policy π with $\pi(a \mid x)$ being the probability that the judge decides a when she observes x . Let y be the outcome, which may or may not depend on a . In this particular case, a is either release or jail. And y is appears for trial or not. If we accept the tenets of decision theory, there is also a utility function $U(a, y)$ defined on which the judge bases her decision.

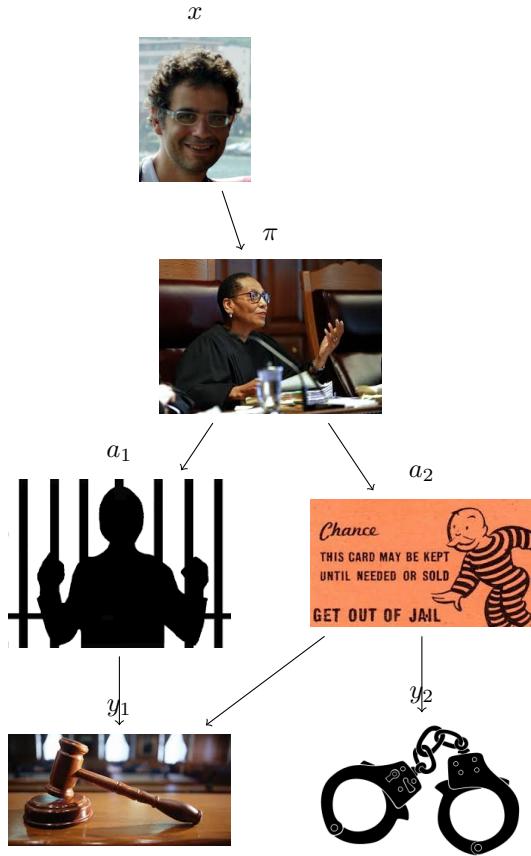


Figure 4.10: The bail decision process, simplified.

4.3.1 Fairness as independence

So how can we reframe the above fairness notions in a more precise way? Both of them involve conditional independence between y, a and a sensitive attribute z , such as race. The first notion says that the actions of the judge (or equivalently, the scores of the algorithm) are *calibrated* with respect to the outcomes. The second says that they are *balanced*, so that were the outcome known to the judge, she would be making a decision independently of the defendant's race. Both of these conditions were discussed in a more restricted setting by

Definition 4.3.1 (Calibration). A policy π is calibrated for parameter θ with respect to z if

$$\mathbb{P}_\theta^\pi(y \mid a, z) = \mathbb{P}_\theta^\pi(y \mid a), \quad \forall a, z. \quad (4.3.1)$$

You will observe that calibration here means that

$$y \perp\!\!\!\perp z \mid a, \theta, \pi$$

i.e. that y is independent of z given the judge's action a , so the distribution of outcomes is the same for every one of our actions no matter what the value of z is.

Definition 4.3.2 (Balance). A policy π is balanced for parameter θ with respect to z if

$$\mathbb{P}_\theta^\pi(a \mid y, z) = \mathbb{P}_\theta^\pi(a \mid y), \quad \forall y, z. \quad (4.3.2)$$

On the other hand, balance means that

$$a \perp\!\!\!\perp z \mid y,$$

i.e. that a is independent of z given the true outcome y .³

4.3.2 Fairness as meritocracy.

A different concept of fairness is meritocracy. For example, if one candidate for a job is better than another candidate, perhaps that candidate should be taken for the job.

Let us consider merit from the point of view of the decision maker, who can either hire ($a_t = 1$) or not hire ($a_t = 0$) the t -th applicant. If the applicant has characteristics x_t and merit y_t , the DM's decision has utility $U(a_t, y_t)$. In order to model meritocracy, we assign an inherent *quality* to y , expressed as an ordering, so that $U(1, y) \geq U(1, y')$ if $y \geq y'$. Assuming $P_\theta(x_t, y_t)$ is known to the DM then clearly she should make the decision by solving the following maximisation problem:

Meritocratic decision

$$a_t(\theta, x_t) \in \arg \max_a \mathbb{E}_\theta(U \mid a, x_t) = \int_y U(a_t, y) \mathbb{E}_\theta(U \mid a_t, x_t) \quad (4.3.3)$$

Here, the notion of meritocracy is defined through our utility function. Although it would be better to consider the candidate's utility instead, this is in practice difficult, because we'd have to somehow estimate each individual's utility function. Finally, we are taking the expectation here is because we may not know for certain what the quality attribute of a given person might be.

4.3.3 Fairness as similarity.

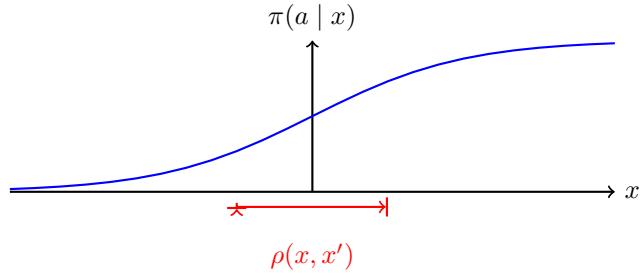
It makes sense to combine the idea of meritocracy with that of similarity. That is, similar people should be treated similarly. This means that we should find a policy π that maximises utility U and makes similar decisions for similar people.

Let \mathcal{X} be equipped with a metric ρ , and let D be a divergence between distributions, such as the KL-divergence. We can then formalise the above intuition as follows:

$$D[\pi(a \mid x), \pi(a \mid x')] \leq \rho(x, x'). \quad (4.3.4)$$

³This definition only really makes sense when y does not depend on a at all. When this is not the case, it's easy to construct a random variable y' that does not depend on a so that y can be written as a function $y(y', a)$. Then we can achieve balance with respect to y' .

This is a so-called Lipschitz condition on the policy, and is illustrated in the figure below.



4.3.4 Bayesian fairness

In both cases, we defined conditional independence for a fixed probability distribution $P_\theta(x, y, z)$ on the various variables. However, this cannot be assumed to be known.

4.4 Project: Credit risk for mortgages

Consider a bank that must design a decision rule for giving loans to individuals. In this particular case, some of each individual's characteristics are partially known to the bank. We can assume that the insurer has a linear utility for money and wishes to maximise expected utility. Assume that the t -th individual is associated with relevant information x_t , sensitive information z_t and a potential outcome y_t , which is whether or not they will default on their mortgage. For each individual t , the decision rule chooses $a \in \mathcal{A}$ with probability $\pi(a_t = a | x_t)$.

As an example, take a look at the historical data in `data/credit/german.data-mumeric`, described in `data/credit/german.doc`. Here there are some attributes related to financial situation, as well as some attributes related to personal information such as gender and marital status.

A skeleton for the project is available at <https://github.com/olethrosdc/ml-society-science/tree/master/src/project-1>. Start with `random_banker.py` as a template, and create a new module `name_banker.py`. You can test your implementation with the `TestLending.py` program.

For ensuring progress, the project is split into two parts:

4.4.1 Deadline 1: September 14

The first part of the project focuses on a baseline implementation of a banker module.

1. Design a policy for giving or denying credit to individuals, given their probability for being credit-worthy. Assuming that if an individual is credit-worthy, you will obtain a return on investment of $r = 0.5\%$ per month. Take into account the length of the loan to calculate the utility through `NameBanker.expected_utility()`. Assume that the loan is either fully repaid at the end of the lending period n , or not at all to make things simple. If an individual is not credit-worthy you will lose your investment of m credits, otherwise you will gain $m[(1 + r)^n - 1]$. Ignore macroeconomic aspects, such as inflation. In this section, simply assume you have a model for predicting creditworthiness as input to your policy, which you can access `NameBanker.get_proba()`.
2. Implement `NameBanker.fit()` to fit a model for calculating the probability of creditworthiness from the german data. Then implement `NameBanker.predict_proba()` to predict the probability of the loan being returned for new data. What are the implicit assumptions about the labelling process in the original data, i.e. what do the labels represent?
3. Combine the model with the first policy to obtain a policy for giving credit, given only the information about the individual and previous data seen. In other words, implement `Namebanker.get_best_action()`.
4. Finally, using `TestLending.py` as a baseline, create a jupyter notebook where you document your model development. Then compare your model against `RandomBanker`.

4.4.2 Deadline 2: September 28

The second part of the project focuses on issues of reproducibility, reliability, privacy and fairness. That is, how desirable would it be to use this model in practice? Here are some sample questions that you can explore, but you should be free to think about other questions.

1. Is it possible to ensure that your policy maximises revenue? How can you take into account the uncertainty due to the limited and/or biased data? What if you have to decide for credit for thousands of individuals and your model is wrong? How should you take that type of risk into account?⁴
2. Does the existence of this database raise any privacy concerns? If the database was secret (and only known by the bank), but the credit decisions were public, how would that affect privacy? (a) Explain how you would protect the data of the people in the training set. (b) Explain how would protect the data of the people that apply for new loans. (c) *Implement* a private decision making mechanism for (b),⁵ and estimate the amount of loss in utility as you change the privacy guarantee.
3. Choose one concept of fairness, e.g. balance of decisions with respect to gender. How can you ensure that your policy is fair? How can you measure it? How does the original training data affect the fairness of your policy? ⁶

Submit a final report about your project, either as a standalone PDF or as a jupyter notebook.

⁴You do not need to implement anything specific for this to pass the assignment, but you should outline an algorithm in a precise enough manner that it can be implemented. In either case you should explain how your solution mitigates this type of risk.

⁵If you have already implemented (a) as part of the tutorial, feel free to include the results in your report.

⁶You do not need to implement any type of fair policy a passing grade, but you should at least try to analyse the data or your decision function with simple statistics.

Chapter 5

Recommendation systems

Structured learning problems involve multiple latent variables with a complex structure. These range from clustering and speech recognition to DNA and biological and social network analysis. Since structured problems include relationships between many variables, they can be analysed using graphical models.

5.1 Recommendation systems



Figure 5.1: The recommendation problem

In many machine learning applications, we are dealing with the problem of proposing one or more alternatives to a human. The human can accept zero or more of these choices. As an example, when using an internet search engine, we typically see two things: (a) A list of webpages matching our search terms (b) A smaller list of advertisements that might be relevant to our search. At a high level,

The recommendation problem

At time t

1. A customer x_t appears. For the internet search problem, x_t would at least involve the search term used.
2. We present a choice a_t . For the matching website, the choice is ranked list of websites. For the advertisements, however, it is typical
3. The customer makes a choice y_t . This might include selecting one or more of items suggested in a_t .

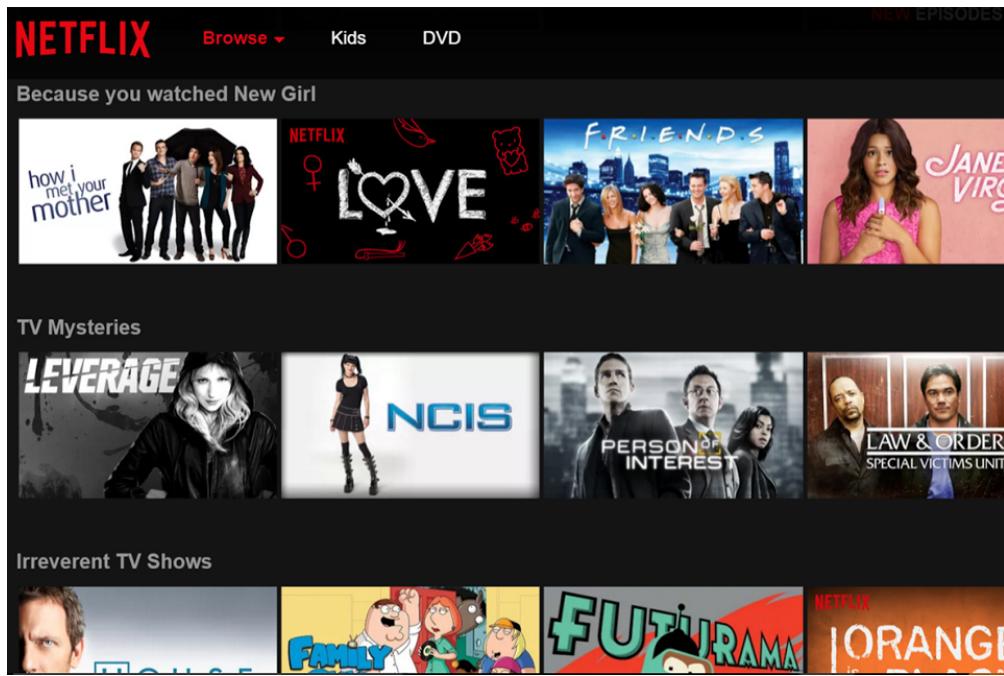


Figure 5.2: The Netflix recommendation problem

EXAMPLE 21. In the case of Netflix and related services, we would like to suggest movies to users which they are more likely to watch, as shown in Figure 5.2. However, how can we tell which movies those can be? It is probably not useful to just recommend them to rewatch a previously watched movie. We need to somehow take into account information across our user database: if somebody watched mostly the same films as you, then maybe you'd be interested in watching those movies she has that you haven't seen.

In the Netflix catalogue, in particular, users also post reviews of the movies they have watched, as shown in Figure 5.3. This allows us to be able to guess the ratings of users from previous user's ratings.

Example: Item-based CF

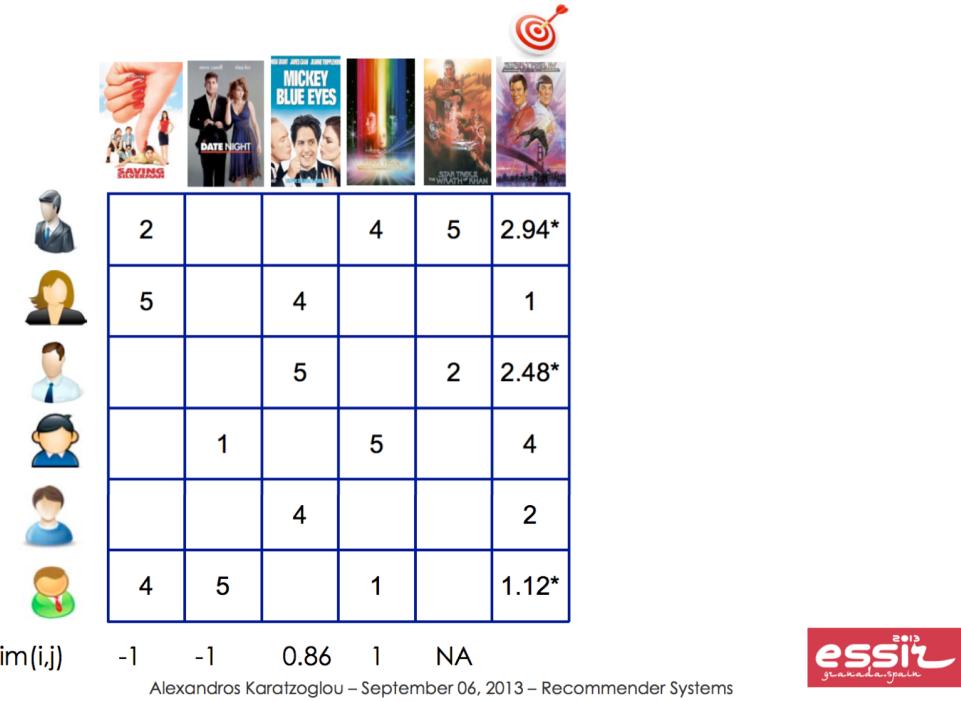


Figure 5.3: User ratings

Predictions based on similarity

Collaborative filtering

- *Similar users have similar tastes.* For example, consider two users t, u who have each watched a set of movies \mathcal{M}_t and \mathcal{M}_u respectively, and $\mathcal{M}_{t,u} = \mathcal{M}_t \cap \mathcal{M}_u$ is the set of common movies. If their ratings are the same for those movies, i.e. $x_{t,m} = x_{u,m} \forall m \in \mathcal{M}_{t,u}$, then it's a good guess that they might have the same ratings for movies they have not both watched.
- That means we can use similar user's *ratings* to predict the ratings for other users. The advantage is that ratings are readily available. The disadvantage is that new users have too few data to be matched to other users.

Content-based filtering.

- Users typically like similar items. For example, a horror movie fan typically rates horror movies highly.

- That means we can one user's ratings and *item information* to predict their ratings for other items. In this scenario

k-NN for similarity

EXERCISE 11. • Define a distance $d : \mathcal{X}^M \times \mathcal{X}^M \rightarrow \mathbb{R}_+$ between user ratings.

- Apply a *k*-NN algorithm to prediction of user ratings from the dataset.

Preferences as clusters

As a simple model, we can assume that each person belongs to a *type*. Every type has the same preferences over films. In the simplest possible model, a user of type c_i that has watched a movie m will rate the film deterministically $x_{c,m}$. More generally, we can assume the following model.

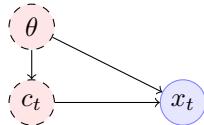


Figure 5.4: Preference model

Preference model

- User t .
- User type $c_t \in \mathcal{C}$. For simplicity, we can think of there being a finite number of types $\mathcal{C} = \{1, \dots, n\}$.
- User ratings \mathbf{x}_t with $x_{t,m} \in \mathcal{X}$ rating for movie m . As an example, $\mathcal{X} = \{0, 1, 2, 3, 4, 5\}$, with 0 meaning no rating given. This is important, since
- Preference distribution $P_\theta(\mathbf{x}_t = \mathbf{x} | c_t = c)$. Typically, we can assume that ratings are independent given the type

$$P_\theta(\mathbf{x}_t = \mathbf{x} | c_t = c) = \prod_{m=1}^M P_\theta(x_{t,m} = x_m | c_t = c),$$

so that a single (vector) parameter $\theta_{c,m}$ will describe the distribution of ratings for a particular movie m and type c , i.e. $P_\theta(x_{t,m} = x_m | c_t = c) = P_\theta(x_{t,m} = x_m | c_t = c)$.

5.2 Clustering

Clustering is the problem of automatically segregating data of different types into clusters. When the goal is *anomaly detection*, then there are typically two clusters. When the goal is *compression* or *auto-encoding* then there are typically as many clusters as needed for sufficiently good accuracy.

Clusters as latent variables

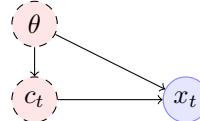


Figure 5.5: Graphical model for independent data from a cluster distribution.

The clustering distribution

The learning problem is to estimate the parameter θ describing the distribution of observations x_t and clusters c_t .

$$x_t \mid c_t = c, \theta \sim P_\theta(x|c), \quad c_t \mid \theta \sim P_\theta(c)$$

Given a parameter θ , the clustering problem is to estimate the probability of each cluster for each new observation.

$$P_\theta(c_t \mid x_t) = \frac{P_\theta(x_t \mid c_t)P_\theta(c_t)}{\sum_{c'} P_\theta(x_t \mid c_t = c')P_\theta(c_t = c')}$$

Bayesian formulation of the clustering problem

- Prior ξ on parameter space Θ .
- Data $x^T = x_1, \dots, x_T$. Cluster assignments c^T unknown.
- Posterior $\xi(\cdot \mid x^T)$.

Posterior distribution

The data we obtain do not include the cluster assignments, but we can still formulate the posterior distribution of parameters given the data.

$$\xi(\theta \mid x^T) = \frac{P_\theta(x^T)\xi(\theta)}{\int_\Theta P_{\theta'}(x^T) d\xi(\theta')}, \quad P_\theta(x^T) = \sum_{c^T \in \mathcal{C}^T} \underbrace{P_\theta(x^T \mid c^T)}_{\text{Cluster Density}} \underbrace{P_\theta(c^T)}_{\text{Cluster prior}} \quad (5.2.1)$$

We simply need to expand the data-dependent term to include all possible cluster assignments. This is of course not trivial, since the number of assignments is exponential in T . However, algorithms such as Markov Chain Monte Carlo can be used instead.

Marginal posterior prediction

$$P_\xi(c_t | x_t, x^T) = \int_{\Theta} P_\theta(c_t | x_t) d\xi(\theta | x^T)$$

EXAMPLE 22 (Preference clustering). The learning problem is to estimate the parameter θ describing the distribution of observations x_t and clusters c_t . In this example, we can assume

$$\mathcal{C} = \{1, \dots, C\}, \quad x_{t,m} \in \{0, 1\}.$$

This means that all movies are either watched or not, and we'd simply want to predict which movie somebody is likely to watch. This allows us to use the following simple priors, splitting the parameters in two parts $\theta = (\theta_1, \theta_2)$.

Model family

$$P_{\theta_1}(c_t = c) = \theta_{1,c}, \quad c_t \sim \text{Multinomial}(\theta_1) \quad (5.2.2)$$

$$P_{\theta_2}(x_{t,m} = 1 | c_t = c) = \theta_{2,m,c} \quad x_{t,m} | c_t = c \sim \text{Bernoulli}(\theta_{2,m,c}) \quad (5.2.3)$$

Since everything is discrete, it makes sense that we can use a Multinomial model for the cluster distribution and a Bernoulli model for whether or not a movie was watched. Now we only need to specify a useful prior for each one of those. The standard priors to use, are a Beta prior for the Bernoulli and the Dirichlet for the Multinomial, as they are conjugate.

Prior

$$\theta_1 \sim \text{Dirichlet}(\gamma), \theta_2 \sim \text{Beta}(\alpha, \beta) \quad (5.2.4)$$

Typically $\gamma = (1/2, \dots, 1/2)$ and $\alpha = \beta = 1/2$ to allow for the possibility of nearly deterministic behaviour.

5.3 Social networks

Social networks afford us another opportunity to take a look at data. We can use connections between users to infer their similarity: if two users are connected, then they are more likely to have similar preferences.

Network model

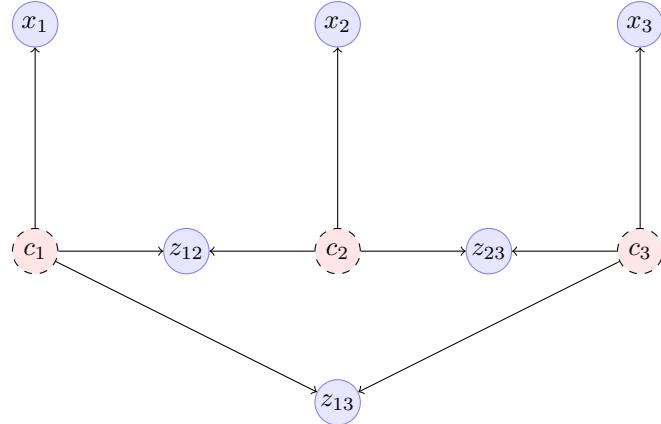


Figure 5.6: Graphical model for data from a social network.

In the model seen in Figure 5.6, each user t is characterised by their cluster membership c_t and emits data x_t . Users t, u are connected when $z_{t,u} = 1$.

5.4 Sequential structures

The simplest type of structure in data is sequences. Examples include speech, text and DNA sequences, as well as data acquired in any sequential decision making problem such as recommendation systems or robotics. Sequential data is always thought to arise from some Markovian processes, defined below.

Markov process

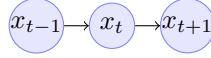


Figure 5.7: Graphical model for a Markov process.

Definition 5.4.1 (Markov process). A Markov process is a sequence of variables $x_t : \Omega \rightarrow \mathcal{X}$ such that $x_{t+1} \mid x_t \perp\!\!\!\perp x_{t-k} \forall k \leq 1$.

DNA data

Hidden Markov model

Frequently the sequential dependency is not in the data itself, but in some hidden underlying markov process. In that case, the hidden variable x_t is the *state* of the process. The observed variable y_t is simply an observation.

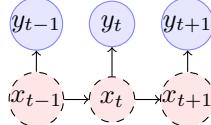


Figure 5.8: Graphical model for a hidden Markov model.

$$\begin{array}{ll} P_\theta(x_{t+1} \mid x_t) & \text{(transition distribution)} \\ P_\theta(y_t \mid x_t) & \text{(emission distribution)} \end{array}$$

For any given parater value θ , it is easy to estimate the probability distribution over states given the observations $P_\theta(x^T \mid y^T)$. As an example, if y^T is raw speech data and x^T is a sequence of words, and θ are the parameters of our speech model, then we can obtain probabilities for every possible sequence of words that was uttered. Frequently, though, in speech recognition we are only interested in the most likely seuence of words. This makes the problem simple enough to be solved instantaneously by modern cellphones.

HMM modelling of DNA data

As a more detailed example, consider hidden Markov models for DNA data....

Chapter 6

Bandit problems

nothing

6.1 Introduction

This unit describes the very general formalism of Markov decision processes (MDPs) for formalising problems in sequential decision making. Thus a *Markov decision process* can be used to model stochastic path problems, stopping problems, reinforcement learning problems, experiment design problems, and control problems.

We begin by taking a look at the problem of *experimental design*. One instance of this problem occurs when considering how to best allocate treatments with unknown efficacy to patients in an adaptive manner, so that the best treatment is found, or so as to maximise the number of patients that are treated successfully. The problem, originally considered by ??, informally can be stated as follows.

We have a number of treatments of unknown efficacy, i.e. some of them work better than the others. We observe patients one at a time. When a new patient arrives, we must choose which treatment to administer. Afterwards, we observe whether the patient improves or not. Given that the treatment effects are initially unknown, how can we maximise the number of cured patients? Alternatively, how can we discover the best treatment? The two different problems are formalised below.

EXAMPLE 23. Consider k treatments to be administered to T volunteers. To each volunteer only a single treatment can be assigned. At the t -th trial, we treat one volunteer with some treatment $a_t \in \{1, \dots, k\}$. We then obtain a reward $r_t = 1$ if the patient is treated and 0 otherwise. We wish to choose actions maximising the utility $U = \sum_t r_t$. This would correspond to maximising the number of patients that get treated over time.

EXAMPLE 24. An alternative goal would be to do a *clinical trial*, in order to find the best possible treatment. For simplicity, consider the problem of trying to find out whether a particular treatment is better or not than a placebo. We are given a hypothesis set Ω , with each $\omega \in \Omega$ corresponding to different models for the effect of the treatment and the placebo. Since we don't know what is the right model, we place a prior ξ_0 on Ω . We can perform T experiments, after which we must make a decision whether or not the treatment is significantly better than the placebo. To model this, we define a decision set $\mathcal{D} = \{d_0, d_1\}$ and a utility function $U : \mathcal{D} \times \Omega \rightarrow \mathbb{R}$, which models the effect of each decision d given different versions of reality ω . One hypothesis $\omega \in \Omega$ is true. To distinguish them, we can choose from a set of k possible experiments to be performed over T trials. At the t -th trial, we choose experiment $a_t \in \{1, \dots, k\}$ and observe outcome $x_t \in \mathcal{X}$, with $x_t \sim P_\omega$ drawn from the true hypothesis. Our posterior is

$$\xi_t(\omega) \triangleq \xi_0(\omega | a_1, \dots, a_t, x_1, \dots, x_t).$$

The reward is $r_t = 0$ for $t < T$ and

$$r_T = \max_{d \in \mathcal{D}} \mathbb{E}_{\xi_T}(U | d).$$

Our utility in this can again be expressed as a sum over individual rewards, $U = \sum_{t=1}^T r_t$.

Both formalizations correspond to so-called *bandit problems* which we take a closer look at in the following section.

6.2 Bandit problems

The simplest bandit problem is the stochastic n -armed bandit. We are faced with n different one-armed bandit machines, such as those found in casinos. In this problem, at time t , you have to choose one *action* (i.e. a machine) $a_t \in \mathcal{A} = \{1, \dots, n\}$. In this setting, each time t you play a machine, you receive a reward r_t , with fixed expected value $\omega_i = \mathbb{E}(r_t | a_t = i)$. Unfortunately,

you do not know ω_i , and consequently the best arm is also unknown. How do you then choose arms so as to maximise the total expected reward?

Definition 6.2.1 (The stochastic n -armed bandit problem.). This is the problem of selecting a sequence of actions $a_t \in \mathcal{A}$, with $\mathcal{A} = \{1, \dots, n\}$, so as to maximise expected utility, where the utility is

$$U = \sum_{t=0}^{T-1} \gamma^t r_t,$$

where $T \in (0, \infty]$ is the horizon and $\gamma \in (0, 1]$ is a *discount factor*. The reward r_t is stochastic, and only depends on the current action, with expectation $\mathbb{E}(r_t | a_t = i) = \omega_i$.

In order to select the actions, we must specify some *policy* or decision rule. This can only depend on the sequence of previously taken actions and observed rewards. Usually, the policy $\pi : \mathcal{A}^* \times \mathbb{R}^* \rightarrow \mathcal{A}$ is a deterministic mapping from the space of all sequences of actions and rewards to actions. That is, for every observation and action history $a_1, r_1, \dots, a_{t-1}, r_{t-1}$ it suggests a single action a_t . However, it could also be a stochastic policy, that specifies a mapping to action distributions. We use the following notation for stochastic history-dependent bandit policies,

$$\pi(a_t | a^{t-1}, r^{t-1}) \quad (6.2.1)$$

to mean the probability of actions a_t given the history until time t .

How can we solve bandit problems? One idea is to apply the Bayesian decision-theoretic framework we have developed earlier to maximise utility in expectation. More specifically, given the horizon $T \in (0, \infty]$ and the discount factor $\gamma \in (0, 1]$, we define our utility from time t to be:

$$U_t = \sum_{k=1}^{T-t} \gamma^k r_{t+k}. \quad (6.2.2)$$

To apply the decision theoretic framework, we need to define a suitable family of probability measures \mathcal{F} , indexed by parameter $\omega \in \Omega$ describing the reward distribution of each bandit, together with a prior distribution ξ on Ω . Since ω is unknown, we cannot maximise the expected utility with respect to it. However, we can always maximise expected utility with respect to our belief ξ . That is, we replace the ill-defined problem of maximising utility in an unknown model with that of maximising expected utility given a distribution over possible models. The problem can be written in a simple form:

$$\max_{\pi} \mathbb{E}_{\xi}^{\pi} U_t = \max_{\pi} \int_{\Omega} \mathbb{E}_{\omega}^{\pi} U_t d\xi \omega. \quad (6.2.3)$$

The difficulty lies not in formalising the problem, but in the fact that the set of learning policies is quite large, rendering the optimisation infeasible. The following figure summarises the statement of the bandit problem in the Bayesian setting.

Decision-theoretic statement of the bandit problem

- Let \mathcal{A} be the set of arms.
- Define a family of distributions $\mathcal{F} = \{P_{\omega,i} | \omega \in \Omega, i \in \mathcal{A}\}$ on \mathbb{R} .

- Assume the i.i.d model $r_t \mid \omega, a_t = i \sim P_{\omega,i}$.
- Define prior ξ on Ω .
- Select a policy $\pi : \mathcal{A}^* \times \mathbb{R}^* \rightarrow \mathcal{A}$ maximising

$$\mathbb{E}_\xi^\pi U = \mathbb{E}_\xi^\pi \sum_{t=0}^{T-1} \gamma^t r_t$$

There are two main difficulties with this approach. The first is specifying the family and the prior distribution: this is effectively part of the problem formulation and can severely influence the solution. The second is calculating the policy that maximises expected utility given a prior and family. The first problem can be resolved by either specifying a subjective prior distribution, or by selecting a prior distribution that has good worst-case guarantees. The second problem is hard to solve, because in general, such policies are history dependent and the set of all possible histories is exponential in the horizon T .

6.2.1 An example: Bernoulli bandits

As a simple illustration, consider the case when the reward for choosing one of the n actions is either 0 or 1, with some fixed, yet unknown probability depending on the chosen action. This can be modelled in the standard Bayesian framework using the Beta-Bernoulli conjugate prior. More specifically, we can formalise the problem as follows.

Consider n Bernoulli distributions with unknown parameters ω_i ($i = 1, \dots, n$) such that

$$r_t \mid a_t = i \sim \text{Bernoulli}(\omega_i), \quad \mathbb{E}(r_t \mid a_t = i) = \omega_i. \quad (6.2.4)$$

Each Bernoulli distribution thus corresponds to the distribution of rewards obtained from each bandit that we can play. In order to apply the statistical decision theoretic framework, we have to quantify our uncertainty about the parameters ω in terms of a probability distribution.

We model our belief for each bandit's parameter ω_i as a Beta distribution $\text{Beta}(\alpha_i, \beta_i)$, with density $f(\omega \mid \alpha_i, \beta_i)$ so that

$$\xi(\omega_1, \dots, \omega_n) = \prod_{i=1}^n f(\omega_i \mid \alpha_i, \beta_i).$$

Recall that the posterior of a Beta prior is also a Beta. Let

$$N_{t,i} \triangleq \sum_{k=1}^t \mathbb{I}\{a_k = i\}$$

be the number of times we played arm i and

$$\hat{r}_{t,i} \triangleq \frac{1}{N_{t,i}} \sum_{k=1}^t r_k \mathbb{I}\{a_k = i\}$$

be the *empirical reward* of arm i at time t . We can let this equal 0 when $N_{t,i} = 0$. Then, the posterior distribution for the parameter of arm i is

$$\xi_t = \text{Beta}(\alpha_i + N_{t,i} \hat{r}_{t,i}, \beta_i + N_{t,i} (1 - \hat{r}_{t,i})).$$

Since $r_t \in \{0, 1\}$ the possible states of our belief given some prior are \mathbb{N}^{2n} .

In order for us to be able to evaluate a policy, we need to be able to predict the expected utility we obtain. This only depends on our current belief, and the state of our belief corresponds to the state of the bandit problem. This means that everything we know about the problem at time t can be summarised by ξ_t . For Bernoulli bandits, sufficient statistic for our belief is the number of times we played each bandit and the total reward from each bandit. Thus, our state at time t is entirely described by our priors α, β (the initial state) and the vectors

$$N_t = (N_{t,1}, \dots, N_{t,i}) \quad (6.2.5)$$

$$\hat{r}_t = (\hat{r}_{t,1}, \dots, \hat{r}_{t,i}). \quad (6.2.6)$$

At any time t , we can calculate the probability of observing $r_t = 1$ or $r_t = 0$ if we pull arm i as:

$$\xi_t(r_t = 1 \mid a_t = i) = \frac{\alpha_i + N_{t,i}\hat{r}_{t,i}}{\alpha_i + \beta_i + N_{t,i}}$$

So, not only we can predict the immediate reward based on our current belief, but we can also predict all next possible beliefs: the next state is well-defined and depends only on the current state. As we shall see later, this type of decision problem is more generally called a Markov decision process (Definition 7.1.1). For now, we shall more generally (and precisely) define the bandit process itself.

6.2.2 Decision-theoretic bandit process

The basic bandit process can be seen in Figure 6.2(a). We can now define the general decision-theoretic bandit process, not restricted to independent Bernoulli bandits.

Definition 6.2.2. Let \mathcal{A} be a set of actions, not necessarily finite. Let Ω be a set of possible parameter values, indexing a family of probability measures $\mathcal{F} = \{P_{\omega,a} \mid \omega \in \Omega, a \in \mathcal{A}\}$. There is some $\omega \in \Omega$ such that, whenever we take action $a_t = a$, we observe reward $r_t \in \mathcal{R} \subset \mathbb{R}$ with probability measure:

$$P_{\omega,a}(R) \triangleq \mathbb{P}_{\omega}(r_t \in R \mid a_t = a), \quad R \subseteq \mathbb{R}. \quad (6.2.7)$$

Let ξ_1 be a prior distribution on Ω and let the posterior distributions be defined as:

$$\xi_{t+1}(B) \propto \int_B P_{\omega,a_t}(r_t) d\xi_t(\omega). \quad (6.2.8)$$

The next belief is random, since it depends on the random quantity r_t . In fact, the probability of the next reward lying in R if $a_t = a$ is given by the following marginal distribution:

$$P_{\xi_t,a}(R) \triangleq \int_{\Omega} P_{\omega,a}(R) d\xi_t(\omega). \quad (6.2.9)$$

Finally, as ξ_{t+1} deterministically depends on ξ_t, a_t, r_t , the probability of obtaining a particular next belief is the same as the probability of obtaining the corresponding rewards leading to the next belief. In more detail, we can write:

$$\mathbb{P}(\xi_{t+1} = \xi \mid \xi_t, a_t) = \int_{\mathcal{R}} \mathbb{I}\{\xi_t(\cdot \mid a_t, r_t = r) = \xi\} dP_{\xi_t,a}(r). \quad (6.2.10)$$

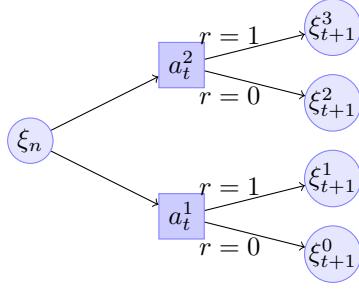


Figure 6.1: A partial view of the multi-stage process. Here, the probability that we obtain $r = 1$ if we take action $a_t = i$ is simply $P_{\xi_t,i}(\{1\})$.

In practice, although multiple reward sequences may lead to the same beliefs, we frequently ignore that possibility for simplicity. Then the process becomes a tree. A solution to the problem of what action to select is given by a backwards induction algorithm similar to that given in Section ??.

$$U^*(\xi_t) = \max_{a_t} \mathbb{E}(r_t | \xi_t, a_t) + \sum_{\xi_{t+1}} \mathbb{P}(\xi_{t+1} | \xi_t, a_t) U^*(\xi_{t+1}). \quad (6.2.11)$$

The above equation is the *backwards induction* algorithm for bandits. If you look at this structure, you can see that next belief only depends on the current belief, action and reward, i.e. it satisfies the Markov property, as seen in Figure 6.1. Consequently, a decision-theoretic bandit process can be modelled more generally as a Markov decision process, explained in the following section. It turns out that backwards induction, as well as other efficient algorithms, can provide optimal solutions for Markov decision processes.

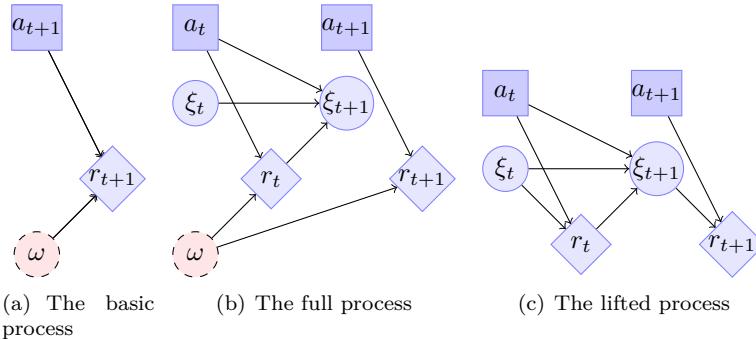


Figure 6.2: Three views of the bandit process. Figure 6.2(a) shows the basic bandit process, from the view of an external observer. The decision maker selects a_t , while the parameter ω of the process is hidden. It then obtains reward r_t . The process repeats for $t = 1, \dots, T$. The decision-theoretic bandit process is shown in Figures 6.2(b) and 6.2(c). While ω is not known, at each time step t we maintain a belief ξ_t on Ω . The reward distribution is then defined through our belief. In Figure 6.2(b), we can see that complete process, where the dependency on ω is clear. In Figure 6.2(c), we marginalise out ω and obtain a model where the transitions only depend on the current belief and action.

In reality, the reward depends only on the action and the unknown ω , as can be seen in

Figure 6.2(b). This is the point of view of an external observer. However, from the point of view of the decision maker, the distribution of ω only depends on his current belief. Consequently, the distribution of rewards also only depends on the current belief, as we can marginalise over ω . This gives rise to the decision-theoretic bandit process shown in Figure 6.2(c). In the following section, we shall consider Markov decision processes more generally.

6.3 Experiment design

Chapter 7

Markov decision processes

7.1 Markov decision processes and reinforcement learning

Bandit problems are one of the simplest instances of reinforcement learning problems. Informally, speaking, these are problems of learning how to act in an unknown environment, only through interaction with the environment and limited reinforcement signals. The learning agent interacts with the environment through actions and observations, and simultaneously obtains rewards. For example, we can consider a rat running through a maze designed by an experimenter, which from time to time finds a piece of cheese, the reward. The goal of the agent is usually to maximise some measure of the total reward. In summary, we can state the problem as follows.

The reinforcement learning problem.

The reinforcement learning problem is the problem of *learning* how to act in an *unknown* environment, only by **interaction** and **reinforcement**.

Generally, we assume that the environment μ that we are acting in has an underlying state $s_t \in \mathcal{S}$, which changes with in discrete time steps t . At each step, the agent obtains an observation $x_t \in \mathcal{X}$ and chooses actions $a_t \in \mathcal{A}$. We usually assume that the environment is such that its next state s_{t+1} only depends on its current state s_t and the last action taken by the agent, a_t . In addition, the agent observes a reward signal r_t , and its goal is to maximise the total reward during its lifetime.

Doing so when the environment μ is unknown, is hard even in seemingly simple settings, like n -armed bandits, where the underlying state never changes. In many real-world applications, the problem is even harder, as the state is not directly observed. Instead, we may simply have some measurements x_t , which give only partial information about the true underlying state s_t .

Reinforcement learning problems typically fall into one of the following three groups: (1) Markov decision processes (MDPs), where the state s_t is observed directly, i.e. $x_t = s_t$; (2) Partially observable MDPs (POMDPs), where the state is hidden, i.e. x_t is only probabilistically dependent on the state; and (3) stochastic Markov games, where the next state also depends on the move of other agents. While all of these problem *descriptions* are different, in the Bayesian setting, they all can be reformulated as MDPs, by constructing an appropriate belief state, similarly to how we did it for the decision theoretic formulation of the bandit problem.

In this chapter, we shall restrict our attention to Markov decision processes. Hence, we shall not discuss the existence of other agents, or the case where we cannot observe the state directly.

Definition 7.1.1 (Markov Decision Process). A Markov decision process μ is a tuple $\mu = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$, where \mathcal{S} is the *state space* and \mathcal{A} is the *action space*. The *transition distribution* being $\mathcal{P} = \{P(\cdot | s, a) | s \in \mathcal{S}, a \in \mathcal{A}\}$ is a collection of probability measures on \mathcal{S} , indexed in $\mathcal{S} \times \mathcal{A}$ and the *reward distribution* $\mathcal{R} = \{\rho(\cdot | s, a) | s \in \mathcal{S}, a \in \mathcal{A}\}$ is a collection of probability measures on \mathbb{R} , such that:

$$P(S | s, a) = \mathbb{P}_\mu(s_{t+1} \in S | s_t = s, a_t = a) \quad (7.1.1)$$

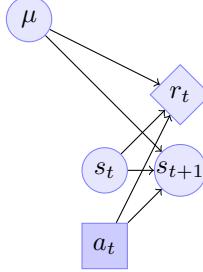
$$\rho(R | s, a) = \mathbb{P}_\mu(r_t \in R | s_t = s, a_t = a). \quad (7.1.2)$$

For simplicity, we shall also use

$$r_\mu(s, a) = \mathbb{E}_\mu(r_{t+1} | s_t = s, a_t = a), \quad (7.1.3)$$

for the expected reward.

Of course, the transition and reward distributions are different for different environments μ . For that reason, we shall usually subscript the relevant probabilities and expectations with μ , unless the MDP is clear from the context.



Markov property of the reward and state distribution

$$\mathbb{P}_\mu(s_{t+1} \in S \mid s_1, a_1, \dots, s_t, a_t) = \mathbb{P}_\mu(s_{t+1} \in S \mid s_t, a_t) \quad (7.1.4)$$

$$\mathbb{P}_\mu(r_t \in R \mid s_1, a_1, \dots, s_t, a_t) = \mathbb{P}_\mu(r_t \in R \mid s_t, a_t) \quad (7.1.5)$$

where $S \subset \mathcal{S}$ and $R \subset \mathcal{R}$ are reward and state subsets respectively.

Figure 7.1: The structure of a Markov decision process.

Dependencies of rewards. Sometimes it is more convenient to have rewards that depend on the next state as well, i.e.

$$r_\mu(s, a, s') = \mathbb{E}_\mu(r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s'), \quad (7.1.6)$$

though this is complicates the notation considerably since now the reward is obtained on the next time step. However, we can always replace this with the expected reward for a given state-action pair:

$$r_\mu(s, a) = \mathbb{E}_\mu(r_{t+1} \mid s_t = s, a_t = s) = \sum_{s' \in S} P_\mu(s' \mid s, a) r_\mu(s, a, s') \quad (7.1.7)$$

In fact, it is notationally more convenient to have rewards that only depend on the current state:

$$r_\mu(s) = \mathbb{E}_\mu(r_t \mid s_t = s). \quad (7.1.8)$$

For simplicity, we shall mainly consider the latter case.

The agent. The environment does not exist in isolation. The actions are taken by an agent, who is interested in obtaining high rewards. Instead of defining an algorithm for choosing actions directly, we define an algorithm for computing policies, which define distributions on actions.

The agent's policy π

$$\begin{array}{ll} \mathbb{P}^\pi(a_t | s_t, \dots, s_1, a_{t-1}, \dots, a_1) & \text{(history-dependent policy)} \\ \mathbb{P}^\pi(a_t | s_t) & \text{(Markov policy)} \end{array}$$

In some sense, the agent is defined by its *policy* π , which is a conditional distribution on actions given the history. The *policy* π is otherwise known as a *decision function*. In general, the policy can be history-dependent. In certain cases, however, there are optimal policies that are Markov. This is for example the case with additive utility functions. In particular, the utility function maps from the sequence of all possible rewards to a real number $U : \mathcal{R}^* \rightarrow \mathbb{R}$, given below:

Definition 7.1.2 (Utility). Given a horizon T and a discount factor $\gamma \in (0, 1]$, the utility function $U : \mathcal{R}^* \rightarrow \mathbb{R}$ is defined as

$$U(r_0, r_1, \dots, r_T) = \sum_{k=0}^T \gamma^k r_k. \quad (7.1.9)$$

It is convenient to give a special name to the utility starting from time t , i.e. the sum of rewards from that time on:

$$U_t \triangleq \sum_{k=0}^{T-t} \gamma^k r_{t+k}. \quad (7.1.10)$$

At any time t , the agent wants to find a policy π *maximising* the *expected total future reward*

$$\mathbb{E}_\mu^\pi U_t = \mathbb{E}_\mu^\pi \sum_{k=0}^{T-t} \gamma^k r_{t+k}. \quad (\text{expected utility})$$

This is so far identical to the expected utility framework we had seen so far, with the only difference that now the reward space is a sequence of numerical rewards and that we are acting within a dynamical system with state space \mathcal{S} . In fact, it is a good idea to think about the *value* of different states of the system under certain policies, in the same way that one thinks about how good different positions are in chess.

7.1.1 Value functions

A value function represents the expected utility of a given state, or state-and-action pair for a specific policy. They are really useful as shorthand notation and as the basis of algorithm development. The most basic of those is the state value function.

State value function

$$V_{\mu,t}^\pi(s) \triangleq \mathbb{E}_\mu^\pi(U_t | s_t = s) \quad (7.1.11)$$

The state value function for a particular policy π can be interpreted as how much utility you should expect if you follow the policy starting from state s at time t , for the particular MDP μ .

State-action value function

$$Q_{\mu,t}^{\pi}(s, a) \triangleq \mathbb{E}_{\mu}^{\pi}(U_t \mid s_t = s, a_t = a) \quad (7.1.12)$$

The state-action value function for a particular policy π can be interpreted as how much utility you should expect if you play action a , at state s at time t , and then follow the policy π , for the particular MDP μ .

It is also useful to define the optimal policy and optimal value functions for a given MDP. In the following, a star indicates optimal quantities. The *optimal policy* π^*

$$\pi^*(\mu) : V_{t,\mu}^{\pi^*(\mu)}(s) \geq V_{t,\mu}^{\pi}(s) \quad \forall \pi, t, s \quad (7.1.13)$$

dominates all other policies π everywhere in \mathcal{S} .

The *optimal value function* V^*

$$V_{t,\mu}^*(s) \triangleq V_{t,\mu}^{\pi^*(\mu)}(s), \quad Q_{t,\mu}^*(s) \triangleq Q_{t,\mu}^{\pi^*(\mu)}(s, a). \quad (7.1.14)$$

is the value function of the optimal policy π^* .

Finding the optimal policy when μ is known

When the MDP μ is known, the expected utility of any policy can be calculated. Therefore, one could find the optimal policy by brute force, i.e. by calculating the utility of every possible policy. This might be a reasonable strategy if the number of policies is small. However, there are many better approaches. First, there are iterative/offline methods where an optimal policy is found for all states of the MDP. These either try to estimate the optimal value function directly, or try to iteratively improve a policy until it is optimal. The second type of methods tries to find an optimal policy online. That is, the optimal actions are estimated only for states which can be visited in the future starting from the current state. However, the same main ideas are used in all of these algorithms.

7.2 Finite horizon, undiscounted problems

The conceptually simplest type of problems are finite horizon problems where $T < \infty$ and $\gamma = 1$. The first thing we shall try to do is to evaluate a given policy for a given MDP. There are a number of algorithms that can achieve this.

7.2.1 Policy evaluation

Here we are interested in the problem of determining the value function of a policy π (for $\gamma = 1, T < \infty$). All the algorithms we shall consider can be recovered from the following

recursion. Noting that $U_{t+1} = \sum_{k=1}^{T-t} r_{t+k}$ we have:

$$V_{\mu,t}^{\pi}(s) \triangleq \mathbb{E}_{\mu}^{\pi}(U_t | s_t = s) \quad (7.2.1)$$

$$= \sum_{k=0}^{T-t} \mathbb{E}_{\mu}^{\pi}(r_{t+k} | s_t = s) \quad (7.2.2)$$

$$= \mathbb{E}_{\mu}^{\pi}(r_t | s_t = s) + \mathbb{E}_{\mu}^{\pi}(U_{t+1} | s_t = s) \quad (7.2.3)$$

$$= \mathbb{E}_{\mu}^{\pi}(r_t | s_t = s) + \sum_{i \in \mathcal{S}} V_{\mu,t+1}^{\pi}(i) \mathbb{P}_{\mu}^{\pi}(s_{t+1} = i | s_t = s). \quad (7.2.4)$$

Note that the last term can be calculated easily through marginalisation.

$$\mathbb{P}_{\mu}^{\pi}(s_{t+1} = i | s_t = s) = \sum_{a \in \mathcal{A}} \mathbb{P}_{\mu}(s_{t+1} = i | s_t = s, a_t = a) \mathbb{P}^{\pi}(a_t = a | s_t = s).$$

This derivation directly gives a number of *policy evaluation algorithms*.

Direct policy evaluation Direct policy evaluation is based on (7.2.2), which can be implemented by Algorithm 3. One needs to *marginalise out* all possible state sequences to obtain the expected reward given the state at time $t+k$ giving the following:

$$\mathbb{E}_{\mu}^{\pi}(r_{t+k} | s_t = s) = \sum_{s_{t+1}, \dots, s_{t+k} \in \mathcal{S}^k} \mathbb{E}_{\mu}^{\pi}(r_{t+k} | s_{t+k}) \mathbb{P}_{\mu}^{\pi}(s_{t+1}, \dots, s_{t+k} | s_t).$$

By using the Markov property, we calculate the probability of reaching any state from any other state at different times, and then add up the expected reward we would get in that state under our policy. Then $\hat{V}_t(s) = V_{\mu,t}^{\pi}(s)$ by definition.

Unfortunately it is not a very good idea to use direct policy evaluation. The most efficient implementation involves calculating $P(s_t | s_0)$ recursively for every state. This would result in a total of $|\mathcal{S}|^3 T$ operations. Monte-Carlo evaluations should be considerably cheaper, especially when the transition structure is sparse.

Algorithm 3 Direct policy evaluation

```

1: for  $s \in \mathcal{S}$  do
2:   for  $t = 0, \dots, T$  do
3:      $\hat{V}_t(s) = \sum_{k=t}^T \sum_{j \in \mathcal{S}} \mathbb{P}_{\mu}^{\pi}(s_k = j | s_t = s) \mathbb{E}_{\mu}^{\pi}(r_k | s_k = j).$ 
4:   end for
5: end for

```

7.2.2 Monte-Carlo policy evaluation

Another conceptually simple algorithm is Monte-Carlo policy evaluation shown as Algorithm 4. The idea is that instead of summing over all possible states to be visited, we just draw states from the Markov chain defined jointly by the policy and the Markov decision process. Unlike direct policy evaluation the algorithm needs a parameter K , the number of trajectories to generate. Nevertheless, this is a very useful method, employed within a number of more complex algorithms.

Algorithm 4 Monte-Carlo policy evaluation

```

for  $s \in \mathcal{S}$  do
  for  $k = 0, \dots, K$  do
    Choose initial state  $s_1$ .
    for  $t = 1, \dots, T$  do
       $a_t \sim \pi(a_t | s_t)$  // Take action
      Observe reward  $r_t$  and next state  $s_{t+1}$ .
      Set  $r_{t,k} = r_t$ .
    end for
    Save total reward:
    
$$\hat{V}_k(s) = \sum_{t=1}^T r_{t,k}.$$

  end for
  Calculate estimate:
  
$$\hat{V}(s) = \frac{1}{K} \sum_{k=1}^K \hat{V}_k(s).$$

end for

```

Remark 7.2.1. The estimate \hat{V} of the Monte Carlo evaluation algorithm satisfies

$$\|V - \hat{V}\|_\infty \leq \sqrt{\frac{\ln(2|\mathcal{S}|/\delta)}{2K}} \quad \text{with probability } 1 - \delta$$

Proof. From Hoeffding's inequality (??) we have for any state s that

$$\mathbb{P}\left(|\hat{V}(s) - V(s)| \geq \sqrt{\frac{\ln(2|\mathcal{S}|/\delta)}{2K}}\right) \leq \delta/|\mathcal{S}|.$$

Consequently, using a union bound of the form $P(A_1 \cup A_2 \cup \dots \cup A_n) \leq \sum_i P(A_i)$ gives the required result. \square

The main advantage of Monte-Carlo policy evaluation is that it can be used in very general settings. It can be used not only in Markovian environments such as MDPs, but also in partially observable and multi-agent settings.

7.2.3 Backwards induction policy evaluation

Finally, the backwards induction algorithm shown as Algorithm 5 is similar to the backwards induction algorithm we saw for sequential sampling and bandit problems. However, here we are only evaluating a policy rather than finding the optimal one. This algorithm is slightly less generally applicable than the Monte-Carlo method because it makes Markovian assumptions. The Monte-Carlo algorithm, can be used for environments that with a non-Markovian variable s_t .

Algorithm 5 Backwards induction policy evaluation

For each state $s \in S$, for $t = 1, \dots, T - 1$:

$$\hat{V}_t(s) = r_\mu^\pi(s) + \sum_{j \in S} \mathbb{P}_\mu^\pi(s_{t+1} = j \mid s_t = s) \hat{V}_{t+1}(j), \quad (7.2.5)$$

with $\hat{V}_T(s) = r_\mu^\pi(s)$.

Theorem 7.2.1. *The backwards induction algorithm gives estimates $\hat{V}_t(s)$ satisfying*

$$\hat{V}_t(s) = V_{\mu,t}^\pi(s) \quad (7.2.6)$$

Proof. For $t = T - 1$, the result is obvious. We can prove the remainder by induction. Let (7.2.6) hold for all $t \geq n + 1$. Now we prove that it holds for n . Note that from the recursion (7.2.5) we have:

$$\begin{aligned} \hat{V}_t(s) &= r_\mu(s) + \sum_{j \in S} \mathbb{P}_{\mu,\pi}(s_{t+1} = j \mid s_t = s) \hat{V}_{t+1}(j) \\ &= r(s) + \sum_{j \in S} \mathbb{P}_{\mu,\pi}(s_{t+1} = j \mid s_t = s) V_{\mu,t+1}^\pi(j) \\ &= r(s) + \mathbb{E}_{\mu,\pi}(U_{t+1} \mid s_t = s) \\ &= \mathbb{E}_{\mu,\pi}(U_t \mid s_t = s) = V_{\mu,t}^\pi(s), \end{aligned}$$

where the second equality is by the induction hypothesis, the third and fourth equalities are by the definition of the utility, and the last by definition of $V_{\mu,t}^\pi$. \square

7.2.4 Backwards induction policy optimisation

Backwards induction as given in Alg 6 is the first non-naive algorithm for finding an optimal policy for the sequential problems with T stages. It is basically identical to the backwards induction algorithm we saw in Chapter ??, which was for the very simple sequential sampling problem, as well as the backwards induction algorithm for the decision-theoretic bandit problem.

Algorithm 6 Finite-horizon backwards induction

Input μ , set \mathcal{S}_T of states reachable within T steps.
 Initialise $V_T(s) := \max_a r(s, a)$, for all $s \in \mathcal{S}_T$.
for $n = T - 1, T - 2, \dots, 1$ **do**
 for $s \in \mathcal{S}_n$ **do**
 $\pi_n(s) = \arg \max_a r(s, a) + \sum_{s' \in \mathcal{S}_{n+1}} P_\mu(s' \mid s, a) V_{n+1}(s')$
 $V_n(s) = r(s, a) + \sum_{s' \in \mathcal{S}_{n+1}} P_\mu(s' \mid s, \pi_n(s)) V_{n+1}(s')$
 end for
end for
 Return $\pi = (\pi_n)_{n=1}^T$.

Theorem 7.2.2. *For T -horizon problems, backwards induction is optimal, i.e.*

$$V_n(s) = V_{\mu,n}^*(s) \quad (7.2.7)$$

Proof. Note that the proof below also holds for $r(s, a) = r(s)$. First we show that $V_t \geq V_t^*$. For $n = T$ we evidently have $V_T(s) = \max_a r(s, a) = V_{\mu, T}^*(s)$. Now assume that for $n \geq t + 1$, (7.2.7) holds. Then it also holds for $n = t$, since for any policy π'

$$\begin{aligned} V_t(s) &= \max_a \left\{ r(s, a) + \sum_{j \in \mathcal{S}} P_\mu(j | s, a) V_{t+1}(j) \right\} \\ &\geq \max_a \left\{ r(s, a) + \sum_{j \in \mathcal{S}} P_\mu(j | s, a) V_{\mu, t+1}^*(j) \right\} \quad (\text{by induction assumption}) \\ &\geq \max_a \left\{ r(s, a) + \sum_{j \in \mathcal{S}} P_\mu(j | s, a) V_{\mu, t+1}^{\pi'}(j) \right\} \\ &\geq V_t^{\pi'}(s). \end{aligned}$$

This holds for any policy π' , including $\pi' = \pi$, the policy returned by backwards induction. Then:

$$V_{\mu, t}^*(s) \geq V_{\mu, t}^{\pi}(s) = V_t(s) \geq V_{\mu, t}^*(s).$$

□

Remark 7.2.2. A similar theorem can be proven for arbitrary \mathcal{S} . This requires using \sup instead of \max and proving the existence of a π' that is arbitrary-close in value to V^* . For details, see[?].

7.3 Infinite-horizon

When problems have no fixed horizon, they usually can be modelled as infinite horizon problems, sometimes with help of a *terminating state*, whose visit terminates the problem, or discounted rewards, which indicate that we care less about rewards further in the future. When reward discounting is exponential, these problems can be seen as undiscounted problems with random and geometrically distributed horizon. For problems with no discounting and no termination states there are some complications in the definition of optimal policy. However, we defer discussion of such problems to Chapter ??.

7.3.1 Examples

We begin with some examples, which will help elucidate the concept of terminating states and infinite horizon. The first is shortest path problems, where the aim is to find the shortest path to a particular goal. Although the process terminates when the goal is reached, not all policies may be able to reach the goal, and so the process may never terminate.

Shortest-path problems

We shall consider two types of shortest path problems, deterministic and stochastic. Although conceptually very different, both problems have essentially the same complexity.

Consider an agent moving in a maze, aiming to get to some terminating goal state X . That is, when reaching this state, the agent cannot move anymore, and receives a reward of 0. In general, the agent can move deterministically in the four cardinal directions, and receives a

negative reward at each time step. Consequently, the optimal policy is to move to X as quickly as possible.

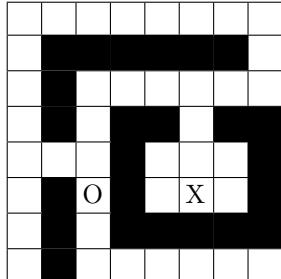
14	13	12	11	10	9	8	7
15		13				6	
16	15	14	4	3	4	5	
17			2				
18	19	20	2	1	2		
19		21	1	0	1		
20		22					
21		23	24	25	26	27	28

Properties

- $\gamma = 1, T \rightarrow \infty$.
- $r_t = -1$ unless $s_t = X$, in which case $r_t = 0$.
- $\mathbb{P}_\mu(s_{t+1} = X | s_t = X) = 1$.
- $\mathcal{A} = \{\text{North, South, East, West}\}$
- Transitions are deterministic and walls block.

Solving the shortest path problem can be done simply by looking at the distance of any point to X . Then the reward obtained by the optimal policy starting from any point, is simply the negative distance. The optimal policy simply moves to the state with the smallest distance to X .

Stochastic shortest path problem with a pit Now assume the shortest path problem with stochastic dynamics. That is, at each time-step there is a small probability ω that move to a random direction. In addition, there is a pit O , that is a terminating state with a reward of -100 .



Properties

- $\gamma = 1, T \rightarrow \infty$.
- $r_t = -1$, but $r_t = 0$ at X and -100 at O and episode ends.
- $\mathbb{P}_\mu(s_{t+1} = X | s_t = X) = 1$.
- $\mathcal{A} = \{\text{North, South, East, West}\}$
- Moves to a random direction with probability ω . Walls block.

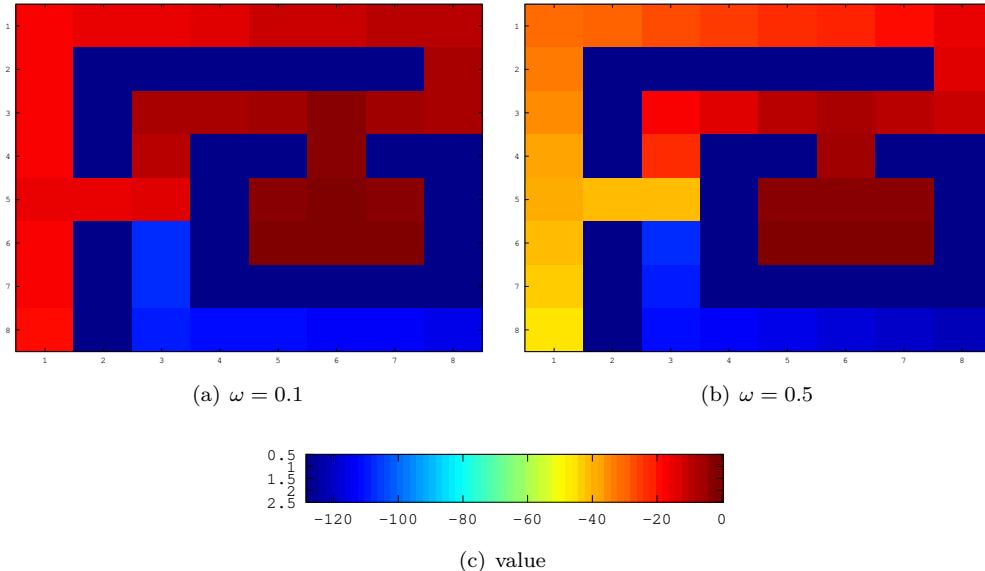


Figure 7.2: Pit maze solutions for two values of ω .

Randomness changes the solution significantly in this environment. When ω is relatively small, it is worthwhile (in expectation) for the agent to pass past the pit, even though there is a risk of falling in and getting a reward of -100 . In the example given, even starting from the third row, the agent prefers taking the short-cut. For high enough ω , the optimal policy avoids approaching the pit. Still, the agent prefers jumping in the pit, than being trapped at the bottom of the maze forever.

Continuing problems

Finally, many problems have no natural terminating state, but are continuing *ad infinitum*. Frequently, we model those problems using a utility that discounts future rewards exponentially. This way, we can guarantee that the utility is bounded. In addition, exponential discounting also has some economical sense. This is partially because of the effects of inflation, and partially because money now may be more useful than money in the future. Both these effects diminish the value of money over time. As an example, consider the following inventory management problem.

EXAMPLE 25 (Inventory management). There are K storage locations, and each location i can store n_i items. At each time-step there is a probability ϕ_i that a client tries to buy an item from location i , where $\sum_i \phi_i \leq 1$. If there is an item available, when this happens, you gain reward 1. There are two types of actions, one for ordering a certain number u units of stock, paying $c(u)$. Further one may move u units of stock from one location i to another location j , paying $\psi_{ij}(u)$.

An easy special case is when $K = 1$, and we assume that deliveries happen once every m timesteps, and each time-step a client arrives with probability ϕ . Then the state set $\mathcal{S} = \{0, 1, \dots, n\}$ corresponds to the number of items we have, the action set $\mathcal{A} = \{0, 1, \dots, n\}$ to the number of items we may order. The transition probabilities are given by $P(s'|s, a) = \binom{m}{d} \phi^d (1 - \phi)^{m-d}$, where $d = s + a - s'$, for $s + a \leq n$.

7.3.2 MDP Algorithms

Let us now look at three basic algorithms for solving a known Markov decision process. The first, *value iteration*, is a simple extension of the backwards induction algorithm to the infinite horizon case.

Value iteration

In this version of the algorithm, we assume that rewards are dependent only on the state. An algorithm for the case where reward only depends on the state can be obtained by replacing $r(s, a)$ with $r(s)$.

Algorithm 7 Value iteration

```

Input  $\mu, \mathcal{S}$ .
Initialise  $\mathbf{v}_0 \in \mathcal{V}$ .
for  $n = 1, 2, \dots$  do
    for  $s \in \mathcal{S}_n$  do
         $\pi_n(s) = \arg \max_{a \in \mathcal{A}} \{r(s, a) + \gamma \sum_{s' \in \mathcal{S}} P_\mu(s' | s, a) \mathbf{v}_{n-1}(s')\}$ 
         $\mathbf{v}_n(s) = r(s, \pi_n(s)) + \gamma \sum_{s' \in \mathcal{S}} P_\mu(s' | s, \pi_n(s)) \mathbf{v}_{n-1}(s')$ 
    end for
    break if termination-condition is met
end for
Return  $\pi_n, V_n$ .
```

The value iteration algortihm is a direct extension of the backwards induction algorithm for an infinite horizon. However, since we know that stationary policies are optimal, we do not need to maintain the values and actions for all time steps. At each step, we can merely keep the previous value \mathbf{v}_{n-1} . However, since there is an infinite number of steps, we need to know whether the algorithm converges to the optimal value, and what is the error we make at a particular iteration.

Theorem 7.3.1. *The value iteration algorithm satisfies*

- $\lim_{n \rightarrow \infty} \|\mathbf{v}_n - \mathbf{v}^*\| = 0$.
- For each $\epsilon > 0$ there exists $N_\epsilon < \infty$ such that for all $n \geq N_\epsilon$

$$\|\mathbf{v}_{n+1} - \mathbf{v}_n\| \leq \epsilon(1 - \gamma)/2\gamma. \quad (7.3.1)$$

- For $n \geq N_\epsilon$ the policy π_ϵ that takes action

$$\arg \max_a r(s, a) + \gamma \sum_j p(j | s, a) \mathbf{v}_n(s')$$

is ϵ -optimal, i.e. $V_\mu^{\pi_\epsilon}(s) \geq V_\mu^*(s) - \epsilon$ for all states s .

- $\|\mathbf{v}_{n+1} - \mathbf{v}^*\| < \epsilon/2$ for $n \geq N_\epsilon$.

Proof. The first two statements follow from the fixed-point Theorem ???. Now note that

$$\|V_\mu^{\pi_\epsilon} - \mathbf{v}^*\| \leq \|V_\mu^{\pi_\epsilon} - \mathbf{v}_n\| + \|\mathbf{v}_n - \mathbf{v}^*\|$$

We can bound these two terms easily:

$$\begin{aligned}
 \|V_\mu^{\pi_\epsilon} - \mathbf{v}_{n+1}\| &= \|\mathcal{L}_{\pi_\epsilon} V_\mu^{\pi_\epsilon} - \mathbf{v}_{n+1}\| && \text{(by definition of } \mathcal{L}_{\pi_\epsilon} \text{)} \\
 &\leq \|\mathcal{L}_{\pi_\epsilon} V_\mu^{\pi_\epsilon} - \mathcal{L}\mathbf{v}_{n+1}\| + \|\mathcal{L}\mathbf{v}_{n+1} - \mathbf{v}_{n+1}\| && \text{(triangle)} \\
 &= \|\mathcal{L}_{\pi_\epsilon} V_\mu^{\pi_\epsilon} - \mathcal{L}_{\pi_\epsilon} \mathbf{v}_{n+1}\| + \|\mathcal{L}\mathbf{v}_{n+1} - \mathcal{L}\mathbf{v}_n\| && \text{(by definition)} \\
 &\leq \gamma \|V_\mu^{\pi_\epsilon} - \mathbf{v}_{n+1}\| + \gamma \|\mathbf{v}_{n+1} - \mathbf{v}_n\|. && \text{(by contraction)}
 \end{aligned}$$

An analogous argument gives the same bound for the second term $\|\mathbf{v}_n - \mathbf{v}^*\|$. Then, rearranging we obtain

$$\|V^{\pi_\epsilon} - \mathbf{v}_{n+1}\| \leq \frac{\gamma}{1-\gamma} \|\mathbf{v}_{n+1} - \mathbf{v}_n\|, \quad \|\mathbf{v}_{n+1} - \mathbf{v}^*\| \leq \frac{\gamma}{1-\gamma} \|\mathbf{v}_{n+1} - \mathbf{v}_n\|,$$

and the third and fourth statements follow from the second statement. \square

The *termination condition* of value iteration has been left unspecified. However, the theorem *termination condition* above shows that if we terminate when (7.3.1) is true, then our error will be bounded by ϵ . However, better termination conditions can be obtained.

Now let us prove how fast value iteration converges.

Theorem 7.3.2 (Value iteration monotonicity). *Let \mathcal{V} be the set of value vectors with Bellman operator \mathcal{L} . Then:*

1. *Let $\mathbf{v}, \mathbf{v}' \in \mathcal{V}$ with $\mathbf{v}' \geq \mathbf{v}$. Then $\mathcal{L}\mathbf{v}' \geq \mathcal{L}\mathbf{v}$.*
2. *Let $\mathbf{v}_{n+1} = \mathcal{L}\mathbf{v}_n$. If there is an N s.t. $\mathcal{L}\mathbf{v}_N \leq \mathbf{v}_N$, then $\mathcal{L}\mathbf{v}_{N+k} \leq \mathbf{v}_{N+k}$ for all $k \geq 0$ and similarly for \geq .*

Proof. Let $\pi \in \arg \max_\pi \mathbf{r} + \gamma \mathbf{P}_{\mu, \pi} \mathbf{v}$. Then

$$\mathcal{L}\mathbf{v} = \mathbf{r} + \gamma \mathbf{P}_{\mu, \pi} \mathbf{v} \leq \mathbf{r} + \gamma \mathbf{P}_{\mu, \pi} \mathbf{v}' \leq \max_{\pi'} \mathbf{r} + \gamma \mathbf{P}_{\mu, \pi'} \mathbf{v}',$$

where the first inequality is due to the fact that $\mathbf{P}\mathbf{v} \geq \mathbf{P}\mathbf{v}'$ for any \mathbf{P} . For the second part,

$$\mathcal{L}\mathbf{v}_{N+k} = \mathbf{v}_{N+k+1} = \mathcal{L}^k \mathcal{L}\mathbf{v}_N \leq \mathcal{L}^k \mathbf{v}_N = \mathbf{v}_{N+k}.$$

since $\mathcal{L}\mathbf{v}_N \leq \mathbf{v}_N$ by assumption and consequently $\mathcal{L}^k \mathcal{L}\mathbf{v}_N \leq \mathcal{L}^k \mathbf{v}_N$ by part one of the theorem. \square

Thus, value iteration converges monotonically to V_μ^* if the initial value $\mathbf{v}_0 \leq \mathbf{v}'$ for all \mathbf{v}' . If $r \geq 0$, it is sufficient to set $\mathbf{v}_0 = \mathbf{0}$. Then \mathbf{v}_n is always a lower bound on the optimal value function.

Theorem 7.3.3. *Value iteration converges with error in $O(\gamma^n)$. More specifically, for $r \in [0, 1]$ and $\mathbf{v}_0 = \mathbf{0}$,*

$$\|\mathbf{v}_n - V_\mu^*\| \leq \frac{\gamma^n}{1-\gamma}, \quad \|V_\mu^{\pi_n} - V_\mu^*\| \leq \frac{2\gamma^n}{1-\gamma}.$$

Proof. The first part follows from the contraction property (Theorem ??):

$$\|\mathbf{v}_{n+1} - \mathbf{v}^*\| = \|\mathcal{L}\mathbf{v}_n - \mathcal{L}\mathbf{v}^*\| \leq \gamma \|\mathbf{v}_n - \mathbf{v}^*\|. \quad (7.3.2)$$

Now divide by γ^n to obtain the final result. \square

Although value iteration converges exponentially fast, the convergence is dominated by the discount factor γ . When γ is very close to one, convergence can be extremely slow. In fact, ? showed that the number of iterations are on the order of $1/(1 - \gamma)$, for bounded accuracy of the input data. The overall complexity is $\tilde{O}(|\mathcal{S}|^2|\mathcal{A}|L(1 - \gamma)^{-1}$, omitting logarithmic factors, where L is the total number of bits used to represent the input.¹

¹Thus the result is *weakly* polynomial complexity, due to the dependence on the input size description.

Chapter 8

Safety

Nothing here

Bibliography

- [1] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996. URL citeseer.nj.nec.com/breiman96bagging.html.
- [2] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015.
- [3] Evelyn Fix and Joseph L Hodges Jr. Discriminatory analysis-nonparametric discrimination: consistency properties. Technical report, California Univ Berkeley, 1951.
- [4] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, technical report, SRI International, 1998.

The Practice of Reproducible Research

Case Studies and Lessons from the Data-Intensive Sciences

Justin Kitzes, Daniel Turek, Fatma Deniz (Eds.)

Online version downloaded from
<http://www.practicereproducibleresearch.org>

Cite as

Kitzes, J., Turek, D., & Deniz, F. (Eds.). (2018). *The Practice of Reproducible Research: Case Studies and Lessons from the Data-Intensive Sciences*. Oakland, CA: University of California Press.

Table of Contents

Front Matter

Table of Contents

Preface

Introduction

Part I: Practicing Reproducibility

Assessing Reproducibility

The Basic Reproducible Workflow Template

Case Studies in Reproducible Research

Lessons Learned

Building Towards a Future Where Reproducible, Open Science is the Norm

Glossary

Part II: High-level Case Studies

Processing of Airborne Laser Altimetry Data Using Cloud-based Python and Relational Database Tools

The Trade-Off Between Reproducibility and Privacy in the Use of Social Media Data to Study Political Behavior

A Reproducible R Notebook Using Docker

Estimating the Effect of Soldier Deaths on the Military Labor Supply

Developing and Analyzing Exact-Diagonalization Simulations for Quantum Many-Body Systems and Creating a Provenance-Rich Publication from the Results

Validating Statistical Methods to Detect Data Fabrication

Feature Extraction and Data Wrangling for Predictive Models of the Brain in Python

Using Observational Data and Numerical Modeling to Make Scientific Discoveries in Climate Science

Analyzing Bat Distributions in a Human-Dominated Landscape with Autonomous Acoustic Detectors and Machine Learning Models

An Analysis of Household Location Choice in Major U.S. Metropolitan Areas Using R

-
- Analyzing Cosponsorship Data to Detect Networking Patterns in Peruvian Legislators
Using R and Related Tools for Reproducible Research in Archaeology
-
- Achieving Full Replication of our Own Published CFD Results, with Four Different Codes
-
- Reproducible Applied Statistics: Is Tagging of Therapist-Patient Interactions Reliable?
-
- A Dissection of Computational Methods Used in a Biogeographic Study
-
- A Statistical Analysis of Salt and Mortality at the Level of Nations
-
- Reproducible Workflows For Understanding Large Scale Ecological Effects Of Climate Change
-
- Reproducibility in Human Neuroimaging Research: A Practical Example from the Analysis of Diffusion MRI
-
- Reproducible Computational Science on High Performance Computers: A View from Neutron Transport
-
- Detection and Classification of Cervical Cells
-
- Enabling Astronomy Image Processing With Cloud Computing Using Apache Spark
-

Part III: Low-level Case Studies

-
- Software for Analyzing Supernova Light Curve Data for Cosmology
-
- pyMooney: Generating a Database of Two-Tone, Mooney Images
-
- Problem-Specific Analysis of Molecular Dynamics Trajectories for Biomolecules
-
- Developing an Open, Modular Simulation Framework for Nuclear Fuel Cycle Analysis
-
- Producing a Journal Article on Probabilistic Tsunami Hazard Assessment
-
- A Reproducible Neuroimaging Workflow using the Automated Build Tool "make"
-
- Generation of Uniform Data Products for AmeriFlux and FLUXNET
-
- Developing a Reproducible Workflow for Large-scale Phenotyping
-
- Developing and Testing Stochastic Filtering Methods for Tracking Objects in Videos
-
- Developing, Testing, and Deploying Efficient MCMC Algorithms for Hierarchical Models Using R
-

Appendix

-
- Maintaining a Reproducible Database on Political Parties, Elections, and Governments
-
- Developing R Code for the Processing and Analysis of Optic Flow Data
-

All or Nothing! Public Goods Provision under Partial versus Full Decentralization in
Indonesia

The Practice of Reproducible Research

Case Studies and Lessons from the Data-Intensive Sciences

Justin Kitzes, Daniel Turek, Fatma Deniz (Eds.)

This is the open, online version of the book *The Practice of Reproducible Research*, published by the University of California Press. Print copies of the book can be purchased at [this link](#) or from other major booksellers.

This book contains a collection of 31 case studies of reproducible research workflows, written by academic researchers in the data-intensive sciences. Each case study describes how the author combined specific tools, ideas, and practices in order to complete a real-world research project. Emphasis is placed on the practical aspects of how the author organized his or her research to make it as reproducible as possible.

The [Introduction](#) and Part I of the book present general information about working reproducibly and synthesizes common themes from across the case studies. This summary section can be read as a stand alone introduction for beginners wishing to learn more about the general practices of reproducible research. Parts II and III of the book contain the 31 case study chapters themselves.

Please cite *The Practice of Reproducible Research* as:

Kitzes, J., Turek, D., & Deniz, F. (Eds.). (2018). *The Practice of Reproducible Research: Case Studies and Lessons from the Data-Intensive Sciences*. Oakland, CA: University of California Press.

Many of the chapters in this book were written by authors affiliated with one of the three [Moore-Sloan Data Science Environments](#): the [Berkeley Institute for Data Science](#) at UC Berkeley, the [eScience Institute](#) at the University of Washington, and the [Center for Data Science](#) at New York University. The editors and authors are particularly grateful for the financial and intellectual support of the the Gordon and Betty Moore Foundation (Grant GBMF3834 to UC Berkeley) and the Alfred P. Sloan Foundation (Grant 2013-10-27 to UC Berkeley).

The contents of this book are copyright University of California Press. Please feel free to share links to this website and to use these online materials for non-commercial, educational purposes. For other uses, including redistribution or reprinting, please contact [Justin Kitzes](#).

Version History

- v1.2.1 - *repro-case-studies* 5eae781, *repro-case-private* 45dc5a6
- v1.2 - *repro-case-studies* 0b4653f, *repro-case-private* 45dc5a6
- v1.1.1 - *repro-case-studies* e77888d, *repro-case-private* bde1339
- v1.1 - *repro-case-studies* e77888d, *repro-case-private* bde1339 (UC Press Feb3)
- v1.0.2 - *repro-case-studies* ff064d0, *repro-case-private* 31ed797π
- v1.0.1 - *repro-case-studies* e7134cc, *repro-case-private* 5e63c6e
- v1.0 - *repro-case-studies* d5f5783, *repro-case-private* 5e63c6e

Table of Contents

Preface: Nullius in Verba

P.B. Stark

The origins of the scientific method, epitomized by Sir Francis Bacon's work in the early 1600s, amount to insistence on direct evidence. This is reflected in the motto of The Royal Society, founded in 1660: *Nullius in verba*, which roughly means "take nobody's word for it" (The Royal Society, 2016). Fellows of the Royal Society did not consider a claim to be scientifically established unless it had been demonstrated experimentally in front of a group of observers (other fellows), who could see with their own eyes what happened (Shapin & Schaffer, 2011). Over time, Robert Boyle and others developed conventions for documenting experiments in sufficient detail, using prose and illustrations of the apparatus and experimental set up, that the reader could imagine being in the room, observing the experiment and its outcome.

Such observability---visibility into the process of generating results---provides the evidence that the scientific claim is true. It helps ensure we are not fooling ourselves or each other, accidentally or deliberately. It is a safeguard against error and fraud, and a springboard for progress, enabling others to replicate the experiment, to refine or improve the experiment, and to leverage the techniques to answer new questions. It generates and promulgates scientific knowledge *and* the means of generating scientific knowledge.

However, science has largely abandoned that transparency and observability, resulting in a devolution from *show me* to *trust me*. Scientific publications simply do not contain the information needed to know what was done, nor to try to replicate the experiment and data analysis. Peer reviewers and journal editors, the gatekeepers we rely upon to ensure the correctness of published results, cannot possibly vet submissions well, because they are not provided enough information to do the job. There are many reasons for this regression, among them, the rise of Big Science, the size of many modern data sets, the complexity of modern data analysis and the software tools used for data analysis, and draconian limits on the length of articles and even on electronic supplemental material. But as a consequence, most scientific publications provide little scientific evidence for the results they report.

It is impractical or impossible to repeat some experiments from scratch: who can afford to replicate CERN, the Hubble Space Telescope, or the National Health and Nutrition Examination Survey? Some data sets are too large to move efficiently, or contain information restricted by law or ethics. Lack of access to the underlying data obviously makes it impossible to replicate data analysis. But even when the data are available, reliance on

proprietary software or point-and-click tools and failure to publish code make it impossible to know exactly what was done to the data to generate the figures and tables in most scientific publications.

The (unfortunately rare) attempts to replicate experiments or data analyses often fail to support the original claims (Lehrer, 2010; Open Science Collaboration, 2015) Why?

One reason is the interaction between scientific publishing and statistics. Because journals are generally uninterested in publishing negative results or replications of positive results, the emphasis is on "discoveries." Selecting data, hypotheses, data analyses, and results to produce (apparently) positive results inflates the apparent signal-to-noise ratio and overstates statistical significance. The ability to automate many aspects of data analysis, such as feature selection and model selection, combined with the large number of variables measured in many modern studies and experiments, including "omics," high-energy physics, and sensor networks, make it essentially inevitable that many "discoveries" will be wrong (Ioannidis, 2005). A primary defense against being misled by this selection process, which includes *p*-hacking and the "file-drawer effect" (Nuzzo, 2015; Rosenthal, 1979), is to insist that researchers disclose what they tried before arriving at the analysis they chose to report or to emphasize.

I would argue that if a paper does not provide enough information to assess whether its results are correct, it is something other than science. Consequently, I think scientific journals and the peer review system must change radically: referees and editors should not "bless" work they cannot check because the authors did not provide enough information, including making available the software used to analyze the data. And scientific journals should not publish such work.

A crucial component of the chain of evidence is the software used to process and analyze the data. Modern data analysis typically involves dozens, if not hundreds of steps, each of which can be performed by numerous algorithms that are nominally identical but differ in detail, and each of which involves at least some ad hoc choices. If researchers do not make their code available, there is little hope of ever knowing what was done to the data, much less assessing whether it was the right thing to do.

And most software has bugs. For instance, a 2014 study by Coverity, based on code-scanning algorithms, found 0.61 errors per 1,000 lines of source code in open-source projects and 0.76 errors per 1,000 lines of source code in commercial software (Synopsys, 2015). Scientific software is not an exception, and few scientists use sound software engineering practices, such as rigorous testing---or even version control (Merali, 2010; Soergel, 2015). Using point-and-click tools, rather than scripted analyses, makes it easier to commit errors and harder to find them. One recent calamity attributable in part to poor computational practice is the work of Reinhart and Rogoff (2010), which was used to justify economic austerity measures in southern Europe. Errors in their Excel spreadsheet led to

the wrong conclusion (Herndon & Pollin, 2014). If they had scripted their analysis and tested the code instead of using spreadsheet software, their errors might have been avoided, discovered, or corrected before harm was done.

Working reproducibly makes it easier to get correct results and enables others to check whether results are correct. This volume focuses on how researchers in a broad spectrum of scientific applications document and reveal what they did to their data to arrive at their figures, tables, and scientific conclusions; that is, how they make the computational portion of their work more transparent and reproducible. This enables others to assess crucial aspects of the evidence that their scientific claims are correct, and to repeat, improve, and repurpose analyses and intellectual contributions embodied in software artifacts.

Infrastructure to make code and data available in useful forms needs more development, but much is possible already, as these vignettes show. The contributors share how their workflows and tools enable them to work more transparently and reproducibly, and call out "pain points" where new tools and processes might make things easier. Whether you are an astrophysicist, an ecologist, a sociologist, a statistician, or a nuclear engineer, there is likely something between these covers that will interest you, and something you will find useful to make your own work more transparent and replicable.

References

- Herndon, M. A., T., & Pollin, R. (2014). Does high public debt consistently stifle economic growth? A critique of reinhart and rogoff. *Cambridge Journal of Economics*, 38, 257–279.
- Ioannidis, J. (2005). Why most published research findings are false. *PLoS Medicine*, 2(8), e124.
- Lehrer, J. (2010). The truth wears off. *The New Yorker*. Retrieved from <http://www.newyorker.com/magazine/2010/12/13/the-truth-wears-off>
- Merali, Z. (2010). Computational science: . . . Error . . . why scientific programming does not compute. *Nature*, 467, 775–777.
- Nuzzo, R. (2015). How scientists fool themselves – and how they can stop. *Nature*, 526, 182–185.
- Open Science Collaboration. (2015). Estimating the reproducibility of psychological science. *Science*, 349, 943.
- Reinhart, C., & Rogoff, K. (2010). Growth in a time of debt. *American Economic Review*, 100, 573–578.
- Rosenthal, R. (1979). The “file drawer problem” and tolerance for null results. *Psychological Bulletin*, 86(3), 638–641.

Shapin, S., & Schaffer, S. (2011). *Leviathan and the air-pump: Hobbes, Boyle, and the experimental life*. Princeton, NJ: Princeton University Press.

Soergel, D. (2015). Rampant software errors may undermine scientific results. *F1000Research*, 3, 303.

Synopsys. (2015). Coverity scan open source report 2014. Retrieved from <http://go.coverity.com/rs/157-LQW-289/images/2014-Coverity-Scan-Report.pdf>

The Royal Society. (2016). The royal society | history. Retrieved from <https://royalsociety.org/about-us/history/>

Introduction

Justin Kitzes

Think back to the first laboratory science course that you ever took, perhaps a high school or an undergraduate chemistry or biology lab. Imagine sitting down on the first day, in a new room, surrounded by new classmates, in front of a new teacher, and encountering all of the strange sights and smells around you. Perhaps there were jars containing strange substances along the walls, oddly shaped glass and metal equipment, and safety gear to protect you from some mysterious danger.

As you entered this new physical and intellectual environment, preparing to learn the foundational knowledge and skills of a new field of science, what was the first thing that you were taught? Whatever it was, we suspect that it was *not* chemistry or biology. For most of us, the first instructions in a lab course were about how to perform basic tasks like cleaning the equipment, zeroing a balance, labeling a beaker, and recording every step that you performed in a lab notebook.

What did all of these seemingly menial tasks have to do with the science that you were supposed to be learning? Although it may not have been clear right away, these steps were all designed to ensure that, when you did conduct an experiment, you would be confident in the accuracy of your results and be able to clearly communicate what you did to someone else. Together, these two factors would permit someone else to perform the same experiment and achieve the same result, verifying your findings. None of your actual experimental results would have been meaningful, or useful to others, had you not followed these basic procedures and principles.

Now jump forward again to the present, and consider the type of research work that you do today. Almost certainly, you are using methods, tools, and equipment that are significantly more complex than those that you encountered in your first lab course. If you are like most scientists today, your research is also slowly, or not so slowly, shifting away from the traditional "lab bench" of your discipline and into the rapidly expanding world of scientific computing. There is scarcely a scientific discipline today that is not being rapidly transformed by an infusion of new hardware, software, programming languages, messy data sets, and complex new methods for data analysis.

Unfortunately, however, many excellent and accomplished scientists never received even high school or undergraduate-level training in basic scientific computing skills. Many of us struggle along as best we can, trying to write code, work with uncomfortably large data sets,

make correctly formatted figures, write and edit papers with collaborators, and somehow not lose track of which data and which analysis led to what result along the way. These are difficult tasks for someone well-versed in scientific computing, much less for scientists who are trying to pick up these skills on the fly from colleagues, books, and workshops.

In one sentence, this book is about **how to take the basic principles of the scientific method that you learned at the lab bench and translate them to your laptop**. Its core goal is to provide concrete advice and examples that will demonstrate how you can make your computational and data-intensive research more clear, transparent, and organized. We believe that these techniques will enable you to do better science, faster, and with fewer mistakes.

Within the world of scientific computing practice, the techniques that we explore in this book are those that support the goal of *computational reproducibility*. For the purposes of this book, we define computational reproducibility as follows:

A research project is computationally reproducible if a second investigator (including you in the future) can recreate the final reported results of the project, including key quantitative findings, tables, and figures, given only a set of files and written instructions.

Thinking back to that first lab course, this would be equivalent to handing a notebook, a stack of equipment, and some raw materials to a classmate and asking them to arrive at the same result that you did.

There are many reasons why we believe that that practicing computational reproducibility is perhaps the key foundational skill for scientific computing. Perhaps most importantly, working towards computational reproducibility will indirectly require you to follow many general scientific best practices for all of your digital analyses, including recording all steps in your research process, linking a final result back to the initial data and other inputs that generated it, and making all necessary data and inputs available to your colleagues.

Additionally, thinking explicitly about computational reproducibility helps to move the focus of research up a level from individual activities to the entire scientific workflow. This change in perspective is becoming increasingly important as our work becomes so complex that the overarching grand perspective is not always obvious.

Finally, the computational reproducibility of an individual research project can often be substantially increased or decreased by an individual investigator, meaning that the skills that we will discuss in this book can immediately be put into practice in nearly all types of research projects. This level of control contrasts, for example, with more complex issues such as scientific replicability (see Chapter 2), which are more heavily dependent on coordination among many scientists or on institutional actions.

This book is designed to demonstrate and teach how many of today's scientists are striving to make their research more computationally reproducible. The research described in this volume spans many traditional academic disciplines, but all of it falls into what may be called the data-intensive sciences. We define these fields as those in which researchers are routinely expected to collect, manipulate, and analyze large, heterogeneous, uncertain data sets, tasks that generally require some amount of programming and software development. While there are many challenges to achieving reproducibility in other fields that rely on fundamentally different research methods, including the social sciences and humanities, these approaches are not covered here.

This book is based on a collection of thirty-one contributed case study chapters, each authored by a leader in data-intensive research. Each case study presents the specific approach that the author used to attempt to achieve reproducibility in a real-world research project, including a discussion of the overall project workflow, key tools and techniques, and major challenges. The authors include both junior and senior scholars, ranging from graduate students to full professors. Many of the authors are affiliated with one of three Data Science Environments, housed at the University of California Berkeley, the University of Washington, and New York University. We are particularly grateful to the Gordon and Betty Moore Foundation and the Alfred P. Sloan Foundation for supporting these environments, which provided the intellectual space and financial support that made this book possible.

In addition to these contributed case studies, this book also includes synthesis chapters that introduce, summarize, and synthesize the best practices for data-intensive reproducible research. Part I of the book introduces several important concepts and practices in computational reproducibility and reports on lessons learned from the thirty-one case studies. In Chapter 2, *Assessing the Reproducibility of a Research Project*, Rokem, Marwick, and Staneva outline the factors that determine the extent to which a research project is computationally reproducible. In Chapter 3, *The Basic Reproducible Workflow Template*, Kitzes provides a step-by-step illustration of a core, cross-disciplinary reproducible workflow, suitable as a standalone first lesson for beginners and as a means of framing the subsequent case study chapters.

These preliminary discussions are followed by Chapter 4, Turek and Deniz's *Case Studies in Reproducible Research*, which describes the format of the contributed case study chapters and summarizes some of their key features. In Chapter 5, *Lessons Learned*, Huff discusses common themes across the case studies, focusing on identifying the tools and practices that brought the authors the most reproducibility benefit per unit effort and the universal challenges in achieving reproducibility. Ram and Marwick's Chapter 6, *Building Towards a Future Where Reproducible, Open Science is the Norm*, includes a broad discussion of reproducible research in modern science, highlighting the gaps, challenges, and

opportunities going forward. Finally, an extended *Glossary* by Rokem and Chirigati in Chapter 7 defines, describes, and discusses key concepts, techniques, and tools used in reproducible research and mentioned throughout the case studies.

Part I of the book can be read as a standalone introduction to reproducible research practices in the data-intensive sciences. For readers wishing to learn more about the details of these practices, Part II and Part III of the book contain the thirty-one contributed case study chapters themselves, divided into high-level case studies that provide a description of an entire research workflow, from data acquisition through analysis (Part II), and low-level case studies that take a more focused view on the implementation of one particular aspect of a reproducible workflow (Part III).

This book unavoidably assumes some background on the part of readers. To make best use of this book, you should have some experience with programming in a scientific context, at least to the point of writing a few dozen lines of code to analyze a data set. If you are not yet comfortable with this task, many good books and courses on basic programming skills are currently available. We would particularly recommend the online lessons and in-person trainings provided by the groups Software Carpentry and Data Carpentry. In addition to basic programming, we presume that you have at least some familiarity with the basic principles of scientific research, and that you are either a published author of scientific papers yourself or are aspiring to be one shortly.

For those who are relatively new to computational research and reproducibility, we suggest beginning by carefully reading the chapters in Part I of the book and attempting to follow along with the basic workflow template described in Chapter 3, either exactly as presented or as adapted to a new research project of your own choosing. The case study chapters can then be skimmed, with particular attention paid to the high-level workflows in Part II. Chapter 7, the extended glossary, should be referred to regularly when encountering unfamiliar terms and concepts.

For those with more experience in computational research, particularly those who are interested in adapting and advancing their own existing research practices, we recommend focusing first on Chapter 4, *Case Studies in Reproducible Research* and then reviewing all of the case studies chapters themselves. We suggest reading the high-level case studies first, followed by the low-level case studies, with an eye towards identifying particular strategies that may be applicable to your own research problems. The *Lessons Learned* and *Building Towards a Future Where Reproducible, Open Science is the Norm* chapters will be useful in providing a synthesis of the current state of reproducible research and prospects and challenges for the future.

Regardless of your current background and skill set, we believe that you will find both inspiration and concrete, readily-applicable techniques in this book. It is always important to remember that reproducibility is a matter of degrees, and these examples will demonstrate

that while achieving full reproducibility may sometimes be difficult or impossible, much can be gained from efforts to move a research project incrementally in the direction of reproducibility.

Let's get started.

Assessing Reproducibility

Ariel Rokem, Ben Marwick, and Valentina Staneva

While understanding the full complement of factors that contribute to reproducibility is important, it can also be hard to break down these factors into steps that can immediately be adopted into an existing research program and immediately improve its reproducibility. One of the first steps to take is to assess the current state of affairs, and to track improvement as steps are taken to increase reproducibility even more. This chapter provides a few key points for this assessment.

What it means to make research reproducible

Although one of the objectives of this book is to discover how researchers are defining and implementing reproducibility for themselves, it is important at this point to briefly review some of the current scholarly discussion on what it means to strive for reproducible research. This is important because recent surveys and commentary have highlighted that there is confusion among scientists about the meaning of reproducibility (Baker, 2016b, 2016a). Furthermore, there is disagreement about how to define 'reproducible' and 'replicable' in different fields (Casadevall & Fang, 2010; Drummond, 2009; Easterbrook, 2014; Stodden, Borwein, & Bailey, 2013). For example, Goodman, Fanelli, & Ioannidis (2016) note that in epidemiology, computational biology, economics, and clinical trials, reproducibility is often defined as:

the ability of a researcher to duplicate the results of a prior study using the same materials as were used by the original investigator. That is, a second researcher might use the same raw data to build the same analysis files and implement the same statistical analysis in an attempt to yield the same results.

This is distinct from replicability:

which refers to the ability of a researcher to duplicate the results of a prior study if the same procedures are followed but new data are collected.

It is noteworthy that definitions above, which are broadly consistent with usage of these terms throughout this book, are totally opposite to the Association for Computing Machinery (ACM, the world's largest scientific computing society), which take their definitions from the International Vocabulary of Metrology. Here are the ACM definitions:

Reproducibility (Different team, different experimental setup) The measurement can be obtained with stated precision by a different team, a different measuring system, in a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using artifacts which they develop completely independently.

Replicability (Different team, same experimental setup) The measurement can be obtained with stated precision by a different team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same or a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using the author's own artifacts.

We can see the heritage of the definitions of the ACM in literature on physics and the philosophy of science (Cartwright, 1991; Collins, 1984; Franklin & Howson, 1984). In her paper on the epistemology of scientific experimentation, Cartwright (1991), presents one of the first clear definitions of the key terms: 'replicability - doing the same experiment again' and 'reproducibility - doing a new experiment'.

The definition of Cartwright is at odd with our preferred definition, from Goodman et al. (2016). This is because we trace a different ancestry in the use of the term 'reproducible', one that recognises the central role of the computer in scientific practice, with less emphasis on empirical experimentation as the primary means for generating knowledge. Among the first to write about reproducibility in this way is geophysicist Jon Claerbout. He pioneered the use of the phrase 'reproducible research' to describe how his seismology research group used computer files to enable efficient regeneration of the figures and tables in theses and publications (Claerbout & Karrenbach, n.d.). We can see this usage more recently in Stodden, Leisch, & Peng (2014):

Replication, the practice of independently implementing scientific experiments to validate specific findings, is the cornerstone of discovering scientific truth. Related to replication is reproducibility, which is the calculation of quantitative scientific results by independent scientists using the original datasets and methods. Reproducibility can be thought of as a different standard of validity because it forgoes independent data collection and uses the methods and data collected by the original investigator. Reproducibility has become an important issue for more recent research due to advances in technology and the rapid spread of computational methods across the research landscape.

It is this way of thinking about reproducibility that captures most of the variation in the way the contributors to this book use the term. One of the key ideas that the remainder of this chapter explores is that reproducibility is a matter of degree, rather than kind. This means that identifying the factors that can relatively easily and quickly be changed can

incrementally lead to an increase in the reproducibility of a research program. Identifying more challenging points, that would require more work, helps set long-term goals towards even more reproducible work, and helps identify practical changes that can be made over time.

Reproducibility can be assessed at several different levels: at the level of an individual project (e.g., a paper, an experiment, a method or a dataset), an individual researcher, a lab or research group, an institution, or even a research field. Slightly different kinds of criteria and points of assessment might apply to these different levels. For example, an institution upholds reproducibility practices if it institutes policies that reward researchers who conduct reproducible research. Meanwhile, a research field might be considered to have a higher level of reproducibility if it develops community-maintained resources that promote and enable reproducible research practices, such as data repositories, or common data-sharing standards.

This book focuses on the first of these levels, that of a specific research project. In this chapter we consider some of the ways that reproducibility can be assessed by researchers who might be curious about how they can improve their work. We have divided this assessment of reproducibility into three different broad aspects: automation and provenance tracking, availability of software and data, and open reporting of results. For each aspect we provide a set of questions to focus attention on key details where reproducibility can be enhanced. In some cases we provide specific suggestions about how the questions could be answered, where we think the suggestions might be useful across many fields.

The diversity of standards and tools relating to reproducible research is large and we cannot survey all the possible options in this chapter. We recommend that the researcher use the detailed case studies in following chapters for inspiration, tailoring choices to the norms and standards of your discipline.

Automation and provenance tracking

Automation of the research process means that the main steps in the project: transformations of the data -- various processing steps and calculations -- as well as the visualization steps that lead to the important inferences, are encoded in software and documented in such a way that they can reliably and mechanically be replicated. In other words, the conclusions and illustrations that appear in the article are the result of a set of computational routines, or scripts that can be examined by others, and re-run to reproduce these results.

To assess the sufficiency of automation in a project, one might ask:

- Can all figures/calculations that are important for the inference leading to the result be reproduced in a single button press? If not a single button press, can these be produced with a reasonably small effort? One way to achieve this goal is to write software scripts that embody every step in the analysis up to the creation of figures, and derivation of calculations. In assessment, you can ask: is it possible to point to the software script (or scripts) that generated every one of the calculations and data visualizations? Is it possible to run these scripts with reasonably minimal effort?
- Another set of questions refers to the starting point of the calculations in the previous question: what is the starting point of running these scripts? What is required as set-up steps to the calculations in these scripts? If the setup includes manual processing of data, or cumbersome setup of a computational environment, this detracts from the reproducibility of the research.

The main question underlying these criteria is how difficult it would be for another researcher to first reproduce the results of a research project, and then further build upon these results. Because research is hard, and error is ubiquitous (a point made in this context by Donoho and colleagues (2008)), the first person to benefit from automation is often the researcher performing the original research, when hunting down and eliminating error.

Provenance tracking is very closely related to automation (see glossary for definitions). It entails that the full chain of computational events that occurred from the raw data to a conclusion is tracked and documented. In cases in which automation is implemented, provenance tracking can be instantiated and executed with a reasonably minimal effort.

When large data sets and complex analysis are involved, some processing steps may consume more time and computational resources than can be reasonably required to be repeatedly executed. In these cases, some other form of provenance tracking may serve to bolster reproducibility, even in the absence of a fully automatic processing pipeline. Items for assessment here are:

- If software was used in (pre)processing the data, is this software properly described? This includes documentation of the version of the software that was used, and the settings of parameters that were used as inputs to this software.
- If databases were queried, are the queries fully documented? Are dates of access recorded?
- Are scripts for data cleaning included with the research materials, and do they include commentary to explain key decisions made about missing data and discarding data?

Another aspect of provenance tracking is the tracking of different versions of the software, and recording of the evolution of the software, including a clear delineation of the versions of the software that were used to support specific scientific findings. This can be assessed

by asking:

- Is the evolution of the software available for inspection through a publicly accessible version control system? Are versions that contributed to particular findings clearly tagged in the version control history?

Availability of software and data

The public availability of the data and software are key components of computational reproducibility. To facilitate its evaluation, we suggest that researchers consider the following series of questions.

Availability of data

- Are the data available through an openly accessible database? Often data is shared through the internet. Here, we might ask about the long-term reliability of the web address: are the URLs mentioned in a manuscript permanently and reliably assigned to the dataset? One example of a persistent URL is a Digital Object Identifier (DOI). Several major repositories provide these for data sets (e.g., [Figshare](#)). Datasets accessible via persistent URLs increase the reproducibility of the research, relative to use of an individually maintained website, such as a lab group website or a researcher's personal website. This is because when an individually maintained websites changes its address or structure over time, the previously published URLs may no longer work. In many academic institutions, data repositories that provide persistent URLs are maintained by the libraries. These data repositories provide a secure environment for long-term citation, access, and reuse of research data.
- Are the data shared in a commonly used and well-documented file format? For tabular data, open file formats based on plain text, such as CSV (comma separated values) or TSV (tab separated values) are often used. The main benefit of text-based formats is their simplicity and transparency. On the other hand, they suffer from a loss of numerical precision, they are relatively large, and parsing them might still be difficult. Where available, strongly-typed binary formats should be preferred. For example multi-dimensional array data can be stored in formats such as [HDF5](#). In addition, there are also open data formats that have been developed in specific research communities to properly store data and metadata relevant to the analysis of data from this research domain. Examples include the FITS data format for astronomical data (Wells, Greisen, & Harten, 1981), and the NIFTI and DICOM file formats for medical imaging data (Larobina & Murino, 2014).

Proprietary file formats are problematic for reproducibility because they may not be usable on future computer systems due to intellectual property restrictions, obsolescence or incompatibility. However, one can still ask: if open formats are not suitable, is software provided to read the data into computer memory with reasonably minimal effort?

- If community standards exist, are files laid out in the shared database in a manner that conforms with these standards? For example, for neuroimaging data, does the file layout follow the Brain Imaging Data Structure (Gorgolewski et al., 2016) format?
- If data are updated, are different versions of the data clearly denoted? If data is processed in your analysis, is the raw data available?
- Is sufficient metadata provided? The type and amount of metadata varies widely by area of research, but a minimal set might include the research title, authors' names, description of collection methods and measurement variables, date, and license.
- If the data are not directly available, for example if the data are too large to share conveniently, or have restrictions related to privacy issues, do you provide sufficient instructions to obtain equivalent data? For example, are the experimental protocols used to acquire the original data sufficiently detailed?

Availability of software

- Is the software available to download and install? Software can also be deposited at repositories that issue persistent URLs, just like data sets. This can improve the longevity of its accessibility.
- Can the software easily be installed on different platforms? If a scripting language such as Python or R was used, it is better for reproducibility to share the source rather than compiled binaries that are platform-specific.
- Does the software have conditions on the use? For example, license fees, restrictions to academic or non-commercial use, etc.
- Is the source code available for inspection?
- Is the full history of the source code available for inspection through a publicly available version history?
- Are the dependencies of the software (hardware and software) described properly? Do these dependencies require only a reasonably minimal amount of effort to obtain and use? For example, if a research project requires the use of specialized hardware, it will be harder to reproduce. If it depends on expensive commercial software, likewise. Use of open-source software dependencies on commodity hardware is not always possible, but when possible electing to use these increases reproducibility.

Software documentation

Documentation of the software is another factor in removing barriers to re-use. Several forms of documentation can be added to a research repository and each of them adds to reproducibility. Relevant questions include:

- Does the software include a README file? This provides information about the purpose of the software, its use and ways to contact the authors of the software (see more below).
- Is there any function/module documentation? This closely explains the different parts of the code, including the structure of the modules that make up the code; the inputs and outputs of functions; the methods and attributes of objects, etc.
- Is there any narrative documentation? This explains how the bits and pieces of the software work together; narrative documentation might also explain how the software should be installed and configured in different circumstances and can explain what order things should be executed.
- Are there usage examples? This is particularly important for scientific computing, usage examples demonstrate the kinds of transformations, analysis pipelines and visualizations that can be undertaken using the software, and provide a point of departure for new explorations using the software. Systems that allow examples to be routinely run as part of compiling the documentation are particularly useful, because they are automatically updated when the code is updated. One such system that was originally developed as part of the PyMVPA software library (Hanke et al., 2009) has been widely adopted and further developed by many other scientific Python libraries, including scikit-image (Van Der Walt et al., 2014) and scikit-learn (Pedregosa et al., 2011) and is now [its own software project](#).

Software engineering

While not all scientific software needs to apply rigorous software engineering practices, using these practices increases the reproducibility and long-term sustainability of the software, and enables expansion of the software to handle extensions of the work. While a full implementation of these practices may be challenging for smaller projects, an awareness of the problems they are intended to solve can lead to better practices in other areas of the software development process. A few guidelines for assessing the software engineering of a computational research codebase follow.

Software testing is a practice that automates the checking of smaller units of the code, in addition and in support of the automation of the full pipeline, described above (see glossary for a detailed definition and typology of software testing). Questions that can be used to

assess the testing of the code include:

- Is a large proportion of the code covered by automatic testing that verifies that the software runs properly and is relatively error-free? Analysis software is often developed to deal with cases that are common in the data analyzed, and it often implicitly embodies assumptions about these common cases. However, some unusual cases (also called "corner cases" or "edge cases") may appear in the data, and it is important for the software to produce correct results in these cases as well. One might therefore ask: are corner cases covered, in addition to the common cases?
- Is a continuous integration system set up to validate the mechanisms for software installation and to run the full complement of tests? Does this system regularly update the software dependencies, such that the software properly runs on newer versions of these dependencies? Is the system set up to maintain backwards compatibility with older versions of these dependencies, in support of dependent developments?

Further open-source and software engineering practices can help support a community of users. These include:

- [Semantic versioning](#) is a way to communicate about the development of the software, and to allow others to depend on it. Is the software regularly released under a semantic versioning scheme? Are releases communicated widely to the user community? When standard installation channels exist, such as package managers (e.g., apt-get, pip) and repositories (e.g. CRAN, PyPi) exist, are new versions of the software made available through these mechanisms?
- Are there mechanisms in place to report and track bugs in the software? When bugs are fixed, are these fixes reported in release announcements?
- While private communication can be used to help individual users of the software, these modes of communication do not scale very well to a larger community of users. Requiring such private communication sets up barriers for users to reproduce the work. Setting up a public communication channel for users of the software to ask questions about use of the software increases the reproducibility. These can include public mailing lists, forums and/or chat rooms.

Copyright issues and other data encumbrances

Creative work, such as research, is protected by copyright laws. While these laws protect the rights of creators and researchers, they can also impede the distribution and verification of their work. Work that has no license or copyright information is still protected by copyright

law. This prevents others from having any rights to reproduce the work or build upon it. Therefore, the application of an appropriate license is important in increasing the reproducibility of the work.

Data and copyright. While copyright law doesn't generally protect facts or data, it does protect the creative acts that go into selection of the data that goes into a database or compilation. To remove doubt about the copyright status of data, a license needs to be chosen. To assess reproducibility, you can ask:

- If the data is openly accessible to others, is it released under a license that would allow them to use it?
- Is the license permissive enough to allow others to build upon the work and extend it?

One set of licenses that allows data providers to control what potential users of the data may do with the data are the [Creative Commons licenses](#), and open licenses designed specifically for data sharing (Miller, Styles, & Heath, 2008). Stodden (2009) recommends the CC-0 (public domain) license for data to enable maximum flexibility for reuse.

Software and copyright The same questions apply to issues related software and copyright, with slight variations: When sharing the source code of the software for free, researchers are encouraged to provide a license which clarifies the conditions under which this code can be used, without infringing on the copyright of the author. A license which allows anybody to use the software, alter it, build upon it, include it in other software packages, and extend it facilitates reproducibility.

Permissive software licenses would allow all of the above with minimal restrictions (e.g., BSD license, MIT license). BSD licenses are unique in including a specific clause which prevents the use of the name of the software author in future derivatives, which protects the author from negative effects of unwarranted use of their software.

Copyleft licenses allow distribution and modification of the software, but require they are released under the same license. For example, if the original software is open source and free, all its copies and derivatives should be open source and free. Such license clearly restricts the use of the software within proprietary applications. For example, software developed in an academic context with a copyleft license could not be used as part of a commercial package. The GNU General Public Licence (GPL) is an example of a popular copyleft license.

- Does the software have an open-source license?
- Is this license sufficiently permissive to allow others to use the software, reproduce the results and extend them?

Proprietary information and software

Often authors may not make the data or software available due to external restrictions. We might ask the following questions to assess the effect these restrictions might have on reproducibility:

- Is the availability and use of the data encumbered through proprietary, privacy or ethical restrictions? (For example, due to presence of sensitive personal information, or customer activity records.).
- Are there trade restrictions, or issues of national security that prohibit the open distribution of the data?
- Is the software closed-source or limited in its accessibility due to funding regulations (governmental restrictions, industrial sponsor requirements, etc.)?

Although these conditions obviously limit the degree of reproducibility that might be possible, there are options to improve the reproducibility of this kind of research. For example, a simulated dataset can sometimes be provided that mimics the key attributes of the real dataset. Where the software is restricted, authors are encouraged to provide sufficient information about key algorithms so that future studies might be executed on openly available data with more accessible software.

Open reporting of results

Crucial to reproducing a study is providing sufficient details about its execution through reports, papers, lab notebooks, etc. Researchers usually aim to publish their results in journals (or conference proceedings) with the aim to broadly distribute their discoveries. However, the choice of a journal may affect the availability and accessibility of their findings. Open access journals allow readers to access articles (usually online) without requiring any subscription or fees. While open access can take many forms, there are two common types of open access publication:

green access - the journal charges a subscription fee to readers for access to its contents, but allows the author to post a version of their article (preprint/postprint) on an electronic print website such as [arXiv](#), [EPrints Archive](#), on their own website, or on a institutional repository.

gold access - the journal does not charge any fees to readers, and makes a freely accessible online version of the article available at the time of publishing. Usually the author pays an article processing charge to enable free access by readers.

Clearly gold access journals provide the easiest and most reliable access to the article. However, since there are no subscription fees to cover publishing costs at gold open access journals and articles, the author is required to pay. Often the amount is over a thousand dollars per article. Authors should check with their institution whether it provides funds for covering such fees. As a compromise, journals sometimes have an embargo on open access (delayed open access), i.e. there is a period of time during which the article cannot be freely accessed, after which either the journal automatically makes the article available or the authors are allowed to self-archive it.

Green open access is an attractive approach to making articles openly available because it is affordable to both readers and authors. According to a study of a random sample of articles in 2009 (Björk, Welling, Laakso, Majlender, & Guðnason, 2010), approximately 20% of the articles were freely accessible (9.8 % on publishers' websites and 11.9% elsewhere through search). A more recent larger study (Archambault et al., 2013) indicates that 43% of Scopus indexed papers between 2008 and 2011 were freely available by the end of 2012. It has been also shown that there is a substantial growth in the proportion of available articles. However, there are still many articles which have been given a green light for access, but they have not been self-archived. Thus it is important for authors to understand the journal's publishing policy and use the available resources (within their field, institution, and beyond) to make their work accessible to a wide audience. Many research-intensive universities, usually via the libraries, provide services to help researchers self-archive their publications.

There are many other methods for sharing research online at different stages of the work (before final results are even available). Preregistration of the hypotheses that are being tested in a study can prevent overly flexible analysis practices and HARKing (hypothesizing after results are known (Kerr, 1998)), which reduce the reproducibility of the results reported. Regular public updates can be achieved through electronic lab notebooks, wiki pages, presentation slides, blog posts, technical reports, preprints, etc. Sharing progress allows for quick dissemination of ideas, easy collaboration, and early detection and correction of flaws. Storing preliminary results and supplementary materials in centralized repositories (preregistration registries, public version control repositories, institutional reports) have potential to improve the discoverability and the availability lifespan of the works. Some important questions researchers can ask when evaluating publishing solutions include:

- Is this electronic publishing platform going to be available in 2 years? In 5 years? Longer?
- Can a simple web search on the topic recover a link to the publication and related materials?

Taking into account the sustainability and the ease of access of these solutions in the decision process is integral to improving the research reproducibility. There is also empirical evidence that publication in open access promotes the downstream use of the scientific

findings, as evidenced by an approximately 10% increase in citations (Hajjem, Harnad, & Gingras, 2006) (and see also <http://opcit.eprints.org/oacitation-biblio.html>).

Conclusion

This chapter has attempted to outline the factors that determine the extent to which a research project is computationally reproducible. We have surveyed three different aspects where reproducibility can be assessed: automation and provenance tracking, availability of software and data, and open reporting of results. For each topic we provide a set of questions for researchers to consider about their own work and stimulate discussion on how computationally reproducibility can be improved. There are many more questions that could be asked, but we have tried to confine ourselves to questions that are relevant to key hurdles in improving reproducibility. We have observed these questions to be key points in making our own work more reproducible, and in assessing the work of our peers.

A key theme of this chapter is that there are many degrees of reproducibility. Computational reproducibility exists on a long spectrum from completely irreproducible research to complete computational reproducibility, with data, software and results all available for scrutiny, use and further exploration. Our hope is that by raising these questions and discussing some of the options, researchers can identify ways to move their work a little further along the spectrum towards improved reproducibility. We recommend a pragmatic approach to assessing and improving reproducibility, making incremental improvements from project to project, keeping an eye on the shifting norms of the field and the evolving standards and norms for data formats, metadata, repositories, etc. Over time, some of the specific suggestions we have offered here may fall out of fashion or be replaced by superior options. However, the general principles that we focus on with our questions for are likely to endure beyond the technical details, and serve as useful prompts for assessing reproducibility well into the future.

References

- Archambault, E., Amyot, D., Deschamps, P., Nicol, A., Rebout, L., & Roberge, G. (2013). *Proportion of open access peer-reviewed papers at the european and world levels—2004–2011*. Science-Metrix. Retrieved from http://www.science-metrix.com/pdf/SM_EC_OA_Availability_2004-2011.pdf
- Baker, M. (2016a). 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604), 452–454.
- Baker, M. (2016b). Muddled meanings hamper efforts to fix reproducibility crisis. *Nature News*. <http://doi.org/doi:10.1038/nature.2016.20076>

Björk, B.-C., Welling, P., Laakso, M., Majlender, H., P., & Guðnason, G. (2010). Open access to the scientific journal literature: Situation 2009. *PLoS ONE*, 5(6).

<http://doi.org/10.1371/journal.pone.0011273>

Cartwright, N. (1991). Replicability, reproducibility, and robustness: Comments on harry collins. *History of Political Economy*, 23(1), 143–155. Journal Article.

Casadevall, A., & Fang, F. C. (2010). Reproducible science. *Infection and Immunity*, 78(12), 4972–4975. <http://doi.org/10.1128/IAI.00908-10>

Claerbout, J. F., & Karrenfach, M. (n.d.). Electronic documents give reproducible research a new meaning. Conference Paper, Society of Exploration Geophysicists.

Collins, H. M. (1984). When do scientists prefer to vary their experiments? *Studies in History and Philosophy of Science Part A*, 15(2), 169–174. Journal Article.

Donoho, D. L., Maleki, A., Rahman, I., Shahram, M., & Stodden, V. (2008). *15 years of reproducible research in computational harmonic analysis*. Department of Statistics, Stanford University.

Drummond, C. (2009). Replicability is not reproducibility: Nor is it good science. *Proc. Eval. Methods Mach. Learn. Workshop 26th ICML, Montreal, Quebec, Canada*. Retrieved from <http://cogprints.org/7691/7/icmlws09.pdf>

Easterbrook, S. M. (2014). Open code for open science? *Nature Geosci*, 7(11), 779–781. Journal Article. <http://doi.org/10.1038/ngeo2283>

Franklin, A., & Howson, C. (1984). Why do scientists prefer to vary their experiments? *Studies in History and Philosophy of Science Part A*, 15(1), 51–62. Journal Article.

Goodman, S. N., Fanelli, D., & Ioannidis, J. P. A. (2016). What does research reproducibility mean? *Science Translational Medicine*, 8(341), 341ps12–341ps12. <http://doi.org/10.1126/scitranslmed.aaf5027>

Gorgolewski, K. J., Auer, T., Calhoun, V. D., Craddock, R. C., Das, S., Duff, E. P., ...

Poldrack, R. A. (2016). The brain imaging data structure: A standard for organizing and describing outputs of neuroimaging experiments. *bioRxiv*. <http://doi.org/10.1101/034561>

Hajjem, C., Harnad, S., & Gingras, Y. (2006). Ten-year cross-disciplinary comparison of the growth of open access and how it increases research citation impact. *arXiv Preprint Cs/0606079*.

Hanke, M., Halchenko, Y. O., Sederberg, P. B., Hanson, S. J., Haxby, J. V., & Pollmann, S. (2009). PyMVPA: A python toolbox for multivariate pattern analysis of fMRI data. *Neuroinformatics*, 7(1), 37–53.

Kerr, N. L. (1998). HARKing: Hypothesizing after the results are known. *Personality and Social Psychology Review*, 2(3), 196–217.

Larobina, M., & Murino, L. (2014). Medical image file formats. *Journal of Digital Imaging*, 27(2), 200–206.

Miller, P., Styles, R., & Heath, T. (2008). Open data commons, a license for open data. *LDOW*, 369.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others. (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12, 2825–2830.

Stodden, V. (2009). Enabling reproducible research: Open licensing for scientific innovation. *International Journal of Communications Law and Policy*, 13, 1–25.

Stodden, V., Borwein, J., & Bailey, D. H. (2013). Setting the default to reproducible. *Computational Science Research. SIAM News*, 46, 4–6.

Stodden, V., Leisch, F., & Peng, R. D. (2014). *Implementing reproducible research*. CRC Press.

Van Der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., ... Yu, T. (2014). Scikit-image: Image processing in python. *PeerJ*, 2, e453.

Wells, D. C., Greisen, E. W., & Harten, R. H. (1981). FITS - a Flexible Image Transport System, 44, 363.

The Basic Reproducible Workflow Template

Justin Kitzes

The core of this book consists of a set of thirty-one contributed case studies, each showing an example of a scientific workflow that was designed, at least in part, to achieve the goal of reproducibility. These case studies are concerned mainly with the goal of computational reproducibility, the ability of a second researcher to receive a set of files, including data, code, and documentation, and to recreate or recover the outputs of a research project, including figures, tables, and other key quantitative and qualitative results.

The thirty-one case studies in this volume describe a wide variety of research projects, disciplines, methods, and tools. Behind this diversity, however, all of the case studies share many key principles and practices in common. In this chapter, we describe what we view as the basic, underlying reproducible research workflow that any scientist should master before continuing on to the complexities described in the case study chapters.

To demonstrate this basic workflow, this chapter walks through a complete, concrete example of perhaps the simplest realistic data-intensive research project: a regression analysis of a single tabular data set. This example is designed to provide useful background for understanding the case studies later in this book. It will also provide a self-contained introduction to the practice of reproducible research for beginning readers looking for a template to adapt to their own research needs. We particularly encourage beginning readers to work along interactively with the example in this chapter to get a feel for how a reproducible workflow can be implemented.

We begin this chapter with a general overview of three key practices that are needed to make any research project, no matter how simple, computationally reproducible. This is followed by a high-level overview of the basic reproducible research workflow. We then provide an extended example of how this workflow can be implemented in a simple research project. We conclude with some additional considerations that arise when transitioning from this simple workflow template to more complex workflows, such as those described in the contributed case study chapters.

Three Key Practices

Chapter 2 described a set of questions that can be used to assess, at a relatively fine grained level, the extent to which a research project is reproducible. At a higher level, we can summarize these recommendations in three general practices that arise repeatedly throughout all stages of a research project:

1. Clearly separate, label, and document all data, files, and operations that occur on data and files
2. Document all operations fully, automating them as much as possible, and avoiding manual intervention in the workflow when feasible
3. Design a workflow as a sequence of small steps that are glued together, with intermediate outputs from one step feeding into the next step as inputs

At a beginning level, the first of these practices largely involves placing files in a clear directory structure and creating metadata to describe them. The second is met by writing code, or scripts, to perform each step automatically, or where this is not possible, documenting all manual steps needed to complete a task at a level that would allow a second researcher to unambiguously repeat them. The third is met through the overall workflow design, especially a clear conceptualization of the different operations that need to occur sequentially and how they support each other.

Although not described in the example below, most of the contributed case studies in this book use version control software as a tool for following the first two practices above. In short, version control is used to capture a snapshot of all of a project's files at any moment in time, allowing a researchers to easily review the history of the project and to manage future changes. Version control also provides a means of documenting and tracking changes to project files in a systematic and transparent manner.

In our experience, however, many beginners find version control more difficult to learn than the other steps described below, and thus we have chosen not to include it in this basic workflow template. However, once you feel comfortable with this basic workflow, we recommend that you progress to one of the many online tutorials that can help you learn to use version control systems. We particularly recommend the tutorials on `git` available from Software Carpentry.

The Stages of the Basic Reproducible Workflow

The basic reproducible research workflow can be divided into three main stages: data acquisition, data processing, and data analysis. These three stages are preceded by activities related to system setup, and are succeeded by steps that automate the entire

workflow as much as possible. While steps such as project brainstorming and publication may also be a key part of a research workflow, the tasks that relate to ensuring a project's reproducibility fall primarily within these stages.

Before beginning a data-intensive computational research project, a computer system with the tools necessary to complete the analysis must be located and set up. These activities can be more or less involved, depending primarily on the researchers level of access to the computer and the programming language that will be used for the analysis.

The first stage of the basic workflow is data acquisition, input, or creation. This stage commonly consists of collecting data from a primary source, such as field observation, experimental research, or surveys. However, it also may include acquiring data from an existing source, through web-scraping or communication with other researchers, or generating data via simulation. Regardless of the method, the end result of this first stage is raw data.

The second stage involves processing or cleaning of the data produced in the first stage. Depending on the tools used and the author's strategies, this stage may include tasks such as manual data entry, visual data review, or systematic data manipulation or filtering using scripts or other software. At the completion of this second stage, the relevant data is digitized, cleaned, and fully prepared for analysis. Although this stage is often treated as minor, or less important, than the other two stages surrounding it, we have found that this stage often requires as much intellectual energy, and as many difficult decisions, as the other stages in this workflow.

The third stage is data analysis. The most common form of data analysis is formal statistics, but other activities in this stage include data visualization, assessing the performance of particular algorithms, and extending the data to address a hypothesis or draw a scientific conclusion. The defining attribute of this stage is that it analyzes, in some manner, the clean data produced in the second stage, and produces the desired scientific products of the research, generally quantitative results in the form of figures and tables that are incorporated into manuscripts, talks, and other forms of communication.

Finally, following the three central stages, the reproducibility of a project can be greatly enhanced through the creation of a single controller script that can automatically execute all three stages to produce a finished result. When this type of "push button" workflow is unrealistic or impossible to achieve due to project constraints, detailed documentation of all non-automated steps should be created.

Setup

The setup activities that precede the three core stages of a reproducible workflow consist first of gaining access to a computer, or several computers, to use for a project. For this simple example, we will presume that the entire analysis will occur on a personal computer for which the researcher has full administrator access.

There are generally three classes of tools that must be installed at this stage. The first of these is a shell or terminal program that allows access to the command line. The second is a plain text editor or a development environment that can be used to write code in a chosen language. The third is software allowing the user to write and execute code in a chosen a programming language. Alternatively, researchers may choose to use an integrated workflow program, such as [VisTrails](#), [Taverna](#), or [Kepler](#), although this approach will not be discussed here.

For the basic workflow that follows, Mac or Linux users can use the pre-installed Terminal program on their systems, while Windows users can work at the Command Prompt. All users should install a plain text editor, of which many are available for each platform. Finally, the examples below will make use of the R language, and users should download and install a recent version of [R](#).

More detailed information on the above installation steps, as well as basic tutorials on how to use these tools, can be found in the [Software Carpentry lessons](#).

Stage 1: Data Acquisition

The first stage in most data-intensive workflows involves the acquisition of raw data. For this example, we'll imagine a study in which we have collected field data on tomatoes being grown as part of an agricultural experiment.

Table 1 reports hypothetical measurements of the total yield of tomatoes, in kilograms per plant, produced by four plants in each of three fields having no management after planting (N), conventional management with fertilizers and pesticides (C), or organic management (O). The third column indicates whether substantial insect damage was noted on the plant leaves at the time of harvest. Of the fifteen plants marked for sampling, two of them, denoted with `NA` in the mass column, were killed before bearing fruit.

Table 1: Sample tomato data set

Field	Weight	Insect
N	5.8	Y
N	5.9	N
N	1.6	Y
N	4.0	Y
N	2.9	Y
C	12.4	N
C	11.5	N
C	9.3	N
C	NA	N
C	12.1	N
O	9.9	N
O	6.7	N
O	10.6	Y
O	3.7	Y
O	NA	N

This data should be entered into a spreadsheet program and saved as a CSV file. CSV files are a commonly used plain text format for storing tabular data, in which each row of a table is on a separate line and data for each column are separated by a comma. Plain text formats are often preferable to program-specific formats, such as XLSX, as they are more easily readable by a variety of software and by other researchers who may wish to work with this data.

Once this file is created, it should be given a name and saved in a useful location. Naming conventions vary widely between researchers, but in small projects such as this one, we recommend using names that usefully describe a file's contents, even if these are somewhat long. This table, for example, might be saved as `raw_yield_data.csv`. To avoid the possibility of errors later in the workflow, spaces, periods, and slashes should not be used in file names.

At the same time that data are saved, a metadata file should also be created and saved with it. The purpose of the metadata file is to document the source of the data and any relevant information about it. While many disciplines have standards for metadata, a minimal metadata file consists of a simple text file that describes, in plain English, where the data came from and what it describes. Such a file, which we can save as `README.txt` alongside the data file, might contain information like the following.

Data collected by undergraduate assistants to Prof John Smith at the Berkeley Field Station. All plants were located in Field 3 and chosen for measurement when approximately 12" tall. Yields were recorded in August 2015.

Field codes indicate no treatment (N), conventional (C), and organic (O). Yield is in kg, with NA indicating a plant that died prior to yield measurement. Insect damage assessed visually, Y indicates more than 25% loss of leaf area.

The question then arises of where these two files, as well as all of the subsequent files that will be part of the project, should be saved. A common convention is to place all project files in a single directory, with a single layer of subdirectories for different types of files, such as data, source code, analysis results, etc. A structure such as the below, with all files and subfolders contained in a single folder called `tomato_project`, provides a useful starting point for simple projects.

```
|-- tomato_project
|   |-- data_raw
|   |   |-- raw_yield_data.csv
|   |   |-- README.txt
|   |-- data_clean
|   |-- results
|   |-- src
```

Stage 2: Data Processing

Once raw data has been collected and placed in a project directory, it nearly always requires some form of processing or cleaning before it can be used in an analysis. This step may involve removing invalid data, subsetting the original data, removing outliers, and other similar steps. The best approach for processing a raw data set is, of course, dependent on the questions that a researcher hopes to answer with this data and the particular type of analysis planned for Stage 3.

In this example, inspection of the raw data table revealed two plants without yield measurements, which we may wish to remove from the data before any further analysis. Given a goal of eventually conducting a two-sample t-test comparing the conventional to the organic yields, we also know that we can remove the no treatment plants from the table at this stage. For a small table such as this one, removing these rows is not strictly necessary, although such subsetting can improve the efficiency of subsequent analysis of larger data sets.

To make this stage fully reproducible, every step taken to process the data must be recorded with detail fine enough that only one processed data set could result from the combination of the raw data and the set of instructions. The simplest and the recommended way to

accomplish this is to encode the instructions for data processing as computer code, in a script, that will read in the raw data, execute various processing and cleaning operations, and save the resulting processed data as a new file.

Particularly for small tabular data, it can be tempting to skip this coding step, and instead open the file in a graphical editor, such as a spreadsheet program, delete the rows or columns that are not needed, and save the resulting file. In some instances, particularly where data files are stored in a proprietary format that can only be opened by certain programs, this manual approach may be the only option. Manual data processing, however, is prone to error and makes the "push button" automated workflow described later impossible.

As is the case with all research tasks, if this step must be done manually, ensure that the processed data file is accompanied by a very detailed human readable description, saved in a text file like the metadata file, that describes every operation performed on the raw data, to the level of what menu was selected and what button clicked in what order. Remember that if someone who you have never met cannot exactly, with 100% accuracy, reproduce the processed data file from the raw data and instructions, then this step is not fully reproducible. In many ways, this instruction file is itself similar to code, although it is intended to be executed by a human reader rather than by a computer.

For this tomato yield data, we can readily write a short script that will read the raw table, remove the rows with `NA` yields and those with a field code of `N`, and save the resulting processed data. The following R commands will perform these operations.

```
yield_data <- read.csv("yield_data.csv")
clean_yield_data <- na.omit(raw_yield_data[raw_yield_data$Field != "N", ])
write.csv(clean_yield_data, "clean_yield_data.csv")
```

While exploring the data, the commands above can be entered interactively into an interpreter window. Once a procedure for data processing has been identified, however, all of these commands should be placed in a separate file that when executed, will read the raw data, process it, and save the resulting processed data file. This ensures definitively that all necessary steps to reproduce this stage of the workflow were recorded properly and can be easily repeated at will.

In the simple directory structure described earlier, scripts and other code are saved in the `src` subfolder. To ensure that a script in the `src` directory will locate and save the appropriate files in the appropriate folders, we can modify the code above to the below, which modifies the locations where the files are read and written. Note that we have also added comments describing what each line of code is intended to do.

```
### Read in the raw data, assuming we are working in the src directory
raw_yield_data <- read.csv("../data_raw/raw_yield_data.csv")

### Clean the data by removing rows with NA and where 'Field' == N
clean_yield_data <- na.omit(raw_yield_data[raw_yield_data$Field != "N", ])

### Write the clean data to disk
write.csv(clean_yield_data, "../data_clean/clean_yield_data.csv")
```

The commands above, when saved as a script `clean_data.R` in the `src` subfolder, will read the table `raw_yield_data.csv` from the `data_raw` subfolder, clean it, and save the resulting cleaned table as `clean_yield_data.csv` in the `data_clean` subfolder. The cleaned data are placed in a different subfolder from the raw data to ensure that the original, raw data are never confused with any derived data products. Ideally the raw data files should never be altered, with all changes and modifications saved to a separate file. This will ensure that you can always go back to the original data if you make a data processing decision that you regret.

To execute this script, navigate to the `src` subfolder in a terminal window and run the command `r clean_data.R`. For more information on working at the command line, see the [Software Carpentry shell tutorial](#).

The project directory should now look like the following.

```
|-- tomato_project
|   |-- data_raw
|   |   |-- raw_yield_data.csv
|   |   |-- README.txt
|   |-- data_clean
|   |   |-- clean_yield_data.csv
|   |-- results
|   |-- src
|   |   |-- clean_data.R
```

Stage 3: Data Analysis

Once data are checked in and processed, the third stage of the basic reproducible workflow is data analysis. There are, of course, many different types of analyses that may be employed here and many different types of outputs that can result, including text-based results, tables, and figures. For this example, we'll perform an unpaired two sample t-test to determine whether the mean tomato yield per plant is significantly different in the conventional and organic fields.

As with data processing, data analysis may be done manually using graphical tools, such as a spreadsheet program. This is not recommended due to the difficulty of accurately capturing all of the minute details needed to allow a second researcher to exactly repeat the analysis without errors. Data analysis may also be performed interactively, with code entered into a "live" interpreter window until a final result is reached and saved. This step is often important as a means of exploration to determine what commands should be used for the analysis. Once interactive tools have been used to explore possible approaches, however, we strongly recommend that all commands needed to perform the data analysis be placed in separate file that will save the results when executed.

The code below should be saved in a script titled `analysis.R` in the `src` directory. When run, it will read the cleaned data table, perform the desired t-test, and save the summarized results of the test in the `results` subfolder as a plain text file `test_results.txt`. Although not applicable here, any other results, such as tables and figures, should also be saved in the `results` subfolder.

```
### Load clean data, assuming we are in the src directory
clean_yield_data <- read.csv("../data_clean/clean_yield_data.csv")

### t-test of Weights by Field type: is there significant difference in
### tomato yield in the different fields?
t_test_Weight_Field <- with(clean_yield_data, t.test(Weight ~ Field))

### Write test result to plain text file
capture.output(t_test_Weight_Field, file = "../results/test_results.txt")
```

Note that several comments describing the analysis steps are included in the code above. Although the relatively simple commands here do not require extensive explanation, comments should be used liberally in all code files, as we have demonstrated in the examples here. While the code itself describes *what* operation is performed, comments should be used to describe *why*, and in a larger sense *how*, a desired analysis is being conducted. While the code itself is designed to reproduce the quantitative results of an analysis, code comments and other documentation are designed to help another researcher reproduce the thought process that went into structuring and writing code in a particular way.

At the conclusion of this stage, after the script `analysis.R` has been run in the same manner as the previous `clean_data.R` script, the project directory will appear as follows.

```
|-- tomato_project
|   |-- data_raw
|   |   |-- raw_yield_data.csv
|   |   |-- README.txt
|   |-- data_clean
|   |   |-- clean_yield_data.csv
|   |-- results
|   |   |-- test_results.txt
|   |-- src
|   |   |-- analysis.R
|   |   |-- clean_data.R
```

The `test_results.txt` file indicates that there is no detectable significant difference between the yields in the conventional and organic fields ($p = 0.104$).

Automation

At this stage, the reproducible workflow is essentially complete. We have written code that, when executed, will read and process our raw data table and save both a cleaned data table and the final results of our analysis. Most importantly, the final result of our analysis, the p-value for the comparison of the conventional and organic yields, can be reproduced by any researcher who has access to the original data and the code that we have written.

To make this workflow even easier to reproduce, a controller or driver script can be added to execute, in one step, all of the various subcomponents of the entire workflow. In this simple example, our workflow has only two steps that can be performed automatically: executing `clean_data.R` to generate the cleaned data table, and then executing `analysis.R` to perform the statistical test.

To create a single entry point that will perform our entire analysis, we can create a shell script, `runall.sh`, that we can save in the `src` directory. For this simple example, the script only contains two lines.

```
r clean_data.R
r analysis.R
```

To test out this controller script, delete the contents of the `data_clean` and the `results` directory to simulate giving a colleague only your raw data and code. From the command line, navigate to the `src` directory and run the command `sh runall.sh` to see the intermediate and final results of the workflow regenerate.

In addition to supporting reproducibility, the creation of a "push button" workflow like this has a second related side benefit, which is ensuring that any generated results are linked directly back to specific known data sets and analysis parameters. We and many of our colleagues

have been known to finish working on real projects, delete all results precisely as described above, and rerun the entire workflow using a controller script. This final step ensures that all results used in subsequent interpretation and presentation were, in fact, generated from the latest data and code in the project directory.

Conclusion

While some real world workflows are nearly as simple as the one shown here, many projects will be more complex, perhaps substantially so. The most immediate extension of the template shown here would be the need to accommodate a greater variety of file types, including many types of code files, several categories of results, binary executables, and documentation. From an organizational perspective, an additional level of subfolders can be created within folders such as `src` and `results` to organize these additional files. Subfolders such as `doc` and `bin` within the main project directory can be used to house files related to documentation, including manuscripts, and compiled binaries.

Beyond the addition of more project files, more complex projects will require more complex workflows that allow, for example, files to be shared across multiple projects, the same analysis to be run on multiple data sets or parameter combinations, analyses to be run on remote computers, etc. Many of these additional complexities are discussed in the contributed case studies in this volume.

When moving beyond the tools and techniques described above, we first recommend that you learn to integrate version control software into your workflow. Tutorials for software such as `git` are readily available online.

A second possible direction would be to try using a literate programming approach. This approach involves creating a single source document in a language such as Markdown or LaTeX, or using a "notebook" interface such as one provided by Sage or Jupyter, that contains text describing the analysis directly alongside code, figures, tables and other results of our report. In this framework, the single source document can be executed to run the code and obtain results alongside narrative description and documentation. This approach provides a self-contained file of text and code that is convenient for circulating to other readers by email or submitting for publication.

In closing, we note once again that the structure of this basic reproducible workflow, particularly the division of the workflow into the three core stages plus setup and automation, underlies all of the more complex case studies described in this volume. We encourage researchers, both beginning and advanced, to use the template in this chapter as a basic foundational framework for understanding, organizing, and creating reproducible workflows as part of real world research projects in the data-intensive sciences.

Case Studies in Reproducible Research

Daniel Turek and Fatma Deniz

Having discussed the context and the general practices of reproducible research, we will now shift focus to a collection of concrete examples of scientific research workflows, all of which strive to attain a high degree of reproducibility. These case studies of reproducible research are the foundation for our study of approaches and current best practices for achieving computational reproducibility. By studying these real-world examples, we are able to draw conclusions regarding the tools, software, and current trends of reproducible scientific research.

In this chapter, we begin by introducing the concept and format of the case studies, including the motivating factors behind the general framework of a case study. Next, we describe the methods and process of collecting the case studies from researchers spanning a range of scientific disciplines. The case studies themselves shed light upon a natural classification into two distinct categories. This classification is described, and an index of the case studies is provided. As a high level summary, we next present broad descriptions and summary statistics of the case studies. These provide insight into the currently most common tools and methodologies facilitating reproducible research. Finally, we provide some suggestions for reading the case study chapters to attain a deeper understanding of these examples. These suggestions are intended to help readers identify ideas and insights for crafting their own reproducible scientific research workflows.

What is a case study?

A case study is a comprehensive description of the computational workflow that a researcher used to complete a single, well-defined scientific research project. Each case study describes how particular tools, ideas, and practices have been combined to support reproducibility. Emphasis is placed on the *how*, rather than the *why* or *what*, of reproducible research. Each case study can be viewed as one approach among many possibilities for how a researcher approached the challenge of reproducibility.

Each case study follows a consistent, standardized format. Each begins with a short biography of the author, including their affiliation, discipline, and a brief abstract describing the subject of their case study. The body of each case study consists of the three core

sections: a workflow narrative accompanied by a flowchart diagram, a discussion of the most important tools and achievements of the workflow, and a discussion of the most significant problems encountered in achieving reproducibility.

The workflow narrative and diagram are the heart of each case study. The diagram outlines the project in a manner similar to a circuit diagram: boxes represent steps in the process, and arrows represent the flow of information into subsequent steps. Most diagrams are built around combinations of specialized tools, version controlled repositories, databases, scripts, and end products such as statistical conclusions, functional software, or scientific publication. The workflow narrative ties closely to the diagram, and explains various stages and flow of information shown in the diagram. The narrative provides an opportunity for authors to discuss topics such as the appropriate use of tools, how various steps were automated, the history of raw data, and whether the software that is used for analysis is publicly available with sufficient documentation and testing.

Following the workflow narrative and diagram, each case study highlights the main successes of reproducible research from the project. This *Key Benefits* section describes the ways in which following this reproducible workflow has improved the author's research, often by making it more efficient, transparent, and trustworthy in addition to more reproducible. This section may also discuss how the project benefited from the reproducible or open-source nature of other projects and how other researchers could reuse portions of the workflow.

Finally, in the *Pain Points* section, each case study reflects on the most troublesome obstacles encountered in the pursuit of reproducibility. These challenges may have been successfully navigated, or may still remain. Examples include data sets that could not be made publicly available, legacy code inherited from other scientists, or difficulties in collaborating with other scientists without experience or interest in reproducible research. These troublesome aspects should be equally as instructive as the successes and key tools, since they highlight the practical hurdles to producing fully reproducible research.

Case studies may also include a *Key Tools* section, which specifically points out any software or other tools that helped achieve a reproducible workflow. And finally, some case studies address several optional questions, which touch on the broader context of reproducible research and its challenges. Where provided, answers to these questions are included at the close of the case study. The optional questions posed to each author were:

- What does "reproducibility" mean to you?
- Why do you think that reproducibility in your domain is important?
- How or where did you learn about reproducibility?

- What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?
- What do you view as the major incentives for doing reproducible research?
- Are there any best practices that you would recommend for researchers in your field?
- Would you recommend any specific resources for learning more about reproducibility?

This format for case studies was designed largely before eliciting the case studies from contributing authors. This format was selected to serve several purposes. Foremost, the workflow narrative and diagram are intended to provide a clear visualisation of the end-to-end scientific workflow, as well as the author's commentary and description of this process. Either alone would not provide a comprehensive idea of their approach to achieving reproducibility. Second, the remaining sections are designed to clearly distinguish important aspects of the researchers' approach to reproducibility. While similar information may also appear in the workflow narrative, the *Key Benefits*, *Pain Points*, and *Key Tools* sections isolate these concepts, and force each author to reflect clearly on the strengths and weaknesses of their approach. Combined, these sections provide a comprehensive view of authors approach and experiences in their quest to achieve reproducibility.

Collecting the case studies

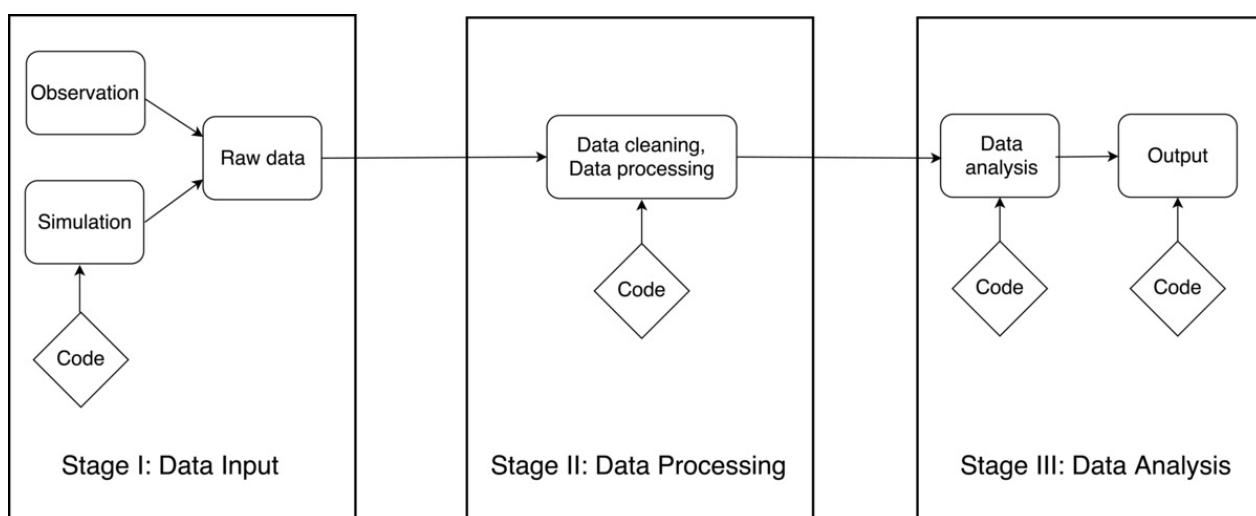
The process of collecting case studies was coordinated by a core group from the Berkeley Institute for Data Science, at the University of California, Berkeley. This process of collecting case studies spanned a period of approximately six months.

Initially, the core group drafted a general framework of a reproducibility case study. At its inception, this consisted only of the workflow diagram and accompanying narrative. Members of this group each wrote a case study describing one of their own research projects. After examining these initial submissions, a formal template for a case study was created. This consisted of the introductory biographical questions for each author, a description and guidelines for the narrative and diagram, and a set of questions regarding various aspects of reproducibility.

This template was later distributed to attendees of a Reproducibility Workshop hosted at the Berkeley Institute for Data Science. One session of this workshop gave attendees the opportunity to draft a case study describing their own research. Although attendees only had a few hours to work on their submissions during the workshop, the majority took additional time after the workshop to finalize their case study. A third and final round of case studies was later elicited through personal requests to leading scientific researchers.

Classification and Index

As described in the last chapter, a data-intensive research workflow can be divided into three main stages: data input/acquisition, data processing, and data analysis and outputs. The first stage represents data acquisition, input, or creation. Regardless of the source of the data (via collection, simulation, or otherwise), the final result of this stage is one or more raw data sets. The second stage includes both cleaning and processing of this raw data. This can include many different tasks such as consolidating, organizing, or digitizing, the output of which is a cleaned dataset fully prepared for the third stage. Finally, the third stage includes all statistical analyses, visualizations, and the creation of output products. This may frequently result in scientific publication, but many other forms of output are possible, such as software tools, public repositories, scientific conclusions, or actionable insights. An outline of a fully generic scientific workflow into these three distinct stages is shown in Figure 1.



Using this three-stage taxonomy, the case studies naturally fell into one of two broad categories. The first we called "high-level" case studies, which describe a complete scientific workflow involving all three stages. These generally provide a lighter treatment of each stage, and contain fewer technical details. The second category is called "low-level" case studies, which consists of those case studies describing only one or two of these three stages. These low-level examples generally provide a more detailed or technical treatment of the various stages. Low-level case studies are further classified by which stage(s) they describe.

Using this classification, we present in Table 1 an index of all case studies contained in this book. Each case study is classified as either high-level or low-level, and according to the scientific discipline from which it is drawn. This index is intended to help guide readers in their exploration of the case studies.

Table 1: Guide to case study chapters

Author	Discipline	Topic
HIGH LEVEL		

Anthony Arendt	Applied Physics	Impact of glacial melt on rising global sea levels
Pablo Barberá	Political Science	Studying political polarization on social media websites
Carl Boettiger	Theoretical Ecology	Forecasting and decision-making in ecological systems
Garret Christensen	Economics	Causal impacts of military history on soldier recruitment
Jan Gukelberger	Physics	Diagonalization simulations for quantum systems
Chris Hartgerink	Applied Statistics	Validating statistical methods to detect data fabrication
Chris Holdgraf	Neuroscience	Feature extraction for predictive models of the brain
David Holland	Applied Mathematics	Climate change and melting of the great ice sheets
Justin Kitzes	Ecology	Analyzing bat activity using autonomous acoustic detectors
Andy Krause	Civil Science	Analysis of US household locations in metropolitan areas
José Manuel Magallanes	Political Science	Using bill cosponsorship data to detect political trends
Benjamin Marwick	Anthropology	Understanding prehistoric hunter-gatherer behaviour
Olivier Mesnard	Aerospace Engineering	Full replication of computational fluid dynamics results
K. Jarrod Millman	Statistics / Psychology	Assessing reliability for human classification of autism
K.A.S. Mislan	Environmental Science	Comparison of blood-oxygen binding characteristics
Kellie Ottoboni	Statistics / Public Health	Analyzing association of salt consumption and mortality
Karthik Ram	Data Science	Developing tools to support stages of reproducible research
Ariel Rokem	Neuroscience	MRI studies of brain structure and function
Rachel Slaybaugh	Nuclear Engineering	Numerical methods to study neutral particle interactions
Daniela Ushizima	Image Processing	Devising machine vision and pattern recognition algorithms

Zhao Zhang	Computer Science	Image processing with cloud computing using Apache Spark
LOW LEVEL		
Kyle Barbary	Cosmology	Analyzing supernova data to measure universe expansion
Fatma Deniz	Image Processing	Generating two-tone Mooney images to study brain activity
Konrad Hinsen	Molecular Biophysics	Analysis of molecular dynamics trajectories for biomolecules
Kathryn Huff	Nuclear Engineering	Simulation framework for nuclear fuel cycle analysis
Randy LeVeque	Applied Mathematics	New approaches to probabilistic tsunami hazard assessment
Tara Madhyastha	Neuroscience	Neuroimaging workflow using automated build tool
Gilberto Pastorello	Computer Science	Data processing pipelines and data management solutions
Russell Poldrack	Neuroscience	Analysis of neuroimaging, behavioral, and metabolomic data
Valentina Staneva	Mathematics	Developing stochastic filtering methods for tracking objects
Daniel Turek	Statistics	Developing and testing efficient statistical algorithms

Trends among the case studies

Despite representing a wide range of scientific disciplines, many similarities exist between the various case studies. Here, we summarize several of the main trends and the emergent characteristics which can be observed. This includes a summary of the main languages used for computational research, trends in data sharing and version control, and other high level properties.

This book contains a total of 31 contributed case studies of reproducible workflows. Among them, 21 are high-level workflows describing the end-to-end process including data input or acquisition, data processing, and data analysis. The remaining 10 case studies are low-level workflows, which provide greater detail on one or two of these stages. Approximately one third of the low-level case studies discuss data input or acquisition (Stage 1), half describe

data processing (Stage 2), and half discuss data analysis (Stage 3). Note that some low-level case studies cover two of these stages, for example both data processing and data analysis.

Each of the 31 case studies represent a data-centric computational scientific workflow, and therefore describes various tools or languages for data management, data processing, or scientific computing. Although myriad computational tools are described, a few are extremely widely represented among the case studies. In particular, 17 of the 31 case studies (55%) make use of Python, an open-source, high-level and general-purpose programming language. This accurately reflects the current popularity of Python, thanks to its rapid development cycle, the high readability of Python code, and the extremely wide range of applications supported by Python. The next strongest representation is of R, an open source programming language for statistical computing, which is used in 13 of the 31 case studies (42%). This is an accurate representation of the wide-spread use of R among data analysts, and generally the statistics community at large, as R is now considered the primary ecosystem for statistical computing. Following Python and R, a vast range of other programs and computational tools have a comparatively modest representation among the case studies. To name just a few of the more mainstream tools, these include C/C++, MATLAB, Julia, Scala, Java/JavaScript, and oftentimes custom-developed software, although this listing is far from comprehensive.

Appropriate use of version control is a key aspect of modern reproducibility. This applies equally for software development and the computer code underlying computational workflows. Older (centralized) version control systems were more cumbersome for users, but the recent introduction of git and GitHub have made version control more accessible for smaller-scale projects. The vast majority (over 80%) of the case studies make use of git and GitHub for version controlling the development of software or analysis code, which represents one of the strongest trends among the case studies. A number of the remaining case studies explain that the nature of the workflow is not appropriate for version control, for example when describing a protocol for data management. Further, a few case studies make use of other version control software -- for example, Bitbucket or SVN -- but these represent a small minority.

In support of transparency and reproducibility, there is an on-going shift within academic communities in support of open data and data sharing. Indeed, 19 of the 31 case studies (over 60%) make use of publicly available data, or themselves describe the process of making their data publicly available. However, an open-data policy is not universally practiced, as in some disciplines the extreme overhead of data collection deters scientists from openly sharing it. That is an unfortunate reality in some fields, for example cosmology, astrophysics, or neuroscience, but the current trend among the scientific and academic communities is strongly moving towards the use of open data.

There is also a clear trend in the output medium of the case studies, although we believe this may be an artifact of the contributing group of authors rather than of reproducible workflows in general. The collection of case studies was drawn from the academic community, where primary emphasis is placed on scientific publication. All but a few of the case study workflows culminate in producing a scientific manuscript intended for peer-reviewed publication. Perhaps more important, slightly over one third of the case studies also describe a second output. This is typically manifested as a software product, or an analysis algorithm intended for wider use. Other, less common, secondary outputs include data management pipelines, or interactive websites.

Reading the case studies

As readers consider the design of their own reproducible scientific workflow, a wealth of knowledge and experience is available in the case studies presented at the end of this book. However, reading the case studies may be daunting, as many are technical and may assume familiarity with computational tools or specific application domains. For this reason, we now provide some suggestions for reading the case studies.

We encourage readers who are new to reproducible research to begin by skimming through the high-level case studies, which provide a general overview of research workflows from a variety of disciplines. This will provide a general idea of what is contained in the case studies, and may highlight disciplines that have faced, or have solved, similar challenges to those faced by the reader. Ecologists and cosmologists, for example, both often work with high-resolution spatial data, while neuroscientists and empirical economists may encounter similar issues surrounding data anonymization.

As readers become familiar with the format and presentation of the case studies, they might next consider a detailed reading of the case studies drawn from the most closely related disciplines to their own. In these examples, the nature of the scientific research is more likely to be familiar to the reader. In addition, they are likely to give an idea of what tools, challenges and approaches are being used in one's own discipline.

Finally, the motivated reader is encouraged to undertake detailed readings of both high-level and low-level case studies which address the tools or issues most closely related to your own research. Case studies will invariably discuss technical tools, topics, and methods that will not be familiar to you. Rather than including explanations of these technical concepts in each chapter, we have provided descriptions of the most common terms and tools in a technical glossary at the end of the book. Readers are encouraged to refer to this glossary frequently while reading through the case studies.

It is important to note that each case study is a problem-specific example of a reproducible workflow. Rather than attempting to recreate any particular workflow, ideas should be selected from a variety of case studies to create your own customized approach to reproducibility. However readers decide to navigate the collection of case studies, they should keep in mind that every case study has some useful insights to offer -- including those drawn from unrelated disciplines. We encourage readers to study a variety of the workflows presented, since this approach is most likely to give a flavor of the common techniques and best practices generally applicable to scientific research.

Lessons Learned

Kathryn Huff

Although the case study authors came from a variety of research backgrounds, a set of themes emerged out of this collection of their workflows. Similar struggles arose despite differing scientific fields (ecology, neuroscience, astronomy, nuclear engineering) and nearly irrespective of preferred programming language (i.e. R, Python, C++, Matlab). This chapter will summarize some of the common themes among the case studies, both painful and positive.

It should be noted that the sample of chapter contributors is not a representative sample of scientists in these research fields. Indeed, these scientists contributed to the book because they are particularly interested in open science and reproducibility. Accordingly, we can imagine that where these scientists have pain points, many of their colleagues may give up on reproducibility outright.

Some key findings include the optimistic observation that git and GitHub are nearly ubiquitous among the case study authors. Additionally, we saw that among respondents, scripting analysis wherever possible is widely accepted as essential. For both reasons, plain text file formats were preferred.\ Testing and continuous integration was seen as crucial to maintaining reproducibility by those who have integrated these steps into their process, but quite a few respondents didn't mention either practice as part of their workflow. Finally, some of the most successful approaches were those that fundamentally recognize and adapt to the "ubiquity of error" that the scientific method defends against (Donoho, Maleki, Rahman, Shahram, & Stodden, 2009).

Obstacles to reproducibility included issues with humans, computers, and the institutions that both inhabit. The case studies made clear, for example, that humans must be incentivized to spend time on tasks intended solely for reproducibility. This can be complicated by skill variation and disparate tool familiarity within research groups. But, even when the tools are used, a lack of access to restricted data or hardware can hobble reproducibility efforts of even the most determined scientists. Similarly, portability of one's workflow is still a challenge for those intent on openness, since packaging - especially installation of dependencies - remains a critical stumbling block to sharing and extending work.

In the following sections, this chapter will discuss the lessons we learned from the case studies. First, this chapter will briefly mention how various scientists perceive reproducibility, then it will focus on the pain points. Next we will make note of oft-mentioned workflow tools, and finally this chapter will note some novel ideas that the case study authors had up their sleeves.

The Meaning of Reproducibility

The case study authors were prompted to give many perspectives as they prepared their case studies. One was "Define what the term *reproducibility* means to you..."

Some authors teased out quite a bit of the subtlety embedded in this semantic question. K. Jarrod Millman, Kellie Ottoboni and Philip Stark, for example, broke down reproducibility into four distinct types.

1. *Computational reproducibility and transparency*, which emphasizes code documentation.
2. *Scientific reproducibility and transparency*, which emphasizes documentation of scientific decisions and accessibility of data.
3. *Computational correctness and evidence*, which emphasizes automated testing and validation.
4. *Statistical reproducibility*, which emphasizes transparency of data analysis the logical path to scientific conclusions.

Most authors, however, expressed some flavor of either computational reproduciblity or replicability.

Computational Reproducibility

There was general agreement among most authors about at least one aspect of what reproducibility means: that when provided with identical source code, input data, software, and computing environment configurations, that an independent party can exactly reproduce the results of the original work -- especially published results. This is described in our glossary as *computational reproducibility*.

This aspect of reproducibility was articulated particularly well by the following case study authors -- although each definition has its own interesting subtleties:

Jan Gukelberger:

In general, given a publication (in a refereed journal), source codes and raw data (which might be available publicly or in the institute's repositories), an expert from my field should be able to understand, and in principle repeat, every step of the study from the running of the correct version of the simulation code to the final results presented in the published paper.

Justin Kitzes:

I consider a study to be (computationally) reproducible when I can send a colleague a zip file containing my raw data and code and he or she can push a single button to create all of the results, tables, and figures in my analysis.

Andy Krause:

"Reproducibility" means that a subsequent interested party can openly access the data, code, analytical workflow and data provenance to re-create the research (and ideally produce identical results) WITHOUT consulting the original researcher(s).

These echo a well-established perspective on reproducibility (Donoho et al., 2009; Stodden, 2010; G. Wilson et al., 2014) that is evolving as a community norm through checklists and pledges such as the "Reproducibility PI Manifesto" (Barba, 2012). A few of our case study authors have taken this pledge in which a PI vows to adopt practices that add a level of sustainability and extensibility to reproducible work:

1. Teaching group members about reproducibility
2. Maintaining all code and writing under version-control
3. Carrying out verification and validation and publishing the results
4. For main results in a publication, sharing data, plotting scripts, and figures under CC-BY
5. Uploading preprints to arXiv at the time of submission of a paper
6. Releasing code no later than the time of submission of a paper
7. Adding a "Reproducibility" statement to each publication
8. Keeping an up-to-date web presence

The importance of this sustainable, extensible kind of reproducibility was noted by Kyle Barbary:

To me, reproducibility has two facets: the availability of usable software (preferably under an open-source license), and the availability of data (preferably in both raw and reduced forms). Together, these should give an outsider the ability to reproduce the results of a study from start to finish. I separate these two aspects because each can be beneficial without the other. For example, even without releasing data, it can still be quite beneficial to release software. If released under an open-source licence, this provides a different flavor of reproducibility - the ability to reproduce an algorithm described in a paper and use and improve that algorithm in subsequent work.

Replicability

When the final conclusions can be confirmed based on a different experiment, scientists consider this validation of the result. In this vein, Valentina Staneva distinguishes between exact and approximate reproducibility:

"Exactly reproducible" - when a result can be regenerated exactly as suggested given the same set of inputs and parameters.

"Approximately reproducible" - when a result or similar performance can be generated with similar or different methods than the one proposed on the same or possibly slightly different data.

Some have used the term "replicability" for this approximate reproducibility. Ariel Rokem put it this way:

A higher standard, sometimes called 'replicability' would be to require that the same conclusions be reached if another group of researchers were to do the same experiments, and implement the same ideas in their analysis. Reproducibility does not guarantee replicability [Leek and Peng, 2015]. Some may even argue that reproducibility and replicability may sometimes be in conflict, because implementation errors can be propagated in reproduction, but not in replication [Peng2009, Baggerly2005].

Validation of a scientific result is achieved in this way when one can repeat the scientific work with a new method or implementation and draw the same conclusions.

Pain Points

We also asked the case study authors which features of their workflows presented challenges to reproducibility. Irrespective of the type of reproducibility being sought, we hoped that these pain points would reveal areas of particular need - workflow bottlenecks where innovation might improve the experience of reproducible science. The following

sections highlight some of these frustrating, time consuming, opaque, or fragile obstacles and mentions when they may represent high priority needs for better tools and improved strategies.

People and Skills

Research teams are diverse. Computational skills especially vary dramatically from one researcher to another even within the same lab. The blinding pace of innovation in software tools means that even well-prepared collaborators can't expect to always keep up with the newest tools. Manuscript preparation software, database formats, and version control systems used by one scientist may be equally modern but nonetheless incompatible with the software stack familiar to their collaborators.

When the case study authors reported that the bottleneck to adopting practices was related to a diversity of skills, the indication was universally that the process might have been more efficient or reproducible were there a greater and more homogeneous distribution of tool familiarity among their research group members. Time was wasted when simple tasks like communicating results or simultaneously editing a manuscript were crippled by one or more collaborators unfamiliar with tools used by their colleagues.

The concern also extended far beyond mere efficiency. One case study author noted that if a collaborator is unable to use the tools that are being employed, then they are at risk of being disenfranchised from the scientific process. This disenfranchisement is especially ethically problematic if a collaborator is unable to directly or simultaneously edit a co-authored manuscript due to their lack of familiarity with the processing tools (e.g. LaTeX.)

A scientist unwilling to disenfranchise their collaborators could certainly elect to use more widely used tools, accepting frustration with inefficiency as the price of collaboration.

However, the price is often paid in reproducibility as well when those widely-used, lowest-common-denominator tools conflict with reproducibility goals. This is especially the case with tools such as Microsoft Word, Excel, or Matlab which were noted as particularly problematic fallbacks, as their closed-source GUI-based nature is fundamentally fragile to reproducibility issues.

So, in the interest of both reproducibility and efficiency, some case study authors were inclined to proceed with the use of preferred tools (e.g. LaTeX) nonetheless. Those scientists largely saw the pain point caused by a\ difficulty of communication with and understanding from their peers. The ethical quandary for those scientists competed with the commitment to more effectively communicate their results (reproducibly and transparently) with the larger scientific community -- even if it had the effect of hobbling communication internally.

Need: Better education of scientists in more reproducibility-robust tools.

Need: Widely used tools should be more reproducible so that the common denominator tool does not undermine reproducibility.

Dependencies, Build Systems, and Packaging

Just as scientists "stand on the shoulders of giants," our software perches upon forests of dependency trees. A single step in our workflow may rely on dozens of libraries and scientific software packages which may each, in turn, rely on many other libraries and packages.

Accordingly, the first obstacle for use, sharing, and adoption of any software stack or analysis workflow is often the battle to simply get the workflow running on a different machine than that on which it was created. This first obstacle easily becomes the last for busy scientists, and halts reproducibility in its tracks. If another scientist can't even install LAPACK with your special compiler flags, they have no hope of building, installing, or running your software pipeline. Reproducing or extending your work becomes an unreachable dream.

This packaging problem varies in magnitude and complexity from field to field. Where some software may require a cross platform build system capturing the compiler flags for weaving together a fleet of system libraries, other software simply requires a download and some documentation of the steps to execute. These case studies spanned the gamut therein.

Many case study authors noted that their data analysis pipeline relied on a specific computational platform or build environment. Their dependencies may be limited to certain platforms or features of the build environment may need to be customized for various options to function. These environment issues quickly become too complex to manage as a manual feat and must be packaged in a robust, cross-platform way if they have any hope of succeeding.

Case study authors packaged their work in myriad ways. Lightweight strategies were often fragile to cross-platform-configuration issues (e.g. bash scripts and makefiles). More robust solutions (e.g. virtual machines), however, are usually more clunky and often less transparent. While a one-click-download runnable virtual machine may be the most reliable option for replicating a simulation, it is also the most opaque to the user. Compromise solutions such as configuration and build systems (e.g. CMake) are often simultaneously clunky and transparent. Notably, there are subtle differences between these solutions. In particular, build and configuration systems are often more bespoke (and fragile) than broader (and often rigid) packaging systems like conda/bundler/packrat.

It's now thirty years after the invention of autotools, but cross platform configuration, build, and packaging systems are not yet a solved problem. Somehow, scientific software developers still await a robust and universal solution. Thankfully, there is hope. The tech

industry, facing similar issues, has developed portable container management systems (Bernstein, 2014) such as [Docker](#) and [Kubernetes](#). These technologies are enabling a new generation of scientific packaging tools for reproducible and open science (e.g. [tmpnb](#), [binder](#), ReproZip (Chirigati, Shasha, & Freire, 2013), Code Data Environemnt (Guo, 2012), etc.).

Need: Improved configuration and build systems for portably packaging software, data, and analysis workflows.

Hardware Access

The build system situation is a special small-scale case of a larger problem of variable hardware access. At that end of the spectrum, there are challenges getting a workflow running on your collaborator's laptop in addition to your own. At the other extreme is access to high performance computing hardware or unique experimental devices.

The Large Hadron Collider is often brought up as an example of an experiment so large that it will never be repeated in a different experimental location on a comparable device, so in some ways it fails the verifiability test. Many fields suffer this on a small scale at the data collection stage, where, for example, an experiment can't be reproduced by just anyone; it must be reproduced by someone with a similarly configured MRI machine.

This hardware access issue was noted at the data collection step in the case studies, but it was also noted at the analysis step. That is, when scientific simulations or large scale data analysis is conducted on very high performance, high capacity, or high throughput computer systems, it is similarly vulnerable to being irreproducible. Not only is access to such resources limited and the hardware often unique, but the sheer amount and variety of metadata likely needed to reproduce results without complication is often enormous. While high throughput computing is being democratized by cloud computing resources, the capability computing of high performance computing machines, necessary for some applications, is not yet replaced by those services. Even cloud computing has limitations, as noted by case study author Arendt, whose collaborators found connecting to cloud servers was limited by bandwidth constraints in Alaska.

Need: Reproducibility at scale for high performance computing. A detailed discussion of this need can be found at (Hunold & Träff, 2013)

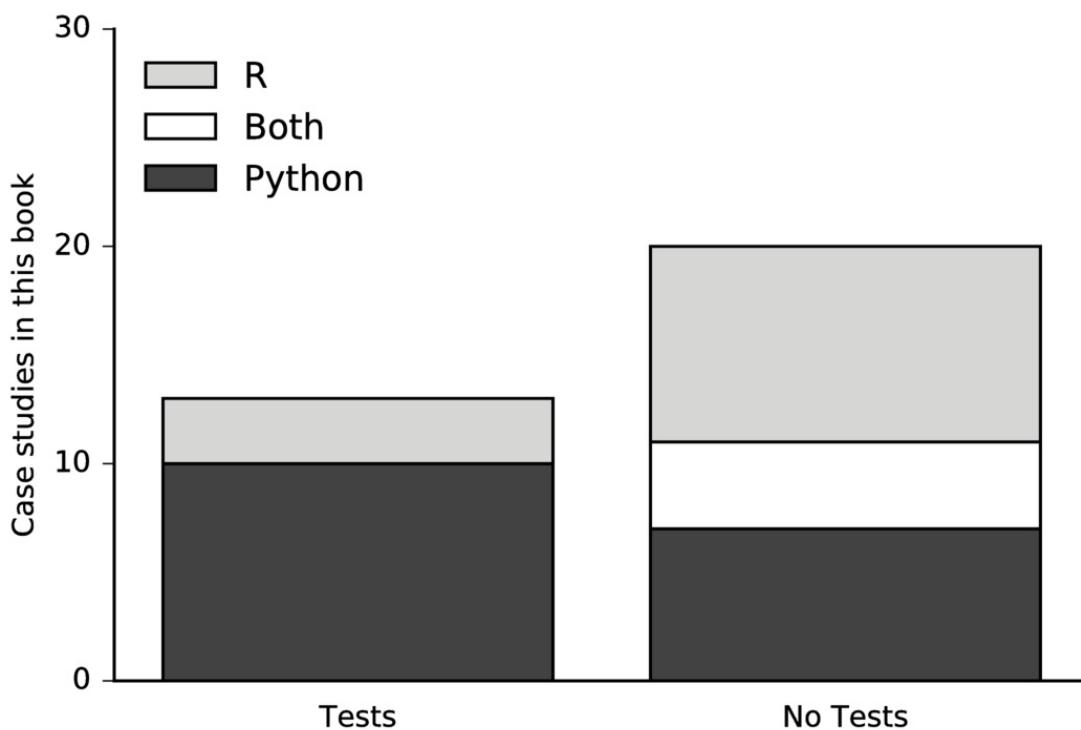
Need: Standardized hardware configurations and experimental procedures for limited-availability experimental apparatuses.

Testing

Many case study authors who developed code beyond simple scripts discussed testing that code systematically. This practice, in addition to being hygienic and improving robustness, is a type of self-check on reproducibility. Software tests and continuous integration in particular allow software authors to automate frequent checking that code consistently performs as expected even as new features are added.

Experiences varied. Some consider tests to be a core element in their workflows and emphasized unit testing -- comprehensive atomic tests at the function level -- for providing confidence that their work can be reproduced. Other case study authors perform integration tests before trusting a result, but not as an active part of development. In all cases, when tests were mentioned in the case studies, the authors were convinced of their utility at saving time and energy. However, not all case studies mentioned tests at all. One is left to wonder what is stopping those and other scientists from adopting these practices. Perhaps they are discouraged by the perceived effort of unit testing or the trade-offs of time spent now versus time saved later. In many cases, a lack of familiarity with unit testing may be the barrier.

It is also worth noting that the use of testing frameworks varied by language more than by scientific domain. In particular, the case study authors who used primarily Python reported testing at a much higher rate than case study authors who rely primarily on R. This is summarized in the figure below.



Of course this is not a statistically significant sample, so conclusions about these communities are somewhat premature, but resulting conversations have indicated to us that this difference in community adoption of testing practice may indeed be present.

Need: Better understanding of why researchers don't respond to the delayed incentives of unit testing as a practice.

Need: Norms encouraging greater adoption of unit testing irrespective of programming language.

Publishing

The most universally shared step in the research process is publication. In a literate programming sense, papers can be integrally automated and "runnable." For the majority of the case studies, however, the production of the research paper capturing the work was reported to be somewhat independent of the science. In other cases a more literate programming method was adopted through the use of Jupyter notebooks or judicious use of rmarkdown.

The workflows were very tool-driven in the sense that the tools used defined the way that the workflow progressed. The factions within the case studies included a LaTeX-based group, a knitr/rmarkdown/sweave contingent, and a frustrated Microsoft Word contingent.

Microsoft Word track changes deserves a special place in this discussion, so ubiquitous is its use worldwide and so consistent were the experiences of the four who mentioned it. Each scientist that mentioned Word had effectively two things to say:

- (1) We used it because a collaborator couldn't figure out LaTeX and
- (2) Track Changes certainly tracks changes, but is frustrating because the merging limitations mean that edits must be made in series rather than in parallel.

A reproducible paper is a large and varied task, perhaps demanding its own separate workflow.

Need: Broader community adoption around publication formats that allow parallel editing (i.e. any plain text markup language that can be version-controlled in a distributed manner.). Tools such as Overleaf and SageMathCloud are a beginning toward making LaTeX more approachable, but greater adoption is needed.

Data Versioning

Mere data storage is not always sufficient for the purpose of reproducibility. Occasionally, data may need to be versioned so that changes can be tracked, evaluation and cleaning steps can be rewound, and work can be extended.

Case study authors often either noted they were not versioning their data or noted that they were struggling to find a good way to do so. Since this challenge -- versioned storage of larger (or more varied, or high velocity) data -- is currently being encountered in the "big data" age of the software industry, active innovation in industry and new solutions are already being implemented.

Tools being developed to streamline the process of data versioning include GitHub Large File Service, Dat, git annex, datalad and others. While these operate in different ways, they typically involve compression, tracking of changes, and efficient retrieval.

Need: Greater scientific adoption of new industry-led tools and platforms for data storage, versioning, and management.

Time and Incentives

Perhaps the most vexing impediment to reproducibility the case study authors and their collaborators suffered from was a lack of time, incentives, or both.

Some case study authors, perhaps as a symptom of their recognition of the importance of reproducibility, were incentivized by confidence in the efficiency of these practices. Many noted the time they saved when repeating calculations, making modifications to analysis, and extending past work.

Conversely, the sentiment that ``time and efforts spent on creating reproducible research are not very well rewarded" (case study author Dr. Valentina Staneva) was echoed by a few authors. This need for additional reward, support, and recognition for reproducible work is an institutional infrastructural issue especially in the academy where metrics for promotion and tenure are tied explicitly to papers and often fail to account for reusable software.

But, while the promotion and tenure process is in need of modernization, funds-granting organizations are moving faster. Private foundations like Moore, Sloan, and Helmsley are leading the charge by supporting large initiatives directed at scientific software reproducibility and transparency. Similarly, government institutions like NSF, NIH, and (less quickly) DOE, are incorporating requirements for openness and data planning as well. Similarly, some journals are implementing data and software submission requirements to incentivize reproducibility at the publication stage. The efficacy of these increased standards for publication and funding, however, depend fundamentally on the enforcement mechanism.

For the incentives to compel action, paper referees must be willing to give due diligence by trying to run the submitted code and grant performance reviewers must be similarly be willing to review data management plan compliance.

Need: Increased community recognition of the benefits of reproducibility.

Need: Incentive systems where reproducibility is not only self-incentivizing.

Data restrictions

In the same way that transparent analysis is core to reproducing scientific work, access to raw data can also be essential for reproducibility. Indeed, it can be necessary for confirming conclusions during review and exploring alternative methods during extension by other scientists. In some fields, however, data access is legally restricted. Some such data restrictions concern human subjects research, such as survey and private medical data. Some restrictions concern national security, such as the restriction of export controlled nuclear data or risk map data. Researchers in these fields are therefore limited in their ability to do completely open science, but can often, behind the export control or IRB wall, share analysis methods with colleagues who do have access to the data.

Need: Standards around scrubbed and representational data so that analysis can be investigated separate from restricted data sets.

Actionable Recommendations

We are not the first to discuss reproducibility. Nor shall we be the last. Thus, many themes were repeated among the recommendations of the scientists:

- version control your code
- open your data
- automate everywhere possible
- document your processes
- test everything
- use free and open tools

Less common refrains that are already well established best practices within the current reproducibility climate include:

- avoid excessive dependencies
- when dependencies can't be avoided, package their installation

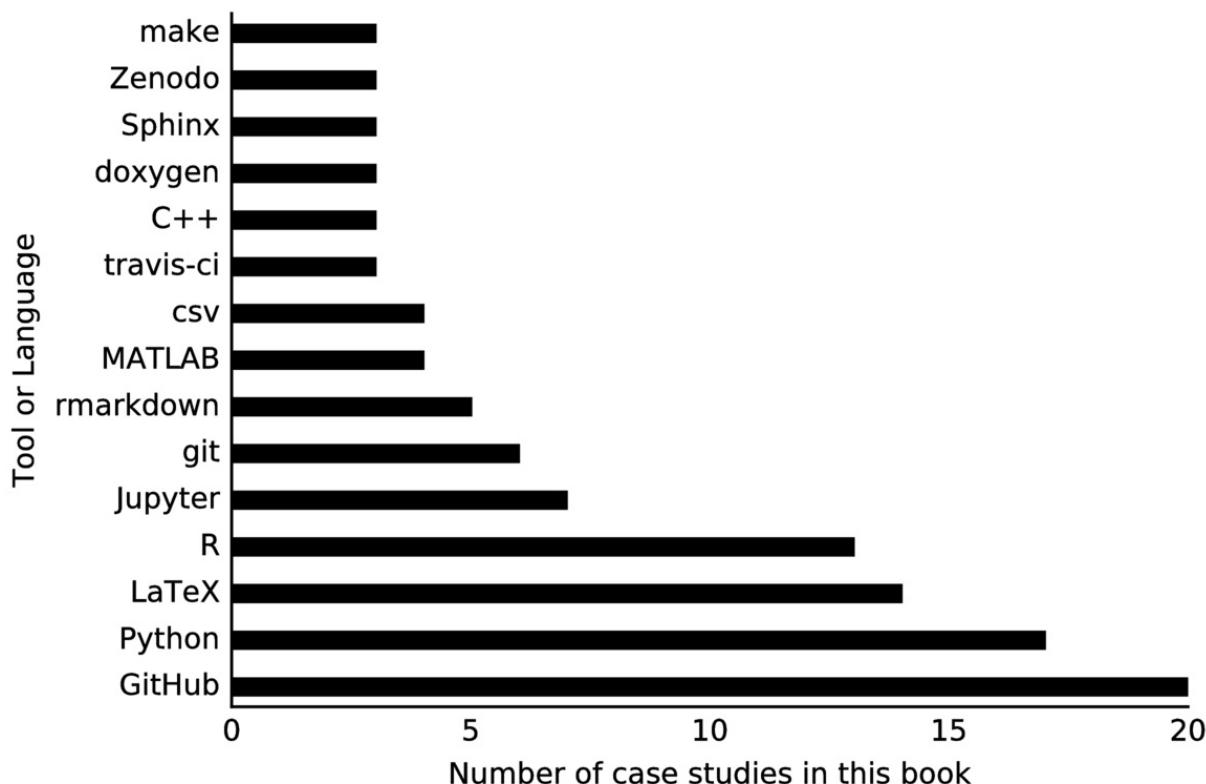
- host code on a collaborative platform (e.g. GitHub)
- get a Digital Object Identifier for your data and code
- avoid spreadsheets, plain text data is preferred ("timeless," even)
- explicitly set pseudorandom number generator seeds
- workflow and provenance frameworks may be too clunky for most scientists

At the core of many of these issues are human concerns around incentives and delayed return on investment. Education and community development will be needed to solve those issues where tool improvements fail to help.

Tools Used

The vast majority of scientists used a programming language such as R or Python to churn through analysis and automate processing.

Nearly all reported use of GitHub-based version controlled repositories at the core of their research work. Additionally, scientists often cited use of an ecosystem of tools appropriate for their work. The following sections categorize tools by their purpose, rather than by the ecosystem (e.g. R vs. Python) that they are found in most commonly.



Publishing

For publishing, the case study authors improved their reproducibility with What You See Is What You Mean (WYSIWYM) mark-up languages. Some preferred LaTeX/Overleaf, while others preferred the R/Knitr/RMarkdown/Sweave ecosystem. In combination with a text editor and distributed version control, both reproducibility and simultaneous collaboration are improved by these plain text mark-up languages.

Data Handling

The scientists used many different formats and systems for storing and cleaning their data. Some storage systems include both hierarchical (e.g. HDF5) and relational (e.g. SQL) database systems. Of particular note, the R community representation among these case studies boasted use of the RStudio IDE as part of the way they streamline access to the collection of R tools which enable some of these data repository solutions as well as data cleaning and exploration tasks.

Additionally, emerging ``data lakes'' for archived storage and retrieval such as Dataverse (King, 2007) were also mentioned. Case study authors even noted that specialized data lakes exist for certain scientific domains. Neurovault (Gorgolewski et al., 2016) was mentioned for neuroscience, but similar solutions exist other fields as well (e.g. Dryad (H. C. White, Carrier, Thompson, Greenberg, & Scherle, 2008) for ecology).

Testing Frameworks

In the Python ecosystem, tests can be run with frameworks such as [nose](#) or [unittest](#). In C++, one can use [GoogleTest](#). In both Python and C++ projects, testing frameworks were mentioned by multiple case study authors.

Although very few R users mentioned unit testing their code, the language does have options for unit test frameworks, with the `testthat` package (Wickham, 2011) being the most widely-used unit testing framework. Other available packages exist as well, such as the `RUnit` package.

Continuous Integration

Although few of the case study authors mentioned continuous integration, its use was lauded as essential. Reproducible practices are easiest to adopt when they require no time from the scientist. Even better, practices that save time are even easier to adopt. Continuous integration is just such a practice.

To get scientists to regularly run the tests for their software on a variety of platforms, don't require any effort of the scientist. That is, outsource the task of building and running the tests to the computer with a continuous integration system. Essential to production software,

continuous integration servers like [CTest](#), [Travis-CI](#), [Jenkins](#), [Bathlab](#), and many others enable scientists to focus on implementation without worrying about checking constantly whether they're introducing bugs. If they introduce a bug, the continuous integration server will notice and send out an email or publish a status report.

DOI Management

A primary incentive for scientists is citation. Accordingly, the efforts they put toward reusable workflows will only seem worthwhile if those data, scripts, libraries, and analyses can be cited. Thus arises the DOI. Many methods exist for putting a citeable, persistent, digital stamp on one's code and data.¹ Common services for archiving digital objects and providing DOIs were mentioned in the case studies including Zenodo, Figshare, and the Open Science Framework.

Other Recommendations

Perhaps most interesting among all of the recommendations are some insights that were only noted by one or two of the scientists. Some of these recommendations may be impactful if they see broader adoption.

Post Flurry Refactoring

Chapter authors Randall LeVeque and Rachel Slaybaugh each expressed their own version of the following idea:

Make a habit of cleaning up code used to produce final results so that it's well documented and all the necessary steps are clearly laid out. Then run through them from scratch if possible to insure that it works. Even if you don't plan to share it with others, your future self will thank you.

This kind of workflow doubles down on the importance of documentation and clarity for users (including your future self). By emphasizing good practices during the day-to-day, this kind of workflow ensures that code is consistently useable by its author during development. By similarly employing best practices during the release or publication stage, this type of workflow effectively double-checks the reproducible nature of the work.

This workflow concept is especially enamoring because it recognizes the humanity of the scientist, who in a day-to-day work environment may let a few tasks necessary for full reproducibility to slip through the cracks. It then corrects for that element of human frailty by budgeting time at the wrapping-up stages to correct for that human error and solidify the process, like a time capsule for the future.

Standardized Data Formats

In the case studies, it was mentioned that a GIS standard is needed to help unify work with maps and geospatial data. This unification of work through collectively adopted standards applies to other fields as well. When a proliferation of standards complicates collaboration within a scientific domain, community agreement on a single standard can allow reproducibility across research groups.

A number of fields have long since successfully adopted data format standards, such as the Flexible Image Transfer System (FITS) files used in astronomy or the evaluated nuclear data files (ENDF) used in nuclear physics. Efforts are ongoing to standardize formats in other fields, where formats may be absent, insufficient, or (possibly worse) proliferant. For example, there is work in the emergent, data-driven field of neuroimaging to establish a standard for neurological images and see that standard adopted across the field (the Brain Imaging Data Structure).

It's worth noting, though, that even when a scientific domain has a community standard, it may not translate well in interdisciplinary work, when internal domain norms around data formats may hamper efforts to communicate. This can be ameliorated if domain standards are more universal standards, based perhaps on common data formats (e.g. SQL databases, HDF5, plain text).

Need: Community adoption for file format standards within some domains.

Need: Domain standards which translate well outside of their own scientific communities.

Conclusion

Many positive lessons came from this set of case studies. One was the reassurance that git and GitHub as well as automation in general are now ubiquitous among the case study authors - scientists seeking reproducibility. Accordingly, 'timeless' plain text file formats are preferred for code and small scale data. A return to transparent open formats based in plain text bodes well not just for reproducibility, but also for its siblings openness and transparency.

Some core needs that were identified include:

- Better education of scientists in more reproducibility-robust tools.
- Widely used tools should be more reproducible so that the common denominator tool does not undermine reproducibility.
- Improved configuration and build systems for portably packaging software, data, and analysis workflows.

- Reproducibility at scale for high performance computing.
- Standardized hardware configurations and experimental procedures for limited-availability experimental apparatuses.
- Better understanding of why researchers don't respond to the delayed incentives of unit testing as a practice.
- Greater adoption of unit testing irrespective of programming language.
- Broader community adoption around publication formats that allow parallel editing (i.e. any plain text markup language that can be version-controlled).
- Greater scientific adoption of new industry-led tools and platforms for data storage, versioning, and management.
- Increased community recognition of the benefits of reproducibility.
- Incentive systems where reproducibility need not be self-incentivizing.
- Standards around scrubbed and representational data so that analysis can be investigated separate from restricted data sets.
- Community adoption for file format standards within some domains.
- Domain standards that translate well outside of their own scientific communities.

While building and installation of dependencies remains a critical stumbling block, the most universal problems are human in nature, ranging from establishing basic toolkit familiarity within a team to motivating reproducible workflows according to their long term benefits. Similarly, though community norms both in and across domains are still in need of unification, many scientists interested in reproducibility are converging on a set of best practices for reproducible, open, robust science. Finally, some solutions - especially those related to human incentives - await institutional changes.

References

- Barba, L. A. (2012). Reproducibility PI Manifesto.
<http://doi.org/https://dx.doi.org/10.6084/m9.figshare.104539.v1>
- Bernstein, D. (2014). Containers and Cloud: From LXC to Docker to Kubernetes. *IEEE Cloud Computing*, 1(3), 81–84. <http://doi.org/10.1109/MCC.2014.51>
- Chirigati, F., Shasha, D., & Freire, J. (2013). Reprozip: Using provenance to support computational reproducibility. In *Presented as part of the 5th USENIX Workshop on the Theory and Practice of Provenance*. Retrieved from
<https://www.usenix.org/conference/tapp13/technical-sessions/presentation/chirigati>

Donoho, D. L., Maleki, A., Rahman, I. U., Shahram, M., & Stodden, V. (2009). Reproducible Research in Computational Harmonic Analysis. *Computing in Science & Engineering*, 11(1), 8–18. <http://doi.org/10.1109/MCSE.2009.15>

Gorgolewski, K. J., Varoquaux, G., Rivera, G., Schwartz, Y., Sochat, V. V., Ghosh, S. S., ... others. (2016). NeuroVault. org: A repository for sharing unthresholded statistical maps, parcellations, and atlases of the human brain. *NeuroImage*, 124, 1242–1244. Retrieved from <http://www.sciencedirect.com/science/article/pii/S1053811915003067>

Guo, P. (2012). CDE: A tool for creating portable experimental software packages. *Computing in Science & Engineering*, 14(4), 32–35. Retrieved from <http://scitation.aip.org/content/aip/journal/cise/14/4/10.1109/MCSE.2012.36>

Hunold, S., & Träff, J. L. (2013). On the State and Importance of Reproducible Experimental Research in Parallel Computing. *arXiv:1308.3648 [Cs]*. Retrieved from <http://arxiv.org/abs/1308.3648>

King, G. (2007). An introduction to the Dataverse Network as an infrastructure for data sharing. *Sociological Methods & Research*, 36(2), 173–199. Retrieved from <http://smr.sagepub.com/content/36/2/173.short>

Stodden, V. (2010). The Scientific Method in Practice: Reproducibility in the Computational Sciences. *SSRN Electronic Journal*. <http://doi.org/10.2139/ssrn.1550193>

White, H. C., Carrier, S., Thompson, A., Greenberg, J., & Scherle, R. (2008). The Dryad data repository: A Singapore framework metadata architecture in a DSpace environment. *Universitätsverlag Göttingen*, 157. Retrieved from <http://www.oapen.org/download?type=document&docid=353956#page=173>

Wickham, H. (2011). Testthat: Get started with testing. *The R Journal*, 3(1), 5–10. Retrieved from <http://imagic.com/eLibrary/ARCHIVES/GENERAL/JOURNALS/R110623I.pdf#page=5>

Wilson, G., Aruliah, D. A., Brown, C. T., Chue Hong, N. P., Davis, M., Guy, R. T., ... Wilson, P. (2014). Best Practices for Scientific Computing. *PLoS Biol*, 12(1), e1001745. <http://doi.org/10.1371/journal.pbio.1001745>

Building Towards a Future Where Reproducible, Open Science is the Norm

Karthik Ram and Ben Marwick

The traditional boundaries between domain researcher and scientific programmer have been blurring rapidly over the past decade. Pressing societal issues such as global climate change, disease outbreaks, endangered species conservation, and drug discovery cut across traditional scientific silos. Successfully answering such interdisciplinary problems will require researchers to not only access and process ever-increasing quantities of data but also leveraging them in the context of their domain expertise. The cost of collecting this data is also dropping, and new technologies in every aspect of our lives now enable cheap and easy collection of high volumes of highly diverse data. As a result, scientific endeavors have come to rely on massive amounts of data being analyzed with a disparate set of tools and technologies.

Another consequence of the high volumes of data and increasing diversity of software tools is that scientists now produce a vast array of research products such as data, code, algorithms, in addition to traditional publications (Heather A Piwowar & Vision, 2013). Yet, until recently, funding agencies such as the US National Science Foundation did not consider any outputs beyond traditional peer-reviewed publications, as credit-worthy outcomes. While some fields, such as astronomy and high energy physics, have long recognized the importance of making the entire research pipeline publicly available, this is far from normal in most areas of science. In the last decade, many areas of science have had high-profile cases of non-reproducible research. Well-publicised retractions include Diederik Stapel in social psychology, Anil Potti in cancer research, Carmen Reinhart and Kenneth Rogoff in economics, and Marc Hauser in evolutionary biology. In addition, large-scale efforts to reproduce biomedical (Begley & Ellis, 2012) and psychological experiments (Open Science Collaboration, 2015) suggest that the prevalence of non-reproducible research has been underestimated, resulting in news headlines declaring a 'reproducibility crisis' in science. The issue of reproducibility is particularly timely given the recent rise in retractions from high profile journals (Van Noorden, 2011). While some aspects of this crisis are due to bad agents, there are also broader systemic problems that result in the production of non-reproducible research. In this chapter we briefly survey some of the gaps, challenges, and opportunities for improving the reproducibility of research.

Gaps: Reproducibility is hard

For many scientists, generating reproducible research is difficult because of the diversity of hardware and software in their workflow. For example, consider an analytical instrument that outputs data in a particular format, which then needs to be transformed and rearranged in several ways before being input into a sequence of several different specialized computer programs for analysis. As the data is moved between each program - we can call this between space a 'gap' - additional manual inspection, readjustment and perhaps combination with other data is required. Gaps result from disconnected tools that have been combined to suit a specific research problem. The problems of handling the data in the gaps are typically solved by bespoke methods that are unique to each group or individual, using tools that were never intended for scientific research (e.g. `Make`), and are rarely produced with the intention of making them public. The custom and expedient nature of these gap-filling methods make it difficult to capture the entire workflow to enable other researchers to reproduce the result. Because of the high diversity of research problems and tools across different areas of science, attempts to integrate these into a single platform have had limited uptake outside of bioinformatics, where many of these pipeline frameworks were first developed (Leipzig, 2016).

Outside of bioinformatics, some researchers are filling these gaps by using literate programming style that allows programming code and narrative text to be interwoven within a single document. One example of this is the work of FitzJohn et al. (2014), who combined the R package knitr with `Make`, among other tools, to create a self-contained and self-documenting workflow for their ecological study. A similar example is the archaeological study by Clarkson et al. (2015), who also used knitr to combine narrative text and programming code to process data from diverse sources. Clarkson et al. also used Docker to provide a self-contained computational environment for their workflow, so that their key software dependencies could be bundled into their research repository with the data. This example is described in more detail in Marwick (2016).

We believe that the use of knitr in these two examples is part of a broader trend in the adoption of executable notebooks in science broadly. An executable notebook is a framework that allows narrative text (and its accessories, such as citations, figures, tables, etc.) and programming code that generates the figures and tables to be interwoven in a single source document. Among R users knitr (a descendant of Sweave) is currently the dominant tool for producing executable notebooks. For Python users there is Jupyter, which can also be used with other programming languages. Our hope is that executable notebooks will be the solution to the problem of gaps in research workflows.

Two other key elements of filling the gaps in the scientific workflow are training for scientists in efficient computer programming, and infrastructure for sharing and collaborating with code. Great progress has been made in these areas, with organizations such as Software Carpentry and Data Carpentry developing and delivering volunteer-led training workshops to researchers across the sciences. Their lesson materials are open source and online to

enable self-study for researchers unable to attend workshops in person. The infrastructure for sharing and collaborating has been made available by services such as GitHub and BitBucket. These services, based on the Git version control system, allow researchers to share their code, organize contributions to scientific software projects, and discover code produced by other researchers (Ram, 2013). In our view, the increase in demand by researchers for training in programming, and the rising popularity of GitHub as public repository for scientific code, reflect a trend toward increasing openness in the scientific process, and in the reproducibility of research.

Challenges: Changing the incentives

Traditional incentives in science prioritize highly cited publications of positive, novel, tidy results. The practice of enabling the reproducibility of those results to be assessed by making the data and code publicly available is not part of the traditional incentives of science. However, individual researchers can gain significant personal benefits for their open science efforts. While preparing and depositing data into an easily discoverable repository requires an upfront time investment, there are numerous benefits to doing so. The National Science Foundation (NSF), for example, requires a data management plan as part of the proposal (Donnelly & Jones, 2010) and also count these endeavors under their merit guidelines (NSF, 2012). Further, authors who share data alongside publications are also likely to be cited more (Heather A. Piwowar, Day, & Fridsma, 2007) and benefit from alternate metrics which are strongly correlated with citations (Heather A Piwowar & Vision, 2013). Citation benefits have been demonstrated for code sharing in research publications (Vandewalle, 2012).

The citation advantage from sharing research data has been demonstrated in numerous disciplines. Henneken and Accomazzi (2011) analysed 3814 articles in four astronomy journals and found that articles with links to open datasets on average acquired 20% more citations than articles without links to data. Restricting the sample to papers published since 2009 in The Astrophysical Journal, Dorch (2012) found that papers with links to data receiving 50% more citations per paper per year, than papers without links to data. In 1,331 articles published in Paleoceanography between 1993 and 2010, Sears (2011) found that publicly available data in articles was associated with a 35% increase in citations. Similar positive effects of data sharing have been described in the social sciences. In 430 articles in the Journal of Peace Research, articles that offered data in any form, either through appendices, URLs, or contact addresses were on average cited twice as frequently as an article with no data but otherwise equivalent author credentials and article variables (Gleditsch & Strand, 2003).

It is clear that researchers in a number of different fields benefit from a citation advantage for their articles that include publicly available datasets. In addition to increased citations for data sharing, Pienta et al. (2010) found that data sharing is associated with higher publication productivity. They examined 7,040 NSF and NIH awards and concluded that a research grant award produces a median of five publications, but when data are archived a research grant award leads to a median of ten publications. These studies suggest the investment of effort in improving reproducibility by sharing data can have payoffs in the traditional incentive system. These efforts are also advantageous in the broader, but very slow, shift in incentives that favor reproducibility over novelty that we sense is occurring in some fields.

The incentivisation of novelty has led to widespread anxiety that sharing of data will result in getting one's own research scooped, and a lack of appropriate rewards for time spent documenting and sharing methods (Heather A. Piwowar et al., 2007). Even when there is an appreciation for open science, the technical challenges such as lack of appropriate skills and knowledge of best practices can hinder this process. By addressing both the cultural and technical challenges we can create a community of practice that would ensure that data sharing is the norm rather than the exception (Birnholtz & Bietz, 2003).

An important step forward in establishing norms for sharing data and using shared data is Daniel Kahneman's (2014) 'reproducibility etiquette'. He proposes that researchers intending to use an open dataset or code repository contact the original authors. When working with code written by others, he especially recommends having a discussion with the authors of the code. The purpose of this to give them a chance to fix bugs or respond to issues you have identified before you make any public statements (Eglen et al., 2016). He also recommends citing code and data in an appropriate fashion. In addition, researchers should also pay close attention to the license agreements attached to specific pieces of code, software, and data products as they unambiguously state the conditions under which such work can be used, adapted, and redistributed (Morin, 2012). Although this is a simple and non-technical detail, we expect that if these values become normalized than the common anxiety of sharing code and data will diminish, and more researcher will feel comfortable to make their work more reproducible.

Making one's research meaningfully reproducible is a significantly more involved effort than merely sharing a handful of scripts and datasets via open repositories (FitzJohn et al., 2014; Mesnard & Barba, 2016). Such activities represent the first of a series of rigorous steps necessary to make a research product truly reproducible. Many of the challenges lie in the analysis phase where the provenance of all inputs and dependencies need to be carefully tracked using automated workflows. It would be naive to suggest that researchers can make their work fully reproducible by following a few simple steps. Even when experienced

computational researchers such as FitzJohn et al and Mesnard et al began their study with full reproducibility in mind, challenges around inadequate tooling and workflow complexity made the task quite hard.

Despite such roadblocks, rapid improvements in tools and workflow technology will continue to lower barriers to reproducibility across various disciplines. In the meantime, any level of reproducibility brings us closer to overcoming the challenges.

Opportunities: The promise of open science

Science is in the midst of a dramatic transformation that is being driven by increasing access to large amounts of heterogeneous data. The long-established model where sole researchers collect and analyze their own data will no longer be the dominant approach and instead be replaced by one where disparate datasets from multiple sources are used. It is now widely accepted in many scientific disciplines that existing datasets can be used to solve novel problems not anticipated by the original investigator (Faniel & Zimmerman, 2011; Nielsen, 2012; Whitlock, McPeek, Rausher, Rieseberg, & Moore, 2010). Such open data can serve as a research accelerator, enabling scientists to rapidly collaborate on knowledge creation and synthesis efforts (Neylon, 2012). A similar pattern of collaboration and reuse is also emerging across the scientific software stack as is evident in the case studies described in this book. A rich suite of open source tools are rapidly lowering barriers to collaborations across disparate domains and institutions and helping accelerate the rate of scientific discovery in ways previously unimagined.

This new era of open science is enabling a community of practice that allows collaborations to scale more easily while various links in the chain of scientific reasoning to be used in different contexts. Part of the reason why scientific workflows are not properly curated or shared are an artifact of the way the credit system currently works in science. Due to insufficient incentives to share, original investigators spend very little time on activities other than publishing. As a result, valuable data, code and critical details on implementation are prone to disappearing or becoming less useful over time (Michener, Brunt, Helly, Kirchner, & Stafford, 1997). However the scholarly landscape is changing to provide both the incentives and means for increased data sharing.

Until recently, researchers who put time and effort into documenting and sharing data and details of their analysis were considered outliers. Now the scholarly landscape is in the midst of a revolution, and among the emerging changes are new incentive mechanisms for reporting research impact. For example, altmetrics (H. Piwowar, 2013) track influence of research outputs and data products outside of the traditional citation framework, providing more ways to measure success. Organizations and repositories including DataCite, figshare, Zenodo, Dryad, DataONE, and others provide the means for data to be cited independent of publications. Papers that share data are more likely to receive citations (Heather A. Piwowar

et al., 2007), and people who collect and deposit well-curated data can receive measurable recognition for their efforts. This is especially important as the scientific community is calling for data citation to be part of the tenure and promotion practice (Parsons, Duerr, & Minster, 2010).

Once a critical mass of scientists share their data and code, it would serve as a multiplier effect and allow disparate groups of researchers to rapidly solve problems such as climate change, (need a few other applications from other domains) (Peterson et al., 2002). We see these collaborations resulting from sharing data and code as one of the great opportunities to come from reproducible research.

Discussion

Our discussion so far has focused on the role of the researcher, and the gaps, challenges and opportunities they face. However, there are a few other key groups that are relevant to changing the norms to enhance the reproducibility of research.

Many funders such as the National Science Foundation (NSF) and National Institutes of Health (NIH) have long maintained data sharing requirements although they have been rarely enforced (Borgman, 2012). However, recent changes to funding policies have made these requirements more stringent and explicit. As of 2011, new NSF proposals require a data management plan (Donnelly & Jones, 2010). This plan requires details on how the data will be documented and where it would be deposited upon completion of the effort.

Many fields in science are in the midst of a data revolution and have adapted to the emerging challenges to varying degrees. At one extreme, disciplines such as astrophysics have fully embraced data driven science by developing and supporting the infrastructure, computational methods, and the culture to derive the most value from the data they generate (Venugopal, Buyya, & Ramamohanarao, 2006). At the other, many data-rich disciplines still lack the culture or the practice to leverage or benefit from past endeavors. Funding agencies can serve as sources of change for these disciplines where cultural change is slow.

A second group for whom reproducible research provides new opportunities are research libraries. Concerns about reproducibility now transcend individual disciplines, and there is a need for research institutes and university campuses to provide resources to support reproducible research. Researchers need information on what tools and services are available for reproducible research, and how they can get training for these. Libraries are becoming sensitive to this need, and some have started providing guides to data management planning, software tools for reproducible research, and training sessions. Two particularly good examples that we are aware of are the University of Utah Library [Reproducibility of Research](#) resource and the NYU Libraries' [Guide to reproducibility](#).

Journal editors are a third group in the research community that have important opportunities to enact change in support of reproducibility. For example, journal editors could increase the importance of reproducibility by requiring (and enforcing) mandatory full data and code deposition, encouraging and even soliciting replication studies, and supporting reviewers who attempt to reproduce studies while reviewing the paper. Several journals have introduced new guidelines for authors and made specific proposals that attempt to address the problems of non-reproducible research (Begley & Ioannidis, 2015). We see this opportunity for editors to support reproducibility as part of a broader cultural change, one occurring at a generational scale, but that will substantially change the way we share our research outputs.

Conclusion

In this chapter we've surveyed some of the gaps, challenges and opportunities relating to reproducible research. We believe that for the majority of researchers there are now mature software solutions to the joining the gaps of a complex workflow. We are starting to see convergence in several disciplines on executable notebooks as one type of software for tackling the challenges of reproducible research. Reproducible research can provide benefits in the traditional incentive system, but our view is that some of the most compelling opportunities are in how incentives - and the practice of science more generally - can be changed by groups such as funding agencies, journal editors and libraries. Finally, we see opportunities for researchers in the form of new and more diverse research collaborations, equipped with uniquely large datasets to take problems of general interest and wide benefit to humanity. Our observations are that the pace of changes toward more reproducible research is accelerating, but that these are changes of a generational scale and so training, persistence, and optimism are vital to support the technical and policy efforts.

References

Begley, C. G., & Ellis, L. M. (2012). Drug development: Raise standards for preclinical cancer research. *Nature*, 483(7391), 531–533. Journal Article. Retrieved from <http://dx.doi.org/10.1038/483531a>

Begley, C. G., & Ioannidis, J. P. (2015). Reproducibility in science: Improving the standard for basic and preclinical research. *Circulation Research*, 116(1), 116–126. <http://doi.org/10.1161/CIRCRESAHA.114.303819>

Birnholtz, J. P., & Bietz, M. J. (2003). Data at work: Supporting sharing in science and engineering. In *Proceedings of the 2003 international acm siggroup conference on supporting group work* (pp. 339–348). ACM.

- Borgman, C. L. (2012). The conundrum of sharing research data. *Journal of the American Society for Information Science and Technology*, 63(6), 1059–1078.
- Clarkson, C., Smith, M., Marwick, B., Fullagar, R., Wallis, L. A., Faulkner, P., ... others. (2015). The archaeology, chronology and stratigraphy of madjedbebe (malakunanza ii): A site in northern australia with early occupation. *Journal of Human Evolution*, 83, 46–64.
- Donnelly, M., & Jones, S. (2010). Template for a data management plan. *Digital Curation Centre*. Retrieved July, 12, 2010.
- Dorch, S. (2012). On the citation advantage of linking to data: Astrophysics. Retrieved from <https://halshs.archives-ouvertes.fr/hprints-00714715/>
- Eglen, S., Marwick, B., Halchenko, Y., Hanke, M., Sufi, S., Gleeson, P., ... Poline, J.-B. (2016). Towards standard practices for sharing computer code and programs in neuroscience. *bioRxiv*. <http://doi.org/10.1101/045104>
- Faniel, I. M., & Zimmerman, A. (2011). Beyond the data deluge: A research agenda for large-scale data sharing and reuse. *International Journal of Digital Curation*, 6(1), 58–69.
- FitzJohn, R. G., Pennell, M. W., Zanne, A. E., Stevens, P. F., Tank, D. C., & Cornwell, W. K. (2014). How much of the world is woody? *Journal of Ecology*, 102(5), 1266–1272. <http://doi.org/10.1111/1365-2745.12260>
- Gleditsch, N. P., & Strand, H. (2003). Posting your data: Will you be scooped or will you be famous? *International Studies Perspectives*, 4(1), 72–107. <http://doi.org/10.1111/1528-3577.04105>
- Henneken, E. A., & Accomazzi, A. (2011). Linking to data - effect on citation rates in astronomy. *CoRR*, *abs/1111.3618*. Retrieved from <http://arxiv.org/abs/1111.3618>
- Kahneman, D. (2014). A new etiquette for replication. *Social Psychology*, 45(4), 310.
- Leipzig, J. (2016). A review of bioinformatic pipeline frameworks. *Briefings in Bioinformatics*. <http://doi.org/10.1093/bib/bbw020>
- Marwick, B. (2016). Computational reproducibility in archaeological research: Basic principles and a case study of their implementation. *Journal of Archaeological Method and Theory*, 1–27.
- Mesnard, O., & Barba, L. A. (2016). Reproducible and replicable cfd: It's harder than you think. *arXiv*, 1605.04339.
- Michener, W. K., Brunt, J. W., Helly, J. J., Kirchner, T. B., & Stafford, S. G. (1997). Nongeospatial metadata for the ecological sciences. *Ecological Applications*, 7(1), 330–342.

Morin, J. A. S., Andrew AND Urban. (2012). A quick guide to software licensing for the scientist-programmer. *PLoS Comput Biol*, 8(7), 1–7.

<http://doi.org/10.1371/journal.pcbi.1002598>

Neylon, C. (2012). Science publishing: Open access must enable open use. *Nature*, 492(7429), 348–349. Retrieved from <http://dx.doi.org/10.1038/492348a>

Nielsen, M. (2012). *Reinventing discovery: The new era of networked science*. Princeton University Press.

NSF. (2012). US NSF - Dear Colleague Letter - Issuance of a new NSF Proposal & Award Policies and Procedures Guide (NSF13004). Retrieved from

http://www.nsf.gov/pubs/2013/nsf13004/nsf13004.jsp?WT.mc_id=USNSF_109

Open Science Collaboration. (2015). Estimating the reproducibility of psychological science. *Science*, 349(6251). <http://doi.org/10.1126/science.aac4716>

Parsons, M. A., Duerr, R., & Minster, J.-B. (2010). Data citation and peer review. *Eos, Transactions American Geophysical Union*, 91(34), 297–298.

Peterson, A. T., Ortega-Huerta, M. A., Bartley, J., Sánchez-Cordero, V., Soberón, J., Buddemeier, R. H., & Stockwell, D. R. (2002). Future projections for mexican faunas under global climate change scenarios. *Nature*, 416(6881), 626–629.

Pienta, A. M., Alter, G. C., & Lyle, J. A. (2010). The enduring value of social science research: The use and reuse of primary research data. Retrieved from http://deepblue.lib.umich.edu/bitstream/handle/2027.42/78307/pienta.Alter_lyle_100331.pdf

Piwowar, H. (2013). Altmetrics: Value all research products. *Nature*, 493(7431), 159–159. <http://doi.org/10.1038/493159a>

Piwowar, H. A., & Vision, T. J. (2013). Data reuse and the open data citation advantage. *PeerJ*, 1, e175.

Piwowar, H. A., Day, R. S., & Fridsma, D. B. (2007). Sharing detailed research data is associated with increased citation rate. *PLoS ONE*, 2(3), e308.

<http://doi.org/10.1371/journal.pone.0000308>

Ram, K. (2013). Git can facilitate greater reproducibility and increased transparency in science. *Source Code for Biology and Medicine*, 8(1), 7.

Sears, J. (2011). Data sharing effect on article citation rate in paleoceanography. In *AGU fall meeting abstracts* (Vol. 1, p. 1628).

Van Noorden, R. (2011). The trouble with retractions. *Nature*, 478(7367), 6–8. <http://doi.org/10.1038/478026a>

- Vandewalle, P. (2012). Code sharing is associated with research impact in image processing. *Computing in Science and Engineering*, 14(4), 42–47.
- Venugopal, S., Buyya, R., & Ramamohanarao, K. (2006). A taxonomy of data grids for distributed data sharing, management, and processing. *ACM Computing Surveys (CSUR)*, 38(1), 3.
- Whitlock, M. C., McPeek, M. A., Rausher, M. D., Rieseberg, L., & Moore, A. J. (2010). Data archiving. *The American Naturalist*, 175(2), 145–6. Retrieved from <http://www.jstor.org/stable/10.1086/650340>

Glossary

Ariel Rokem and Fernando Chirigati

Like other technical areas, the topic of computational reproducibility has its own terminology and jargon. The terms range from key concepts of the field, that are important when defining the parameters of reproducible research, to specific techniques and practices that are used in upholding computational reproducibility. Finally, there is a plethora of technical tools and practices that are mentioned throughout this book. In this chapter, we provide some important definitions to help clarify the terms, techniques, and tools that are mentioned throughout the case studies and the other chapters.

Key Concepts

Scientific Experiment

A **scientific experiment**, or simply **experiment**, is a procedure carried out to validate or refute a hypothesis. In our modern times, many stages in a research project are done partially or entirely through the use of computer programs and processes, and that involve digital data that is consumed (**input data**) and produced (**output data**). This may include studies in which the experiments themselves are computational in nature. In this case, the experiment is often modelled as a **pipeline** (or **dataflow**): a sequence of **steps** that are connected by the flow of data, where the output data of a step is used as input data for the following step. A step can be represented by a computer program or a sequence of programs (a sub-pipeline), and it transforms the data it consumes as part of the procedure.

Reproducibility

Reproducibility is a cornerstone of science. Definitions vary greatly across scientific disciplines, but the meaning that we find most prevalent is the 'calculation of quantitative scientific results by independent scientists using the original datasets and methods' (Stodden, Leisch, & Peng, 2014). The goals of reproducibility go beyond duplicating someone else's investigation: it also entails having reproducibility for yourself, defeating self-deception in scientific results (Ioannidis, 2005; Nuzzo, 2015), and extending another researcher's methods to build your own work. Reproducibility is a matter of degree, not of

kind. We say that research is reproducible if reproducibility applies to the results to some extent. That is, some of the corresponding experiments and scientific methods are deemed to be reproducible.

Empirical and Computational Reproducibility

We can define different types of reproducible research as follows, adapted from Stodden (2014): empirical reproducibility, and computational reproducibility.

Empirical reproducibility entails communicating the procedure, protocols, equipment, and observations related to the experiment, but does not require making the computational assets (code and data) used during the research publicly available. This is often a minimal standard in science: published manuscripts contain descriptions and static figures and plots, and scientists need to follow these in hopes of building upon past research.

In **computational reproducibility**, in addition to the published manuscripts, the computational assets used to test all the hypothesis and derive the results are made available, which allows the computational processes to be reproduced verbatim and, in some cases, re-used. These assets may include, but are not limited to: the input data, either in extension (raw data) or in intension (a script that generates the data); the software (in binary or in source code); and the computational environment (computational dependencies and operating system information).

The notions of **verification** and **validation** are also commonly used when referring to reproducibility (Stodden et al., 2013): verification is concerned with the code solving the problem it claims to solve, while validation is concerned with the results being consistent with observations of the phenomenon being studied. In this sense, empirical reproducibility helps in the validation process, whereas computational reproducibility helps in the verification process, since the experiment execution can be investigated in more details.

With respect to the verification process, computational reproducibility helps identify if the code is not broken, and also pinpoint any statistical issues that may invalidate the results. For instance, *p-hacking* is a common bias in science where researchers select data or statistical analyses until non-significant results become significant. By having the data and all the artifacts, including the full chain of research events, one could tweak the different variables and vary the original analysis to detect how robust and significant the claims are.

Reproducibility Modes

Reproducibility can also be defined with respect to how results can be reproduced. Some of the terms often used in this regard are replicability, approximate reproducibility, and modular reproducibility.

Like reproducibility, there are conflicting definitions of **replicability** across different scientific domains. In some areas of science this is a synonym for **exact reproducibility**: the reproduced results are exact the same (meaning the exact same numbers) as the ones presented and discussed in the corresponding published manuscript. The computational assets, such as software, configuration parameters, and hardware, must be ideally the same to ensure replicability. Replicability guarantees reproducibility, but not the converse (Leek & Peng, 2015). In other areas, replicability can refer to a prior study being duplicated using the same procedures but with new data (Stodden et al., 2014).

Approximate reproducibility is related to having results that are similar to (and not the same as) the ones produced in the original experiment run. This often includes varying configuration parameters and input files to better verify how robust the experiment is, and perhaps simulating some steps that are harder to replicate. For example, an experiment that involves parallel and distributed computation may depend on the availability of massive servers for its replication; these steps can then be simulated or conducted at smaller scale to make its reproducibility feasible (Hunold & Träff, 2013). Also, some experiments are intrinsically difficult to replicate, such as the ones that require non-deterministic steps (random number generation) and access to third-party servers (code that is on servers cannot be controlled by researchers). Note that the reproduced results need to be consistent to the original ones to allow others to validate the experiment.

When an experiment supports **modular reproducibility**, its different steps and components can be reproduced individually, i.e., the experiment does not need to be reproduced in its entirety. For instance, if a single binary is shared for the entire experiment, it may be hard to reproduce only some of its steps; however, if the source code is made available, researchers will have more flexibility to use the parts of the experiment they want/need. Modular reproducibility allows **reusability**: the experiment can be more easily re-used for other purposes, thus making it possible for others to modify and build upon the original work.

Reproducibility Coverage

Another important aspect in reproducibility is its **coverage** (Freire, Bonnet, & Shasha, 2012): some experiments may not be reproduced in its entirety, including the ones that rely on data derived by third-party Web services or special hardware, or that require non-deterministic computational processes. But such experiments can, sometimes, be partially reproduced. For instance, if an experiment uses data that is derived by special or proprietary hardware, the data derivation process may not be reproducible, but the downstream analyses that use these data may be reproduced by others if the original data is made available.

Provenance

As the volume of digital data increases and the complexity of computational processes that manipulate these data grows, it is becoming increasingly important to manage their **provenance**. The Oxford English Dictionary defines provenance as *the source or origin of an object; its history and pedigree; a record of the ultimate derivation and passage of an item through its various owners*. Provenance helps determine the value, accuracy, and authorship of an object.

Computational provenance enables data products derived by computational processes to be interpreted and understood (Freire, Koop, Santos, & Silva, 2008). By examining a sequence of steps that led to a result, it is possible to obtain insights into the chain of reasoning used in the production of this result; to verify that the steps were performed according to acceptable procedures; and to identify what the inputs to the experiment were and where they came from.

Provenance is a critical ingredient for reproducibility (Freire & Silva, 2012). Providing detailed information about the provenance of results of an experiment tells about both the data and the sequence of steps that generated the findings. Through this information, it is possible to detect the required components of the experiment, and this facilitates the task of making it reproducible. The availability of such provenance information not only makes it possible to replicate the findings, but it also makes it easier to re-use and extend a result (by changing inputs and modifying the sequence of steps); in other words, detailed provenance allows modular reproducibility.

Provenance can be **describable** or **executable**. Describable provenance entails having a full description of the experiment (textual description, or a graph detailing all the steps) that serves to communicate in detail the computational aspects that one need to know to reproduce each step. Executable provenance, on the other hand, entails having an executable asset that can be directly used to reproduce the experiment (a binary or a scientific workflow (Davidson & Freire, 2008)).

There are different **provenance components** that must be captured to ensure the reproducibility of an experiment (Chirigati, Shasha, & Freire, 2013):

- **Data** entails the original input data used to execute the experiment, and the original output data to compare the raw results. Sharing the intermediate data (data produced by intermediate steps of the pipeline) may also be useful if some steps cannot be reproduced (see *coverage* in *Reproducibility*).
- **Process** entails all the computational programs and scripts used to execute the experiment. As mentioned before, this can be done by sharing either the source code or the binaries, which influences reusability.

- **Environment** entails all the assets belonging to the computational environment where the experiment was originally executed, which includes information about the operating system (e.g.: name and kernel version), hardware architecture (e.g.: 32 or 64 bits, number of computational nodes), and computational dependencies (e.g.: library and software packages on which the experiment depends to run). This component is important to allow an experiment to be portable to other computers, especially if they have different software and hardware systems.

Techniques

Version Control

Version control is a set of practices and tools originally used in software development to track the versions of software. These tools monitor, track and store changes to files within a circumscribed part of the file system, often referred to as a **repository**. The first generation of these systems are referred to as **centralized version control** systems (these include the **Concurrent Versions System (CVS)** and **Subversion (SVN)**). These systems rely on the existence and setup of a centralized server that stores the history of the changes to the code. In contrast, **distributed version control** systems (such as **git**, and **Mercurial**) do not depend on the presence of a centralized server. The history is instead stored together with the files in each user's computer. To facilitate collaboration and coordination of work on different users' computers, centralized servers are nevertheless often used as a common point for 'push' and 'pull' operations that synchronize the history between repositories stored on different computers and to merge work that is concurrently done on different files, or different parts of the same files by different users. Centralized servers can be set up on websites, and such websites offer other features. For instance, they display and allow browsing of the files in a virtual file-system-like website, and they provide web pages that can be used to browse the files in the repository, without downloading them. In addition, these websites provide for collaboration and communication among users (such as **bug-trackers**, pages in which errors in the code, or "bugs", can be reported and addressed). The use of version control tools in science has risen in recent years, with many large collaborative projects and institutions (e.g., CERN, LSST, and NCBI) using the services of websites such as **GitHub** to distribute and collaborate on software.

Literate Programming

Computer programs are read many more times than they are written (Wulf, 1977). Considering this fact, Knuth (Knuth, 1984) proposed that instead of focusing on computer programs as only a set of instructions to the computer, the focus of a computer program should be to explain what is (supposed to be) achieved through these instructions. This shift

in focus implies a more thoughtful approach to descriptive details of the software, such as function and variable names, and a substantial focus on documentation. In a research context, computer programs are embedded within documents, such as scientific papers. This practice is also described as 'literate computing', 'literate statistical programming', 'literate data analysis', and 'literate statistical practice', in recognition of the adoption of literate programming methods from a software development context into a data analysis context. Several systems, such as **knitr** and **Jupyter** allow the writing of documents, including papers, with the code embedded or interleaved with the text.

Data Publication

Full access to the computational assets that led to previously reported results are essential for **computational reproducibility**. **Data publication** (also known as "**data sharing**") refers specifically to public availability of the data that was used (as distinct from the software, for example). If the data is stored digitally, this can be done by sending the data to specific collaborators, by creating digital copies, or making files available over the Internet. It can be done by uploading the data to publicly available websites that either can be accessed unencumbered, or require agreement to certain terms and conditions of use. In some cases, data size also limits the possibility of data publication and it is more practical to send physical copies of the data (for example, the so-called "connectome in a box" (Poldrack & Gorgolewski, 2014), distributed by the NIH-funded Human Connectome Project, which is a hard-drive version of large collections of human MRI data). Other limitations may include restrictions due to participant privacy (the **HIPAA**, or Health Insurance Portability and Accountability Act, enacted into law in the United States in 1996, restricts the information that can be made public about participants in research data; other similar laws apply elsewhere). While data could be considered factual information that cannot be copyrighted, research data often undergoes several steps of transformation before it can be useful: it is collected, aggregated, and manipulated, using significant investment of time or resources. Thus, it could represent an original and creative expression of the source (or "raw") data and may be considered copyrightable intellectual property. For data sharing to achieve its goals of reproducibility, it is therefore important to consider and define appropriate conditions of license to potential users when sharing data.

Munging

Research data is often quite "messy". This means that it is not immediately tractable to the standard statistical analysis without additional steps (Milliken, 2006).

Data munging (also known as **data cleaning**) refers to the application of transformations to the data to bring from a "messy" state to a "tidy" state (Wickham, 2014). This may include filtering operations (exclusion of certain observations that contain missing values),

aggregation, and integration of data from different sources. According to some estimates, data munging is one of the principal activities of individuals conducting data analysis across different sectors, including research in both industry and academia (Dasu & Johnson, 2003; Lohr, 2014).

Figuratively, people speak collectively of these transformations and data "janitorial" work as data "munging". This word stems from either the English word "mung", which refers to a messy mixture of things (originally, a mixture of graines) (Oxford English Dictionary, 2016a), or the word "munge", referring to "eating greedily and noisily" (Oxford English Dictionary, 2016c) (possibly related to the word "munching"). More rarely, it also refers to wiping of a person's nose (Oxford English Dictionary, 2016b), which could be a reference to the act of cleaning itself. Alternatively, this is derived from the acronym MUNG, meaning "mash until no good" (or recursively, "mung until no good"). To maintain reproducibility of these steps, **provenance tracking** must be used to maintain the transformations and intermediate states of the data.

Software Testing

There are several types of testing to be considered:

1. **Unit testing**: This type of testing focuses on the operations of individual parts of the software ("units"). One rule of thumb is that unit testing should not require disk input/output, or access to the network. Unit testing works best when coupled with modular software design. In scientific software, unit testing takes the form of verification of known results from a specific function.
2. **Integration testing**: This type of testing focuses on testing the combination of different parts of a system. For example, verifying that the outputs of one part of the system can be ingested as inputs by other parts of the system to produce reasonable results.
3. **Regression testing**: This type of testing focuses on testing that previous results of a computation are maintained over time. This is useful to assess parts of the software for which it is hard to write unit tests. For example, parts of the software that contain random number generation can be tested to not deviate from a prior stored result by more than a certain factor.
4. **End-to-end testing**: This type of testing verifies if the operations of an entire system, under realistic conditions, produce desired results. For example, an analysis pipeline that starts with raw experimental data (considered representative of the actual data that the system is designed to analyze) transforms and munges this data, and results in some statistical analysis. Testing an entire workflows is considered end-to-end testing (see also **continuous integration**, below).

Continuous Integration

In software development, **integration** refers to the steps taken at different stages of development to harmonize the operations of different parts of systems made up of small parts. The integration of new features into a software system can cause unexpected changes in its behavior. This is addressed by **software testing**: if the existing software has sufficient **test coverage** — that is, the tests exercise all the different parts of the software, and exercise a sufficiently broad range of scenarios: corner cases, handling of extreme and unusual values, etc. — then integration of a new piece of software would be evaluated against the expected behavior of the software. To make the process of integration easier, many advocate doing it *early and often* (Duvall, Matyas, & Glover, 2007). For integration testing to be **continuous**, automated systems can be configured to run the **test suite** of the software system (the full set of tests) each and every time a change to the software is introduced. Such publicly available systems include [Travis](#) and [CircleCI](#). These services integrate well with websites that provide version control repositories, such as GitHub or Bitbucket, where new contributions to the software from collaborators can be set to trigger a run of the test suite on a publicly accessible server. Continuous integration on a remote server also help make sure that the dependencies of the software are well-defined, and protects against problems that arise from changing these software dependencies by triggering a test-failure whenever these dependencies change.

Workflow Management

Many scientific projects rely on the execution of several steps of data processing, including data munging and different steps of data analysis. Workflow management systems help distribute and orchestrate the work that needs to be done on the computational resources that are available, but also helps in *tracking provenance* of the results, by storing details of the data, the process, and the executions that take place during the analysis (Davidson & Freire, 2008).

File Format Standards

Scientific data is saved in a myriad of file formats. A typical file format might include a **file header**, describing the layout of the data on disk, **metadata** associated with the data, and the data itself, often stored in binary format. In some cases (e.g., **CSV (or comma-separated value) files**), data will be stored as text. The danger of proliferation of file formats in scientific data lies in the need to build and maintain separate software tools to read, write and process all these data formats. This makes interoperability between different practitioners more difficult, and limits the value of data sharing, because access to the data in the files remains limited.

Licensing

In most countries in the world, creative work is protected by copyright laws. International conventions, and primarily the Berne Convention of 1886, protect the copyright of creators even across international borders for 50 years after the death of the creator. This means that copying and using the creative work is limited by conditions set by the creator, or another copyright holder. For example, in many cases musical recordings may not be copied and further distributed without the permission of the musician, or of the production company that has acquired the copyright from the musician. Facts about the universe that are discovered through research are not subject to copyright, but the collection, aggregation, analysis and interpretation of research data may be considered creative work, and could be protected by copyright laws. Thus, the consumption of research publications is governed by copyright law. Furthermore, even data sharing is often governed by copyright laws, because the compilation of data to be shared often requires a creative effort. Another case of research-relevant copyrighted products is software that is developed in the course of research. In all of these cases, if license terms are not explicitly specified, the work is considered to be protected as "all rights reserved". This means that no one but the creator of the work can use the work unencumbered. For software this means that copying and further distribution of the software is prohibited. Even running the software may be restricted. The exact selection of a license is beyond the scope of this section, but depends on your intentions and goals with regard to the software (Fogel, 2005; Hunter, 2004; Rosen & Einschlag, 2004).

Virtualization and Environment Isolation

Software often requires other software to run properly. The software and hardware elements that are required to properly run a program are known as the **software dependencies**. Because of differences in hardware and operating systems, and because of conflicting dependencies between different programs, the creation and maintenance of software environments that have all the dependencies for a software system is cumbersome, and may require substantial system administration expertise. Pre-configured software environments that include all of the dependencies, software, and sometimes also the data needed for an analysis can be provided through systems that present the user a virtual machine (or VM) that runs in an isolated manner. These systems for virtualization include **VirtualBox** and **Vagrant**.

These systems rely on the ability to store an entire virtual machine as a file that can be copied, and launched within other machine's environment. In addition, some systems provide programmatic virtualization, and dependency management, through the creation of minimal virtual machines referred to as "containers". This includes the **Docker** system, which allows not only storing and publishing light-weight virtual machines, but also provenance tracking and version control of containers. Conflicting software dependencies

can also be managed through systems that isolate a computational environment by setting the parts of the file system that are visible, including the parts of the file system into which versions of dependency libraries are installed. In Python environment, isolation can be achieved through the use of virtual environments such as **virtualenv** and **conda**.

Tools

Programming Language and Related Tools

C/C++

C is one of the most widely used programming languages of all time. Designed to be a compiled language, C was used to re-implement the Unix operating system, and many high-level languages were implemented in C, including Python. C++ is an extension of C that provides support for object-oriented capabilities, and it has become one of the most widely used object-oriented languages, especially for large scale and high performance applications.

Go

[Go](#) is a compiled programming language developed at Google, mostly used in some of the Google's production systems.

IPython

[IPython](#), or Interactive Python (Pérez & Granger, 2007), is a command shell that allows interactive computing for Python, including tab completion, history (provenance capture), parallel computing tools, and support for interactive data visualization.

Java

[Java](#) is a programming language that is compiled into Java bytecode and run on a Java Virtual Machine (JVM), which ensures that all implementations are interoperable in different environments.

JavaScript

[JavaScript](#) is an interpreted programming language extensively used for World Wide Web content production, alongside HTML and CSS.

Jupyter

[Jupyter](#) is a Web application that allows users to create and share *notebooks*, documents that contain live and dynamic code. The Jupyter project evolved from the original IPython, generalizing the interactive environment from being Python-specific to supporting over 40 programming languages.

Python

[Python](#) is a general-purpose interpreted programming language. While Python has a comprehensive standard library, [PyP](#) (the Python Package Index) allows users to search for and download a number of additional Python packages and libraries. Many of these packages are remarkably popular and widely used in different sciences, including:

- [NumPy](#): this library provides support for large, multi-dimensional arrays and matrices, as well as implements a plethora of high-level mathematical functions that operate on these arrays and matrices. NumPy also allows the definition of arbitrary data types, which facilitates the integration with other libraries and tools.
- [SciPy](#): this library builds on top of NumPy to provide many high-level and efficient numerical routines mainly for numerical integration and optimization.
- [matplotlib](#): this library provides 2D plotting procedures for Python.
- [scikit-learn](#): this library provides support for a variety of machine learning algorithms, including classification, regression, clustering, dimensionality reduction, and model selection techniques. scikit-learn is built on top of NumPy, SciPy, and matplotlib.
- [scikit-image](#): this library provides support for a collection of image processing algorithms. Similar to scikit-learn, it is built on top of NumPy, SciPy, and matplotlib.
- [pandas](#): this library brings to Python many data analysis functionalities, including high-level data manipulation tasks (selecting, filtering, slicing, sorting, grouping, plotting, etc.)
- [MNE](#): this library includes a Python package for processing electroencephalography and magnetoencephalography data.
- [Nipype](#): this library provides a uniform interface for creating workflows that integrate a collection of neuroimaging software and applications.

R

[R](#) is a widely used interpreted programming language for statistical computing, data analysis and visualization, with its popularity largely increasing in diverse scientific fields during the past few years (Tippmann, 2014). There is a large and vibrant community of scientists using and developing software in R, with over 8000 packages contributed to the [Comprehensive R](#)

Archive Network. These packages are free to download and extend the functionality of R by adding specialized statistical algorithms, visualization techniques and file handling methods. The following R packages are worth noticing:

- *knitr*: this library provides support for dynamic report generation: R code can be evaluated on the fly to generate documents (PDF, HTML or MS Word files) that automatically include the results of the R analysis.
- *knitcitations*: this library extends knitr by allowing users to add citations to the dynamic reports.
- *dplyr*: this library includes high-level functions for data manipulation tasks that resemble database-like queries (selecting, filtering, and summarizing the data).
- *stringr*: this library provides tools for manipulating text, using regular expressions and character strings.
- *caret*: this library provides an extensive suite of tools for training regression and classification models
- *ggplot2*: this library provides data visualization procedures for R.
- *Rcpp*: this library enables R functions to call C++ code for high performance computing.
- *devtools*: this library includes functions to simplify the development of a new R package.
- *testthat*: this library includes functions to set up unit testing for the code.

RStudio

RStudio is an integrated development environment (IDE) for R that includes both desktop and web server versions. Its code editor provides syntax highlighting, tab-completion, indenting, and definitions. It includes a debugging console, breakpoints, an environment panel, history, tracebacks, and integrated R help and documentation. It supports 2d and 3d visualizations, data display, and data manipulation. Knitr, markdown, and git are deeply integrated into RStudio, enabling version controlled programming via R markdown documents.

Ruby

Ruby is an interpreted programming language commonly used in Web development, and its syntax is broadly similar to that of Python.

Scala

[Scala](#) is a programming language intended to be compiled to Java bytecode and executed on a JVM. Java and Scala are interoperable in the sense that libraries from one language can be used inside the other language.

Documentation Generators

Doxygen

[Doxygen](#) is a tool that automatically generates documentation (in different formats) from annotated source code, supporting a number of different programming languages.

Read the Docs

[Rea](#) is a hosting service for software documentation. The service facilitates the process of generating documentation for the different versions of the code, Read the Docs can be set up to automatically build the documentation whenever a new version of the code is generated.

Roxygen

[Roxygen](#) is a Doxygen-like system for R.

Sphinx

[Sphinx](#) is a tool that can generate documentation in many different file formats.

Pandoc

[Pandoc](#) is a tool that can convert between many different file formats, including LaTeX, HTML, Microsoft Word documents, and Markdown files.

Version Control

Bitbucket

[Bitbucket](#) is a repository hosting service for two distributed version control systems: git and [Mercurial](#). Similar to GitHub, it provides a Web-based interface to facilitate the collaboration in a project.

Git

[Git](#) is a distributed version control system that has become [widely used](#) in the past few years.

GitHub

[GitHub](#) is a git repository hosting service: developers maintain their git repositories on the Web. It provides a Web-based interface, as well as a desktop application, to facilitate the collaboration with other people in the same project. GitHub has numerous features, including, among others, forking, issue tracking, pull requests, and wikis.

SVN

[Subversion](#), or SVN, is a centralized version control system.

Data Munging and Analysis

Apache Hadoop

[Hadoop](#) is a popular framework for distributed processing of large datasets across clusters of computers. Hadoop uses the map-reduce programming model for scaling up to multiple machines. Apache HDFS is the distributed file system used to store input, intermediate, and output data.

Apache Spark

[Spark](#) is a framework for distributed processing that, in contrast to Hadoop, provides in-memory primitives that can achieve better performance for a number of applications.

Connectome Workbench

The [Connectome Workbench](#) is a tool that provides multiple resources for mapping neuroimaging data.

MATLAB

[MATLAB](#) is a numerical computing environment and also a programming language widely popular for data and statistical analysis. It provides many useful features, especially for data management, matrix manipulation, and plotting.

Microsoft Excel

[Excel](#) is a spreadsheet system developed by Microsoft that has many different features, including graphing tools and support for a macro programming language.

MongoDB

[MongoDB](#) is a database system that has been widely used recently, in particular for distributed stores. Instead of storing data in multiple relational structures—such as in traditional relational systems — MongoDB is document-oriented, it stores data in a minimal number of documents.

pandas

[panda](#) is a Python library that has many data analysis functionalities, including high-level data manipulation tasks (selecting, filtering, slicing, sorting, grouping, plotting, etc).

SEPlib

[SEPlib](#) is a distributed software package for seismic data processing, including seismic processing routines, a graphics library, and a IO subroutine library.

Stata

[Stata](#) is a commercial data analysis and statistical analysis software.

Data Visualization

Adobe Photoshop

[Adobe Photoshop](#) is a popular commercial graphics editor, providing a plethora of features to compose and manipulate graphics.

D3

[D3](#) is a JavaScript library used for manipulating data and creating 2D interactive information and data visualizations.

ggplot2

[ggplot2](#) is a data visualization library for R.

matplotlib

[matplotlib](#) is a popular 2D plotting library for Python.

Workflow and Provenance Management

EUPS

[EUPS](#) is a version management tools that tracks the exact project computational dependencies.

Make

[GNU Make](#) and [CMake](#) are tools commonly used to build and derive executable programs from source file. These utilities obtain the dataflow of how to build a program from files called *makefiles*.

VisTrails

[VisTrails](#) is an open-source scientific workflow system that provides support for simulations, data exploration, and visualization, while having many capabilities for provenance capture, management, and analytics.

Software Testing and Continuous Integration

BuildBot

[BuildBot](#) is a Python-based continuous integration tool that automates the process of building and testing software projects.

CircleCI

[CircleCI](#) is a hosted continuous integration service for Web and mobile applications that, similar to Travis CI, can be used to automatically build and test projects hosted at GitHub.

Coveralls

[Coveralls](#) is a tool that automatically identifies the test coverage in a project, showing which parts of the code are not covered by the test suite.

devtools

[devtools](#) is a library that contains a series of functions to facilitate package development for R.

Google Test

[Google Test](#) is a unit testing library for C++ developed and used by Google.

Jenkins

[Jenkins](#) is a Java-based continuous integration tool that automates the process of building and testing software projects.

JIRA

[JIRA](#) is a commercial software for bug tracking, issue tracking, and project management.

Nose

[nos](#) is a Python library that implements functions to assist in writing and running software tests.

nose2

[nose2](#) — a successor to nose — is a unit testing library for Python.

testthat

[testtha](#) is a unit testing library for R.

Travis CI

[Travis CI](#) is a hosted, distributed continuous integration service that can be used to automatically build and test projects hosted at GitHub. If the service is configured, every new commit to the GitHub repository triggers Travis CI, which tries to build the project and run tests. Travis CI is available for a number of different languages.

Virtualization and Environment Isolation

Amazon EC2

[Amazon EC2](#) is a Web service that provides compute infrastructure in the cloud. Virtual environments can be created, launched, and terminated as needed, and users pay by the hour for active servers.

Docker

[Docker](#) is a tool that automates the deployment of applications inside software containers, which are much lighter than virtual machines: containers are isolated but share the operating system, and, when appropriate, binaries and libraries as well. [boot2docker](#) is a Linux distribution made specifically to run Docker containers.

Vagrant

[Vagrant](#) is a tool used to create and configure virtual environments, such as virtual machines and Docker containers.

Virtualenv

[Virtualenv](#) is a tool that creates isolated Python environments. This allows multiple Python projects that have different (and sometimes conflicting) dependencies to coexist in the same computer.

Data Sharing and Repositories

Amazon S3

[Amazon S3](#) is a service for online file storage on the cloud. S3 has been widely used for Web hosting, image hosting, and storage for backup systems.

arXiv

[arXiv](#) is a repository of electronic preprints of scientific publications, and is widely used in the fields of mathematics and physics.

CrossRef

[CrossRef](#) is an official Digital Object Identifier (DOI) Registration Agency. A DOI is often assigned to a publication or research data so that it can be uniquely identified, and therefore, citable. Services like Dataverse and figshare automatically generate DOI's for data that is uploaded to their systems.

Dataverse

[The Dataverse Project](#) is a repository for sharing, citing, and archiving research data. It offers support for backups, recovery, data discovery and cataloging, metadata extraction, and preservation.

Docker Hub

[Docker Hub](#) is a service for building and shipping Docker containers. Docker Hub allows integration with GitHub and BitBucket, as well as collaboration between different users, among other features.

Dropbox

[Dropbox](#) is a service that hosts files on the Web as well as synchronizes files across different platforms. Dropbox also has file versioning features, where users may revisit old versions of their files without losing any work.

figshare

[figshare](#) is a repository for sharing and citing research data (results and manuscripts).

Flickr

[Flickr](#) is a service to host and share images and videos on the Web. It is widely popular among photo researchers and bloggers.

Mendeley and Zotero

[Mendeley](#) and [Zotero](#) are both Web services and desktop applications for managing and sharing research publications.

NeuroVault

[NeuroVault](#) is a repository for sharing statistical maps, parcellations, and atlases of the human brain.

Zenodo

[Zenodo](#) is a repository for sharing and citing research results, including data and publications.

Document Authoring

LaTeX

[LaTeX](#) is a word processor and a document markup language commonly used for writing research publications. In contrast with Microsoft Word, LaTeX is not a WYSIWYG editor: the document needs to be compiled to generate the finished product.

Microsoft Word

[Word](#) is a document and word processing software developed by Microsoft. Microsoft Word is a WYSIWYG editor, while editing, the content onscreen appears in a form that is similar to its appearance as a finished product (WYSIWYG stands for “What You See Is What You Get”).

Overleaf

[Overleaf](#) is an online platform for collaborative writing and publishing using LaTeX, with an integrated real-time preview that closely resembles a WYSIWYG editor.

File Formats

API

An API (or application programming interface) are elements of the design of a software system that allows programmers to use the system to build applications out of it. For example, a software library API will be the design of functions and objects in the library that can be combined together to create new functions and objects.

CSV

The CSV (“Comma Separated Values”) file format stores data in a tabular fashion in plain text. This format is often used to transfer data between applications.

DO

A DO file is a Web-base Java program that is run by a Web server.

Dockerfile

A Dockerfile is a file that has a set of instructions and commands for building a Docker container.

FIF

A FIF (“Fractal Image File”) file stores images in fractals, which can be resized without losing image quality.

HDF5

The HDF5 file format is designed to store and organize large amounts of data. Different data models can be specified for storing data, including multidimensional arrays and tables.

ipynb

An ipynb file represents an IPython notebook document.

JSON

JSON is a data-interchange format that is both human- and machine-readable, storing and transmitting data as attribute-value pairs. It has been widely used recently, largely replacing XML.

Markdown

A Markdown file contains data in a simple markup language that facilitates the conversion from plain text to HTML and other formats. Common extensions for this file include *md* and *Rmd* (the latter represents an R Markdown file where R code is included among the text).

netCDF

The netCDF ("network common data format") file format is machine-independent format commonly used for sharing array-oriented scientific data.

PDF

The PDF (“Portable Document Format”) file format is commonly used to display documents in an interoperable manner.

RAID

RAID (redundant array of independent disks) is a system that confers robustness to data storage through redundancy across sub-partitions. Every bit of data is stored in at least two different partitions, such that if any given partition fails, it can be swapped out without incurring data loss.

SQL

A SQL (“Structured Query Language”) file contains a series of database queries to analyze and manage tables in a database. These queries are represented by statements written in SQL, a programming language designed for managing data in relational databases systems.

SVG

The SVG (“Scalable Vector Graphics”) file represents graphics using an XML-based format that offers support for interactivity and animation.

VT

A VT file stores a VisTrails workflow and its corresponding provenance.

XML

An XML (“Extensible Markup Language”) file stores data in XML, which is a markup language that encodes documents in a format that is both human- and machine-readable. XML is known to provide interoperability among different applications.

References

- Chirigati, F., Shasha, D., & Freire, J. (2013). ReproZip: Using Provenance to Support Computational Reproducibility. In *Proceedings of the 5th usenix workshop on the theory and practice of provenance* (pp. 1:1–1:4).
- Dasu, T., & Johnson, T. (2003). *Exploratory data mining and data cleaning*. John Wiley & Sons.
- Davidson, S. B., & Freire, J. (2008). Provenance and Scientific Workflows: Challenges and Opportunities. In *Proceedings of the 2008 acm sigmod international conference on management of data* (pp. 1345–1350).
- Duvall, P. M., Matyas, S., & Glover, A. (2007). *Continuous integration: Improving software quality and reducing risk*. Pearson Education.
- Fogel, K. (2005). *Producing open source software: How to run a successful free software project*. O'Reilly Media, Inc.

Freire, J., & Silva, C. T. (2012). Making Computations and Publications Reproducible with VisTrails. *Computing in Science Engineering*, 14(4), 18–25.

Freire, J., Bonnet, P., & Shasha, D. (2012). Computational Reproducibility: State-of-the-art, Challenges, and Database Research Opportunities. In *Proceedings of the 2012 acm sigmod international conference on management of data* (pp. 593–596).

Freire, J., Koop, D., Santos, E., & Silva, C. T. (2008). Provenance for Computational Tasks: A Survey. *Computing in Science and Engineering*, 10(3), 11–21.

Hunold, S., & Träff, J. L. (2013). On the State and Importance of Reproducible Experimental Research in Parallel Computing. *CoRR*.

Hunter, J. (2004). Why we should be using BSD. Accessed: 2015-10-25. Retrieved from http://nipy.org/nipy/faq/johns_bsd_pitch.html

Ioannidis, J. P. A. (2005). Why Most Published Research Findings Are False. *PLoS Med*, 2(8), e124. <http://doi.org/10.1371/journal.pmed.0020124>

Knuth, D. E. (1984). Literate programming. *The Computer Journal*, 27(2), 97–111.

Leek, J. T., & Peng, R. D. (2015). Opinion: Reproducible research can still be wrong: Adopting a prevention approach. *Proceedings of the National Academy of Sciences*, 112(6), 1645–1646. <http://doi.org/10.1073/pnas.1421412111>

Lohr, S. (2014, August 17). For big-data scientists, 'janitor work' is key hurdle to insights. *New York Times*. Retrieved from <http://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html>

Milliken, G. (2006). Messy data. In S. Kotz (Ed.), *Encyclopedia of statistical science*. Hoboken, NJ: Wiley.

Nuzzo, R. (2015). How scientists fool themselves – and how they can stop. *Nature*, 526(7572), 182–185.

Oxford English Dictionary. (2016a). Mung, n.1 and adj. Oxford University Press. Retrieved from <http://www.oed.com/view/Entry/113400>

Oxford English Dictionary. (2016b). Munge, v.1. Oxford University Press. Retrieved from <http://www.oed.com/view/Entry/123777?rskey=KZFDs3&result=1>

Oxford English Dictionary. (2016c). Munge, v.2. Oxford University Press. Retrieved from <http://www.oed.com/view/Entry/252110?rskey=KZFDs3&result=2>

Pérez, F., & Granger, B. E. (2007). IPython: A system for interactive scientific computing. *Computing in Science and Engineering*, 9(3), 21–29. <http://doi.org/10.1109/MCSE.2007.53>

Poldrack, R. A., & Gorgolewski, K. J. (2014). Making big data open: Data sharing in neuroimaging. *Nat. Neurosci.*, 17(11), 1510–1517.

Rosen, L., & Einschlag, M. (2004). *Open source licensing*. Prentice Hill.

Stodden, V. (2014). What Scientific Idea is Ready for Retirement? Reproducibility. *Edge.org*. Retrieved from <http://edge.org/response-detail/25340>

Stodden, V., Bailey, D. H., Borwein, J., LeVeque, R. J., Rider, B., & Stein, W. (2013). Setting the Default to Reproducible: Reproducibility in Computational and Experimental Mathematics. *ICERM Workshop Report*. Retrieved from http://stodden.net/icerm_report.pdf

Stodden, V., Leisch, F., & Peng, R. D. (2014). *Implementing reproducible research*. CRC Press.

Tippmann, S. (2014). Programming tools: Adventures with R. *Nature*, 517(7532), 109–110. <http://doi.org/10.1038/517109a>

Wickham, H. (2014). Tidy data. *J. Stat. Softw.*, 59(10).

Wulf, W. A. (1977). Some thoughts on the next generation of programming languages. *Perspectives on Computer Science*, 217–234.

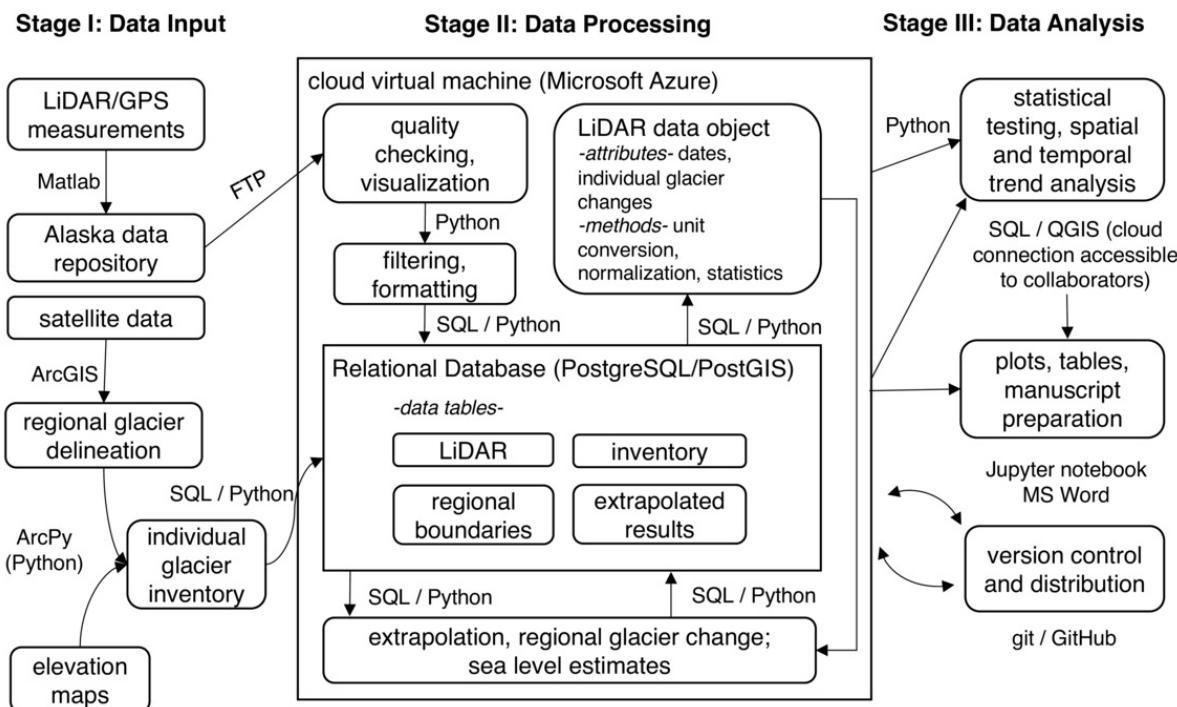
Processing of Airborne Laser Altimetry Data Using Cloud-based Python and Relational Database Tools

**Anthony Arendt, Christian Kienholz,
Christopher Larsen, Justin Rich and Evan
Burgess**

My name is Anthony Arendt and I hold a joint appointment as a Senior Research Scientist at the Applied Physics Laboratory, and a Research Fellow at the eScience Institute, University of Washington. I am part of a research team that studies the impact of glaciers on rising global sea levels, with a focus on the glaciers of Alaska and northwestern Canada. During the past 20 years my colleagues at the University of Alaska Fairbanks have been measuring the elevation changes of Alaska's glaciers using Light Detection and Ranging (LiDAR) data collected from a small aircraft. Our LiDAR system consists of a laser range finder and Global Positioning System (GPS) that measures the precise elevation along the centerline of the glacier surface. By repeating these observations through time, we estimate total changes in mass of each observed glacier, and then extrapolate these data to unmeasured glaciers based on information acquired from satellite imagery. From this we produce detailed maps of the spatial distribution of glacier mass change and the total contribution of these ice masses to global ocean change.

During the 20 year duration of the project the data analysis has evolved from manual manipulation of text files, to a semi-automated workflow that integrates Geographic Information System (GIS), relational database and Python tools within a cloud computing framework. Here we describe the workflow which culminated in a recent publication ([Larsen et al., 2015](#)). Core developers of the software include Evan Burgess, Christian Kienholz, Justin Rich, Anthony Arendt and Christopher Larsen.

Workflow



The workflow begins with annual field data collection that produces both GPS positional and LiDAR point cloud data, both in industry standard binary formats. Commercial proprietary software is used to process the data into four dimensional point observations (x, y, z and time), which are then further processed using Generic Mapping Tools ([GMT](#)) into gridded 10 m digital elevation models. These elevation maps are then subtracted from maps acquired at an earlier time to obtain a change in elevation along the flight line, using Matlab scripts. These results are stored in text files, with the file name describing the glacier name and start and end dates, and are located on a server at the University of Alaska Fairbanks. In a separate step, we assemble satellite imagery and regional digital elevation models for the Alaska region. We use [ArcGIS](#), a commercial GIS software package, to manually digitize the glacier extent. ArcGIS provides a set of vector manipulation tools that enable our technicians to trace glacier outlines from satellite imagery. ArcGIS commands can also be scripted using the ArcPy library. We automate a series of GIS commands using ArcPy to calculate the distribution of glacier surface area with elevation for each of approximately 27,000 glaciers in Alaska.

The majority of our data processing and analysis occurs on a single Microsoft Azure Linux Virtual Machine (VM) that hosts a spatially enabled Relational Database (RDB). We find that hosting an RDB in the cloud is a core element of our reproducible workflow. Our RDB provides rapid query capabilities so that much of our spatial and temporal averaging can be

carried out using efficient database algorithms. Our cloud hosting enables colleagues to make direct connections to the database to access spatial data using their local GIS software. We use the open source [PostgreSQL](#) database with the [PostGIS](#) extension, to which we ingest point, line and polygon geospatial datasets. Relevant tables include:

- *inventory*: polygons of each glacier in Alaska with attributes of surface area, glacier type (whether terminating on land or ocean), name
- *regional boundaries*: polygons of outlines of mountain ranges or climatic zones over which we perform regional extrapolations
- *LiDAR*: measurements of elevation and volume changes on surveyed glaciers
- *extrapolated results*: final estimates of the volume change of every glacier in Alaska

Each time we acquire new altimetry observations we run a Python script to connect via secure FTP to the server in Fairbanks and search for new text files across the directory structure. We use the Python [Pandas](#) library as an interface between our text file and RDB data objects. Specifically, once we ingest the data into a Pandas DataFrame, we can employ a series of methods to generate simple plots and export the data directly to our PostgreSQL database. We use similar Python tools to create and update the *inventory* and *regional boundaries* tables, for example to accommodate changes in surface area as glaciers retreat.

The *LiDAR data object* is the foundation of all subsequent processing and analysis. The data object is created via a function call with parameters describing a single or a regional grouping of glaciers. Each instance of the data object has predefined attributes enabling users to rapidly acquire elements of the raw data in the *LiDAR* table. For example, a user can issue a request to the *LiDAR* table for a specific glacier, returning a data object whose attributes contain that glacier's elevation change, area, and other statistical information. The data object also has several methods that handle the majority of the standard data processing and filtering workflow. These methods include algorithms that carry out unit conversions, normalize the data, calculate statistics and perform mass change calculations for each glacier. To perform these calculations we issue Structured Query Language (SQL) commands to the database from within our Python scripts.

In a final processing step, we utilize the grouping functionality of the LiDAR object methods to generate average elevation change profiles across glaciers grouped by type or by spatial location. For example, we found similar elevation change distributions across glaciers with similar terminus types (i.e. whether terminating in land or in water). Therefore we generated LiDAR objects averaged over *type* groupings, as queried from our *inventory* table, thereby returning a single estimate of elevation change versus elevation. In a final step, we invoked a function that regionally extrapolates these profiles to the unmeasured ice masses stored in the *inventory* table, based on their distribution of area with elevation. This returns a dictionary of ice mass changes by group, as well as an optional new database table

containing mass change estimates for each of the 27,000 ice polygons in the region (table *extrapolated results*). All functions and methods run quickly, with the exception of steps involved in building *extrapolated results*, which takes about 15 minutes to run.

To analyze results and distribute our findings we host a permanent instance of a Jupyter notebook on our Linux VM and provide access to project team members. The notebooks, as well as the core Python scripts used to generate results, are also located in a GitHub repository. The notebook also contains markdown to provide metadata at each step in the analysis. Collaborators with experience writing SQL code can have direct access to the PostgreSQL database to perform their own queries. Other collaborators more familiar with GIS tools can connecting directly to the geospatially encoded tables to generate their own maps.

Pain points

Our team brought together researchers with different backgrounds and approaches to data handling and processing. The processing of raw LiDAR and GPS data is performed by a different group than the one handling the GIS and extrapolation portion of the project, and each uses different software tools. We dealt with this problem by creating standardized files at different stages of the processing chain. For example, the LiDAR/GPS team produced a stack of files processed to the point where they could be used for extrapolation, which were then ingested to the geospatial database. A challenge here is the data are replicated in multiple locations, requiring careful version control.

Another challenge is that some of our collaborators encountered problems when attempting to connect directly to our cloud computing resources. One issue is that Alaska has limited internet bandwidth so that transfer of data between Alaska and commercial cloud providers is slow. Another challenge is that many US government agencies have firewalls that restrict direct traffic with our cloud database services. Therefore our collaborators in government and/or those located in Alaska had to set up duplicate versions of our databases, creating challenges with version control and project management.

Key benefits

Our workflow provides a mechanism to continually update our analysis as new data arrive. Our project is funded for several more years, and we are now in a position to regenerate key figures and update sea level estimates every time we acquire new datasets. This will greatly diminish the time it takes for us to provide stakeholders with updated information on the status of Alaska glaciers and their contribution to sea level. Also, our data are uniquely dynamic, and must accommodate not only new data but changes to the base inventory as

glacier geometries (area and elevation) change over time. By having all our inventory data in relational tables we can update individual polygons and account for the feedback effects of glacier area on mass balance.

Our workflow also provides a stable foundation that can accommodate changes in team composition over time. As students and technicians join and leave the project we can have them use and contribute to a repository of scripts, rather than having to reinvent things from scratch.

We are well positioned to explore our data in ways not previously possible. New collaborators are joining our team and making direct connections to our database, generating complex queries that are exploring what climatic and geometric factors may be driving the glacier mass changes we are observing in the field. Other similar LiDAR observation programs do not provide access to relational databases, limiting researchers' ability to perform spatial and temporal queries.

Key tools

Hosting our resources in a cloud environment played a vital role in making our workflow reproducible. The cloud enabled us to co-locate our scripts with the observations, enabling rapid processing and minimizing the need to transfer files. Additionally, using a relational database to store our geospatial datasets provided efficient methods for us to explore a wide range of spatial relationships in our datasets.

Questions

What does "reproducibility" mean to you?

Reproducibility is a crucial component of our workflow due to the dynamic nature of our monitoring campaign, and the need to constantly update the position and elevation of glaciers as they change in response to climate. We achieve reproducibility through:

- Maintaining consistency in the input datasets
- Utilizing a series of scripts to automate data ingest and filtering
- Storing raw and filtered/processed data in a relational database
- Generating data objects that handle typical data analysis functions
- Scripting all manuscript figures in Jupyter notebooks

Why do you think that reproducibility in your domain is important?

Glaciology has become highly interdisciplinary in the past decade: oceanographers, climatologists, geodesists and glaciologists must integrate knowledge to close the sea level budget. Also, data from remote glacier regions is sparse, so any data we collect needs to be made available. By generating reproducible workflows we have a greater capacity to share information and to better understand exactly how each research team is processing their datasets.

How or where did you learn about reproducibility?

I learned these techniques through coursework, a visiting scientist appointment at Microsoft Research, and through self-directed learning.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

The inability of non-specialists to make full use of our tools occasionally requires us to revert back to non-reproducible methods in order to get things done in a timely fashion. We are working to solve this problem by building lightweight Application Programming Interfaces enabling collaborators to access some of the core elements of our workflow through simple web protocols.

What do you view as the major incentives for doing reproducible research?

Within a research team, major incentives include: increased transparency in methods, increased accountability and ability to check for errors in processing, a reduction in spin-up time as new members join the team, and an ability to minimize duplication of effort. Between the team and other collaborators/stakeholders, we see major benefits in the ability to share and visualize results, and in our capacity to perform cross-disciplinary research.

Are there any best practices that you'd recommend for researchers in your field?

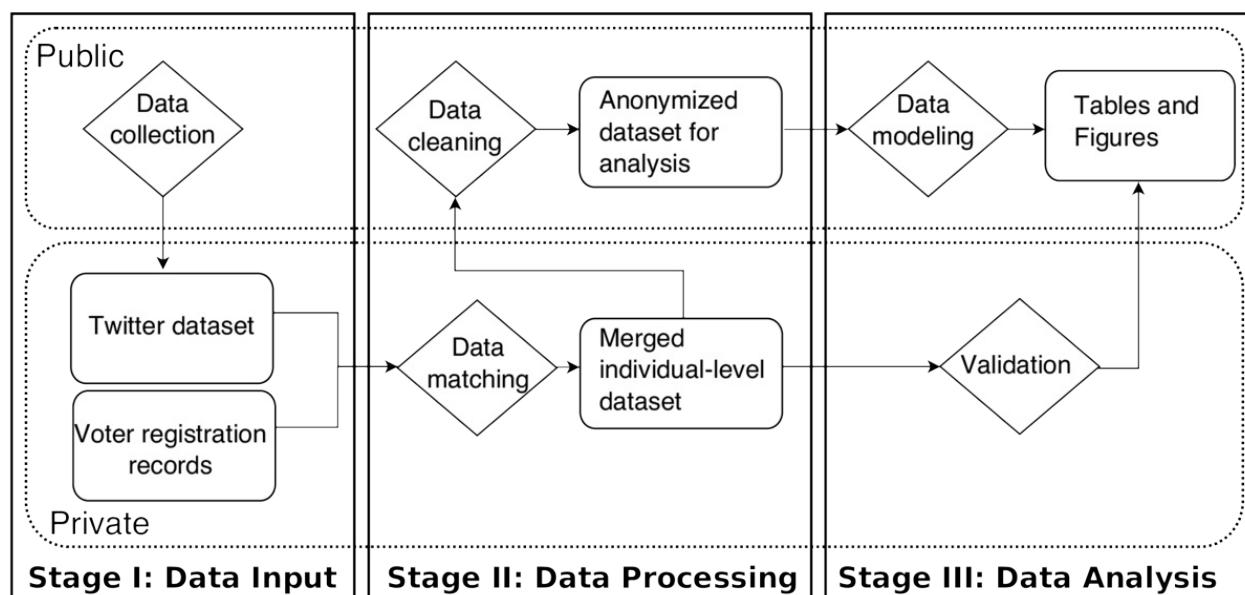
We recommend development and adherence to standards in geospatial data formats and distribution protocols.

The Trade-Off Between Reproducibility and Privacy in the Use of Social Media Data to Study Political Behavior

Pablo Barberá

My name is Pablo Barberá, and I am a political scientist who applies computational methods to the study of political and social behavior. I joined the School of International Relations at the University of Southern California as an Assistant Professor in 2016, after spending a year as a Moore-Sloan Fellow at the Center for Data Science in New York University. The workflow I describe here corresponds to part of my dissertation research, whose aim is to study political polarization on social media websites. In particular, here I focus on the research process that led to an article published in the journal *Political Analysis* in 2015, which presents a new method to estimate the political ideology of Twitter users based on the structure of their personal networks.

Workflow



An important concern in the design of projects involving social media data should be to guarantee that the private records of the individuals in the sample are protected, while ensuring that every step is reproducible. In the analysis described here, I only rely on publicly available information -- in particular, information about Twitter profiles and voting registration records in the state of Ohio, which is used for validation purposes. However, the goal of the project is to infer a sensitive latent trait about each Twitter users -- their

ideological position, on a scale from very liberal to very conservative. This is information that most users are not aware could be inferred based only on their public personal information, which raises the question of whether the concept of informed consent in sharing users' data -- as defined in the Terms of Service that users accept when they sign up for a Twitter account -- applies in this context as well.

To achieve the goal of reproducibility while adequately protecting users' data, all the analysis in the study takes place at two levels, private and public, as shown in the diagram in this chapter. The private level includes all the original data, which will be processed and merged, and then anonymized so that it can be included in the replication materials. These datasets will not be released, but they can be acquired by other researchers from their original sources. The second level contains all the R code and output (tables and figures), as well as the anonymized version of the dataset, which allows partial replication of the results in the paper even without access to the full dataset. After finalizing the project, the materials in this level were released in public repositories on GitHub and Dataverse.

As shown in the workflow diagram, the first step in the project was to collect a dataset that would allow me to reconstruct the networks of a sample of Twitter users. In particular, I compiled from Twitter's API the lists of followers of a set of around 500 political accounts in the U.S., which includes legislators, candidates, media outlets, etc. Then, I identified the list of users who follow at least three of these political accounts -- this will be the sample in the study. Finally, for each of these users, I also extracted their profile information, including their approximate location, which was parsed into geographic coordinates using the Data Science Toolkit. This data collection step was conducted using R tools developed by the Social Media and Political Participation Lab. All the R code used in this step was made public, but the complete Twitter dataset was stored privately in order to comply with Twitter's terms of service regarding data sharing.

The second step in the workflow involves two types of data processing tasks. First, the user-level information from Twitter was matched with publicly available voting registration records from the state of Ohio, which includes information about the party that each voter is registered with. A Twitter user was matched with a voter whenever there was a perfect and unique match of first name, last name, and county between these two datasets. This information will be used in the validation step in order to assess whether the ideology estimates that result from this method are correlated with offline measures of behavior, such as the number of times that a given voter has participated in a party primary election. The second part of the data processing task involves cleaning the Twitter dataset and building the networks that will be scaled in order to obtain estimates of their ideological positions. In particular, here I constructed an adjacency matrix that indicates whether each of the users in the sample follows each of the political accounts.

After these two steps are completed, I generated an anonymized version of both datasets. The anonymization was achieved by replacing Twitter and voter unique IDs by randomly generated numeric IDs. This will allow researchers to replicate every step of the analysis after this point, but without being able to identify the individuals in the sample.

In the data modeling step, the adjacency matrix that represents this network was scaled using the [STAN software for Bayesian modeling](#) using R. The model that was implemented was similar in nature to other latent space models applied to social networks. It builds upon the assumption that the existence of a following link between users and political accounts is inversely related to their distance on a latent ideological dimension. In other words, the intuition of the model is that users tend to follow political accounts that they perceive to be close to their own ideological position. This method returns estimates of the latent positions of both users and political accounts. The output of the model was carefully validated using a variety of offline measures of ideology for both types of actors, including roll-call votes in Congress, aggregate measures of ideology from surveys, and individual-level voting records. One of the strongest results of the paper is that individuals predicted to have the most extreme positions are those that most frequently vote in primary elections -- a clear indication that strength of ideological identities is correlated with strength of partisanship.

After conducting the analysis and validation, the last part of the project consisted of producing a series of tables and figures that summarize the dataset, describe the main results of the paper, and offer a graphical representation of the validation process. All figures and tables were generated using R. Throughout this process, I documented exactly what datasets were required to generate each figure, being careful to ensure that only the code and data available in the public level of the project were required in order to replicate them. These tables and figures were then integrated into the manuscript, written using LaTeX.

Pain points

The replication data and replication code were released in different platforms; the code in GitHub, the data in Dataverse. GitHub provides the ability to track changes in the code, and makes it easy to collaborate. Their online interface is easy to use, which reduces the entry costs for other researchers interested in forking the replication materials for their own projects. However, at the moment GitHub does not allow pushing files over 100 MB. Storing smaller files within this limit is not recommended either, since every change to this file is also stored in the repository. Dataverse, on the other hand, provides a free platform to store large files, with some built-in analysis tools, as well as some basic versioning system. However, it lacks the social layer of GitHub, the ability to collaborate, and a good interface to see differences between files using version control. As a result, at the moment there doesn't exist a single platform that combines the advantages of these two.

A problem that is more specific to the workflow described here is the difficulties in ensuring that the anonymization of private records is complete. As described above, replacing the original Twitter user and voter IDs with randomly generated numbers is an approach that in theory ensures anonymity. In practice, however, it might be possible to discover the identity of some of these individuals using only some of the other variables. For example, if there are unique patterns of following behavior in the dataset (e.g. only one individual follows all the political accounts in the dataset except for Barack Obama), another researcher could succeed at discovering her identity. These are edge cases, which may not occur in the dataset here, but it is a concern if the goal is to guarantee the privacy of all individuals in the sample. Recent developments in the field of cryptography, such as differential privacy, provide promising new methods to improve these research practices.

Key benefits

Most published studies that use social media datasets to study human behavior do not provide replication datasets. This is unfortunate because it represents an important obstacle towards ensuring reproducible scientific practices and limits the use of these materials for learning purposes, but it is also understandable, as the restrictive policies of social media companies imply that researchers need to devote a significant amount of time towards ensuring compliance with these policies. My hope is that the workflow described here can become a blueprint for future replication datasets in this field.

Questions

What does "reproducibility" mean to you?

A study is reproducible when a researcher external to that particular project, but familiar with the literature and methods, is able to obtain identical results by using the same datasets and following the same procedures as those described as in the research output, be it a published article or book. Researchers should also produce replication code and a lab book with more precise details about the analysis conducted as part of the study. However, this should be in addition to the description of the research process in the publication, since the output of running code may depend on software versions, for example. There is also the possibility that a set of results is not "correct," and simply the product of errors in the code or software bugs. In other words, being able to run a piece of code and obtain identical results as those described in a published output is not a necessary nor a sufficient condition for reproducibility.

When applied to studies that rely on social media data, the concept of reproducibility is slightly more nuanced. The Terms of Service of social networking platforms like Twitter or Facebook restrict the distribution of datasets obtained through their Application

Programming Interfaces (APIs) for privacy reasons. These companies have taken steps towards enforcing this requirement, including contacting researchers to request they take down replication datasets, even if they were used for research purposes only. The trade-off between ensuring individual privacy and allowing reproducibility is even more evident when social media datasets are combined with survey data or other individual records, as in the case I describe here. In these instances, reproducing a published study implies the additional steps of querying the API to reconstruct the original dataset and matching it with the individual records, which is inefficient and not always possible. The workflow I describe here represents my best attempt towards addressing these challenges and ensuring that other researchers can reproduce my results.

A Reproducible R Notebook Using Docker

Carl Boettiger

My name is [Carl Boettiger](#). I'm a theoretical ecologist in [UC Berkeley ESPM](#) working on problems of forecasting and decision-making in ecological systems. My work involves developing new computational and frequently data-intensive approaches to these problems.

My workflow seeks to provide a way to capture & reproduce the day-to-day workings of a computational ecologist using freely available platforms (e.g. GitHub, Travis CI, Docker Hub) and open source software (`R`, RStudio, `git`, `docker`, `jekyll`) in the format of an online, open lab notebook. I have tweaked and adapted this workflow over the past 5 years, often experimenting with new technology. Other researchers have frequently told me how they have adopted parts of this approach, but rarely in an identical way.

My general approach to an open lab notebook has been described previously (Gewin, 2013, Mascarelli (2014)), while I focus on documenting more details of the workflow here. When possible, I have sought to leverage general-purpose tools rather than custom solutions: for instance, I organize project directories using the R package format, as described in Gentleman & Temple Lang (2007) and [rrrpkgs](#) project, rather than introduce my own custom structure. Nevertheless, my current system no doubt remains too complex, specialized, esoteric and even fragile to be easily adopted by others. Rather, I encourage the reader to focus on specific elements or modules that look most practical, as others have done.

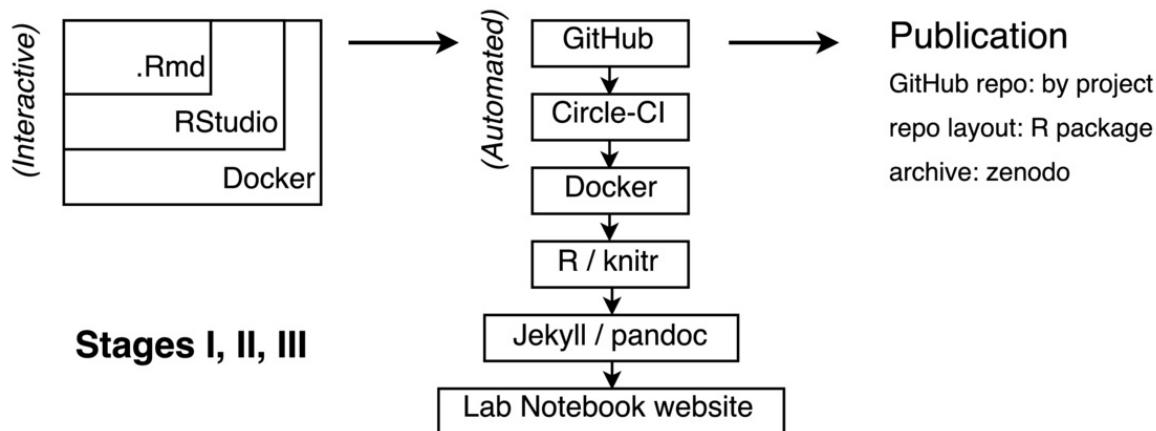
Workflow

Lab Notebook

GitHub repo: by year

repo layout: Jekyll

archive: figshare



A note to the reader: The following description is meant as a high level overview, which leans heavily on several powerful and well-developed tools and workflows including git/GitHub, docker/DockerHub, .Rmd /RStudio, and others. Table 1 provides a concise reference where a reader can learn more about these tools and their use.

Table 1: Tools used in workflow

Tool	Description / purpose	Website
git	Version control software	https://git-scm.org
GitHub	Online repository for sharing code managed with git	https://github.com
docker	Containerization software for portable computational environments	https://docker.com
DockerHub	Central hub for building and distributing docker containers	https://hub.docker.com
RStudio	IDE for editing R and Rmd files	https://rstudio.com
Rmd	Dynamic documentation format for R language	http://rmarkdown.rstudio.com
pandoc	convert between document formats	http://pandoc.org
servr	An R package for combining jekyll with Rmd	http://yihui.name/knitr-jekyll/
jekyll	Static website generator closely integrated with GitHub	https://jekyllrb.com
Circle CI	Flexible continuous integration software for executing scripts pushed to GitHub	https://circleci.com
figshare	Permanent data archiving platform	http://figshare.com
zenodo	Permanent archiving platform (handles code/software well)	http://zenodo.org

Interactive workflow

My daily workflow on an active project simply involves opening a new `.Rmd` document with the day's date in my lab notebook. In this file, I write the code, text, equations, and other elements of my work (see diagram, top left).

At the heart of my workflow is the dynamic documentation tool `knitr`. `knitr` is an R package that is tightly integrated into RStudio and R-markdown, or `.Rmd` format it supports for integrating code, documentation, equations, figures and other components of research into a single document. Its key feature is the ability to "knit" or "execute" the document, resulting in the code blocks being run and their output figures, tables, and so forth being displayed in the document. Text and code are written together in the popular, simple, and flexible markdown format, which is widely recognized by other tools (e.g. GitHub, a widely used code repository, and Jekyll, a ruby-based static website generator). Markdown is easily converted into other formats by `pandoc`, a conversion engine integrated into RStudio (and other popular platforms such as Jupyter) which can generate LaTeX, HTML, Microsoft Word and other document formats. This flexibility is useful later in turning my `.Rmd` files into either HTML pages for my laboratory notebook or into other formats suitable for traditional journal publication.

During active research, I often find it impractical to clearly separate out the stages of Data Input (Stage I), Data Processing (Stage II), and Data Analysis (Stage III). I merely strive to have all of these stages coded and explained in the `.Rmd` document.

I write / edit this `.Rmd` file inside an instance of RStudio which runs inside a Docker container, which in turn may be running on my laptop, an Amazon Web server, or even an NSF super-computing cluster depending my needs that day. RStudio is a popular integrated development environment for R users which can be run in server mode through a web browser. Docker is a popular containerization tool which allows one to create a portable image of one's entire software environment that can be easily moved around between different computers, regardless of architecture. I believe this has major implications for addressing common problems in reproducibility, as I have described more fully elsewhere (Boettiger, 2015). A Dockerfile in my notebook provides an executable recipe for building this computational environment on top of existing, general-purpose Docker images maintained by the [Rocker project](#).

Automated workflow

At regular intervals I "commit" my notebook in `git` and "push" this progress to GitHub, a widely used version control system and public repository for code and other digital material. This triggers the automated build portion of my workflow, illustrated in the center of the diagram. A Continuous Integration platform ([CircleCI](#) in my case, as the more widely used platform, [Travis](#), did not support Docker execution until much more recently) detects this commit, and begins to execute and assemble my code.

The CI platform begins by pulling down a public image of my computational environment, itself built automatically by Docker Hub from the Dockerfile in my repository. A separate Docker volume container can also be pulled from the Hub which contains results cached by

knitr for any code too intensive to run on the (free, public) CI platform.

As the notebook is already organized as a Jekyll repository, just with `.Rmd`-formatted posts instead of plain `markdown`, existing tools (see `servr`, Table 1) can easily execute the R code and format it as a new post in the notebook. Jekyll templates make it easy to add semantic metadata to the post automatically including bibliographic information, links to version history, commit hash, modification date and so forth. At this time a given exploration might not have a particular project connected to it -- it might build from several existing projects, a paper I'm reading, or represent an entirely new exploration. I use categories and tags in the notebook to associate the post with relevant projects or themes, which makes it easier to come back to. (Figuring out appropriate tags is harder than it sounds!)

Each year I archive the GitHub repository that contains that year's notebook on figshare, adding the DOI badge to the repository's README.

Project finalization / publication

Eventually multiple entries will relate to the same project. At this point, I frequently want to reuse code first developed in a previous entry. This is my signal that it is time to create a new project on GitHub. (Figuring this out is much harder than it sounds!) I create a new public GitHub repo using a name that matches a tag in the relevant notebook entries. In the `R/` directory I store functions that provide these reusable bits. For non-trivial functions, I try and develop unit tests (in the `/tests` directory) -- these usually come directly from the interactive tests I write in the notebook when first creating these functions. I also add minimal Roxygen documentation to the functions I create, usually just to remind me what the input and outputs are. Data goes in the `/data` directory; or more frequently, as R scripts that either simulate or download and clean the data from external sources.

Notebook pages do not load these functions as a single package -- as the package is constantly changing this is unlikely to continue to work anyway. Instead, they source in the script directly from the version-explicit links on GitHub. (I learned this the hard way). This avoids the burden of making sure the 'package' is always installable, it just serves as a convenient organizational skeleton.

I continue to develop, test, and explore results in the pages of the notebook, adding and modifying functions as necessary. This usually involves plenty of mistakes and dead ends that are captured in the history of an individual page (when I modify an existing workflow to correct the results) or are left as dead (or incompletely explored) ends in the various pages of the notebook under that category.

Once the work has coalesced around a particular set of ideas and results appropriate for a single manuscript, I begin drafting the manuscript as a `.Rmd` file in the GitHub repository, often based on `.Rmd` files from the notebook. The `rticles` package from RStudio provides

a template system which makes it easy to render `.Rmd` files into `pdf` articles for various journal formats.

When preparing for submission, I upload a copy of the manuscript (in `tex` format, generated from the `Rmd`) to the [arXiv](#) and configure the Zenodo permanent archive which connects automatically to GitHub, much like a Continuous Integration service. Zenodo then generates a permanent archive with a unique Digital Object Identifier (DOI) every time it detects a new 'release' on GitHub. GitHub releases are part of the `git` tag system and can be used to signal a new version of software or publication of a paper. A DOI badge from Zenodo is then displayed on the GitHub repository.

The reader is encouraged to view any of the real-world examples of this process in the repositories of my recent projects, such as <https://github.com/cboettig/nonparametric-bayes>, or in the pages of my online lab notebook at <http://carlboettiger.info/lab-notebook>.

The frequency of these steps is highly variable -- from many commits a day to gaps of months. See my GitHub commit history for a more realistic answer. In addition, although most of my research projects involve others, I am the only researcher committing to my lab notebook, just as we see in paper notebooks. The final research product will see more direct involvement by others.

Pain points

Knowing when to refactor and how to avoid fragile and opaque design. A good reproducible workflow should be like good software: built from simple, easy-to-understand modules that do one task well. Most reproducible workflows, mine included, can too readily resemble most scientific spaghetti code: pieces tacked together over the years because they got the job done. The best way to make a workflow or code understandable is to *refactor* it after it works, breaking it into well-defined, well-tested modules with clear input and output.

Pretending research can be written like this from the start is fiction, but just capturing all the messiness provides none of the abstraction that makes something more re-usable and reliable. I don't have a good solution for how to do this though -- refactoring is demanding and offers few incentives.

Key benefits

A key benefit of this approach is making my work portable and scalable. By making it easy to reproduce my computational environment and analyses, it suddenly becomes much easier to re-run an analysis on a cloud machine or cluster if it proves too large for my local system.

A second benefit has been the ability to explore research ideas more easily. New ideas often build on old ones, and the dread of having to remember how some old stuff worked in the first place before tinkering with it to explore something new was often enough to make me turn to something easier.

Key tools

I believe any of the tools mentioned in Table 1 could be of use to a broader audience. I have tried to place the more general near the top -- GitHub and Docker address very general issues in computational reproducibility, justifying their wide adoption. These tools can be inserted into many common workflow patterns without requiring significant re-tooling.

For R users, RStudio has made the Rmd format far more practical as an authoring environment, both for websites (e.g. with `servr` package) and journal articles (`rticles` package). However, these tools may require both a bigger shift from existing strategies and offer a smaller benefit.

The particular pattern I have used to chain this together with CI, etc, is probably less generally applicable, and has a higher learning curve than the afore-mentioned tools.

Questions

What does "reproducibility" mean to you?

Reproducibility in this context is 'computational reproducibility.' It means a good-faith effort to make sure that the analysis can produce qualitatively identical results while running on comparable hardware. This means certain things do not need to be reproduced: e.g. how long the code takes to run may vary by hardware and operating system, but this is okay. Nor am I not concerned with bitwise identical results, nor with necessarily reproducing stochastic random draws -- rather, I expect conclusions from reproducible results to be robust to the details of stochastic seed or choice of random number generator.

I am also concerned that reproducibility is modular -- that individual components of the analysis can be reproduced (and thus recombined or otherwise modified), and not merely provide a black box that can only replicate final outputs without variation or adjustment.

Lastly, I think it is important to identify *who* should be able to reproduce the analysis. Like the paper itself, the analysis requires a certain degree of expertise to understand, and I do not expect that individuals with no familiarity with programming, statistics, or scientific process can reproduce the analysis. However, I do expect that researchers with some scientific background in my area (e.g. the broadest readership of the journal in which it is published)

and with minimal familiarity with the R language or similar computing langauge can reproduce the overall results after suitable investment of time and effort in reading the documentation.

Why do you think that reproducibility in your domain is important?

Reproducibility makes results more reliable, and more importantly, makes it easier to extend, test, and build upon existing results. Ultimately this makes it easier for an individual to build on their own work and the work of others, making for faster, better science.

How or where did you learn about reproducibility?

Independent study of examples, experimentation, and reading, and connecting with other researchers sharing similar interests through the internet and social media.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

Not a standard practice. In the short-term it takes more time. It may also increase the probability of errors in your work being discovered.

What do you view as the major incentives for doing reproducible research?

Making research easier to do. Reproducible research facilitates collaboration, particularly with myself. It improves my confidence in my own results and helps me build more efficiently on work that I have already done.

Are there any best practices that you'd recommend for researchers in your field?

Adopting tools that are widely used within my field (and others) for reproducibility. These include: GitHub, Docker, rmarkdown.

Would you recommend any specific resources for learning more about reproducibility?

The documentation linked in Table 1 would be a great place to start on any of the individual tools. Additionally, see the reproducible research workshop developed by NESCent:
<https://github.com/Reproducible-Science-Curriculum>

References

Boettiger, C. (2015). An introduction to Docker for reproducible research, with examples from the R environment. *ACM SIGOPS Operating Systems Review*, 49(1), 71–79.
<http://doi.org/10.1145/2723872.2723882>

Gentleman, R., & Temple Lang, D. (2007). Statistical Analyses and Reproducible Research. *Journal of Computational and Graphical Statistics*, 16(1), 1–23.
<http://doi.org/10.1198/106186007X178663>

Gewin, V. (2013). Turning point: Carl Boettiger. *Nature*, 493(7434), 711–711.
<http://doi.org/10.1038/nj7434-711a>

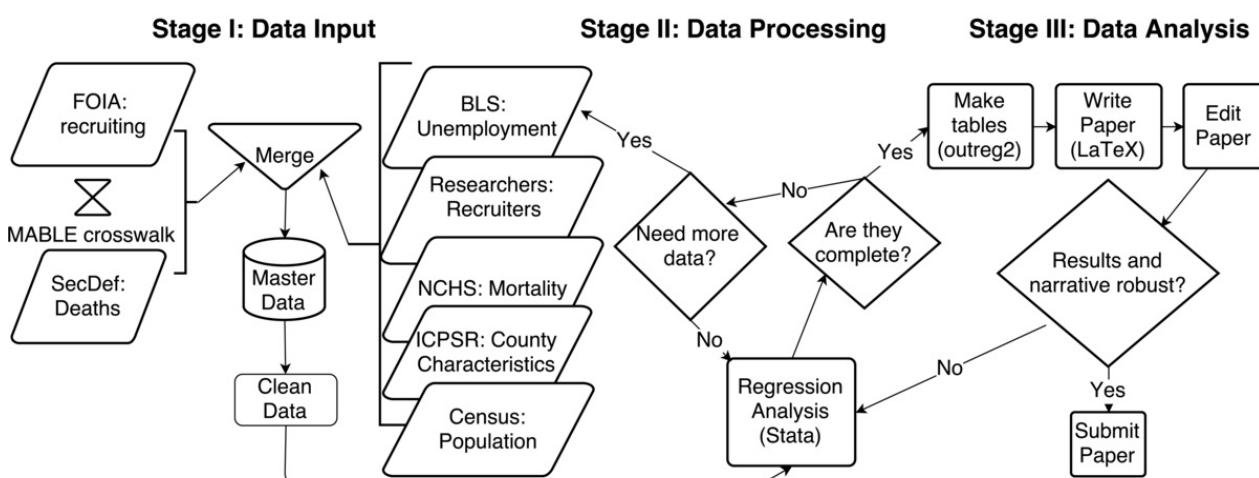
Mascarelli, A. (2014). Research tools: Jump off the page. *Nature*, 507(7493), 523–525.
<http://doi.org/10.1038/nj7493-523a>

Estimating the Effect of Soldier Deaths on the Military Labor Supply

Garret Christensen

My name is Garret Christensen. I am currently a project scientist at the [Berkeley Initiative for Transparency in the Social Sciences](#) and a fellow at the [Berkeley Institute for Data Science](#). I work on questions of program impact evaluation in labor and development economics. I conducted the research described below as part of my dissertation work in economics at UC Berkeley, beginning in 2010. This research is on the effect of deaths of US soldiers in Iraq and Afghanistan on military recruiting. I use panel data methods with fixed effects to try and identify the causal impact of the death of a US soldier in Iraq or Afghanistan on recruiting in the soldier's home county.

Workflow



This project started with the idea, which I got from reading the newspaper, and hearing anecdotes about recruiting stations being overwhelmed after 9/11, and seeing popular reaction to the battle of Fallujah a few years later. The next step was obtaining the main data; a colleague just happened to have relevant data--the universe of US enlisted military recruits--through a Freedom of Information Act.

The other half of the main data is deaths of soldiers from Iraq and Afghanistan. I obtained this data from a [public Defense Department website](#). Perhaps unsurprisingly, the original website is no longer operable. Luckily I still have, and have archived on Dataverse, the original dataset I downloaded and used. Updated versions of the data are also still [available publicly](#). To merge the recruits and deaths data, I used Missouri Census Data Center's

[MABLE/Geocorr](#) to construct a geographic crosswalk. This uses census geographies, so I used the 2000 Census version, but sadly I don't think I recorded every exact option used when constructing this crosswalk.

I used Stata to merge this data together as well as to do all subsequent statistical analysis. No, Stata is not open source, but it's what most economists use. I did all my work in script (.do) files, and with the 'version' command, theoretically any other user should be able to produce the same results. The code was version controlled, but only after a fashion by updating script files with the date as part of the file name. Old versions of files just got dumped into an archive folder, where they were kept permanently.

The merged data was analyzed using the `xtpoisson` (poisson) and `xtreg` (linear) regression algorithms in Stata. Regression tables were output to tab delimited plain text files using the user-written '`outreg2`' command, edited in Excel, saved as .pdfs, and then included in the LaTeX file that made up my paper. Clearly, Excel is the antithesis of reproducible, but I didn't change numbers in the tables, just formatting. Next on my to-do list with the paper is to cut out this clunky step and do directly from Stata to LaTeX. I think the only reason I started out this way is because I wasn't comfortable enough with LaTeX to figure it out.

Regression output was by no means complete the first time. I looped back numerous times to add more data from other sources, such as the Bureau of Labor Statistics, the Cenus, ICPSR, etc. This would require updating the merging and analysis code, reformatting tables, and changing the text in the paper that refers to specific output. Unfortunately that process is still ongoing because the paper has yet to be accepted at a journal.

Pain points

Given that this research is based on observational data, and I did not pre-specify my statistical analysis plan, I highly doubt that any other researcher who looked at my original raw data (or even my cleaned final data) would agree on the exact set of regression specifications that I should include in my paper. For the sake of transparency, however, I try to include a nearly-exhaustive set of alternative specifications in a lengthy appendix to the paper. For example, all the results are available using both log-linear and Poisson regression specifications.

Regarding the data, even though I am very grateful to staff at the Defense Manpower Data Center who provided me with the data, I'm less than confident that multiple identical FOIA requests for the exact same data would result in identical datasets, since I don't have access to the original data, and have no way to verify if the dataset I was given was the true universe of observations I requested. Perhaps that's just an issue with all original data--you'll never be able to go back to all the homes in the census and check their answers, but you do have the ability of downloading the data from the census server. I don't have access to the

Defense Manpower Data Center's servers, but if I make available the data they gave me, does that mean we're reproducible? What if, as actually happened, you notice a completely implausibly low number of a certain type of observation in what is supposedly the universe of such observations?

Lastly, I'd say that my code is fairly well-documented, though it took a lot of work to get it there. Reading through it, I hope that other researchers could understand what the code is doing. There isn't yet a readme file but there is one master .do file that should (in theory) be able to reconstruct everything I've done from scratch. I have worked on the project for several years, with a few large breaks since economics journals can take 6 months to make decisions. I had to go back and extensively re-examine code that I no longer remembered. Having done that a few times, the code now seems fairly well-documented. This process would have been much easier had I kept a research log. I've had to open dozens of dated versions of the same file to find the last one written before a major change, which would have been much easier with version control or a research log. **### Key benefits**

I would say that use of specific version control software is relatively new to the social sciences. When I began this work in 2010, I had never heard of git. I just used the method I learned from my adviser: include the date in the name of files, and every time you make significant changes to a script file (called a ".do file" in Stata), change the date. Using a distributed version control system (DVCS), as I do now, is a significant improvement.

Key tools

An excellent reproducibility tool to use is using the [outreg2](#) (or [estout](#)) user-written commands in Stata to automatically turn regression output into journal-formatted tables. Although I use these commands, at present I have a clunky two-step process to first output the tables as .csv files before editing the formatting slightly then including the tables in my ultimate paper written in LaTeX. Ideal would be to use these commands to output the tables directly as .tex files, and include them in my paper file.

Questions

What does "reproducibility" mean to you?

Reproducibility to me is the ability of other researchers to get the same results as in my paper. In the weakest form, this would simply be for other researchers to be able to download my final datasets, run my final analysis code with nothing more than a file path name change or two, and get the exact results that are in my paper. A better version of reproducibility would be for other researchers to download my original raw datasets--the major two of which I have made publicly available using [Harvard's Dataverse](#)--redo my

extensive merging and cleaning of data, and then get the same results. Even better would be for others to go through the same Freedom of Information Act (FOIA) request process from the Office of the Secretary of Defense that a colleague and I did, redo the merging, redo my analysis, conduct the analysis they themselves see fit, and get the same results. I have some concerns about that, which are described below, but I've done the best I can, and rest on the assumption that any missing data is not correlated in a way that biases my estimates, and that I was thorough enough in my analysis that my results are robust to other forms of analysis.

Why do you think that reproducibility in your domain is important?

Significant errors have been discovered in high profile published economics research. In one sense, economics is doing well because many top journals require data sharing, so it's actually possible for these errors to be discovered since replicators have access to data. But without a systematic replication or code-checking of analysis, we still don't know what fraction of research suffers from these problems. Should we throw out the baby with the bathwater? I don't think so, but we don't know right now.

Also, even when economists share their data, they very rarely share their raw data, and all of their cleaning code, and instead only share their final data and analysis. We're humans, so we're probably making some coding mistakes that go unnoticed.

How or where did you learn about reproducibility?

My graduate adviser Edward Miguel taught me the simple method of version control with file names while I was working on a project of his as a graduate research assistant.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

A lot of the advances in transparency and reproducibility in economics are coming from the medical sciences, where randomized control trials are nothing new. But RCTs are still a very small minority in economics. Like my work here, most work is observational. Economics only created the [AEA RCT Registry](#) in 2013, and there has been no serious discussion of registration of observational work. Should we register observational work? Should we pre-register our statistical analysis plans for observational work? This is all uncharted territory in economics.

What do you view as the major incentives for doing reproducible research?

Being reproducible requires extra up-front costs. In the long run, the benefits should outweigh the costs, because when someone comes along and wants to extend or replicate my research, they won't find any embarrassing errors in my work.

Are there any best practices that you'd recommend for researchers in your field?

Comment the hell out of your code so you know what you were doing when a journal makes a decision on your submission after 6 months. Save all your analysis files using version control. Use a one-click workflow to incorporate your tables directly into your paper so you don't lose track of output.

Would you recommend any specific resources for learning more about reproducibility?

J. Scott Long's [*The Workflow of Data Analysis using Stata*](#)

My [*Manual of Best Practices in Transparent Social Science Research*](#)

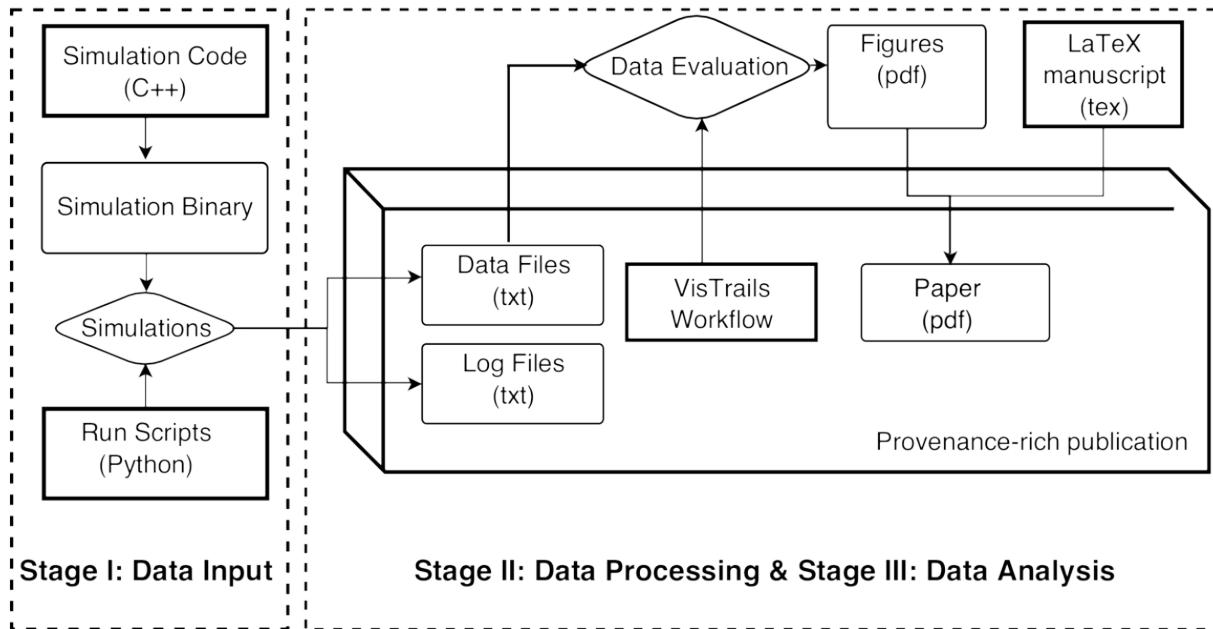
Turning Simulations of Quantum Many-Body Systems into a Provenance-Rich Publication

Jan Gukelberger and Matthias Troyer

My name is Jan Gukelberger, I am a computational condensed-matter physicist, who recently completed his PhD at the Institute for Theoretical Physics, ETH Zurich. This case study describes a project I worked on during my first year as a PhD student (2010-2011). This case study was conducted together with my advisor, Matthias Troyer.

Broadly speaking, the project's goal was to characterize a family of quantum many-body model systems. A specific model is described by a large matrix, the Hamiltonian, and its physical properties can be deduced from the lowest-lying eigenvalues (energies) and the corresponding eigenvectors (quantum states). Therefore I wrote a C++ program that would build the Hamiltonian matrix for a given set of model parameters, run an iterative diagonalization algorithm, and output the corresponding properties. Analysis of the results produced by this program for different parameters yielded a deeper understanding of the studied model family and corroborated analytical results obtained by colleagues. The analytical and numerical results were finally published together in [Phys. Rev. B 85, 045414 \(2012\)](#).

Workflow



Since the simulations may require a large amount of compute resources (on clusters or large workstations), it is usually not feasible or desirable to re-run the whole process in one go. We therefore typically adopt a two-step approach: The output of the simulation runs is treated as primary/raw data, which is archived along with log files containing detailed information about source code version, execution environment, and input parameters. The evaluation and transformation of this raw data to the final results (typically figures with plots) should then be as easily repeatable as possible, ideally with a single push of a button or script execution.

In this study, we opted to publish the raw data as supplementary information on the publisher's (APS) web server and provide workflow files for the [VisTrails](#) system, which would retrieve the raw data from the server and recreate the figures contained in the paper. VisTrails is an open-source scientific workflow and provenance management system which was used for the data evaluation and plotting tasks in the project. This way, any reader can inspect in detail and rerun all steps of our data analysis.

At the beginning of the project is the development of a simulation code in C++. Once the code is ready, it is used to explore the properties of the physical model under study. For this purpose, it is compiled and run with different input parameters on different systems (workstations and clusters). Because the simulation code is adapted and expanded continuously over the course of the study, it is essential to record what version of the code produced which results. To this end, we use a run script (Python), which records the code version (subversion revision), input parameters for the run, as well as details about the build configuration (compiler, libraries, etc.) and the system environment in which the code is run (host, date & time, dynamic libraries, etc.). All these details are written to a log file next to the data file that contains the results of the simulation. Both are semantically linked to each other by having the same file name, up to the extension (.dat and .log).

These output files constitute the raw data, which is collected on a desktop system for evaluation. The evaluation typically loads data files from several simulations (corresponding to different input parameters), computes some numerical transformations of the data, and finally produces one or more figures (pdf files) with data plots. We code the evaluation process in VisTrails workflows, making use of the [ALPS](#) package (delivered with VisTrails), which contains many utility routines for common processes like data transformations, fitting and plotting. We generally aim for a separate workflow (VT file) for each figure. This increases modularity and makes the development of the workflows easier, but implies that several VT files need to be opened and executed if all figures are to be recreated.

Finally, the manuscript of the paper is written in LaTeX, including the figure files created by the VT workflows. LaTeX compilation produces the paper as pdf, which constitutes the central part of the publication.

Publishing the paper together with the raw data and workflows, such that readers could easily inspect and reproduce our data evaluation process, turned out to be a challenge in its own right and required intense interaction with the publisher. Here, the main problem was the need for cross-references between the manuscript, the VT workflows, and the raw data, because the final location of each component only becomes available in the last step of the production process, when the files cannot be changed anymore without manual intervention from the production team. Some aspects of this issue are explained in detail in our report [Publishing provenance-rich scientific papers, Procs. TAPP'11](#). In the end, the publisher was not able to insert links from the figures in the paper to the corresponding workflow files, but only a general reference to the supplementary material section on their server, where all the workflows could be downloaded.

Note: One collaborator actually recreated the figures with a different plotting tool before publication in order to improve their visual appearance. For this purpose, we amended the VT workflows to export the preprocessed data to an external file before plotting. Therefore, the figures presented in the paper are equivalent, but not identical, to the ones created by the VT workflows.

Pain points

Apart from the non-trivial publishing process, the main pain points during the study were connected to the fact that the data evaluation had to be done in the VisTrails GUI and to the opaque-ish VisTrails workflow file format:

- Data evaluation (execution of VT workflows) could not be scripted at that time.
- The evaluation could not be run on a cluster/via ssh.

- Version management was harder because viewing differences between versions was not as easy as looking at the diff file for a Python script.
- This also made the synchronization of workflows between different machines (e.g. laptop and workstation) less straightforward.

When now inspecting the "reproducible publication" on the APS server, three years after publication, some mid- to long-term issues become obvious, because both the used software and the publisher's infrastructure is evolving. Continuous testing and maintenance of the published instructions and workflows would be needed in order to keep up with the changes:

- The instructions we provided in the supplementary materials section accompanying the article do not work out of the box with the current VisTrails version: In the most recent stable VisTrails release at the time of writing (2.2), the ALPS package is broken and needs to be patched with the latest (not-yet-released) ALPS version. Otherwise initialization of the ALPS package fails and the workflow cannot be executed.
- The APS journals were not able to guarantee a long-term stable location for supplementary material. In fact, the URL has already changed, such that the workflows fail to fetch the raw data from the APS server, unless the URL is fixed manually in each workflow. For one specific example, the original location http://prb.aps.org/epaps/PRB/v85/i4/e045414/dyl_ladder_gap.zip has been changed to http://journals.aps.org/prb/supplemental/10.1103/PhysRevB.85.045414/dyl_ladder_gap.zip. Also, the cause of the error is not easy to fix for the uninitiated, because the DownloadFile module actually succeeds (it downloads the html file shown at the old URL), but the subsequent UnzipDirectory module fails with the message "BadZipFile: File is not a zip file". Hence we, the authors, need to prepare new workflows with adapted URLs and send them to the publisher for replacing the original ones whenever their infrastructure changes.

For these reasons I would now prefer to publish a self-contained archive containing the raw data and a script with minimal dependencies in a wide-spread language, such as Python, which reruns the analysis and reproduces the figures. This would be more robust with respect to changes in the publisher's infrastructure. Also, backwards compatibility issues might be expected to be solvable more easily in the long run for scripts in a wide-spread language, compared to special purpose solutions like VisTrails/ALPS (no matter how professional and helpful the developers of the software may be at the moment).

Key benefits

One of the most important points is recording exactly what version of the simulation code was run with what kind of input parameters. This excludes some of the worst cases of "non-reproducible results" and should definitely be a standard practice. (I cannot judge how established this practice is in our field because code and log files are rarely published.)

A second point is the actual publishing of raw data and evaluation workflows, allowing any reader to directly inspect all details of the evaluation process -- even those that the authors did not deem important enough (or forgot) to mention in the paper. This is clearly not widespread practice in our field and would be quite desirable in my opinion.

Questions

What does "reproducibility" mean to you?

In general, given a publication (in a refereed journal), source codes and raw data (which might be available publicly or in the institute's repositories), an expert from my field should be able to understand, and in principle repeat, every step of the study from the running of the correct version of the simulation code to the final results presented in the published paper.

How or where did you learn about reproducibility?

Some basic principles are quite evident, but integrating them in an efficient workflow may require some programming/version control experience. I came into contact with the VisTrails software due to a collaboration between our group and the VisTrails developers, aimed at integrating the evaluation tools of the ALPS package (developed within our group) with VisTrails.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

The main challenge is probably making the recording of provenance data as simple as possible, so no-one has an excuse not to do it.

Another point is that simulation codes, raw data, and evaluation tools are rarely published. Most researchers are very reluctant to publish their codes, e.g. because they do not want competitors to publish results produced with their code before they can, or because they feel ashamed of the poor quality of their code. Raw data may be large and in non-standard format. And the evaluation may be performed by a chain of different tools, which makes publishing of the workflow hard.

What do you view as the major incentives for doing reproducible research?

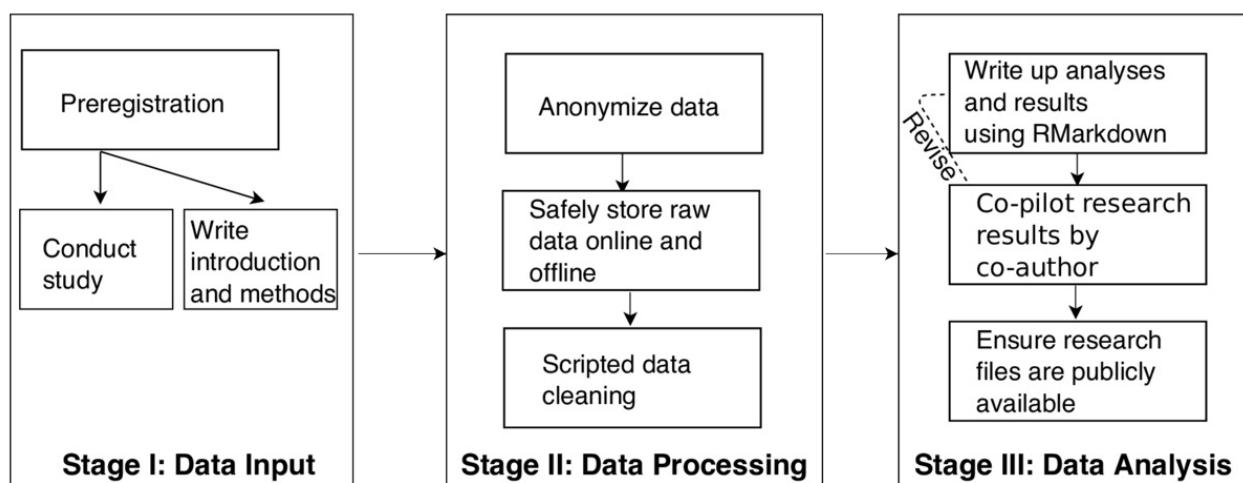
Apart from research ethics and institutional requirements demanding this, the recording of provenance information can make a researcher's life significantly easier when he/she discovers a discrepancy between different sets of results produced during a single study or in studies by different researchers. (Are the discrepancies caused by differences in the code, different input parameters, or data evaluation?)

Validating Statistical Methods to Detect Data Fabrication

Chris Hartgerink

My name is Chris Hartgerink, I am an applied statistician at Tilburg University specializing in detecting potential data fabrication with statistical methods. As a PhD candidate I pay attention to my workflow in order to increase efficiency and ensure it applies modern tools to improve my research. My case study will revolve around a project where I assess the performance of statistical methods to detect both genuine and fabricated data. In this project, I collect genuine datasets and invite researchers to fabricate datasets, to which I apply statistical methods to detect data fabrication.

Workflow



Statistical methods can be applied to detect potential data fabrication, but their validity is unknown. Simulating how researchers fabricate data is impossible because we simply do not know how researchers actually fabricate. Anecdotal summaries of uncovered misconduct cases do exist but are not generalizable. In order to test the performance of statistical methods in an ecologically valid manner, I invited researchers to fabricate data for a study for which we also have genuine data. With this, we can test the validity of a set of statistical methods to detect data fabrication. More specifically, genuine data contains sampling variance, whereas fabricated data frequently contains insufficient sampling variance. An example of such an analysis is testing whether nonsignificant p-values are uniformly distributed, or whether higher p-values occur more frequently than lower p-values (which is theoretically implausible).

At the beginning of the project, we had an initial meeting to detail specific aspects of the project. This meeting is crucial to determine and assign responsibilities, start discussion of ethical considerations, and how the project will be conducted. For this project the main points we discussed were (i) the ethical obligation to guarantee participants that they would remain completely confidential, given that they were technically required to breach ethical standards, and (ii) how to convince the participants that this study had justified reasons to request this behavior. We required the participants to generate data, and decided to not save any identifying information by default and permanently delete the identifying they gave us as soon as practically possible. We decided to motivate the participants with the study goal itself and reward them depending on whether we could detect their data fabrication with statistical methods. Several default points I like to address in this initial meeting:

1. Agree that research files will be publicly shared by default and proper arguments are needed to *not* share research files publicly (e.g., ethics committee restraints).
2. Agree that the publicly shared research files are put in the public domain (licensed Creative Commons 0), to maximize breadth and clarity with respect to reuse rights of the research materials.
3. Agree that the research manuscript will be shared as a preprint upon completion for improved peer feedback, and that the manuscript will be published openly and with an appropriate and clear reuse license only (i.e., CC-BY or CC-0, definitely not CC-BY-Non-Commercial).
4. Role assignment (e.g., project lead, fund raiser, analyst, who will check the analyses).
5. Specifying research project

The project-lead subsequently followed up with a draft description of the project and an initial draft of the research materials. These contents included a technical description of the design, hypotheses and theoretical framework, but also a draft of how the data would be analyzed. This increased transparency in the analytic methods increases accountability amongst authors. Explicating these methods is important because things that seem obvious might not be, and this helps get everyone on the same page (e.g., do we include covariates? Do we agree how covariates are measured?). After sharing these drafts and iteratively revising them, the experiment was submitted to the ethical committee for review, given that the study involves human participants. Considering that each university has different procedures, I will not elaborate on these here.

The files that are drafted after the meeting are included in a [GitHub](#) repository, which allowed for version control of the files, providing a logbook. These files were synchronized to an Open Science Framework (OSF; [osf.io](#)) repository to increase discoverability and shorter URLs to include in the manuscript. Version control can be compared to track changes for computer files, allowing you to go back in time and view what changes were made and

when. With version control a logbook of changes is created, which improves the reproducibility of the research process if anyone is ever interested. My personal experience indicates that this logbook is rarely inspected (except in data audits), but it can serve as a handy reference when somebody asks you when a specific event occurred (e.g., when were the analyses programmed for the first time). Version control is most effective when started immediately after the initial meeting.

After the ethics approval was acquired, the version controlled research files were preregistered. This preregistration is done on the OSF. This preregistration makes an unalterable snapshot of the research files with a timestamp, which provided a confirmation that what we set out to do and expected was indeed *a priori* to actually conducting the study..

Following the preregistration, we actually conducted the study. Because our study ran for several weeks to reach the quorum of participants, I used that time to rewrite the preregistration into the introduction and method sections of the manuscript. Having planned this beforehand, I wrote my preregistration in such a way that it already resembled these sections. However, I can also recommend to be more detailed in the preregistration and subsequently prune it into the manuscript, given that manuscripts typically do not contain all the study details that matter.

When the study was conducted, the raw data were stored in a non-proprietary file format and as read-only files. It is important to ensure the file is read-only, so no accidental adjustments would be made to the file and that I could comply with the data policies (i.e., original file always needs to be retained). Saving these raw data as a non-proprietary format meant that I did not save it as an SPSS, Excel, or other commercial format, but as a clear-text file (e.g., a comma separated value (CSV) file). Clear-text files ensure that the data will remain readable in the future, whereas proprietary file formats might not be. Moreover, with clear-text files other people are not required to acquire commercial software packages to read the data.

After having safely stored the data, I cleaned the data in an automated, scripted manner. I conducted my data cleaning in `R`. I try not to clean data by hand, because I often forget what I have done; scripted data cleaning prevents this entirely. If I do need to manually clean data, the logbook provided by version control is a safety measure to allow the hard route to reproducibility; fully automated data cleaning is in the end the easy route to reproducibility. For example, in this project I had to split responses into separate datapoints, which required a few hours to automate, but doing it manually would have cost me more time and would have made it less reproducible (and more error-prone)

Subsequently I conducted the analyses in `R` with `RMarkdown`. `RMarkdown` allows data analysis and writing to be conjoined into one file. As such, all results were directly generated into the manuscript dynamically. This helped me to prevent errors and to increase the

reproducibility of results. For example, the statistical result $F(1, 12) = 5.43$, $p = .037$ would not be typed in by hand, but automatically generated with `R` code. With this we not only enhance reproducibility, but also prevent human errors; in the previous result $p = .037$ should be $p = .038$, a simple rounding error which is quickly made. For this project, I had written several specific functions to test for uniformity of p-values that were fabricated when there truly was no effect, to analyze the sampling fluctuation of the variances, and to combine these statistical tools to detect data fabrication. Based on these results, genuine and fabricated data collected were classified as genuine or detected with the statistical tools, which indicated the performance of these methods.

Upon completing the analyses, I requested a co-author to check all the analyses and results (what we call co-piloting). These comments lead to changes in the analysis script (e.g., data handling error), which were not a problem given the dynamic `RMardown` manuscript (another benefit: not having to redo all the numbers in the manuscript). After these errors were revised and checked once more, the manuscript and results were (mostly) reproducible. I typically do a final check to see if everything went as planned and whether all analyses can be run on an independent computer (i.e., whether there are unspecified dependencies).

The final step, prior to submission of the manuscript, is to ensure that the analyses corresponded to the preregistration and that all research files were made publicly available. Research has indicated that researchers who preregister analyses frequently report other analyses, indicating that is easy to forget what you actually planned to do at the start. Cross-checking this allows to pick up on these errors in time. Additionally, I have seen several articles where researchers said they made the research files publicly available, where their files were uploaded (e.g., Github) but not yet made public. These final checks ensure that results are according to the preregistration plan and can be accessed by others.

Pain points

The part of a reproducible workflow that I consider particularly painful is that of co-piloting analysis scripts. It shows when a researcher is reproducible but also shows it can be relatively complex to make reproducibility easy. It can sometimes take an entire day to check a colleague's analyses. However, as reproducibility increases co-piloting becomes less strenuous. Additionally, knowing the particularities of checking other people's work helps improve your own reproducibility. This is why I go through hoops to make sure one file is sufficient to get all the results in the manuscript and that dependencies or datafiles do not cause any trouble.

Another effortful aspect of a reproducible workflow is that the project lead often has to enforce reproducibility. I want my research to be reproducible, so I enforce this in my project. Co-authors need not have the same perspective on this and therefore do not feel

responsible for this. As such, you have to ensure that what they do is reproducible as well. If the project has a centralized project lead, this is not a huge problem. However, with more decentralized projects it can cause some difficulty. It requires you to structure the project thoroughly, but requires increasing effort with increasing project complexity (note that increasingly complex projects also have a higher need for reproducibility because of a higher potential for error-making).

Key benefits

My workflow has actively developed in recent years and this has culminated in analysis scripts that can run everything from the script itself. This requires nothing from the person trying to reproduce the results, except to download the script. It can be quite daunting when a researcher shares ten files and you have to find a way through them. It is not sufficient to be transparent. In order to become reproducible, it is highly important to structure your documents such that others, including your future-self, can understand what is going on.

Version control is a benefit within this reproducible workflow, considering that it goes beyond reproducibility of research results but also ensures reproducibility of the research process. My direct colleagues are starting to realize this as well; it is affirming to hear them stress that it helps them increase efficiency by allowing to retrace their steps. I hope that other colleagues will see the value in that sooner rather than later, (e.g., when their data gets audited).

Key tools

I use a set of tools which all have one thing in common: they are based on open formats that are timeless, inclusive, and can be used by anyone who has a computer. These open formats include the data in clear-text files, but also includes software packages that are open-source, whose code is checked by the open-source and academic community (e.g., `R`, `git`). It seems to me that the use of closed software has proliferated throughout the social sciences (where I operate most of the time) without the realization that it is actually hurting the future of science (e.g., irreproducibility of results), but also hurts current-day science. Not everybody can afford a license to SPSS or Microsoft Office, for example. Why exclude those who do not have those funds? Science is an enterprise that should be all-inclusive and not select on financial wealth of individuals or institutions. I try to reaffirm this principle by ensuring that all the tools I use are open-source and can be used by anyone who wants to.

Questions

What does "reproducibility" mean to you?

For me, reproducibility pertains to the reliability of research findings, which both includes direct reproducibility (i.e., can someone else reproduce the results by applying the described method to the same data?) and retest reliability (i.e., if we rerun the study, do we get similar results?). My case study focuses on direct reproducibility, that is, that anyone or a future-me can retrace the steps from the project in such a way that it is understandable and that the results are reliable.

Why do you think that reproducibility in your domain is important?

Scientists are humans and humans make mistakes. By using reproducible practices, we can discover these mistakes and not be led down a research path that is based on a mistake. It is important in my domain, because we preach that science has to be conducted in a reproducible manner.

How or where did you learn about reproducibility?

I got interested in reproducible practices during my master education when my supervisor introduced me to the idea of Open Science. I found myself wondering how to implement it in different stages of the research process, not knowing where to start documenting *during* the research. I learned much from colleagues across the world and across fields with who I discussed ways to be more reproducible (mostly on Twitter, which is a highly valuable resource for researchers).

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

Reproducible research is tiresome when you figure out a new way of doing things and then think that your previous work is incomplete. Additionally, not all colleagues are as enthusiastic about reproducibility and this can lead to discussions (also a good thing) that postpone implementing certain practices. It is very important to get everyone on the same page in the initial meeting on how the project will be managed, such that nobody is met with surprises and potential ambivalence at the end.

What do you view as the major incentives for doing reproducible research?

The main incentive for reproducible research is (future) efficiency. When you know that you can revisit projects from years ago and need at most 30 minutes to find what you are looking for is a major improvement over spending a day looking for that one specific value someone asked about in your email. It also helps revisit previous projects and see what I did, because I frequently unlearn things I require in a new project (e.g., I often forget how to make plots in the `ggplot2` package because I use it too infrequently, and I just reuse code from previous projects).

Are there any best practices that you'd recommend for researchers in your field?

The best practices I recommend any researcher to apply are the following:

1. License your work with an open license (CC-BY or CC-0), explicating free reuse of your materials and manuscript.
2. Script your data handling and analyses as much as possible, such that each step is reproducible.
3. Have a colleague check your analysis code, it is too easy to make mistakes. Not checking analysis code is comparable to not having co-authors proofread the manuscript.
4. Try and create an analysis script that can run automatically, downloading all required files and installing its dependencies. Otherwise, other people are likely to fail in reproducing your results, when they cannot get to the dependencies.

Would you recommend any specific resources for learning more about reproducibility?

I recommend the article by Karthik Ram on using version control in research. It opened my eyes on the use of version control as a project management tool that improves reproducibility at the lowest cost possible. Low threshold version control is available at the OSF, which provides online training tools (see osf.io).

Ram, K. (2013). Git can facilitate greater reproducibility and increased transparency in science. *Source Code for Biology and Medicine*, 8(1), 7.

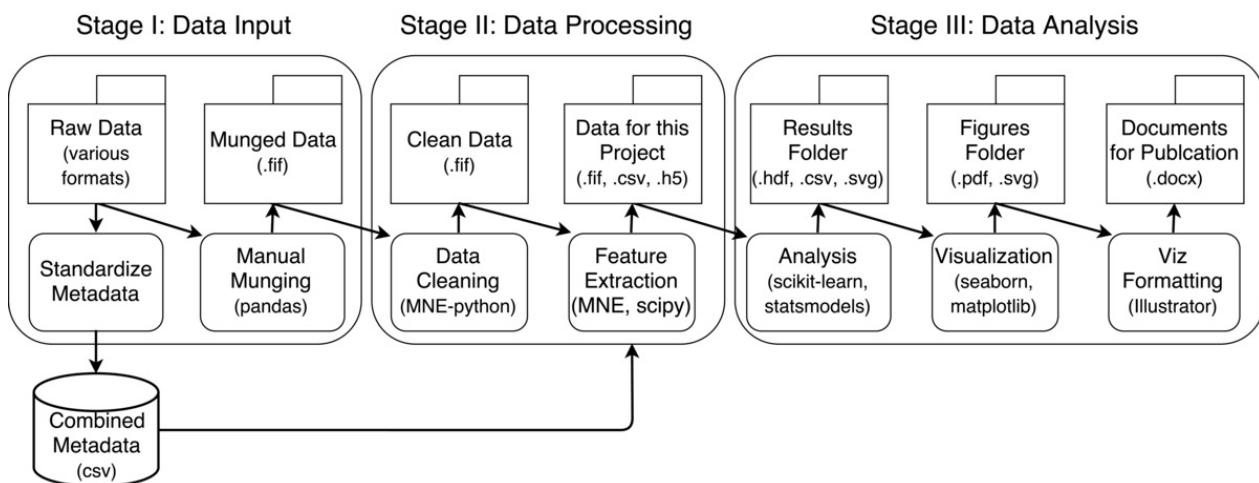
Feature Extraction and Data Wrangling for Predictive Models of the Brain in Python

Chris Holdgraf

My name is Chris Holdgraf, I am a senior graduate student with the Helen Wills Neuroscience Institute at UC Berkeley. My thesis work involves using predictive models to understand how auditory regions of the brain respond to acoustic features.

I am interested in how experience, learning, and assumptions about the world shape the way that we interact with low level features of sound. This involves a lot of computational work, signal processing, and data cleaning, utilizing a number of packages in the Python scientific ecosystem.

Workflow



My workflow involves taking raw data from a variety of sources, doing a few steps of munging on each one separately, combining them into a common structured format, and then doing further processing on this data. Here's a general breakdown:

Wrangling raw data

The raw data for my work involves electrophysiology signals collected from the brains of surgical patients in several sites across the country. This is challenging because the nature of the data is quite different from one location to another. I often get the neural data in many different formats, and a loose collection of metadata (e.g., sensor locations, names,

sampling rates) that can be anything from a PDF of hand-written notes to a structured text file. As such, the first step in my workflow is to wrangle all of this data into some kind of structured format.

First, I put all data into subject-specific folders. Each of these folders has sub-folders for different kinds of data (e.g., "raw", "munged", "meta"). The sub-folders will eventually be populated with data during processing, and the structure is consistent across all subjects so that I can easily parse them with scripts.

Next, I have a Jupyter notebook that is unique for each person, and is designed to take whatever raw format the data is in and turn it into a standardized version. This is called `munge_{subject_name}.ipynb`, and will output a file that I can use for the rest of my analyses. Jupyter notebooks are useful here because each subject is different, and will require a different set of steps to get the data ready. For this reason, I like to have lots of plots that go along with the analysis process, and a record of exactly what code was run to create the munged data for that subject.

Because the data often comes in different formats, I make the output of this step the same format for everyone. I use a Python package for neuroscience electrophysiology called `MNE-python`. This provides a common way of structuring data in order to streamline I/O, processing, and data analysis. I convert all of my raw data into the `.fif` format, which is a standard format for storing electrophysiology data. This means that I can read the data into other platforms (e.g., R or Matlab) fairly easily. The output files are stored in the folder `subject_name/munged`.

The final thing I do in this step is look at the metadata files for this subject, double check that all the values inside are correct (this is done with the munging notebook), and then insert them into a "combined" csv file of metadata. I have a Python script called `create_combined_meta.py` that will look through all the subject folders, find the metadata files that my munging notebook outputs, and turn them all into a single CSV file.

This aggregated CSV has data for all subjects that I have, and makes it much easier to quickly look at information across datasets. To do so, I use `Pandas`. This is a package that lets you represent tabular-style data in memory, and also gives you "database-style" functionality with their objects (e.g., joining two `Pandas` objects with partially-overlapping fields). Doing this necessitates that all of my data is in "tidy" format. This simplifies things, because it means that while I have a separate file for each dataset, I have a single combined file across all subjects for their metadata.

I should note that this is the point where somebody usually suggests using a "proper" database like SQL instead of keeping my data in CSV / FIFF formats. I've found that the overhead added by reformatting my data for something like SQL isn't worth the benefit it

would give. If I were to start a new project - particularly with larger datasets - then I would likely consider a more robust data storage solution like SQL.

Cleaning the data

Once I've created my munged data, I can now use a single script for processing/analyzing all datasets. In the field of cognitive neuroscience, there are common preprocessing techniques that are carried out in order to improve the quality of the data (e.g., a few filtering steps and rejecting channels that are too noisy). I have a `clean_data.py` file that will look through the "munged" folder of each subject, load the data, run the cleaning, and then output the results to the folder `{subject_name}/clean`. This way, I know that any data in the clean folder is ready for further analysis.

At this point, I have cleaned data in each subject's folder, I also have that subject's metadata inserted into a CSV file with all subject information. From now on, I can just load a subject's data file, then load the metadata CSV file for everybody, and query only the rows that belong to the subject that I care about.

Defining a project

When I begin analyzing my data to answer a specific question, I create a new project-specific folder that exists alongside my "data" folder. Each project generally entails a number of analyses, and this is a way to keep them all in the same place. The project folder is structured similarly to the "data" folder. It has a sub-folder for "scripts", for "data", for "results", and for "documents" and any information necessary for publications that come out of this project.

For example, the first thing I might do is create some Python script to extract features of interest. I will put it in `project_name/script/feature_name/extract_feature.py`. The script assumes that there is data for each subject in the "clean" folder. It will pull the data from "clean," extract whatever feature I'm interested in, and then save the result to the project-specific folder, something like `project_name/data/my_feature/{subject_name}_feature.fif`. I parse all subject folders and save files in the same manner.

Storing the extracted features for all subjects in a single project-specific data folder makes it much easier to develop scripts/notebooks to further analyze the results, since I don't need to keep track of which features have been extracted for which subjects. I also develop feature extraction scripts using the Sun-Grid engine (a platform for dividing computation between a cluster of computers) for speeding up my analyses. I can do this relatively easily because the folder structure for each subject is the same.

Running analyses

Now that I have a set of features created for each subject, it is time to run analyses and answer questions. These scripts also exist in the project-specific folder, and assume that there is data in the "project_name/data/my_feature/" folder.

A difficulty that I've had is knowing when to keep your analyses in interactive notebooks vs. Python scripts. I generally pilot my analysis interactively - this lets me do sanity checks and on-the-fly calculations that help me develop the final analysis. Once I have code that does a specific analysis, I will put it in a `.py` script.

The output of this script then goes into a `project_name/results/my_analysis` folder. They may be in the form of PDFs and SVGs for further inspection, or data files (e.g., CSV) representing model results (such as regression coefficients). For anything consisting of lists, numpy arrays, or simple dataframes, I use the `h5io` package, which provides a fast way to read/write collections of data to `hdf5` files (another standardized data storage format).

Finally, I use the outputs of the analysis script to create visualizations and decide whether or not my analysis actually worked. I use another set of notebooks to read in the results, perform last-second wrangling, statistics, etc, and output visualizations. If I have a publication-ready figure in mind, I will create a notebook specifically for that figure in another folder called `project_name/fig_{analysis_name}`.

When creating actual figures for papers, I like to use software like Adobe Illustrator to make sure that my fonts are the proper size, well-spaced, etc. I use visualization notebooks to create high-res PNG versions of my data that have most formatting stripped away (except for the data). These plots are then linked to an Illustrator file, so that they are updated automatically when a new plot is created. This way I can easily arrange my plots and standardize fonts without doing a lot of manual tweaking.

Finally, I use Microsoft Word to write drafts which I put in the "doc" folder. These pull from the figures I've created in the "fig" folder. Ideally I would use text files here with LaTeX, but the team that I work with makes this prohibitive.

A general note

This process has been refined many times over the past year, and the original structure looked very different than this. My original file system had code and data living in totally different places. Moreover, it had project-specific scripts and more general utility scripts living in the same place. The goal of this file structure is to keep data and scripts together when they have a natural pairing, and to separate out my more production-ready functions/modules from "hackier" project-specific scripts. Below is a list of some things that I've learned along the way, and that have guided the development of this system:

1. For any data/code that are project-specific, keep these together in the same general file hierarchy.
2. Rather than creating metadata structures that live next to the data, come up with a "master" metadata template, then store entries from every subject in a CSV file that follows this template. Rather than storing data in separate subject files, include an entry with "subject id" in the metadata file so that you can pull out individual subjects in this way.
3. Utilize other packages whenever possible, particularly with the annoying task of data I/O. In my case, I store all my raw data as '.fif' files, which can be opened easily in Python or Matlab with well-supported third party packages. I also use `pandas` and `h5io` to read / write metadata files, which makes it very easy to slice and dice these files for particular entries that I want.
4. More generally, take a "database" approach to how you store any data. Even though I'm not storing data in a MySQL database per-se, I can still draw inspiration for how this data is organized. Treat data entries as rows, and data features as columns, and then combine / split up the data using pandas database-style syntax (e.g., joins and merges). A great guideline for this is the "tidy" data specification described by Hadley Wickham.
5. Put ad-hoc code in a project-specific folder. Be much pickier about code that you expose in public Python modules for any project. If you think a function is worth generalizing, then move it out of the project folder and to its proper module, and document it extensively.
6. Do all coding with automatic PEP8 and PEP257 checkers. PEP8 is a set of standards for naming conventions, code syntax, using white space, etc to ensure that code is clean and readable. PEP257 is a set of similar guidelines for docstrings. Many "fully-featured" text editors have plugins that automatically check code using these guidelines and highlight errors, which is useful for quickly writing clean code.
7. Make a conscious effort to structure Python scripts differently from Jupyter notebooks. Structure code (and data) such that it lends itself well to scripting, rather than assuming interactive sessions for everything.
8. Use Jupyter notebooks to glance at the data and preliminary results, but move code into scripts as it becomes more refined or complex. This avoids creating a mega notebook with tons of different analyses in it.
9. Structure code so that some scripts live with the data that they operate on. E.g., if you've got a script that only operates on a specific collection of data, renames specific columns in that data, and always saves it to another location relative to the original data,

then create another folder “script” right next to that data folder. Put all data-specific scripts into this folder. This way, you know that scripts operate with relevant data nearby.

10. Finally, this is not specific to this project but has been very useful to me. Find opportunities to contribute to other open-source projects. Open pull requests and learn about how to use the code. The back-and-forths and input you get will make you a much better coder, and your codebase / research will greatly benefit from it.

Pain points

The hardest part of my workflow has been deciding how to balance flexibility and control. On the one hand, you don't want your scripts to be so specific to data that they break as soon as anything changes, but on the other hand creating code that generalizes well and isn't terribly confusing is really difficult to do. In my case, I initially made errors on both of these fronts, but the current structure of my data seems to be more intuitive, easy to maintain, and easy to grow.

Another big issue I've had lies in dealing with different formats of data and information. For example, I want to version control all of the code that I write, but:

- Does that mean that I should create one big repository for all of the code described above? What about a separate repository for each project?
- How should I split up the general modules and functions vs. the code that only lives with a particular project?
- Finally, how should I deal with the fact that there are lots of other "non-code" files living with this file structure (e.g., images)? Should they be version-controlled, or should the code just assume that the data lives in particular folders? What would happen if somebody else copied the code and didn't get the data?

I don't necessarily have good answers for these issues, and I'm still coming up with a solution that makes me happy, but these are some things that I'm thinking about.

Key benefits

The biggest benefit of this system is the fact that messy, subject-specific data is quickly turned into a standardized format that is consistent across all of my subjects. This is useful because it means you can write scripts that analyze the entire dataset without doing a lot of extra customization. Moreover, because the structure of the filesystem is the same for everybody (including things like naming conventions), it is easy to find what you want from each dataset.

Another benefit is the fact that I am storing code along with the data that it operates on. Some people feel strongly that this is a bad idea, but I've found it to be useful so long as the within-project folder structure still makes sense. In previous workflows, I had all of my code in one folder, and all of my data in another folder. This often led to confusions where I was unsure which code operated on what data. It also made it more difficult to connect the steps of preprocessing and feature extraction chains. Now, if I want to know all of the things that have been done to a collection of data files, I just need to look into its corresponding "script" folder.

Finally, by separating out operations that are true for all projects (e.g., data munging and cleaning) and those that are project-specific, the scope of individual projects becomes more clear and easy to follow. I think of the data pipeline as a single tree trunk, where projects branch out from this trunk and do extra things to the data, on top of the base workflow of preprocessing. Now, my file structure more naturally follows this concept.

Key tools

The two most useful tools that I have found are `Pandas` and `MNE-python`. `Pandas` made it much easier to embed metadata with the signals that I analyze. It allowed me to store information from lots of subjects in a single CSV file, and treat it as a "database" by using queries on it. `MNE-python` is a package for electrophysiology in neuroscience written in Python. When I discovered it, I found that it duplicated many of the functions I had already written, and in general did this much better than I had. Moreover, it has a lot of convenience functions for doing I/O, which up until then was a pain to maintain. By using these two packages, I was able to significantly cut down on the amount of custom-written functions that I used to wrangle my data.

Questions

What does "reproducibility" mean to you?

The discussion in this writeup covers the first 6 months of the project. To that extent, my definition of "reproducibility" means actually being able to reproduce my own results (aka, coding for my future self). I ran into a lot of issues to keep things streamlined and understandable in my own head, which made it difficult to interpret my findings. Obviously this would generalize to other scientists trying to reproduce my analyses and work as well.

Why do you think that reproducibility in your domain is important?

Because it's a guiding principle that will make my code more understandable, maintainable, and extendable for others and for myself.

How or where did you learn about reproducibility?

At first it came from teaching a few Software Carpentry classes and reading things online. Lately, I have gotten a lot of help by contributing to the `MNE-python` project, as I've found that going through the pull request process for a well-maintained project is a great way to learn a lot about coding well.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

In my case, legality is a big problem because I'm dealing with medical data that cannot be shared publicly. Another big problem is simply a matter of training and incentive. Right now there is little opportunity to learn how to code well or how to make reproducible science. To make matters worse, I see very little incentive for anyone to actually do so (if they want to be a tenured faculty).

What do you view as the major incentives for doing reproducible research?

Other than the warm fuzzy feeling, I think the biggest advantage is that when you code and organize for other people, you also code and organize for yourself in the future. This makes your life much easier in the long run.

Are there any best practices that you'd recommend for researchers in your field?

Front-load a lot of thinking/planning before you just start creating scripts and functions. Spend a good chunk of time thinking "big picture" early on, then zoom in and build some stuff, then zoom back out and decide if it was a good idea or not. Don't get lost in the weeds.

Would you recommend any specific resources for learning more about reproducibility?

Software Carpentry is a great one, but most other stuff is just scattered on stack overflow unfortunately. I think that finding a good package that has a sweet-spot of contributors (aka, not so few that you don't get feedback, not so many that it's a huge pain to do anything). Try

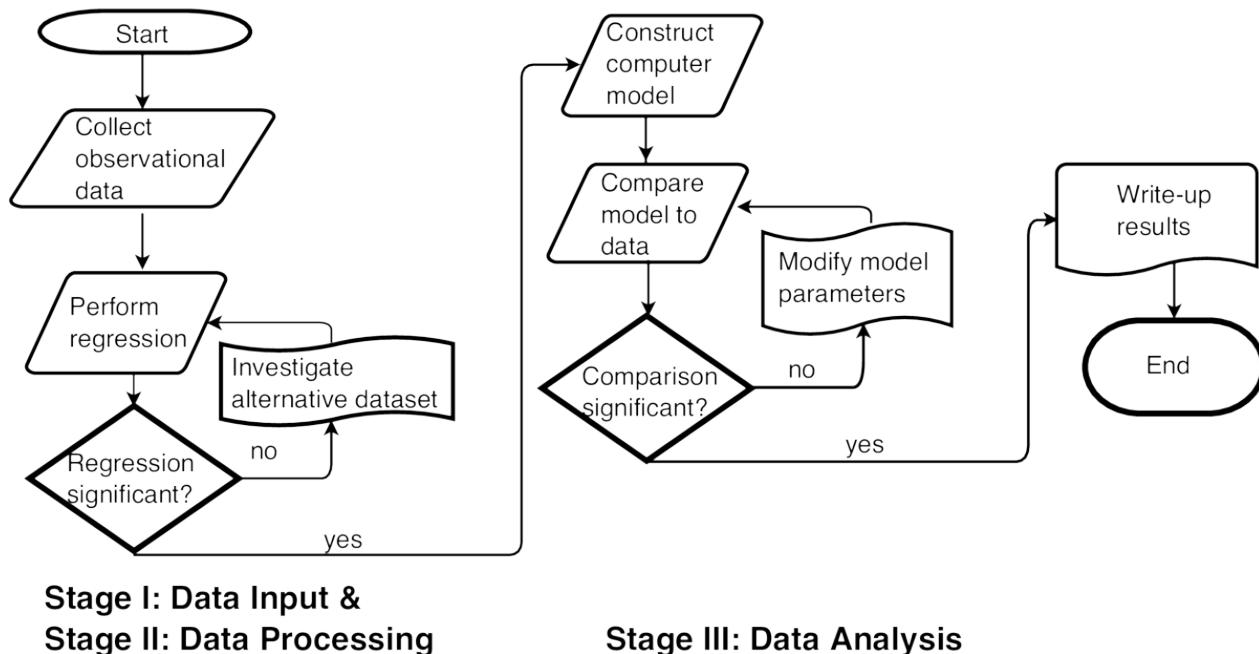
to contribute something via a pull request and learn from the other people in the community. It will be a great way to learn good coding principles. Finally, find a community around you (at the university, at local companies/hackathons, etc) that shares your interests. Spend time learning and teaching with these people.

Using Observational Data and Numerical Modeling to make Scientific Discoveries in Climate Science

David Holland and Denise Holland

My name is David Holland. I am a Professor of Mathematics and Atmosphere Ocean Science at New York University's Courant Institute. I study global sea level rise in a changing climate, specifically, how changes in global weather patterns affect the melting of the great ice sheets, with Denise Holland, who is the Field and Logistics Manager for our field research program.

Workflow



The goal of our workflow design is to use observations and computer models to make new discoveries about the natural environment particularly in the context of climate change. The approach we take in our algorithm design is to generally always look for an interesting phenomena in the observational record first. If we find it there, then we generally proceed to try to simulate it in a computer model as an independent check on the physical plausibility of

the phenomena in question. In this very specific workflow, we will illustrate our approach for the particular question: does the North Atlantic Ocean drive climate change in Western Antarctica?

The very first step in our workflow is to collect all available observational data of North Atlantic Ocean surface temperatures going back in time as far as is possible. The more spatial and temporal data we have, the more robust the results will be. On occasion, different datasets can be contradictory or inconsistent. In this case, we have to make subjective decisions if one dataset is better than the other. If we cannot make this decision, we cannot proceed and the algorithm has to terminate. On the other hand, if two different datasets agree, then we proceed with greater confidence of the quality of our input data. In our study, our analysis of the North Atlantic temperature data strongly showed us that there exists a 60 year period oscillation in the surface temperature in a broad pattern that covers the entire North Atlantic. This is known as the Atlantic Multi-decadal Oscillation (AMO). This result has been previously found by other researchers so our very first step has not only given us confidence in what we are doing, but also reproduces a result from other researchers. All of our science research tends to in fact work this way in the sense that our new findings tend to be built on reproducing previous work by others and then extending that into new, unchartered research areas.

The next dataset we investigate is the surface winds over the Western Antarctic region. In exact analogy with the processing of the North Atlantic surface temperatures above, we proceed here to look for long term trends in winds. We find such a trend and it matches with the North Atlantic Ocean surface temperatures, suggesting one is causing the other. We claim such a relation based on a formal regression calculation which shows our finding is statistically significant but still does not explain which is the cause and which is the effect.

The next step in the algorithm is to employ a physically based numerical model of the global climate system. Using such a model, we can impose the observed North Atlantic Ocean surface temperatures in the model, and the model can simulate the response of the global atmosphere to this imposed ocean forcing. We carry out the simulation and we find that North Atlantic Ocean temperature oscillations drive wind circulation anomalies in Western Antarctica. This is a very surprising, non-intuitive result. We also try to model in the opposite sense, and impose surface wind anomalies in Western Antarctica to see if they drive ocean temperature anomalies in the North Atlantic. The simulation showed that this does not happen. This gives us some confidence to conclude that the direction of flow of climate change is from the North Atlantic to West Antarctica.

At this final stage, having made a new discovery in two independent manners, one purely observational and one purely computer modeling, we are ready to report our findings to the scientific community. This involves a rigorous peer-review process that imposes a number of reproducibility requirements. A section of the manuscript must be devoted to explaining

where all datasets are located and how a reviewer or future reader could access the same datasets we use. Likewise, we are required to describe the computer model we used and how a future researcher can access the same model. While the main scientific article is relatively brief (about 4 printed pages), giving the reader the essential information on what we found and how we found it, we also write a supplementary materials section. This is an exhaustive description of each step we took with our observations and our modeling.

The original data sources that we used for ocean temperatures and atmospheric winds are on-line and available through national climate data repositories. The numerical modeling code is available on-line through national climate modeling centers. The regression calculations and the numerical simulations we preformed are very large and are not stored on-line but are archived at NYHU on hard disks off-line. The regression code is standard and available at many repositories such as part of Matlab. It is well documented, well tested, with many examples. The numerical climate code is used by a large number of people, it is well documented, well tested, with many examples. There are no restrictions on other researchers replicating or confirming our work. We warmly welcome such activity.

As mentioned in our published paper, there is a supplementary document that includes details about the data processing workflow. There is not the actual computer scripts used to perform the regressions nor those to run the numerical climate model. These could in fact be put on-line if there was a repository for such information. However, in our estimation, if someone was to try to reproduce our research it would probably be more natural for them to write their own scripts as this has the additional benefit that they might not fall into any error we may have accidentally introduced in our scripts.

Our published work is citeable as: Li, Xichen, David M. Holland, Edwin P. Gerber, and Changhyun Yoo. "Impacts of the north and tropical Atlantic Ocean on the Antarctic Peninsula and sea ice." *Nature* 505, no. 7484 (2014): 538-542.

Pain points

The most difficult part of reproducing these results is the sheer volume of the datasets involved and the amount of computational storage and time required to complete all these calculations. Transferring large volumes of data from super computers (where the main code is run) to personal computers (where the analysis is generally performed) is an onerous and time consuming task with many failure points. Often data transfer is incomplete, storage disks break or fail, and weeks or months of research time is lost. In such case, one has to simply go back to the start and begin again.

Key benefits

Our approach of independently using observational data and modeling is a stronger approach than that of just using one or the other. Our approach also is to abandon findings that are contradictory between independent observational datasets as this suggests that the data is of inadequate quality to proceed further with any analysis or conclusion. In other words, if you find something interesting in analysis of one dataset, but not in another similar one, despite the temptation to proceed with the interesting finding, one must acknowledge that the contradiction prevents moving forward.

Key tools

We have nothing special to report here but are aware of efforts in the computer science community to better track the workflow stages of a research project. In our project, such software would have been beneficial in that our workflow algorithm could be online and a user could click through it and find the scripts and datasets and models we used.

Questions

What does "reproducibility" mean to you?

"Reproducibility" for me means that someone, anyone, could read my published research, then take the datasets I have archived on the web and use them to reproduce the results I had in my published paper.

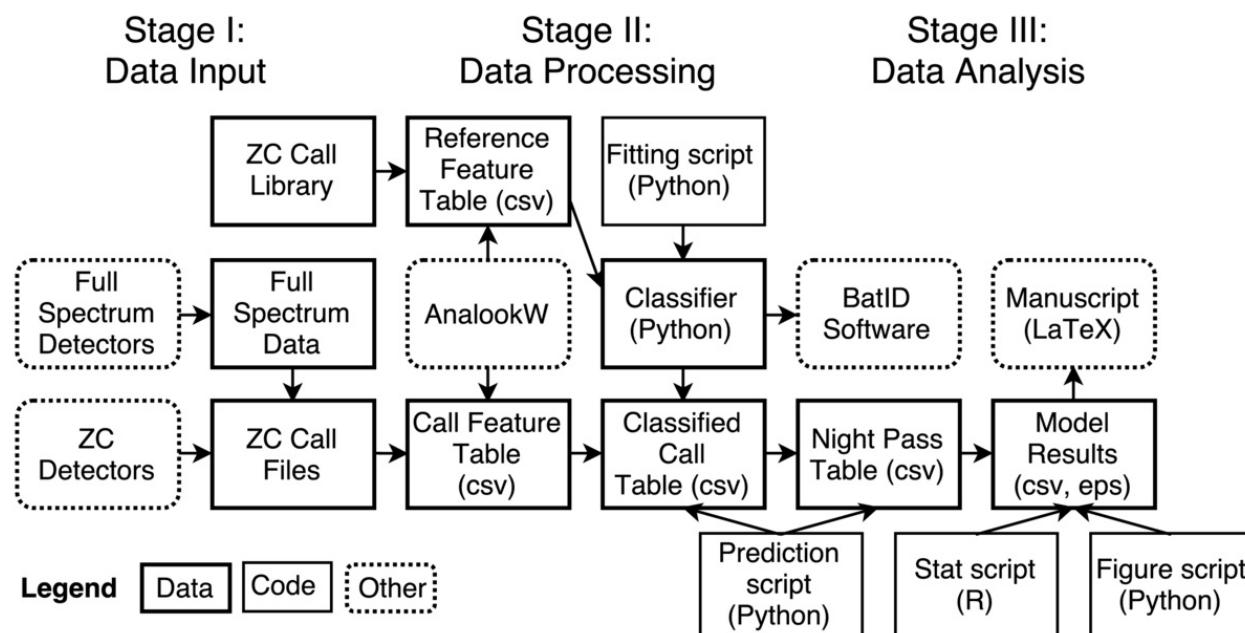
Reproducibility is at the heart of natural science. Without being able to perform an experiment and achieve a certain result, and then to have an independent scientist reproduce the same result, the research finding is murky. Our publications in high-profile journals, such as *Science* and *Nature*, demand that we include a methods section in our papers, as well as carefully document the location of all our datasets on the internet. Some of the research code developed requires years to master. Scientific funding and the number of scientists available to do the work is finite. Therefore not every scientific result can, or should be reproduced. The most important, paradigm shifting results should, however, be reproduced. In the case of climate science, important decisions by world leaders rely on scientific findings. These findings must be robust and reproducible in order to guide energy use policy. In our study, we reached the same conclusion using both purely observational data and then independently through a computer simulation of the same phenomenon. Finding the same result in two completely independent manners gives us confidence in our findings.

Analyzing Bat Distributions in a Human-Dominated Landscape with Autonomous Acoustic Detectors and Machine Learning Models

Justin Kitzes

My name is Justin Kitzes, and I am a quantitative ecologist who studies the effects of land use and climate change on biodiversity distributions. I am currently an Assistant Professor of Biology at the University of Pittsburgh, and I was formerly a Data Science Fellow in the Institute for Data Science at the University of California, Berkeley. The research that I describe below focuses on the spatial ecology of bats in a complex, human-altered landscape. This case study describes the use of acoustic detectors, machine learning methods, and likelihood statistics to examine the effects of three large Northern California highways on bat activity.

Workflow



This study investigated whether several common species of bats showed decreased activity adjacent to three large highways near San Francisco Bay. Activity in this study was defined as the number of ultrasonic foraging calls recorded by autonomous acoustic detectors. The core tasks involved collecting raw bat call data using the detectors, extracting specific

features of the recorded calls, classifying the calls to the species level, and performing statistical analysis on the resulting nightly call counts as a function of predictor variables, including distance from the road. The complete analysis is described in a [manuscript](#) published in *PLoS ONE* in 2014. We later used a similar workflow to conduct a [second study](#), published in *Agriculture, Ecosystems & Environment*, of the predictors of bat activity in vineyard landscapes.

Two different types of acoustic detectors were used, one of which recorded data in zero-crossing format and the other in full spectrum format. The full spectrum data were converted to zero-crossing format using a closed-source utility provided by the detector manufacturer, with conversion parameters selected to produce output similar to the recordings from the native zero-crossing detector. The free, closed-source software AnalookW, developed by an individual researcher who has been active for many years in bat call analysis, was used to filter out files containing only noise and, for the remaining files containing bat calls, extract twelve features describing each call. These features were saved in a csv table.

The calls were then classified by species, which was done using a random forest classifier. A reference library containing zero-crossing calls made by individual bats identified in hand was obtained from a personal contact, and the same twelve features were extracted for these calls using AnalookW. A random forest classifier was trained on this data using the Python package scikit-learn v0.12. Classifier accuracy was evaluated using cross validation and a confusion matrix.

The classifier was then used to identify the recorded calls to the species level, creating a classified call table. This table was summarized into a nightly pass table, which aggregated the calls into passes consisting of multiple, closely spaced calls and summarized the number of passes of each species recorded in each night, at each distance from the road. Environmental and site variables were joined to this pass data to create the final table for statistical analysis.

As functions for fitting generalized linear mixed models (GLMMs) were not available in Python, statistical analysis was carried out in R. Exploratory analysis showed that a Poisson regression was not appropriate for the data, so a negative binomial GLMM was fit to the nightly counts of passes from all species and separately for four common species. The model results were saved as a table that later appeared in the final manuscript. The model result table was then read by a Python script, which created and saved a figure that appeared in the final manuscript. The final manuscript was written in LaTeX and submitted to journals in that format.

In addition to the manuscript, a second output of this project was the open source software [BatID](#), which bundled the classifier object with a browser-based interface to enable non-programmers to automatically classify California bat calls. This software is freely available

for download and has been used by researchers in academia, government, and the private sector.

Pain points

At the beginning of the workflow, two closed-source graphical programs had to be used, one to convert a proprietary data format to the zero-crossing format and the second (AnalookW) to perform feature extraction on the zero-crossing call files. Both of these steps required parameters to be entered into these programs, which I was careful to document manually, as this information can otherwise easily be lost. AnalookW runs only on Windows, which required me (and any analysts wishing to use my later software BatID) to locate a Windows computer to complete this step. Although I write code faster and more accurately in Python, I needed to switch to R for statistical analysis, as the necessary packages were not (and still are not) available for Python. A major headache at the manuscript stage arose because the R statistical functions reported output only as a non-machine-readable text file or as an object, which required me to create the final table, containing coefficients and standard errors for 14 variables across 5 models, by hand.

Once I created and released the BatID software, a problem immediately arose when the scikit-learn package was updated to v0.13, which could not read the classifier object created during my analysis. Additionally, the original BatID package required a user to install a full scientific Python stack, a task that proved difficult for precisely the audience of non-programmers that I was hoping to reach. I eventually used pyinstaller to create a standalone binary executable for Windows, reasoning that users of the software needed a Windows computer anyway in order to run AnalookW as a prior step in the analysis. Creating this distributable binary was not straightforward and took many days of trial-and-error testing and manual tuning.

Key benefits

Of all aspects of the analysis, I am particularly happy about the effort that I put in to creating the BatID standalone classifier software. As the many of my colleagues in ecology are non-programmers or novice programmers, I believe that these types of user-friendly tools are critical to advancing the state of science in my field, as well as to supporting the uptake of new methods by non-profit and agency scientists. I hope that more of my computationally-oriented colleagues will engage in similar activities in the future.

Ironically, of course, in creating a tool for non-programmers, I also created another graphical program that cannot easily be scripted into a workflow such as the one. I attempted to ameliorate this concern in the most recent BatID version by requiring users to create a text file containing all parameters, which is read by the program along with the data file, and

having the program save all results in the same directory as the parameter file, along with a log file. This at least ensures that there is, by default, some record of the program version, time, and parameters used to process the raw data into classified results tables.

Questions

What does "reproducibility" mean to you?

I consider a study to be (computationally) reproducible when I can send a colleague a zip file containing my raw data and code and he or she can push a single button to create all of the results, tables, and figures in my analysis. It can, of course, be quite challenging to achieve this goal with anything short of the simplest scientific workflows.

Why do you think that reproducibility in your domain is important?

I think that reproducibility is particularly important in fields like ecology in which researchers are striving to make increasingly detailed inference and predictions using relatively scarce data. Although I do not have specific evidence to this effect, it seems logical to me that in these "high leverage" types of analyses, small analytical decisions (how data are cleaned, the options passed to optimizers, etc.) could play a disproportionate role in influencing the eventual study conclusions, and thus need to be fully documented and shared. An easy way to guarantee that all of these decisions are recorded is to make one's entire analysis reproducible by others. More broadly, I feel strongly that reproducibility is a basic component of good science. Now that "doing science" requires communicating more detail than can be easily expressed in narrative form in a manuscript, releasing code and data seems completely necessary, where feasible, across all domains.

How or where did you learn about reproducibility?

I started learning these tools through workshops, in particular by taking Python workshops at UC Berkeley and by teaching Software Carpentry workshops with other experienced instructors. I also learned a great deal by working closely for several months with a former student in my Ph.D. lab who had previously worked at Microsoft. I picked up more advanced techniques and ideas mostly through web searches while attempting to get unstuck from issues that arose in my own research and software development.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

A major challenge in ecology continues to be data sharing and access. Many field ecologists, understandably, are reluctant to share their hard-won raw data with other researchers. I suspect that this caution arises both from a sense of professional necessity (i.e., I invested a ton of time collecting this data and I am going to be the one to publish all of the analyses using it) and from the feeling that numbers alone cannot possibly capture all the subtle nuances that were observed in the field and that are important to truly understanding the data, its potential, and its limitations. In particular, information about sampling bias (as derived from choices of study site, sampling techniques, season, missing data, and many other factors) cannot always easily be described in numeric form. I also suspect that many field ecologists recognize that this information often isn't even in any published manuscript, leaving the person who collected the data as arguably the only one truly qualified to analyze it. What data is published and available tends to be relatively small and of heterogeneous format, and thus is often locked up in printed pdf tables and other non-machine-readable formats.

What do you view as the major incentives for doing reproducible research?

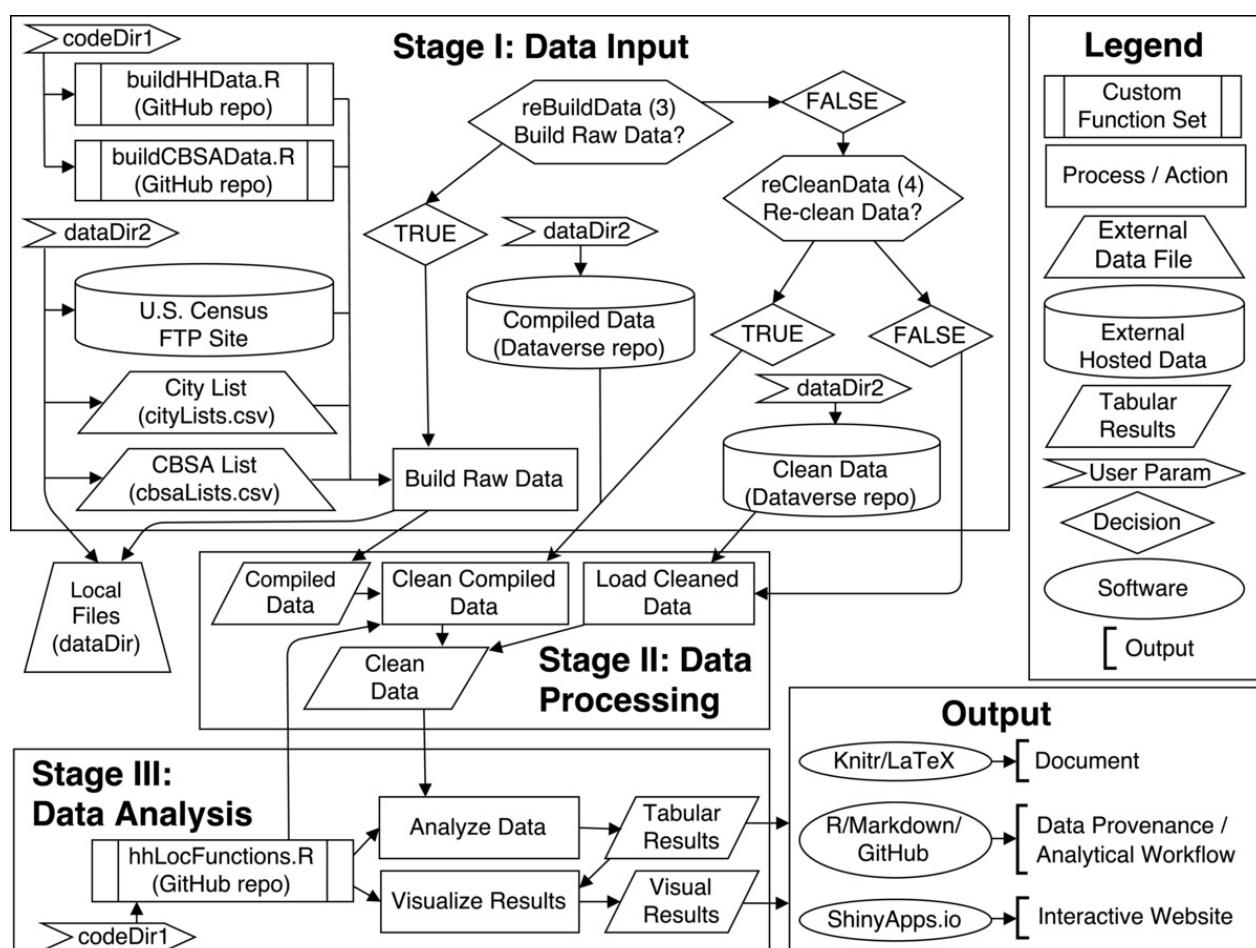
I'm not sure that there are any major external incentives in my field -- certainly, in principle, releasing reproducible research could increase the number of researchers who end up citing your manuscript, but this seems somewhat indirect. Some journals, like PLoS, are now mandating that all novel computer code be uploaded as manuscript supporting information, but it seems that this requirement is not thoroughly checked at this point.

An Analysis of Household Location Choice in Major U.S. Metropolitan Areas Using R

Andy Krause and Hossein Estiri

I am Andy Krause, a Lecturer in Property (Real Estate) at the University of Melbourne. My research focuses on the spatial analysis of real estate markets, particularly in regards to valuation and location. This work was completed with my colleague, Hossein Estiri, Research Fellow at Harvard Medical School. Hossein uses data science approaches to study urban energy and health.

Workflow



This research analyzes the household location choices of American households in the largest 50 metropolitan areas in the United States. Households are broken down by five-year age cohorts (based on the age of the head of the householder) and mapped against the household's distance (census block group level) from the central business district (CBD) of

the metropolitan area in which they reside. In polycentric regions such as Seattle (Tacoma, Bellevue and Everett as alternative CBDs, analyses are conducted on distance to core center as well as secondary centers. An initial paper reporting the results is currently under review.

All data, code and analytical workflow are hosted on-line. Code and analytical workflow, including analytical script and custom function sets, are written in R and found on the project's [GitHub Repository](#). The complete set of raw data is available through the U.S. Census. Users wishing to skip the data compiling and/or cleaning steps can download the compiled or cleaned data from the project's [Dataverse Repository](#).

The *hhLocAnalysis.R* file is the main analysis script and the only file that needs to be executed. Two key path parameters and two key process parameters must be manually set at the beginning of the *hhLocAnalysis.R* script:

1. **codeDir**: Location of the cloned GitHub code repository
2. **dataDir**: Location of the compiled (and/or cleaned) data downloaded from Dataverse
3. **reBuildData**: Do you want to go through the entire data compilation process?
4. **reCleanData**: Do you want to re-clean data?

Additional parameters containing the file names of the downloaded intermediate data and the path to export the results may need to be set also be set prior to executing the script:

1. **rawDataFile**: (Optional). If **reBuildData** is equal to FALSE and **reCleanData** is equal to TRUE then you will need to provide the name of the compiled data file (within the **dataDir**) downloaded from Dataverse.
2. **cleanDataFile**: (Optional). If both **reBuildData** and **reCleanData** are FALSE then you will need to provide the name of the cleaned data file (within the **dataDir**) downloaded from Dataverse.
3. **figurePath**: (Optional) If you intended to output the plots enter directory to export to

This is the extent of manual operations. All other processes run automatically. If the data is fully built (**reBuildData** = TRUE and **reCleanData** = TRUE) this process may take multiple hours. Additionally, the user may change a number of the optional parameters that handle the distance scaling, overall number of metro-regions to analyze, maximum distance from central business district centroid to include in the data and whether or not computational progress is reported.

Stage 1: Data Collection

Based on the parameters selected above the data collection phase of the study either downloads the compiled (**reBuildData**=FALSE and **reCleanData**=TRUE) or cleaned data (**reBuildData**=FALSE and **reCleanData**=FALSE) from the Dataverse repository or compiles all of the raw data directly(**reBuildData**=TRUE). To compile the raw data, files for every county in the fifty largest metropolitan areas are downloaded, unzipped, cleaned and written out as a standardized .csv (comma-separated value) file. This raw data is hosted on the U.S. Census Bureau's FTP site. Custom functions to handle the data acquisition process were written in R and are found in the *buildHHData.R* and *buildCBSAData.R* files in the repository.

Stage 2: Data Processing

If the data is to be recleaned, then the data cleaning functions are employed at this step. In this process observations with missing data are removed and information on the core-based statistical areas (CBSAs) are added to the compiled data. If cleaned data is directly downloaded then this pre-cleaned data is passed forward to the analysis stage.

Stage 3: Data Analysis

The analytical process begins by calculating the location quotient distance profiles. Location quotient profiles measures the proportion of a given household type at a location versus the proportion of that household type in the entire metro region. Location quotients higher than 1 indicate that, relatively speaking, more of a given household exist at a given location than would be expected if households were randomly distributed. The *hhLocFunctions.R* file contains all of the custom functions necessary to calculate and visualize the location quotient results.

Data visualization of the results via a variety of different plotting functions follows. Final results, both tabular and visual, are then combined in an RStudio/Knitr file along with the narrative to create the final document (compiled in LaTeX). The full data provenance is described and hosted on the code repository via a Markdown file. Also note that the collaborative website [Authorea](#) (which offers git-based tracking and LaTeX support) was used by the authors to write the first draft of the narrative portion of the report.

Pain points

There are two major steps that we consider particularly painful. The first is convincing yourself (and co-authors) to take the time to properly document every action and to take the time to fully annotate the analytical workflow. This can be especially difficult when deadlines arise or when co-authors do not see the value in reproducibility. The second is the need to write custom functions that are generalizable. Writing very specific, single use functions can be easy, but are rarely useful in more than a single instance. Good reproducible research contains flexible functions than can accommodate changes or permutations thereby allowing subsequent users to expand or change your original analysis.

The current peer-review process also presents a considerable hurdle to reproducibility. In order to remain anonymous in the review process, we've had to build a set of anonymous code and data repositories and interactive websites for the review process and then switch over to our own repositories after the paper has been accepted. It means a lot of extra work as well as remembering which GitHub account we are signed into at all times. Along this line, judging by usage statistics, reviewers have been uninterested in actually examining the hosted code, data or results.

Key benefits

For us the biggest benefit is efficiency. The first time we do an analysis it usually takes longer than it would take other colleagues, but each time after the time savings multiply. One situation where this is particularly helpful is in responding to peer reviewer comments and requests. Changes to assumptions or sensitivity tests on parameters can be done in a matter of hours (or minutes), not days or weeks. This greatly shortens the re-submittal response time. Related, we constantly find ourselves borrowing old code and re-purposing it, making new analyses easier and faster.

(Andy) Better organization is another benefit. No more folders full of data files with version names and dates. No more mystery fields in a dataset. No more starting all over after forgetting what was previously done. My students and their Excel sheets with dozens of tabs and screen clips from SPSS (or other point and click-based statistical software) remind me of this benefit every semester. I am slowly incorporating more and more reproducibility into my classes, with the intent of breaking some of these bad habits that students have.

(Hossein) Another benefit would be built greater capacity for related research. Beyond the theoretical approach that can be used to study other metropolitan patterns, functions that we built in this research can be applied to facilitate other forms of research using census data. Researchers can adapt these functions to address other purposes. In an ideal scientific world where all research is reproducible, research will be more efficient because of the code that can be shared, re-used, or adapted for research or non-research purposes.

Key tools

The RStudio integrated development environment (IDE) and their related Shiny Apps (interactive web applications) have been a huge help in our reproducible research. If you are an R programmer and want to share your visualizations with non-programmers, we highly recommend these tools from RStudio. Using the IDE allows for easier navigation between multiple scripts, reviewing a history of plots and offering a view of all objects in the current computing environment.

Questions

What does "reproducibility" mean to you?

"Reproducibility" means that a subsequent interested party can openly access the data, code, analytical workflow and data provenance to re-create the research (and ideally produce identical results) WITHOUT consulting the original researcher(s). In this context, "reproducibility" can facilitate the verification of results from a given research project and also accelerate new research discoveries by providing reproducible modules that can be applied in other settings and/or for other purposes.

Why do you think that reproducibility in your domain is important?

(Andy) A majority of quantitative analyses in real estate (both academic and professional) is usually duplicated by numerous parties, widely disseminated and frequently updated; all characteristics that benefit from reproducible analyses. Despite these core facts of the discipline, there is very little, if any, discussion on or attempt to create reproducible research in the field.

(Hossein) In general, the importance of reproducibility in policy-/decision-oriented fields is not clear. It can certainly improve policy research, but one could debate whether or not reproducibility has direct benefits for decision-making.

How or where did you learn about reproducibility?

(Andy) My pre-academic background was as part of a team of expert witnesses in litigation support. In this industry, any analysis that was produced had to be reproducible by the opposition and therefore, my firm was constantly striving to produce more efficiency in their reproducible analyses.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

(Andy) Two challenges exists in the real estate field. First, most data is proprietary and expensive and therefore it is hard to share data. Second, it is a small field that is composed of many senior individuals (both in academia and industry), many of whom are very resistant to change.

(Hossein) In health sciences the biggest concern is around data privacy. For example, research on individual-level patient information can hardly become fully reproducible, within conventional workflows.

What do you view as the major incentives for doing reproducible research?

Doing reproducible research is like installing solar panels in your home. It will cost you at the beginning, but down the road you will get benefits such as time savings, better quality output and the increased opportunity to collaborate/share ideas.

Are there any best practices that you'd recommend for researchers in your field?

No more manual data cleaning. Use code.

Would you recommend any specific resources for learning more about reproducibility?

For collaboration, if you want to get away from writing in LaTeX, you can try [Authorea](#). If you are in Australia, the [University of Melbourne's Research Platforms](#) group offers a number of Research Bazaars, Software Carpentry and Reproducibility-related courses and event. It is open to researchers world-wide.

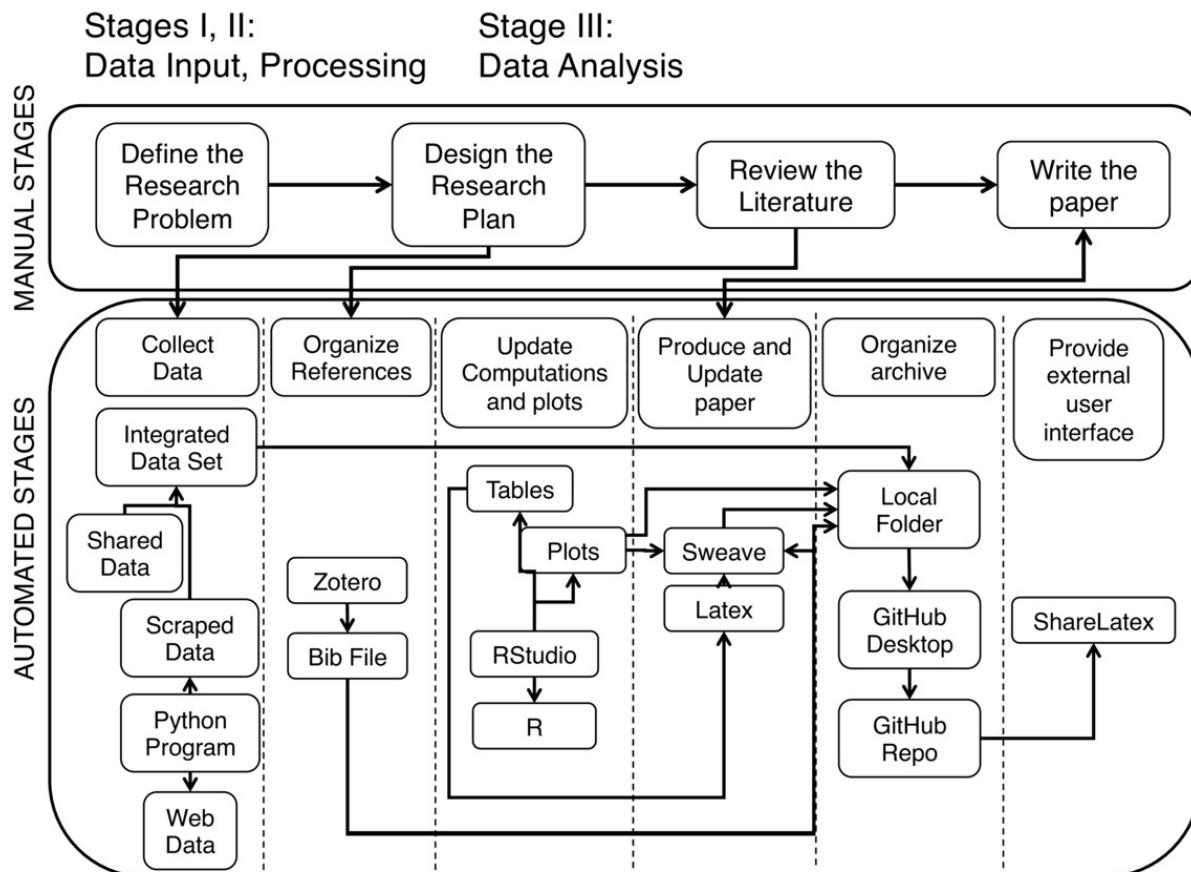
Analyzing Cosponsorship Data to Detect Networking Patterns in Peruvian Legislators

José Manuel Magallanes

My name is José Manuel Magallanes, I am a Senior Data Science Fellow at the eScience Institute of the University of Washington, where I am also a Visiting Professor at the Evans School of Public Policy and Governance (2015 - 2017). Since 2003, I have been Professor of Political Science and Public Policy Methodology at the Catholic University of Peru. My research is related to framing political and policy problems with a computational social science approach. I have dealt with different topics including electoral behavior, public management performance, climate change and social conflict, and legislators behavior. My contribution for this case will be a research carried out on bill cosponsorship data to detect key players, reveal association patterns, anticipate party splitting and detect tactics to get re-elected.

I have a BSc in Computer Science (UNMSM - Peru), a MA in Political Science and Public Management (PUCP - Peru), a Phd in Psychology (UNMSM - Peru) and a Phd in Computational Social Science (George Mason University-USA).

Workflow



The workflow above represents:

a. Manual Stages:

1. Define the Research Problem.
2. Design the Research Plan.
3. Review the Literature.
4. Paper writing.

b. Automated Stages:

1. Collect Data.
2. Organize References.
3. Update computations and plots.
4. Produce and update Paper.
5. Organize archive.
6. Provide external user interface.

A. Manual Stages

a1. **Defining the research problem.** There has been an interest in the political science community in Peru to learn more on the dynamics of their National Congress. Particularly, Peruvian scholars and pundits have been discussing some particular phenomena affecting the congress dynamics:

- The low re-election rate of legislators the previous two elections (~20%).
- The fact that some legislators migrated to other parties during their mandate (party switching).
- The fact that parties that had a good share of seats, ended up splitting their seats (It is worth keeping in mind that Peru has a multiparty system).

Some scholars in the USA have been using bill cosponsorship data as a proxy to understand some of these issues, so I decided to follow a similar approach. However, the complexity of the Peruvian case was higher than the bipartisan American Congress, but the data was less available in the Peruvian case; as this work used one Congress data while there are only five Congress periods available on-line as webpages (no API and no data to download).

This stage was done only once. I was the only one in charge to define the research questions (no co-authors); however, some colleagues participated in informal exchanges of ideas. No particular computational tool was used in this stage.

a2. **Designing the Research plan.** This stage identified the main authors that have worked similar research problems before. The key ingredient in all cases was bill cosponsorship. However, most hypotheses were not the same I had, due to the different political regimes that researchers were focusing on. But, in all cases, bill cosponsorship was considered a good proxy to understand legislator's associative patterns. From this stage it was clear that:

- We would need to write code to extract the information from the Congress of Peru website, as the data was not available for download by any means. This process, also known as *web scraping*, collects poorly structured data from webpages, and gives them a structure that could be used in further computational or statistical analysis.
- To test the hypothesis, the information of a complete Congress would be needed (five years).
- The information from the bills would need to be complemented with the archives of the National Jury of Elections. There, personal information on every legislator is available. This information was downloaded.
- There would be a need for graph or social network techniques.
- The budget available would require the use of free tools.
- There would be a need to share the findings with other scholars.
- This research could be combined with other efforts in another similar countries. There was a need to organize efficiently the process, so that the data and the code could be reused.

This stage was done only once. I was the only one in charge to define the research plan. No particular computational tool was used in this stage.

a3. **Review of the literature.** This step allowed me to identify similar cases and organize my basic set of references. The references were continuously updated along the process. I was in charge of updating, but also got some recommendations from the users I shared my drafts with.

a4. **Write the paper.** As expected, this was a manual step. However, as I describe later, this was supported by different tools. As usual, this step was repeated many times. I was the only contributor.

B. Automated Stages

b1. **Collect Data.** Data was collected from two main sources:

1. [The Congress webpage](#). This website has a webpage for every bill proposed. The webpage has detailed metadata on each bill, including the authors (legislators), which represent the nodes of the network.
2. [The INFOGOB webpage](#). This webpage provided the information needed to organize some attributes of the legislators (nodes).

The INFOGOB webpage is organized in such a way that you can download information for different processes. It also helped me get the political history of every legislator.

The webpage of the Congress was much different. The information needed is visible as webpages, but they do not offer a download service or a mechanism to get the data (known as API - Application Program Interface) in a structured way. For this reason, a code for scraping the website was needed. The code was written in Python, relying mainly on the *beautiful soup* package. I created some extra code to 'clean' the values collected.

So, with INFOGOB, I built the attributes of the legislators; and, with the scraped data, I built the network. Both data sets were merged using Python's *Networkx*. The merged file was saved as a GraphML file and also as a two separate file of edges and nodes, which will ease exporting into R.

This process was done entirely by me. It was the first part of the operational research and took around two weeks. The Python version I used was 2.7, and it was installed via Anaconda. I used the Spyder graphical user interface (GUI) to do the coding.

b2. Organize references. References are a key component of academic writing. In my case, besides papers and books, there was also the need to include webpages, white papers, code, data, and so on. As it is common, there are set of references you know you would use when you start writing, but more come along the process as you exchange ideas with colleagues. In this particular aspect, the use of **Zotero** was very important. It allowed to create a BibTex file to be used later during the paper production process. This text will later be integrated into the LatTex document of this work. Automating this process not only helps you recover the right of citing a work, but also gives you the flexibility to later change the style (APA, Chicago, etc - see [citation list](#)) a particular publisher will require. This was extremely important as this research could be presented in social sciences or computer-science-related conferences.

This process was done entirely by me. This was a continuous process as the paper was written. The desktop version of Zotero was used. The BibTex was saved in the working folder.

b3. Update computations and plots. While data collection and structured datasets were produced in Python, the exploration of the data, the test of hypotheses, and the visualization of results was done in R. I decided to use R for a simple reason: RStudio can combine LaTeX and R in an easier way than Python via its *sweave* library. Sweave differentiates between text and R code; codes are organized in *chunks* that also can interact with the LaTeX code.

This eased the update of the tables and plots produced by the data, as *sweave* documents will rerun the R code and update whatever is needed. This was a crucial part to make this work more reproducible; and also for me, as I could try different *layouts* for the network plot

and pay closer attention to final appearance of the paper.

This process was done entirely by me. This was a repetitive process as the paper was written. A Rnw file was produced using RStudio, which also produced a LaTeX file.

b4. Produce and update Paper. RStudio integrated my writing, the bibliography file, and the tables and plots generated using R into a LaTeX document, which finally would produce a pdf document (RStudio, via *knitr*, instead of *sweave*, can produce also an html document) . Any change in whatever part of the main document or any of the files used was updated in the final product automatically.

It is very important to keep in mind that *sweave* allows LaTeX users to customize all the details in the document, which includes code highlighting or hiding, among other possibilities. I could even present the Python code inside the document as needed.

This process was done entirely by me. This was a continuous process as the paper was written. And in fact, there were many versions that I could share with colleagues. LaTeX complied the R chunks producing tables and plots, and compiled the bibliography into the main document from the BibTex file generated in Zotero.

b5. Organize archive. One of the first steps after the research questions were clear, and before any coding was made, was the creation of a GitHub private repository. This repository was cloned into my laptop, and all the files were organized in this folder, including code, data files, bibliography files and plots. In a way, using a repository that will be online forces you to organize your work and folders since the beginning. Before becoming a GitHub user, preparing the final version of my work took too much time; a good planing when using GitHub will force you to your system of folders ready when you are done with the paper. An additional advantage is the version control power you have when using GitHub, which I had to use just one time, to recover a version that had a code that produced a better plot than one I thought was going to work better. Without it, you need to be commenting and uncommenting code sections which increases, unnecessarily, the coding space.

This process was done entirely by me. This was a continuous process as the paper was written. The GitHub client was used for committing and synchronizing the local repository into GitHub.

b6. Provide external user interface. It was clear during the planing stages that I will need to share my drafts with other colleagues in order to get some feedback and/or discuss further collaboration on this matter. As the paper reflected an step-by-step approach, it would be easier for my colleagues to read the draft paper which included the code chunks, the plot and the tables. For that, I decided to use [ShareLaTeX](#), which can collect the files in the GitHub repository and compile the LaTeX document. So, after I updated the GitHub with my last version, I could also ask ShareLaTeX to update its contents based on the latest document version I had recently pushed into GitHub.

This process was done entirely by me. However, the drafts were shared when most of the processing was finished. This was a continuous process as the paper was written. The selected users created ShareLaTeX accounts to see the LaTeX generated pdf version of my document. I allowed them to write comments in the LaTeX document using ShareLaTeX itself.

C. On the Data, Software and Processing

- **Data:** The raw data as well as the cleaned and aggregated data are online, in a private GitHub repo. The data can be share upon request and instructions are included on how to cite it. The data files have a table-like structure to be easily read into R, but other versions were produced in xml-like format as I thought I may need to use other network visualization programs like Gephi.
- **Software:** The Python code is also in the repository. The R code is embedded in the LaTeX code, and the paper itself describes the algorithms adopted in the paper. Most R chunks make constant use of the data scraped using Python. For GitHub, ShareLaTeX and Zotero, you only need to create and account and download the desktop version.
- **Processing:** The processing of the data is reflected in the Python code flow, and it is online. The Python and R codes are commented extensively. It would be fairly easy for an external researcher to follow the logic of the research and replicate the results, or simply change the data from other country and get all the tables and plots in the final PDF, as R, Python and LaTeX are connected.

Pain points

This work was not producing a blog or a notebook, but a paper. So the most challenging parts were:

1. Produce a quality layout where tables and plots are located in the right place is hard. LaTeX is not exactly what you see is what you get, so you need to learn how to override some default behavior in LaTeX for that.
2. You can become too excited as you learn to use LaTeX, so you start thinking all the time to make it better, and it takes too much extra time because you need to include more LaTeX functions and need to learn how to configure them. It is better to do that after the paper is done.
3. Scraping several webpages takes time, and you learn that your code may only be usable for those particular websites. I scraped many pages, but all came from the same institution, so a project that involves scraping from more than one institution will deal with much more complexity.

4. A particular pain point is the lack of a reproducibility culture in the field I work. Political scientists in my country are not used to reproducible research. In fact, for every key paper that dealt with the kind of data I used, no further instructions were found from the authors or in the authors' webpages. In most cases, it is only mentioned what data was used but no links or other related procedures were clear.

Key benefits

I consider the way I worked allowed me to obtain several benefits:

1. Planing your research in a reproducible way is a great advantage to the scientific community you belong to. But most of all, it forces you to plan your work better.
2. Including version control forces you to have well organized set of folders in your machine.
3. Following a reproducibility approach will allow you escalate your work if more data becomes available or if a colleague wants to make a comparative work. I am sure this is not impossible without this approach, but I am sure that researchers can become much more productive than in the past.
4. Another important benefit is that allowing colleagues to audit your work gives you enough input to make a newer and more robust version of your work.
5. You have the possibility to produce plots with different levels of quality. R allows you to produce simple quality plots and more complex formats. In this case, I was requested a higher resolution of a plot in vectorized format, and I simply recreate the one I had, changing a couple of parameters.

Key tools

LaTeX was a key component in all this research and its reproducibility level. It offers a way to organize the paper and interact with code and data files, including references and plots. This can not be done using Word, as far as I know. LaTeX is not a common software in social scientists in my country. RStudio is also a key ingredient. Its capacity to transform the R chunks and its output (including tables, values and plots) into LaTeX makes the flow and update of research even better. Both LaTeX and RStudio facilitate reproducibility. Without R, the barrier for producing papers is even higher, but it can be done. It gives you more confidence and save you lots of time to update/edit your manuscript, compare to copy, past or inserting procedures in MsWord. The flow is simply great.

Questions

What does "reproducibility" mean to you?

In general, I consider this term means the level of reconstruction of a research that can be achieved by a person foreign to the researcher / research team via the code and data available in some repository. For me, reproducibility is not only that the foreign person can decompress and run and executable file to see the results, but be able to audit the whole process. The less feedback required by the auditor, the more reproducible a work is.

Why do you think that reproducibility in your domain is important?

Because computational social science is still young in many other countries. As in my case, data is just starting to become available, so following, and teaching, the reproducibility approach will benefit the research quality. In public policy, particularly, it will enable stakeholders participation in knowledge creation.

How or where did you learn about reproducibility?

I had no chance to have mentor or courses on this. I just felt the need to organize my work as many tools and data were available for my case. I was afraid that if I did not follow this approach I could easily get lost. Reproducibility demands good research planing, and it pays off.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

If the data you are using is public, I see *less* problem. When the data is not, and you get access with a special permission, legal issues are always present. As for investment, my particular collection of tools are free, so it should not be a problem, unless your funding institution forces you to use particular tools. I also believe that this approach can be very challenging for older generations not used to this. I see less of a problem in younger generations of researchers.

What do you view as the major incentives for doing reproducible research?

The main incentives for me are project organization and ,m.l/. That is, reproducibility requires order, some structure to your work; then, you need to find a way to organize writing, code, files, and so on. I believe that LATEX enables reproducibility too, as it can interact

seamlessly with other software elements, as shown in my work here. Auditability is important as it increases your credibility in the research community, mainly because anybody can follow closely what and how you did your work.

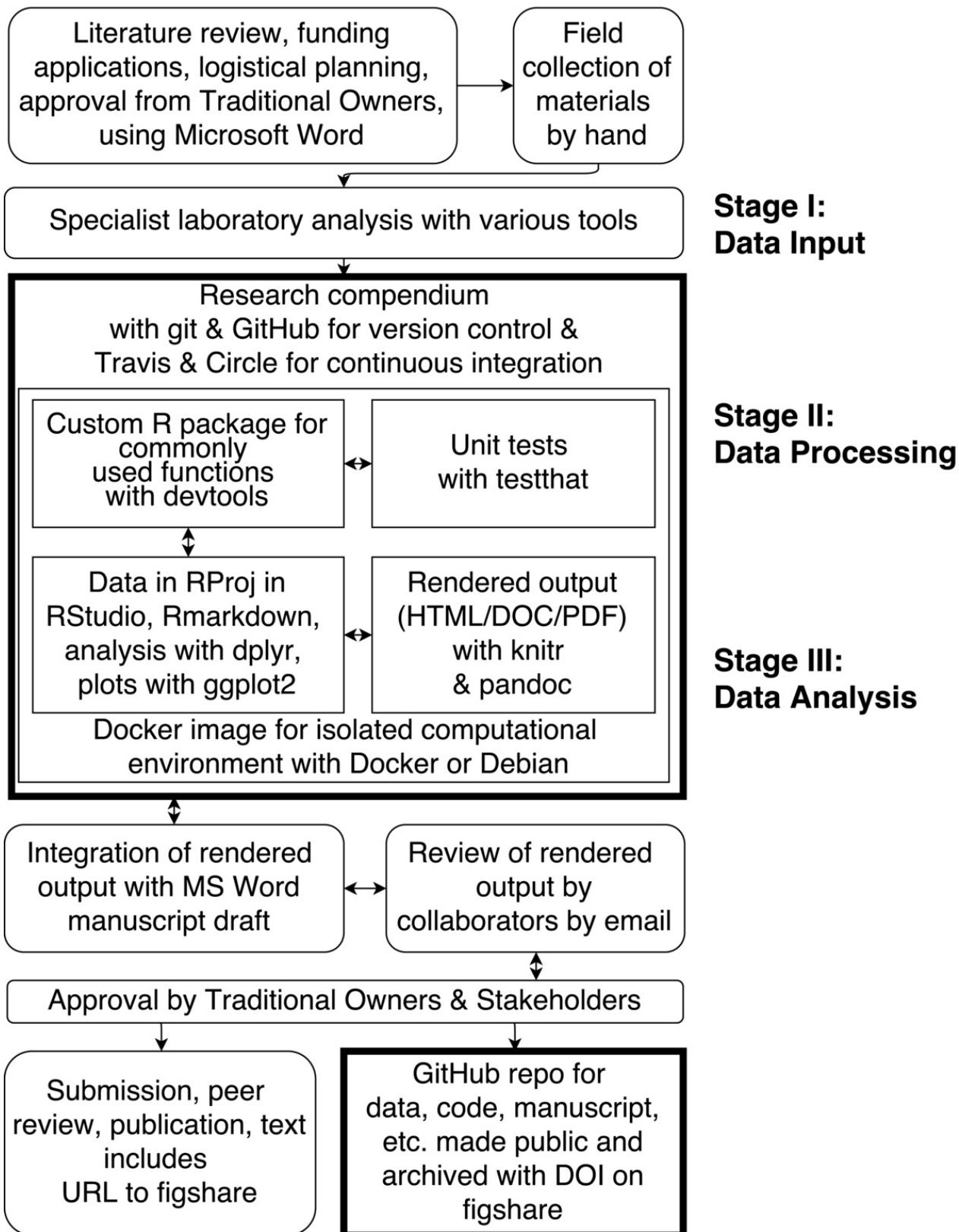
Using R and Related Tools for Reproducible Research in Archaeology

Ben Marwick

My name is Ben Marwick, and I am an Associate Professor of archaeology in the Department of Anthropology at the University of Washington, and a Senior Research Scientist at the University of Wollongong. My research interests include human-environment adaptations during the Pleistocene in Southeast Asia and Australia. My colleagues and I work with stone artefacts, organic and geological remains to understand past human behaviours and their environmental contexts. My narrative here describes one recent project from the initial concept to a specific publication (Clarkson et al. 2015), but the details below are typical of my experience with several projects focused on understanding prehistoric hunter-gatherer behaviour (cf. Marwick et al. 2016; 2017).

In the context of this case study, reproducibility refers to computational reproducibility, which means enabling other researchers and students to combine the code and data that we produce to obtain the same statistical results and data visualizations that we present in our publication. I also expect that the code could be used for empirical reproducibility, where our code is applied to a new dataset to generate substantively similar results to our published results. I have explored these definitions, and the principles that motivate my selection of tools, in more detail in Marwick (2016).

Workflow



The boxes with a bold outline indicate key steps and tools that enabled computational reproducibility in our project.

One recent project I was involved in aimed to excavate Madjedbebe rockshelter, a well-known archaeological site in northern Australia. The purpose of the excavations was to test the findings of previous excavations in the 1990s that uncovered controversial early

evidence for human occupation of Australia. The project was initiated with consultation with stakeholders, including Aboriginal traditional owners of the archaeological site, and a grant application written in Microsoft Word and circulated among the team by email.

The archaeological excavation was conducted with standard modern techniques. These include a combination of direct digital capture of artefact and feature provenance with a total station, digital photography and GIS, and hand-written paper notes using structured site recording forms. These data from these forms was later entered into an Excel spreadsheet.

At the conclusion of fieldwork, post-excavation analysis continued at the home institutions of each of the team members. The tools for data collections and analysis at this stage were according to the norms of each lab, but the final products from most of the team members at this stage were MS Word and Excel files. At this point, work began on a manuscript for publication, which was a MS Word document that was circulated among the authors by email.

As the specialist work concluded, the Excel and Word files were collected into an R Project using [RStudio](#). The spreadsheets were converted to CSV files to ensure they could be accessed independent of any specific software. A research compendium was created, based on a custom R package, following the examples described in [rrpkg](#). This package was written to contain custom functions used often in the analysis. The [devtools](#) package was used to develop the custom R package in RStudio. The [testthat](#) package was used to write tests to ensure the package functions performed as expected while they were being developed. An R markdown file was created as part of the compendium, and edited in RStudio to recompute and extend the analysis and visualizations from the specialist labs, and combine the key pieces of narrative text from the lab reports that contain statistical results. The R markdown file is a kind of lab note book where code and text are interwoven in a single document. It summarizes and extends the work of the team specialists using R script. The code in the R markdown file used several R packages, including [dplyr](#) and [reshape2](#) for data cleaning and analysis, [rioja](#) and [analogue](#) for specialist environmental methods, and [ggplot2](#) for visualization. The runtimes of the analyses are rarely longer than 30 min, so writing code and narrative, and testing are the most time consuming tasks here.

The R package [knitr](#) and the [pandoc](#) program (included with RStudio) was used to execute the R markdown file to inspect the output as the code was being written. A [Docker](#) container was created to create an isolated computational and portable environment for writing the R markdown document and developing the package. The Docker image was backed up on the Docker Hub server

(<https://registry.hub.docker.com/u/benmarwick/mjb1989excavationpaper/>), and tested using continuous integration from [CircleCI](#). All of these components, data files, R markdown file, package files, etc. were all version controlled using git locally and backed-up on a repository at GitHub. The GitHub repository is here <https://github.com/benmarwick/1989-excavation->

[report-Madjebbebe](#), and a snapshot of this repository at the time of acceptance of our 2015 *Journal of Human Evolution* paper is archived on figshare here:

<http://dx.doi.org/10.6084/m9.figshare.1297059>. One of the downsides of using this compendium approach is that most of the work is done by just a few of the team members because not everyone is familiar with (or interested in) the tools.

While the analysis was being developed in the research compendium, a manuscript was being drafted in a MS Word document and circulated among the authors by email, and revised using track changes. The rendered output of R markdown document is also circulated among the authors by email. As the manuscript is updated, and new ideas are incorporated into the analysis, additional code is written, some code abandoned, new plots produced, and others deleted, etc. This is probably the messiest and least ideal part of the workflow as it involves manual updating of the MS Word document with new values and figures from the rendered R markdown document, and two unrelated version control systems (git and track-changes in MS Word). The non-linearity of the process was also challenging, as the authors negotiated how the manuscript and analysis should take shape.

As the review and updating cycle concluded, the manuscript was sent for review by the traditional owners of the land where the archaeological site is located. After this review, which involves some changes to the manuscript, the final draft was prepared for submission. At the same time, the GitHub repository that contains the research compendium was made public and continuous integration from [Travis](#) was added to monitor the effect of changes made during peer review. The compendium was also deposited at [figshare](#) and the persistent URL to the figshare repository was added to the text of the manuscript as a pointer to the data and code that generated the results and visualizations found in the paper. The MIT license was attached to the code (giving others permission to use and reuse the code), the CC0 license was attached to the data (meaning that the data are in the public domain), and a CC-BY license was attached to the text and figures (meaning that the text is free to use with proper attribution to the original authors). These licenses allow flexible reuse of our materials. The paper was then submitted for publication at the *Journal of Human Evolution*. At this point the data and software were openly available online for peer reviewers and others to inspect. The code includes the R package, which has documentation about installing the packages and using the functions, has unit tests, and has machine- and human- readable metadata about dependencies. We have also made available the Docker image that contains the compendium in an Linux environment so that all the dependencies external to R can be included in a single bundle.

Pain points

Some of the most notable pain points include:

- the inefficiencies of duplication of effort in translating the Excel-based analysis into R, and in moving between MS Word and R markdown for drafting the text. This happens because only a few members of the team are familiar with R and related command-line tools.
- the complexities of working on the draft manuscript and updating the analysis as the team explores different options and research directions. This challenges are typical of any large collaborative project, but I think multiplied here because of the 'two universes' of toolkits, with some of the team using Microsoft tools, and others using open source tools. Because of the greater flexibility and efficiency of R over spreadsheets for data analysis, we observed a disempowering of team members who are not familiar with R.
- overall, the research compendium is still quite a complex arrangement of tools and scripts, and productive engagement with it requires a high degree of enthusiasm and a high tolerance for trouble-shooting. This is a barrier for collaborators who don't share my interest in reproducibility. However, I'm optimistic that making and using research compendia will become simpler and more normal, and increasing awareness about reproducibility will motivate researchers to take a greater interest in incorporating these practices into their own work.

Key benefits

Some of the advantages that motivated us to pursue that approach include:

- A detailed human- and machine-readable record of all the steps in the analysis. This takes the methods out of Microsoft Excel, where they are often invisible due to ephemeral point-and-click behaviours, and reconstructs them in R scripts where every step is explicit. This makes it a lot easier to engage with questions like "can we do that again, but change X a little bit?" and "what happens if we add/exclude Y from the analysis?" This kind of exploratory work is most efficiently done using a scripting language because the equivalent work in a spreadsheet often requires redoing numerous manual steps of data manipulation simply to alter one small step in the analysis pipeline.
- An open and transparent record of the analysis for reviewers to inspect, this allows us to say 'we have nothing to hide', and the git repository allows us to show 'we already tried that, and this is what we got' because we have a complete history of our analytical efforts, even those that didn't lead to results included in the publication.
- We have a high degree of confidence that our results are correct. We can rerun the analyses repeatedly in an isolated and well-defined environment and get the same result each time.

- Our data and methods are available for reuse and application to new projects and contexts by us, and by other researchers and students. This saves time for us in the future, and has the potential to increase the impact of our work.
- The uniqueness of our workflow is a double-edged sword because it attracts attention to our project because of its exoticness, but because it's so unfamiliar few people can engage with it or use it in the ways we're hoping. As I developed this workflow I was worried it might be a once-off effort, and that it wouldn't be suitable or sustainable for future projects. Since that time, I've found the opposite to be true -- I've used a similar R-package-as-research-compendium approach as I've described here for subsequent scholarly publications (e.g. Marwick et al. 2016; Marwick et al. 2017). In evolving and simplifying this workflow I've enjoyed substantial gains in efficiency. I've also received a lot of interest in this approach from other groups outside of my discipline who are keen to adopt these practices to improve the reproducibility of their research.

Key tools

The key specialized tool that enhanced the reproducibility of our research is R, and the suite of user-contributed packages that extend its functionality. Many of these add several idioms that greatly improve the ease of use of R, such as dplyr, ggplot2, and knitr. The RStudio program was used to develop the code because it has many built-in conveniences that lower the cognitive burden of package development and coding. Although git and GitHub are not specific to R, use of git is deeply integrated into RStudio, so we consider it part of the R ecosystem. Similarly, Pandoc is a universal document format converter that is not unique to R, but since it is also built into RStudio we consider it part of the ecosystem also.

In addition to R and its ecosystem, we used several popular software engineering tools to help with quality control. These include Docker, a system for lightweight virtual environments (and boot2docker, which enables Docker on Windows and OSX), Travis, which builds and checks our R package each time a commit is made to the GitHub repository, and CircleCI, which is a similar service to build the Docker image and run some simple tests each time a commit is made that changes the dockerfile. We also used these services to render our R markdown documents each time a commit was made, to check that no errors had been accidentally introduced.

While R by itself ('base' R, without contributed packages) is familiar to many social scientists, the packages noted above that introduce powerful modern idioms are less well known. The broader R ecosystem and software engineering tools we used are almost totally unfamiliar to our peers, despite their ubiquity in the software development community. So we see a lot of potential for these tools to be of broader interest, ideally because of the reproducibility they efficiently enable, but likely also because of their novelty.

Questions

Why do you think that reproducibility in your domain is important?

Reproducibility is important because many important steps in our data analysis occur on the researchers' computers, but these steps are often not documented in a way that we can easily access, archive, and communicate with others. The use of software operated by a point-and-click interface is the key problem here. By changing the key analytical tool to a scripting language such as R, we change the nature of our computational work from closed and ephemeral, to open, reusable, and enduring. This makes it a lot easier to show what we've done, why we think the results we present are correct, and enable us and others to reuse and extend our work. These are fundamental for the advancement of science, and with improved reproducibility in our research, we can advance science faster and more reliably.

How or where did you learn about reproducibility?

Most of the reproducible practices in our project were self-taught by a few members in our team adopting practices they've observed in elsewhere, such as ecology and biology. Key resources in this self-teaching include Software Carpentry teaching materials, materials produced by rOpenSci, and instructive scholarly publications and blog posts with code examples, and GitHub repositories written by researchers in other fields who are highly progressive in enabling reproducible research. These include Carl Boettiger, Jenny Bryan, Rich FitzJohn, Karl Broman, and others in the rOpenSci community. Many of the idioms that greatly simplify using R for archaeological data analysis have been contributed by Hadley Wickham and his collaborators on the 'tidyverse' set of packages. The R community on [StackOverflow](#) is a great resource because of their strong emphasis on including reproducible examples in questions posted to the site. Many of the questions and problems I encounter have already been answered in several different ways on StackOverflow, often by highly skilled programmers.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

In my field, where the datasets are usually not problematically large and compute times are not inconveniently long, I consider that most of the technology barriers are sufficiently low to be considered solved. The tools are stable, well-documented, and widely used in other domains, so I don't see any major technical challenges. The key challenge is human - not

everyone in the team has the skills to use the tools that enable reproducible research, and not everyone has the motivation and opportunity to learn. This contributes to the primary logistical challenge, which is the manual integration of project components using traditional low-reproducibility tools and the components that enable high-reproducibility. My suggestions are to ride the wave of generational change, and teach students and early career researchers about reproducible research as a normal part of doing research. This means teaching them to expect that analyses should be done with a scripting language (rather than point and click), and that code and data from other researchers should be openly available for inspection (rather than 'by request', which when requests are made, are often refused or ignored). This is the long game, waiting for generational change, but I think will be more effective than efforts full of sound and fury to change the entrenched behaviours of senior colleagues, who rarely have the time or inclination to learn new tools and workflows.

What do you view as the major incentives for doing reproducible research?

The major incentives are:

- increasing the certainty of the correctness of our results
- increasing the ease of tracking our analysis, and exploring new options
- increasing the impact of our work by increasing the ability of, and likelihood that, other researchers will use our methods, data and results.

Are there any best practices that you'd recommend for researchers in your field?

The generic practices I'd recommend for researchers in my field include:

- making raw data openly available in trustworthy repositories in open formats at the time of publication
- using scripts written in a widely used open source programming language to analyze the data
- making the scripts openly available in trustworthy repositories so that they can be used with the data to generate the figures and statistical results in the publication.

Would you recommend any specific resources for learning more about reproducibility?

- Stodden, V and Miguez, S (2014). Best Practices for Computational Science: Software Infrastructure and Environments for Reproducible and Extensible Research. *Journal of Open Research Software* 2(1):e21, DOI: <http://dx.doi.org/10.5334/jors.ay>
- Gandrud, C. (2013). [Reproducible Research with R and R Studio](#). CRC Press. Chicago
- [Reproducible Science Curriculum](#)
- [Software Carpentry](#)
- [Data Carpentry](#)
- [rOpenSci Reproducible Science Guide](#) (and see the *further readings*)

References cited

Clarkson, C., Mike Smith, Ben Marwick, Richard Fullagar, Lynley A. Wallis, Patrick Faulkner, Tiina Manne, Elspeth Hayes, Richard G. Roberts, Zenobia Jacobs, Xavier Carah, Kelsey M. Lowe, Jacqueline Matthews, S. Anna Florin 2015 The archaeology, chronology and stratigraphy of Madjedbebe (Malakunanza II): A site in northern Australia with early occupation. *Journal of Human Evolution* Volume 83, June 2015, Pages 46–64 DOI: <http://dx.doi.org/10.1016/j.jhevol.2015.03.014>

Marwick, Ben, Hannah G. Van Vlack, Cyler Conrad, Rasmi Shoocongdej, Cholawit Thongcharoenchaikit and Seungki Kwak (2017) Adaptations to sea level change and transitions to agriculture at Khao Toh Chong rockshelter, Peninsular Thailand. *Journal of Archaeological Science* 77:94-108. DOI: <http://dx.doi.org/10.1016/j.jas.2016.10.010>

Marwick, Ben, Chris Clarkson, Sue O'Connor and Sophie Collins (2016) Early modern human lithic technology from Jerimalai, East Timor. *Journal of Human Evolution* 101:45-64. DOI: <http://dx.doi.org/10.1016/j.jhevol.2016.09.004>.

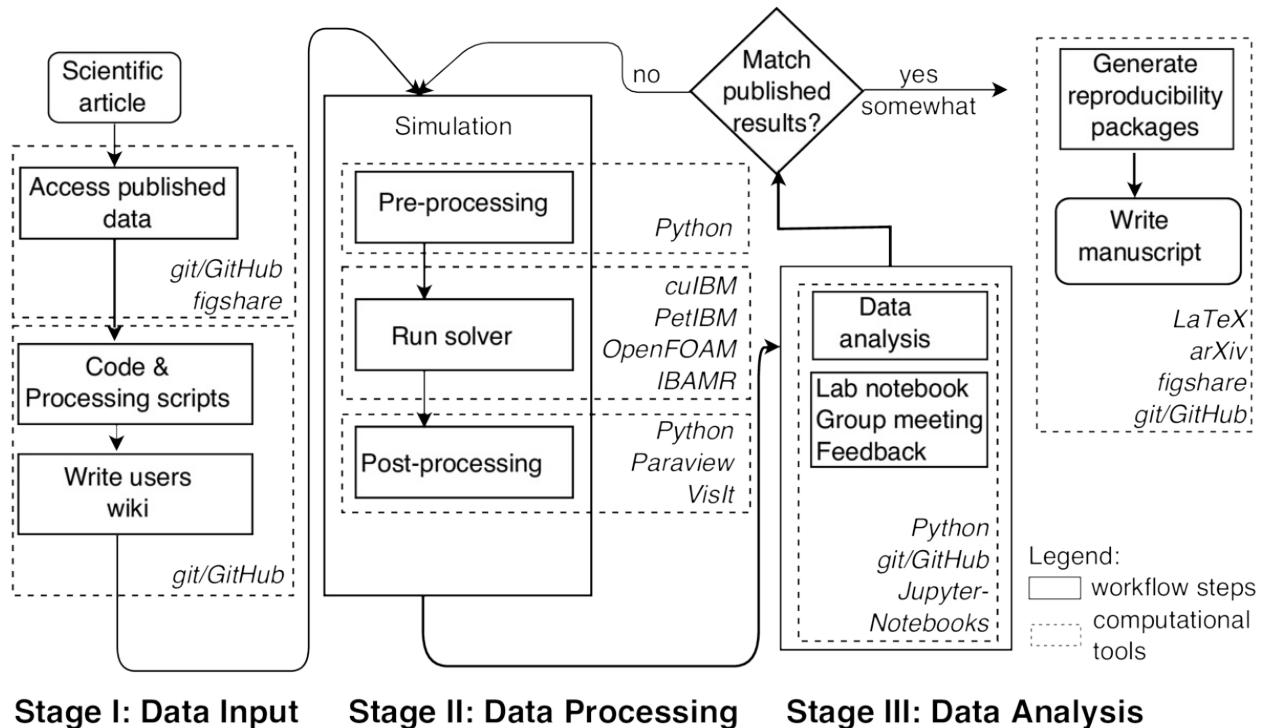
Marwick, Ben (2016). Computational reproducibility in archaeological research: Basic principles and a case study of their implementation. *Journal of Archaeological Method and Theory* 1-27. DOI: <http://dx.doi.org/10.1007/s10816-015-9272-9>

Achieving Full Replication of our Own Published CFD Results, with Four Different Codes

Olivier Mesnard and Lorena A. Barba

We are members of a computational research group led by Prof. [Lorena Barba](#) at the George Washington University in the department of Mechanical and Aerospace Engineering. We do our best to accomplish reproducible research and have for years worked to improve our practices to achieve this goal. According to the "[Reproducibility PI Manifesto](#)," pledged by Barba in 2012, all our research code is under version control and open source, our data is open, and we publish open pre-prints of all our publications. For the main results in a paper, we prepare file bundles with input and output data, plotting scripts and figure, and deposit them in the [figshare](#) repository. This case study describes what happened when we set out to complete a full replication of published results from our own group, using different Computational Fluid Dynamics (CFD) codes: a new code developed in our group, an open-source code developed by another group, and an open-source CFD library.

Workflow



Our research lab has developed over the years a consistent workflow that, we believe, leads to reproducible research. A previous study coming out of our lab, published in Krishnan et al. (2014), already satisfies the criteria of the "Reproducibility PI Manifesto" (Barba, 2012). That work studied the aerodynamics of flying snakes using our code [culIBM](#) for solving the Navier-Stokes equations with an immersed-boundary method. The crux of the study was that, for a particular configuration, the snake's cross-section experiences a lift-enhancement. Here, we describe our effort to achieve full-replication of the main results, using four different Computational Fluid Dynamics (CFD) codes, including [culIBM](#). We encountered failures and difficulties, leading to improvements in our workflow and conclusions about the challenges for reproducibility in a scenario of highly unsteady flow dominated by vorticity (local spinning of the flow).

The first code we used to attempt replication is IcoFOAM: the unsteady laminar solver of the well-known CFD package [OpenFOAM](#). We chose OpenFOAM because it is widely used, open-source, and documented: both code documentation and users' guide are available. With unstructured-mesh finite-volume solvers like IcoFOAM, the mesh generation step is most often what determines the quality of the solution, and we experienced that some meshes resulted in unphysical results. Our first tries led to inconsistent results and we had to replace the mesh-generation tool to get acceptable mesh quality. Setting the boundary condition at the domain outlet was particularly problematic, and made more difficult by lack of documentation for the type of boundary condition we needed. We invested several months of persistent efforts before finally replicating our previous findings (in terms of the lift characteristics) with IcoFOAM.

We then used [IBAMR](#), an open-source library hosted on GitHub that provides several numerical methods for immersed bodies. One of them is specifically designed for non-deforming bodies, which is our situation. Bhalla et al. (2013) published a detailed validation of this method, and some examples are included in the code repository. After many failed attempts, we found that this method requires forcing the fluid to rest everywhere *inside* the immersed-body, not just at the boundary—this is not an intuitive option with immersed-boundary methods. In the end, we can say that the *scientific findings* of Krishnan et al. (2014) have been replicated, but we still see noticeable differences in the details of the flow characteristics.

The [cuIBM](#) and [PetIBM](#) codes are both being developed in our research lab and implement the same immersed-boundary method (Taira & Colonius, 2007). The GitHub code repositories include code documentation with [Doxygen](#), users' documentation (on the GitHub wiki), as well as basic examples and tutorials. cuIBM uses [CUSP](#), an open-source library for sparse linear algebra on a single CUDA-architecture Graphical Processing Unit (GPU). We used cuIBM again to confirm the reproducibility of the published findings in Krishnan et al. (2014). It is important to remark that we had to use the *same version* of the code, with the *same version* of the linear-algebra library to obtain the same numeric answers as before. In fact, our first attempts used a newer version of the CUSP library, and failed to replicate the findings! In PetIBM, we use the [PETSc](#) library to solve the linear systems on a distributed-memory machine. Even though the mathematical formulation in cuIBM and PetIBM is exactly the same, we observed that a different linear-algebra library could change the results. As of this writing, we have been unable to replicate with PetIBM the lift-enhancement feature of the flying snake.

The lessons learned from this case study are sobering. First, the vigilant practice of reproducible research must go beyond the open sharing of data and code. We now use Python scripts to automate our workflow—all scripts are version-controlled, code-documented and accept command-line arguments (to avoid code modification from users). Instead of using GUIs, we call the Python interpreter included in the visualization tools [Paraview](#) and [VisIt](#) to plot the numerical solution. Throughout, Jupyter Notebooks and Markdown files document partial project advances. Second, certain application scenarios pose special challenges. Here, we are working with the Navier-Stokes equations applied to highly unsteady flows dominated by vorticity, a particularly tough application for reproducibility. Third, extra care is needed when using external libraries for iterative solution of linear systems: they may introduce uncertainties.

As we now prepare a manuscript to publish the results of this project, it is being written using LaTeX and version-controlled in its own GitHub repository to facilitate collaboration between authors. To advocate open-science, the manuscript will be first available on arXiv. We will also provide, on the repository figshare, a reproducibility package for all simulations and

figures reported in the manuscript. These packages include the version of the software, the input parameters, information related to machine architecture, and the necessary scripts to run and post-process the simulation.

Pain points

A critical ingredient in a reproducible workflow is keeping a detailed, up-to-date, and version-controlled lab notebook. It is nearly unthinkable that a proper lab notebook for recording computational experiments could be kept without scripting all steps—pre-processing, running, post-processing—and automatically saving command-line inputs. In the project of this case study, we used four different CFD codes in batches of simulations spanning many parameter combinations, resulting in hundreds of runs. The run times varied between 1 and 3 days and the numerical solutions each generated between 3.5 and 16 gigabytes of data. Most of the simulations were run remotely on an HPC cluster at the George Washington University, and the solutions were then moved to several different local desktop machines for post-processing and storage. The lab notebook proved to be vital for tracking all simulations and data. Another aspect of this project that was very time consuming was becoming familiar with new software—it took even longer to familiarize ourselves with codes that offer poor users' documentation. Finally, we also spent considerable time developing automated scripts for analyzing the numerical solutions resulting from different codes (producing different output formats). These scripts, however, are essential to deliver reproducible computational experiments.

Key benefits

In the field of computational fluid dynamics, it can easily take six months or a year to develop software from scratch for solving a specific fluid-flow scenario. On publishing the results, if the authors do not release the code and data used for the study, it leaves any reader hoping to reproduce the results facing a steep time investment. Not surprisingly, studies attempting to reproduce previously published findings are rare. As we have illustrated with our campaign to achieve full replication of our own previous study, there are severe pitfalls and challenges in fluid-flow simulations under unsteady, highly vortical regimes. It is a distinct possibility that many published studies report wrong results. As noted by Leek and Peng (2015), increasing the level of reproducibility of published studies will help uncover flawed research findings. For this reason, the minimum level of reproducibility—making code and data available—is essential for increasing the confidence on any new scientific claims to knowledge generated computationally. Going beyond sharing code and data, full automation and digital recording of experimental campaigns offer the best guarantee of being able to extract scientific value from computational experiments.

Key tools

We use the version-control hosting platform GitHub to support our reproducible workflow. GitHub greatly facilitates collaboration when developing numerical codes and documentation. The platform also allows creating wiki pages for users' documentation. We use GitHub to write manuscripts, to record our group-meetings, and to store teaching materials. We also extensively use Python to automate analysis and post-processing. Progress reports and summaries for discussion in group meetings are best presented using Jupyter notebooks, where textual media is combined with code and visualizations. For a digital record of all steps taken in preparing a simulation and running it, bash scripting is essential. We also use Travis CI for running automated testing of the codes whenever a change is to be merged into the main repository.

Questions

What does "reproducibility" mean to you?

The starting point for our understanding of reproducibility is contained in the pledge "Reproducibility PI Manifesto" (Barba, 2012) which includes these steps:

1. teaching group members about reproducibility;
2. maintaining all code and writing under version-control;
3. carrying out verification and validation and publishing the results;
4. for main results in a publication, sharing data, plotting scripts, and figures under CC-BY;
5. uploading preprints to arXiv at the time of submission of a paper;
6. releasing code no later than the time of submission of a paper;
7. adding a "Reproducibility" statement to each publication;
8. keeping an up-to-date web presence.

Some of these items have to do with making our research materials and methods open access and discoverable. The core of this pledge is releasing the code, the data, and the analysis/visualization scripts. Already this can be time consuming and demanding. Yet, we have come to consider these steps the most basic level of reproducible research. On undertaking a full replication study of a previous publication by our research group, we came to realize how much more rigor is required to achieve this, in the context of computational fluid dynamics of unsteady flows. We use the term "full replication" in the sense presented by Peng (2011), that is, completing an independent study using new methods to collect new data, arriving in the end at the same scientific findings. In computational fluid dynamics, full

replication of the findings can involve using a different code that implements the same numerical method, or a code that implements a different numerical method altogether but solves the same mathematical model. Because we are solving the Navier-Stokes equations—an unsteady and nonlinear model—certain problem scenarios can present particular challenges to replication.

Why do you think that reproducibility in your domain is important?

In computational science, we use simulations and data analysis as tools for the creation and justification of scientific knowledge. This process of knowledge creation, as in all science, must also produce evidence to justify itself. Reproducibility is a way to provide grounds for trusting the scientific findings obtained computationally. Ensuring that a publication (along with the data used to generate the figures) is reproducible makes it easier for others to corroborate (or reject) a scientific hypothesis. Codes and data used to publish results should be version-controlled and open-source to facilitate reproducibility. Donoho and co-authors (2009) mentioned that we develop codes so that they can be used again by strangers and defined strangers as "anyone who doesn't possess our current short-term memory" (including ourselves in some years). We believe that reproducible research can also prevent scientists from "reinventing the wheel" by having to re-create complete software stacks to build from previously published work.

How or where did you learn about reproducibility?

The group's PI, Prof. Lorena Barba, plays an active role in raising awareness about reproducible research. Incoming students joining our research lab must start by learning the different tools mentioned in the "Reproducibility PI Manifesto". The [Software Carpentry Foundation](#) (through workshops and online resources) also contributes to educate our group members and improve our workflow to achieve reproducible research.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

Reproducible research can be time-consuming, requiring rigorous methods and organization. At various moments during the project, we had to pause and ask ourselves if our research was currently reproducible. Often, this was prompted by a conversation or questioning during group meetings. In that sense, a strong collaborative culture in the

research group, and beyond in the wider community of the discipline, are vital to instill reproducibility practices in computational researchers. Lack of systematic and widespread educational programs that emphasize reproducible research is a serious obstacle.

What do you view as the major incentives for doing reproducible research?

Making your research more reproducible—e.g., providing reproducibility packages along with the manuscript—is a way of showcasing your skills, a medium for communicating research more transparently, and an invitation to give feedback on your work. If the research community is inclined to put more effort in doing reproducible research, it would prevent scientists from reinventing the wheel by rewriting software in order to build from your work. In the long run, it saves resources to achieve scientific knowledge growth, both at the level of a community and within a research group.

Are there any best practices that you'd recommend for researchers in your field?

Again, we insist that automating all the computational workflow and diligently maintaining a lab notebook are fundamental to record your research. We try to avoid GUIs as much as possible and prefer to script everything so that analysis can be automated, reproducible, and recorded. This may be time-consuming but surely beneficial in the longer term of a research project.

Would you recommend any specific resources for learning more about reproducibility?

- Barba, L. A. (13 December 2012). "Reproducibility PI Manifesto", 10.6084/m9.figshare.104539. Presentation for a talk given at the ICERM workshop "Reproducibility in Computational and Experimental Mathematics". Published on figshare under CC-BY.
- Donoho, D. L., Maleki, A., Rahman, I. U., Shahram, M., & Stodden, V. (2009). Reproducible research in computational harmonic analysis. Computing in Science & Engineering, 11(1), 8-18.
- Leek, J. T., & Peng, R. D. (2015). Opinion: Reproducible research can still be wrong: Adopting a prevention approach. Proceedings of the National Academy of Sciences, 112(6), 1645-1646.
- Madeyski, L., & Kitchenham, B. A. (2015). Reproducible Research—What, Why and How. Wroclaw University of Technology, PRE W, 8.

- Peng, R. D. (2011). Reproducible research in computational science. *Science* (New York, Ny), 334(6060), 1226.
- [Reproducible Research -- Coursera MOOC](#).
- Sandve, G. K., Nekrutenko, A., Taylor, J., & Hovig, E. (2013). Ten simple rules for reproducible computational research.
- [Software Carpentry](#).
- Software Testing -- [Udacity MOOC](#).
- Stark, P. B. (2015). Science is "show me", not "trust me". [Blog post](#)
- Vitek, J., & Kalibera, T. (2011, October). Repeatability, reproducibility, and rigor in systems research. In Proceedings of the ninth ACM international conference on Embedded software (pp. 33-38). ACM.

References

- Bhalla, A. P. S., Bale, R., Griffith, B. E., & Patankar, N. A. (2013). A unified mathematical framework and an adaptive numerical method for fluid–structure interaction with rigid, deforming, and elastic bodies. *Journal of Computational Physics*, 250, 446–476.
- Krishnan, A., Socha, J. J., Vlachos, P. P., & Barba, L. A. (2014). Lift and wakes of flying snakes. *Physics of Fluids*, 26(3), 031901.
- Taira, K., & Colonius, T. (2007). The immersed boundary method: A projection approach. *Journal of Computational Physics*, 225(2), 2118–2137.

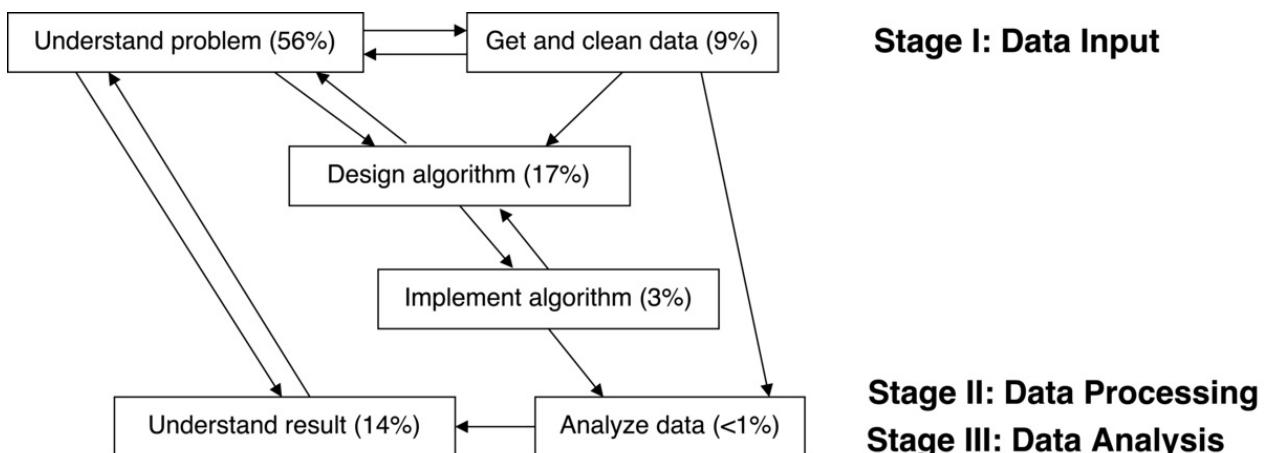
Reproducible Applied Statistics: Is Tagging of Therapist-Patient Interactions Reliable?

K. Jarrod Millman, Kellie Ottoboni, Naomi A. P. Stark and Philip B. Stark

We are three applied statisticians (JM, KO, PS) at UC Berkeley working with a domain specialist (NS) at the University of Pennsylvania. Our case study involves assessing inter-rater reliability (IRR) of the assignment of “tags” applied by human raters to classify interactions during therapy sessions with children on the autistic spectrum.

An extended version of this case study along with the analysis script and results can be found at <https://github.com/statlab/nsgk>.

Workflow



Our project arose from a pilot study NS was working on with Dr. Gilbert Kliman of the Children’s Psychological Health Center in San Francisco. To investigate therapeutic interventions with children on the autistic spectrum, Dr. Kliman and NS collected some observational data (described below). NS approached PS about the data and the problem NS was studying. After investigating the problem further, PS emailed JM and KO a one-page proposal for a stratified permutation test to assess inter-rater reliability using stratified samples. We (JM, KO, PS) had recently begun developing a general purpose Python package for permutation tests, called `permute`, based on our collaborations. PS suggested this would be an interesting example to include.

After coming to an initial understanding of NS's underlying research question and experiment, including how she collected the data, we (JM, KO, PS) cleaned the data, developed a nonparametric approach to assessing IRR appropriate to the experiment, implemented the approach in Python, incorporated the resulting code into our evolving Python package of permutation tests, applied the approach to the cleaned data, documented the code and the analysis, and wrote up the results in LaTeX.

We distinguish the following aspects of our project, which are typical in applied statistics:

- understand problem
- get and clean data
- design algorithm
- implement algorithm
- analyze data
- understand result

Figure 1 shows how each aspect of the project influenced the other aspects and gives estimates of the total person-hours we collectively spent on each aspect of the project. For example, if JM, KO, and PS spent an hour together discussing the problem in a meeting, then that meeting counts as 3 people hours.

We did not keep a detailed record of time spent, but our computational practices provide enough detail about who did what when that we believe our estimates provide an accurate qualitative account of the time demands for each aspect of the project. However, since these are only rough estimates, the reader should focus on the relative differences in the amount of time we spent on each aspect. We have found that researchers (ourselves included) often underestimate the time needed to understand the problem, acquire and clean the data, as well as understand the results, while overestimating the time needed for writing code. We have included our time estimates to give people an idea how “inexpensive” (or expensive) working more reproducibly is, to capture how our group understanding evolved, and in the hope that it might be instructive for students and collaborators.

Since we view computational reproducibility as a cross-cutting concern of all project aspects, we have adopted a set of computational practices, which we (JM, KO, PS) followed (almost) whenever we were working on the project. Exceptions include that we did not record all of our in-person discussions or whiteboard work. However, we endeavored to record summaries of these activities. These computational practices, described in Millman & Pérez (2014), are used widely in the open source scientific Python community. While developed for managing software contributions, these practices are ideal for ensuring computational reproducibility in scientific and statistical research. We will illustrate how we leverage the

software infrastructure and development practices of `permute` to conduct reproducible and collaborative applied statistics research with our colleagues. We discuss the software tools and practices briefly in Key tools and practices below.

Understand problem (80 hours)

The Kliman-Stark research project sought to identify characteristics of effective clinical interactions with children on the autistic spectrum. The project first required developing a set of characteristics that observers could use to “tag” what was happening in each 30-second interval of a therapy session. After they developed a taxonomy of clinical interactions, Kliman and NS had a number of trained raters watch videos of therapy sessions and label each 30-second interval using the collection of tags. For the classification system to be meaningful and useful, different raters must generally agree on whether a given tag applies to a given video segment: there must be inter-rater reliability. Of course, if a tag is never used or is always used, inter-rater reliability will be perfect, but the tag is useless for distinguishing clinical interactions.

That led to a statistical question: how to assess the evidence in the tagged videos that different raters tag interactions the same way? After numerous conversations, it made sense to consider the null hypothesis to be that, conditional on the number of times a given rater applied a given label to a given video, all assignments of that label to time stamps in the video by that rater are equally likely, and the ratings given by different raters are exchangeable (essentially, that raters are independent).

Once PS had an initial understanding of NS’s problem, we (JM, KO, PS) met regularly (approximately weekly, sometimes more) as a team to discuss the project. Initially these discussions involved a lot of work on whiteboards and asking a lot of probing questions. This helped us develop a shared understanding of the problem, understanding that improved by explaining things to one another and by asking hard questions about our planned approach and whether it could address the question of interest. As our understanding of the problem progressed, our work transitioned from working on whiteboards to testing our ideas out on a computer. We often used pair programming at this stage and sometimes we all sat in front of one computer, while one of us typed code in an interactive IPython session. This helped ensure that we all understood the problem well and it also helped us catch errors (typos as well as conceptual misunderstandings).

Get and clean data (13 hours)

The tag data were collected by NS and raters working at her direction. The data comprise ratings of segments of 8 videos by 10 trained raters. Each video is divided into approximately 40 time segments. In each time segment, none, any, or all of 183 types of

activity might be taking place. The raters indicated which of those activities was taking place during each segment of each video.

PS received the data from NS as an Excel spreadsheet that had been entered by hand by NS and an assistant. Understanding the “data dictionary” and vetting for obvious errors entailed several rounds of email between PS and NS before PS had a version of the data that did not have obvious errors. PS then exported the Excel data to comma-separated value (CSV) format. The original data contained personally identifying information. Using regular expressions in an interactive text editor, PS substituted unique numerical identifiers for raters’ names in the CSV file. While this step was not performed reproducibly (i.e., not scripted), it can be checked readily. After PS generated the original anonymized data, JM committed it to our repository and added a data loader with tests to ensure that if the data changed we would know. At this point, we (JM, PS) screened the anonymized data for transcription errors (e.g., duplicate rows). This involved writing a number of quality assurance tools (e.g., to find duplicate consecutive rows), which are now included in `permute`. Once we identified entries incompatible with our understanding of what should be in the data, JM wrote a `sed` script to “correct” the inferred typos. The exact commands used to clean the data are included in the commit corresponding to that cleaning step. After carefully examining the data for potential errors and documenting every change we made and why, we sent the cleaned data and an explanation of what we did to NS to verify that the corrections were appropriate. As a result, we provide the cleaned data in our project repository as well as a careful account of its provenance.

Design algorithm (25 hours)

Although the test we eventually implemented was very similar to the original test proposed by PS at the start of the project, we (JM, KO, PS) spent significant time focused on “problem appreciation,” some of which resulted in considerable simplification of the algorithm used to implement the test. We also developed a more general terminology (see Table 1).

Mapping between terms from our motivating problem (NSGK) and the terms used in our general algorithm (IRR).

NSGK	IRR
183 types of activity	T tags
8 videos	S strata
40 segments/videos	N_S items/strata
10 raters	R raters

We decided to assess rater reliability in identifying (i.e., tagging) each of the 183 types of activity separately, because they are of separate interest. This introduces questions about whether inferences are to be made about each tag separately (per-comparison error rate, PCER) or simultaneously (familywise error rate, FWER), or whether we are concerned with the fraction of tags we conclude are reliable that in fact are not reliable (false discovery rate, FDR). Ultimately, we decided that the PCER was the most relevant error criterion, since the tags are individually interesting. As a “first cut” through the rating scheme, eliminating tags that are clearly not reliable across raters simplifies the scheme and reduces the cognitive burden on raters, because they do not have to keep so many categories of activity in mind. We imagined that if we could eliminate a substantial number of the tags as unreliable, there would be a repeat of the tagging using a different set of raters to validate or refine the results, reducing the rate of “false positives.” On the other hand, incorrectly rejecting tags as unreliable could eliminate a potentially useful predictor of successful therapeutic outcomes, so the FWER seemed far too stringent a criterion. See the Understand result section below for more discussion.

Since each of the videos contained different sessions of therapist-patient interactions, in general rated by different people, we stratified the test by video. A literature search for approaches to assessing IRR led us to conclude that there was no existing suitable method for several reasons: the experiment was stratified; there were multiple raters but not the same set for all videos; and standard methods required indefensible parametric assumptions or population models, which we hoped to avoid. After deciding to use permutation tests, we (JM, KO, PS) then determined that permuting each rater’s ratings within a video, independently across raters and across videos, made sense as the appropriate invariant under the null hypothesis. We chose to use concordance of ratings as our partial test statistic within each stratum. We (JM, PS) derived a simple expression for efficiently computing the concordance. To combine tests across strata, we (JM, KO, PS) used the nonparametric combination (NPC) of tests (Pesarin & Salmaso, 2010) with Fisher’s combining function. Finally, we developed a computationally efficient approach to finding the overall p -value for the NPC test.

Implement algorithm (5 hours)

Once we had a blueprint of the algorithm, KO led the implementation effort. She did most of the coding; JM and PS reviewed the code and discussed the implementation. Following our software development practices, KO also wrote tests for every function she implemented. After a few iterations of coding, testing, and review, KO finalized our implementation and we merged it into `permute`.

KO wrote three functions to implement our general IRR algorithm:

1. a function to compute the IRR partial test statistic from a binary matrix with one row per rater and one column per item;
2. a function to simulate the permutation distribution of the IRR partial test statistic for a matrix of ratings of a single stratum;
3. a function to simulate the permutation distribution of the NPC test statistic by combining the S distributions of the IRR partial test statistic for each of the S strata.

Analyze data (1 hour)

Once we merged KO’s implementation of the general algorithm (including tests) into `permute`, KO wrote a short script (about 50 lines of Python) to analyze the cleaned data from NS.

Since we included the main workhorse functions in `permute`, the analysis script contained only high-level commands:

1. Load the cleaned data
2. For each of the 183 categories of activity:
 - i. For each of the 8 videos:
 - i. Compute the mean and standard deviation of the number of times the tag was applied
 - ii. Compute the IRR partial test statistic
 - ii. Simulate the permutation distribution of the NPC test statistic for each tag combined over the 8 videos, and report a single *p*-value
3. Save the results to a CSV file

Understand result (20 hours)

At a high level, even the summary statistics we computed were useful: some tags were never applied by any rater to any video. Presumably, the tag taxonomy could be simplified by eliminating those tags from the universe of labels, reducing the cognitive burden on the human raters. There were also tags that were used so frequently that high concordance was virtually guaranteed—and therefore high inter-rater concordance was not evidence of inter-rater reliability. This may also imply that any differences in efficacy of therapy are not attributable to whether the corresponding activity is taking place, since it is often taking place, at least in these sessions. Whether it makes sense to keep such tags in the taxonomy

depends in part on subject matter knowledge: are those interactions typical only in the videos in these evaluation data, or are they typical of all therapeutic interventions with children on the autistic spectrum?

At the other extreme, there were tags for which the concordance of use was quite low, but still highly significant. This raises the scientific question of what threshold level of agreement among raters makes a tag interesting or useful, separate from whether the agreement is statistically significant. That is a matter we need to discuss at greater length with the domain specialists. It also points to a frequent situation in statistics: practical significance and statistical significance are not the same thing, and one must consider “fitness for use” when devising summary statistics.

We hope that the concrete findings will lead to a refinement of the taxonomy and additional tests of reliability. We hope that those tests will involve greater automation of data collection and transcription, to eliminate some of the sources of error in the data. Regardless, this work has led to a new nonparametric test for inter-rater reliability, now available publicly in the `permute` package.

Pain points

Given our different backgrounds and experiences we (JM, KO, PS) each found different points in the process challenging. However, for all of us the most challenging aspect -- and the most time-consuming -- was the necessary struggle to understand the scientific question and the experiment well enough to devise an approach to answering the question.

For KO and PS there was a learning curve to master the tools and practices. This involved understanding the data model used by git, acquiring habits such as writing tests for all functions and following a common style guide, and learning to contribute to the project repository indirectly through GitHub’s pull request mechanism. JM was already familiar with the tools and practices, and devoted significant time to teaching KO and PS the workflow. Once mastered, the benefits of these tools and habits outweigh the time and effort spent learning them.

For JM the most painful part of the project was vetting hand-entered data to look for errors and inconsistencies. Not only was this laborious, but it involved inferring what the data should have been without any direct way to ensure that these inferences were correct: the original raters and videos were not available to us. The solution to this pain point is to automate data collection as much as possible. However, when data have already been entered by hand, there is not much that can be done other than being cautious when “fixing” data entry errors and recording every aspect of the data cleaning process.

Key benefits

Since Buckheit & Donoho (1995) popularized the idea of computational reproducibility, applied statisticians have increasingly embraced version control and process automation. Many of our colleagues have made the idea of computational reproducibility central in both the classroom and the lab. Some ask anyone working with them to follow a set of computational practices including version control.

However, the computational practices described in this study (see Key tools and practices) go beyond the standard work habits of our colleagues. Our computational practices provide the following benefits:

1. it reduces the number of errors introduced by new code and changes to existing code
2. it makes it easy to modify the analysis when errors are found, to apply the analysis to new datasets, and so on
3. the process is self-documenting, making it easier to draft a paper about the results or to pick up where we left off after working on something else
4. the methods are abstracted from the analysis and incorporated into a package so that others can discover, check, use, and extend our methods.

Key tools and practices

As part of the development of our software package `permute`, we invested significant effort in setting up a development infrastructure to ensure our work is tracked, thoroughly and continually tested, and incrementally improved and documented. To this end, we have adopted best practices for software development used by successful open source projects (Millman & Pérez, 2014).

Version control and code review

We (JM, KO, PS) use git as our version control system (VCS) and GitHub as the public hosting service for our official `upstream` repository [statlab/permute](#). Each of us has our own copy, or fork, of the `upstream` repository. We each work on our own repositories and use the `upstream` repository as our coordination or integration repository.

This allows us to track and manage how our code changes over time and to review new functionality before merging it into the `upstream` repository. To get new code or text integrated in the `upstream` repository, we use GitHub's *pull request* mechanism. This enables us to review code and text before integrating it. Below, we describe how we automate testing our code to generate reports for all pull requests. This way we can reduce the risk that changes to our code break existing functionality. Once a pull request is reviewed and accepted, it is merged into the `upstream` repository.

Requiring all new code to undergo review provides several benefits. Code review increases the quality and consistency of our code. It helps maintain a high level of test coverage (see below). Moreover, it also helps keep the development team aware of the work other team members are doing. While we are currently a small team and we meet regularly, having the code review system in place will make it easier for new people to contribute as well as capturing our design discussions and decisions for future reference.

Testing and continuous integration

We used the `nose` testing framework for automating our testing procedures. This is the standard testing framework used by the core packages in the scientific Python ecosystem. Automating testing allows us to monitor a proxy for code correctness when making changes as well as simplifying the code review process for new code. Without automated testing, we would have to manually test all the code every time a change is proposed. The `nose` testing framework simplifies test creation, discovery, and execution. It has an extensive set of plugins to add functionality for coverage reporting, test annotation, profiling, as well as inspecting and testing documentation.

Our goal is to test every line of code. For example, not only do we want to test every function in our package, but if a specific function has a conditional branching structure we test each possible execution path through that function. Having tested each line of code increases our confidence in our code and provides some assurance that changes we make do not break existing code. It also increases our confidence that new code works, which reduces the friction of accepting contributions. Currently over 98% of the lines of code in `permute` get executed at least once by our test system.

We often work on several pull requests simultaneously. These pull requests may take several weeks or months before they are reviewed, improved, and accepted in our `upstream` repository. While we are working on one pull request, we may merge several others. Since the underlying code base is changing, each pull request may potentially introduce integration conflicts when we attempt to merge it back into the main line. To mitigate the difficulty in managing these conflicts we employ continuous integration and track our test coverage.

Continuous integration works as follows: Each pull request (as well as a new commit to an existing pull request) triggers an automated system to run the full test suite on the updated code. Specifically, we have configured [Travis CI](#) and `coveralls` to be automatically triggered whenever a commit is made to a pull request or the `upstream` master. These systems run the full test suite using different versions of our dependencies (e.g., Python 2.7 and 3.4) every time a new commit is made to a repository or a pull is requested. Travis CI checks that all the tests pass, while `coveralls` generates a test coverage report so that we can monitor what parts of our code are checked by a test and which are not. This system

checks whether any of our automated tests fail as well as tracks the percentage of our code that is covered by our automated tests. This means that when you review a pull request, you can immediately see whether the proposed changes break any tests and whether the new code decreases the overall test coverage.

Documentation

We use Sphinx as our documentation system and have extensive developer documentation and the foundation for high-quality user documentation. Sphinx is the standard documentation system for Python and is used by the core scientific Python packages. We use Python docstrings and follow the [NumPy docstring standard](#) to document all the modules and functions in `permute`. Using Sphinx and some NumPy extensions, we have a system for autogenerated the project documentation (as HTML or PDF) using the docstrings as well as stand-alone text written in a light-weight markdown-like language, called [reStructuredText](#). This system enables us to easily embed references, figures, code that is auto-run during documentation generation, as well as mathematics using LaTeX.

Release management

Our development workflow ensures that the official `upstream` repository is always stable and ready for use. This means anyone can install our official upstream master at any time and start using it. We also make official releases available as source tarballs and as Python built-packages uploaded to the Python Package Index, or PyPI, with release announcements posted to our mailing list.

By making official releases whenever we reach an important stage of an applied project, we are able to easily recover the exact version of our analysis at a later date. To install the exact version of `permute` used in this case study, type the following command from a shell prompt (assuming you have Python and a recent version of `pip`):

```
$ pip install permute==0.1a2
```

Questions

What does "reproducibility" mean to you?

In this case study, *reproducibility* means:

- *Computational reproducibility and transparency.* We have documented (and scripted) nearly every step of the analysis—from cleaning to coding to code execution—and made the code and documentation publicly available.

- *Scientific reproducibility and transparency.* We documented much of the discussion leading to our decisions to take each step in the analysis. We made the data publicly available in an open format, with an adequate data dictionary.
- *Computational correctness and evidence.* We tested our code thoroughly and in an automated fashion, to have justifiable confidence that the code does what it was intended to do. We made those tests publicly available, so that others can see how we validated our computations.
- *Statistical reproducibility.* We invested time to understand the fundamental problem and the results of our analysis so that we do *not* draw conclusions that are not justified by the data, the manner in which it was acquired, and our domain understanding. We avoided “p-hacking” and other potentially misleading selective reporting, and made all our analyses publicly available.

By keeping all code, text, and data in a public version-controlled repository, we have made our well-documented analysis available for anyone to examine, check, modify, or reuse. We published the data used in our study -- both the original anonymized version as well as our cleaned version including the commands necessary to produce the cleaned version from the anonymized one. In addition to making what we did transparent to anyone who is interested, working in this way means that when errors are found we can identify how and when those errors were introduced. We have written tests for almost all our code, which means we have a high level of confidence that as we change our code we will catch any errors we might have introduced, and can correct them quickly and easily. And since we have automated the process of running our analysis, if errors are identified and corrected, it is easy to rerun the entire analysis from start to finish.

If you have standard tools on your computer and network access, you can run our complete analysis of the cleaned data by typing the following three commands from a Unix shell prompt:

```
$ git clone git@github.com:statlab/nsgk.git  
$ cd nsgk/nsgk  
$ make
```

The first command creates a directory `nsgk` in your current working directory with a copy of the project repository (i.e., a directory with our code, data, and text along with the provenance of these documents). This directory contains this document as well as everything needed to run our analysis. Inside `nsgk/nsgk` there is a `Makefile`, our analysis script `analysis.py`, and the output `results.csv` of that script.

When you enter the command `make`, the following commands will be run:

```
virtualenv -p /usr/bin/python2.7 venv
venv/bin/pip install --upgrade pip
venv/bin/pip install -r requirements.txt
venv/bin/python analysis.py
```

The first command creates a new virtual environment (`venv`) for Python 2.7. Using this new virtual environment (`venv`) the subsequent commands respectively upgrade the Python package manager (`pip`) to the most recent version, install the necessary Python package dependencies (`numpy 1.11.0`, `scipy 0.17.0`, and `permute 0.1a2`), and run the analysis script `analysis.py`.

References

- Buckheit, J. B., & Donoho, D. L. (1995). Wavelab and reproducible research. In A. Antoniadis & G. Oppenheim (Eds.), *Wavelets and statistics*. Springer.
- Millman, K. J., & Pérez, F. (2014). Developing open-source scientific practice. In V. Stodden, F. Leisch, & R. D. Peng (Eds.), *Implementing reproducible research* (pp. 149–183). Chapman; Hall/CRC.
- Pesarin, F., & Salmaso, L. (2010). *Permutation tests for complex data: Theory, applications and software*. John Wiley & Sons.

A Dissection of Computational Methods Used in a Biogeographic Study

K. A. S. Mislan

My name is Allison Smith, I am an ecophysicist and my research focuses on organism-environment interactions in the ocean. In particular, I am interested in forecasting the effects of climate change on marine ecosystems.

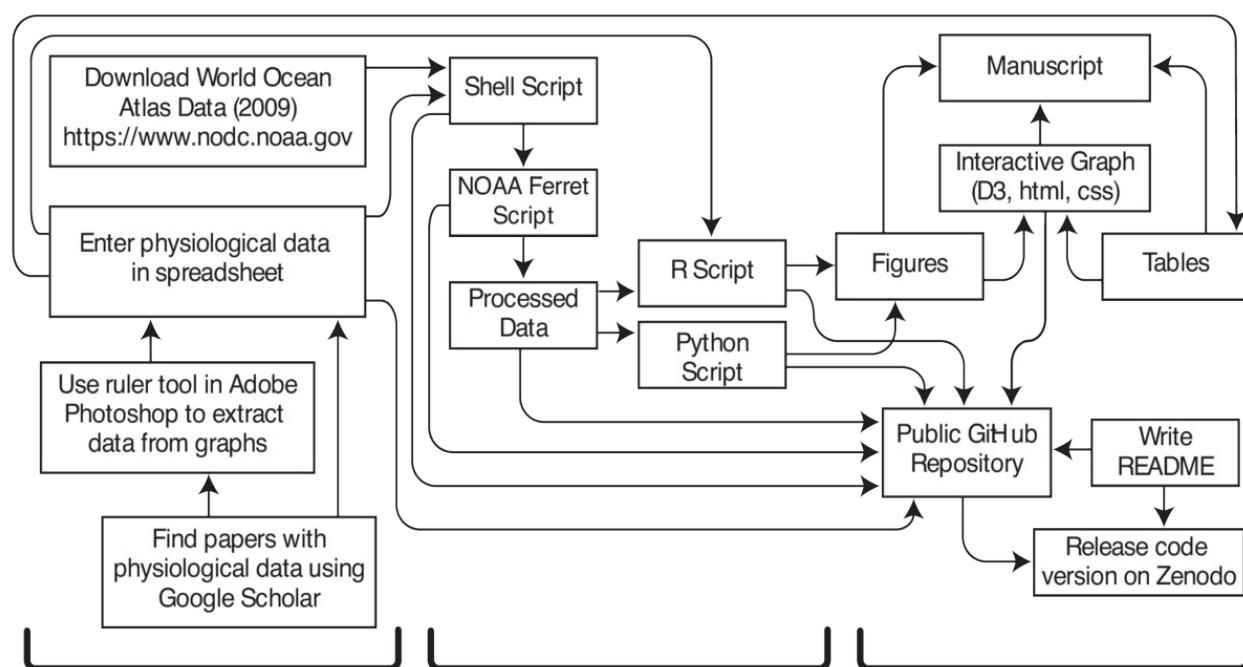
I recently published research on the fundamental niche of pelagic animals in the global ocean. The study included a comparison of blood-oxygen binding characteristics of different species obtained from published papers. Thresholds for blood-oxygen binding characteristics were mapped in the ocean using gridded oceanographic data. My workflow details my process for obtaining and analyzing data for the project. In order to increase the reproducibility of the study, code used for the project was put in a long-term archive.

Research paper: Mislan, K. A. S., Dunne, J. P. and Sarmiento, J. L. (2016), The fundamental niche of blood-oxygen binding in the pelagic ocean. *Oikos*.

<https://doi.org/10.1111/oik.02650>

Code archive: Mislan, K. A. S., Dunne, J. P. and Sarmiento, J. L. (2015). P50 Depth Analysis v1.0. Zenodo. <http://dx.doi.org/10.5281/zenodo.31951>

Workflow

**Stage I: Data Input****Stage II: Data Processing****Stage III: Data Analysis**

Obtaining data from existing resources was the first step. World Ocean Atlas 2009 (WOA09) data is publicly available through the National Oceanic and Atmospheric Administration (NOAA) National Centers for Environmental Information. The WOA09 data is on a geographic grid and available in several different file formats. I downloaded the network common data form (netCDF) file format. The NetCDF file format facilitates access and sharing of scientific data in arrays. There are many tools available to read and manipulate netCDF files. NOAA Ferret is publicly available software for visualization and analysis and has built-in functions designed for efficient processing of geographic data in netCDF file format. One of the objectives of the analysis was to vary two input parameters and determine the effect on geographic characteristics. I automated the process of creating NOAA Ferret scripts with different input parameters by writing a shell script that generated and processed NOAA Ferret scripts (.jnl files). The processed files were saved as netCDF files.

The physiological data were gleaned from published studies found through Google Scholar searches for key words. The data were extracted from the papers and put into a spreadsheet. The data of interest were often available in tables and the text of the papers. In some cases, the data were only available in scatter plots. The ruler tool in Adobe Photoshop was used to manually extract the data from plots. Once all the information from the studies was entered into the spreadsheet, the data from the spreadsheet were saved as a tab-delimited text file. The data were then read into R to determine parameters for the analysis of the oceanographic data. Additionally, the data were plotted in a scatter plot in R and each point in the plot had a number assigned to it. Information about the individual points was put into two tables. However, the numbers on the plot had to be matched to numbers in the two tables to get relevant information from the plot, which was inefficient. Therefore, I created a web-based interactive graph that embeds information from the tables into the scatter plot

using a javascript library called Data-Driven Documents (D3). In the interactive graph, the information for each point is visible when the cursor is placed on the point. The interactive graph also includes options to select different legends for the graph that highlight additional groupings for the points. A link to the interactive graph was included in the publication.

R and Python were used to create the figures although it would have been possible to create all the figures in one or the other of these software packages. I learned R before Python so I tend to do most of my analysis in R because I am most familiar with it. However, I like the tools for making geographic plots in Python so I used Python to make the geographic plots. Then I wrote the paper.

Code is not usually included in the methods section of a paper due to space and formatting constraints. However, the code tells a much more complete story of my analyses. In order to prepare my code for archiving, I made some modifications. I created a folder structure that would make it easy for a user to find files. The folder structure included a folder for code, folders for input files to run the code, output files produced by the code, and graphs produced by the code. I annotated my code while I was writing it, but the annotations were usually short and meaningful only to me. Descriptive annotations were added to the code being archived. I also changed the file paths to be referenced solely within the folder structure (../../) so that the code would be independent of my home directories (/Users/kasmislan/code/project).

My code worked on my computer, but it might not work the same way on a different computer. For example, different operating systems have different methods for rounding numbers. I generated output files for my code on my computer and moved the files to the test files folder. Then I wrote a script that automatically compares output files produced by another user using my code on another operating system to the test files I produced using my code on my operating system. If there are differences, then a user will be able to identify and address them.

The most critical step for archiving my code was writing the documentation (README file). The documentation includes a description of the purpose of the code, references to research articles, a list of software dependencies including versions, clear step-by-step instructions on how to use the code, and clear step-by-step instructions on how to use the test files. I also included a section to acknowledge my funding sources and others who helped with the project.

The final step was to submit the code with the Massachusetts Institute of Technology (MIT) License to Zenodo, a long-term archive. The MIT License has few restrictions which maximizes the ways in which my code can be used and adapted by others. I sent the link to the code archive to the journal so that it could be included in the publication.

Pain points

My primary pain point is that archiving my code takes additional time after I am ready to submit a scientific paper. The most time-consuming steps are associated with making the code usable by someone else on their own computer. As I wrap-up a project, my files are cluttered because I have an exploratory phase as I am analyzing data where I write code that is not ultimately used for the results presented in the paper. I have to identify and organize the relevant code files. Then I have to modify filepaths, annotate the code, create test files, and write a README file with instructions for using the code which also takes additional time. Another pain point is trying to find someone to test the code to make sure that the instructions are comprehensible, and the code runs on other computers and operating systems without errors. This step requires another person to spend time to help me archive my code. In the current scientific research system, archived code is not valued as highly as scientific papers so the extra time spent by scientists and code testers to archive code does not directly translate into greater scientific success. Some academics have hypothesized that papers with archived code are cited more often, but this has not been universally verified. In my experience, my archived code is limited to a specific analysis and, while the availability of code may increase the confidence of the scientific community in my results, I do not think that my archived code is generating more citations of my scientific papers.

Key benefits

Reproducibility has always been an important component of research in my field. In the past, instructions for reproducing research were put in a methods section in journal articles. The increasing importance of code in my field is changing the way reproducibility is accomplished because it is not possible to include code in a methods section. However, it is necessary to have access to the code to reproduce the research.

Key tools

I have always believed that making my code available is important, but, until recently, I was not sure how to do it. GitHub is a "game-changer" for sharing code with others in a reliable, consistent, and discoverable way. After I was introduced to GitHub, I was able to start archiving my code. My code is posted to GitHub and a permanent copy and digital object identifier (doi) are generated by Zenodo.

There are specific instructions on GitHub for releasing code to Zenodo:

<https://guides.github.com/activities/citable-code/>

Questions

What does "reproducibility" mean to you?

Reproducibility means that sufficient descriptive information and resources are provided for someone to be able to repeat the same study. As part of my research as a scientist, I write code to manipulate and visualize large quantities of data to obtain results. I believe that my code should be adapted and made available so that it is usable by others in my field.

Why do you think that reproducibility in your domain is important?

Reproducibility has long been a central tenet of the scientific research process in my field because data from new studies are compared to data from earlier studies. Not so long ago, all the data collected during a study were included in published research articles, and the analyses could be easily described in a materials and methods section of the articles.

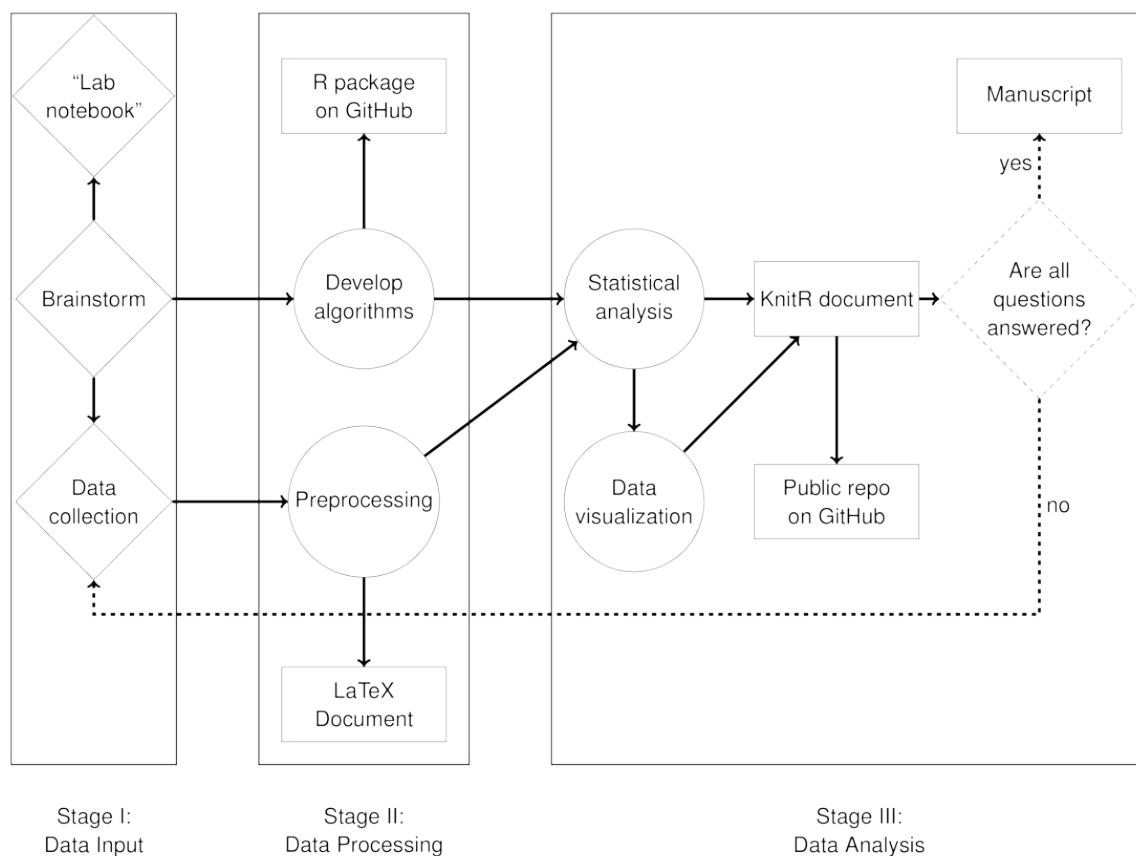
Recent increases in the quantity of data and the concurrent increase in the complexity of the analyses has made it impossible to include the data itself or the same level of descriptive detail in the materials and methods sections of research articles. I think that my domain values reproducibility but the current format for publishing research articles is not able to fully accommodate the modern scientific research process.

A Statistical Analysis of Salt and Mortality at the Level of Nations

Kellie Ottoboni

My name is Kellie Ottoboni. I'm currently a PhD student in the Department of Statistics at UC Berkeley and a Data Science Fellow at the Berkeley Institute for Data Science. My research focuses on nonparametric statistics, causal inference, and applications in health and social sciences. The project I describe in this case study is an investigation of the association between salt consumption and mortality at the level of nations.

Workflow



This project started as a collaboration between my advisor and a professor of public health at UC Irvine, along with one of his students. I became involved after our collaborators had begun putting together the dataset. The data consists of demographic and socioeconomic

variables, as well as gender-specific estimated sodium, alcohol, and tobacco consumption for 38 countries. The data were pulled together from several sources and the variables are described in a text file.

The first step was to decide what data we needed to answer the question: does salt consumption have an effect on a nation's life expectancy at age 30? We decided that the best way to address this would be to consider males and females separately, and to use a country as its own "control" by looking at the changes in life expectancy, alcohol, tobacco, and sodium consumption, and economic variables over time. We had the variables we needed, but gender-specific alcohol data were missing for one year. These couldn't be obtained so we imputed them based on gender ratios in each country and overall consumption that year. We also had to remove countries with missing data on life expectancy, the outcome of interest. In addition to the R code implementing these steps, I described them in a LaTeX report as I went along.

The method of analysis was a novel hypothesis test that I have been developing with my advisor. The premise of the method is to predict the outcome of interest, change in life expectancy, using all covariates *except* the treatment of interest, sodium. If sodium is still related to life expectancy after controlling for known health predictors, then sodium consumption will be associated with the residuals of the model. This is simply a mathematical fact: some of the variation that the model cannot explain will be associated with sodium. But how large of an association is statistically significant? We answered this question using nonparametric permutation tests. After discussing how our approach might answer the problem at hand, I wrote R code for the two main steps in the algorithm: the model selection and the nonparametric test of association. I had already written code for our proposed statistical method in R package format in a public GitHub repository, so I added the new code from this project into the package. I developed the code iteratively by running it on our dataset and checking that the output looked sensible. This isn't the best way to write code: ideally, I would have invented simple test cases and checked my functions against those.

After writing each component of R code, I combined all of the preprocessing, analysis, and plot scripts into an R Markdown file. knitr allows you to compile R Markdown, R code chunks interleaved with markdown text, into a PDF or HTML document. This way, I could send my collaborators the results quickly and in a user-friendly format. I posted all the scripts, data, and the compiled HTML document in a public GitHub repository dedicated to this project.

At this point, we were unsatisfied with the scope of the analysis and wanted to ask more questions. In particular, we wanted to run the same test of association but use tobacco and alcohol in place of sodium. These analyses would serve as a "sanity check" that the method performs as expected. Given that we know that both alcohol and smoking have negative effects on health, we would expect to find a negative association between the model's

residuals and alcohol and tobacco use. The original dataset included alcohol consumption per capita, but no measures of tobacco use. This required our collaborators at UC Irvine to gather this data from an existing journal article. After receiving the tobacco usage data from them and performing the model-based matching analysis with it, we realized that the measure of smoking that we used was not a good measure of a nation's overall tobacco consumption. We again gathered different tobacco data and ran the analysis one last time. Since the R scripts had already been written, redoing the analysis with each version of the data amounted to changing a few lines of code. I kept each version of the data used and named them according to the date when it was sent to me.

We believe that we have done all the statistical analysis that we need to answer the scientific questions that we posed. The analysis portion of the project took about six months to complete. Our collaborators are preparing a manuscript. We are using LaTeX and communicating by email.

The workflow diagram distinguishes three types of activities: thinking and planning steps are marked with diamonds, action and implementation are marked with circles, and documentation and outputs are marked in rectangles. While arrows point to the right and boxes separate the three stages of the workflow, this is a bit artificial. In my experience, there is a lot of iteration involved in applied statistics projects. In addition, the number of nodes in each stage is not reflective of the amount of time spent. The majority of my time was spent planning and documenting. It was quick to make minor changes once the preprocessing and analyses were scripted.

Pain points

A seemingly trivial problem I struggle with is keeping track of files. Using git certainly simplifies the version control aspect of file organization, but that doesn't help when I've forgotten where I put a chunk of code I wrote two weeks ago. Around the beginning of this project, I started keeping a "lab notebook" to organize my thoughts on all the various projects I'm doing. I keep a folder of text files just for myself where I jot down the date, ideas and concerns, notes on what work I've done, and the names of files or folders where I saved that work. It has helped tremendously when I need to remind myself things about a project, and it's also a nice way to archive meeting notes and save ideas that I might want to share with collaborators later on.

A pain point particular to this project was trying to encourage my collaborators to use the GitHub repository. Ultimately, we ended up sending data and results back and forth by email. Pushing updated data and results to the repository would have been more efficient.

The data collection part of the project was opaque. Our repository does not include any scripts used to collect the data from various databases and journal articles or scripts to merge these data sources. At one point, under pressure of a deadline, I manually entered tobacco consumption figures into an Excel spreadsheet. All the preprocessing and analyses of the data are reproducible, but the process of collecting the data is not.

Key benefits

The biggest advantage of a reproducible workflow is efficiency. There were many iterations of the analysis for this project. By having preprocessing, analyses, and results written in the same document, it was easy to make small changes and ensure that they appeared throughout the report.

Incorporating the main functions for this analysis into an R package with larger scope will be beneficial in the future. The functions I wrote for hypothesis testing here are well-documented and uniform in their style, inputs, and outputs. Having them all in a package on GitHub makes it easy for anybody who reads our paper to install the package and replicate the results.

Key tools

RStudio and knitr were key for this workflow. This was my first time using knitr and I am pleased with the quality of the reports I created to share the results with my colleagues. All steps of the data analysis are in the documents, alongside my commentary and explanation of the steps. I hope that this makes the statistical methods transparent to my collaborators and future readers. Additionally, having all the tables and figures in one place will make it easy to put results into the manuscript.

Questions

What does "reproducibility" mean to you?

I think that a data analysis project is reproducible if there are enough breadcrumbs (in the form of code and instructions) for anybody to recreate the analysis from start to finish. In another sense, a project is reproducible if someone can carry out a different analysis on the data and arrive at qualitatively similar conclusions.

Why do you think that reproducibility in your domain is important?

Researchers tend to blame the "reproducibility crisis" on statistics, and in particular p-values. It's our job as statisticians to fight this claim by making statistical analyses as correct and as transparent as possible, so people know exactly where their p-values are coming from.

How or where did you learn about reproducibility?

I learned some of these practices from other students in my department and the rest were self-taught using resources on the internet.

Are there any best practices that you'd recommend for researchers in your field?

Explain every step in the data preprocessing and analysis carefully and thoroughly. Document and comment code liberally. Make code and data publicly available.

Would you recommend any specific resources for learning more about reproducibility?

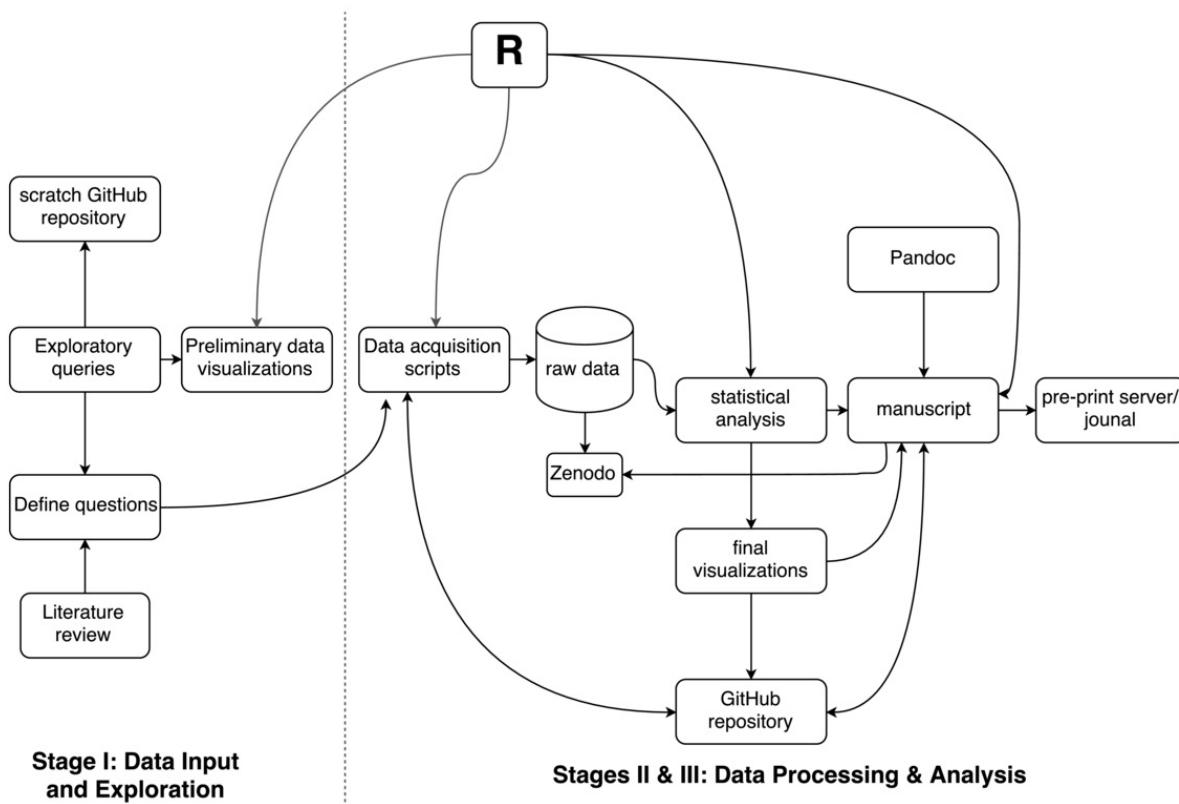
Hadley Wickham's *R Packages* book is an invaluable resource. He demystifies the R package and shows how to use RStudio to make the workflow smooth and efficient.

Reproducible Workflows For Understanding Large Scale Ecological Effects Of Climate Change

Karthik Ram

My name is [Karthik Ram](#) and I am a quantitative food web ecologist by training. I received my PhD in 2009 from the University of California, Davis. Since then I've studied the impacts of climate change on the rate at which the landscape greens up in spring time, and the consequent effects on a large mammal food web, among various other projects. The Yellowstone project was my entry into data science, where I was thrust into the challenges of validating (reproducing) someone else's work before continuing on with my own research. During this time I also encountered various pain points, which led me to create my own version control system before I learned of the existence of git. In my current job I've transitioned from a full-time research scientist to a hybrid role, where I spent part of my time on research activities but the rest on developing tools and workflows to support various stages of a reproducible research workflow which I describe below.

Workflow



Even though this narrative does not describe any specific project that I work on, it captures the general workflow I employ for all my projects. With all of my messy, raw data ready, I spend considerable time on exploratory data analysis during which I generate several visualizations. This process allows me get a sense for any early quality issues with the data and the kinds of steps I will need to employ to make the data usable for analysis. All of the code I use for this process (usually `R`), along with the outputs (rendered markdown and figures using packages such as `ggplot2` and `knitr/rmarkdown`) are committed to a scratch GitHub repository. This allows me to share early insights into my data with my collaborators. This process is highly iterative, and I generate various visualizations (across multiple branches) to gain a better insight into my data. This process takes me a few weeks as I multi task other projects. During this time, project collaborators and various others, including my Twitter followers, provide constructive feedback.

During this process I also document any data cleaning steps I'll need to undertake before I begin any data analysis steps. At this time I also deposit my raw, unprocessed data into a persistent repository such as figshare, or a Zenodo collection, and obtain a permanent identifier. figshare is a private company that provides free data archiving to individual users. Zenodo is an EU funded research archive that allows scientists to deposit various types of digital objects, including software and code.

I simultaneously start writing code to clean my data using a scripted workflow which also involves mostly R. I sometimes use a bash script or two to pre-process the data using old unix tools like `sed` and `awk` but with recent developments in the R toolkit (`data.table`,

`dplyr`, `tidyr`, and `rvest`) I rely less and less on my bash scripts. These scripts are called inside a Make file, which allows me to generate my cleaned datasets at any time with a simple command line call e.g. `make clean_data`. At this time I also start the process of creating a separate library for the project to capture the right versions of my tool dependencies, such that further updates to my computer don't affect the reproducibility of my work. In the case of `R`, I use a library called `packrat` to accomplish this. Once my data are cleaned, I deposit them back at the same identifier on the persistent data repository and include the DOI in the text of my paper.

The data analysis and modeling steps vary based on a project to project basis as some involve simulations on a cluster. For smaller projects, a handful of scripts accomplish this process. If any of my code is reusable in more than one step, I capture these into common functions, and sometimes convert this into a separate package. This allows me to further modularize my code. For projects that involve simulations on clusters, I create scripts that work on smaller examples for local testing, with full version that run on high performance clusters (HPC) and write out my results to disk.

Somewhere along the way, I also begin a Rmarkdown file, i.e. my manuscript, which is merely a markdown file with embedded R code captured inside code fences with some metadata. In addition to including small snippets of code, I am also able to source in larger chunks of code from my scripts without cluttering my document. The manuscript is also rendered from my Makefile frequently. I also include additional code to turn the `Rmarkdown` → `markdown` → `PDF` (using `Pandoc`), which gives me a sense for the how the final manuscript might look like. All of the code, figures, and raw/rendered markdown files are committed to my manuscript's GitHub repository. I have configured git to ignore large intermediate files that could easily be generated again in the future. As a researcher who practices open science, I leave the manuscript publicly accessible in my (or collaborators') GitHub repositories. GitHub now renders both the PDF, and also both the unparsed markdown (RMarkdown) and the markdown files on the browser, allowing anyone to review my work in progress.

For citations, I use the `knitcitations` R package to embed DOIs into my text, which are automatically rendered into full parsed citations and a bibliography during the Make process. For projects that cite content such as blog posts, I rely on Mendeley's bibliography rather than retrieving citations from Crossref using `knitcitations`.

Pain points

The two biggest pain points in my research are related to black box data and unscripted data processing steps. I frequently collaborate with researchers who process one or more chunks of data using proprietary, closed sourced software. Quite often, these steps are also not

scripted, requiring human intervention to update outputs as input data change. This combination of factors results in out of date versions of one or more pieces of input data simply because there was no automated way to determine what steps needed to be rerun.

In an ideal world, all my data would be a few simple queries away, allowing me to write concise scripts to retrieve the raw data before analysis. In reality, my data are a hodgepodge of manually entered data, sensor derived data (often bulk downloads after a mandatory sign in step), and other data retrieved via application programming interfaces (API). I try to alleviate the burden for anyone trying to reproduce my results by depositing all of my raw data and associated scripts into either institutional or other repositories so that others can replicate my research.

Key benefits

The key benefits to the approach I have outlined above are that anyone with the right technical skills can clone all of my code from a repository and re-run all the steps necessary to acquire the raw unprocessed data, munge the data, then run all associated statistical analysis and generate the full manuscript as published in a pre-print server or journal. Depending on the complexity of a paper, this could either be a single step, or a series of steps linked together in a Makefile. My work almost always include instructions in a `README` file.

Key tools

I've outlined my major tools inline but briefly: Programming tools: R, Make, git, Pandoc Services: GitHub, Zenodo, figshare, Mendeley

Questions

What does "reproducibility" mean to you?

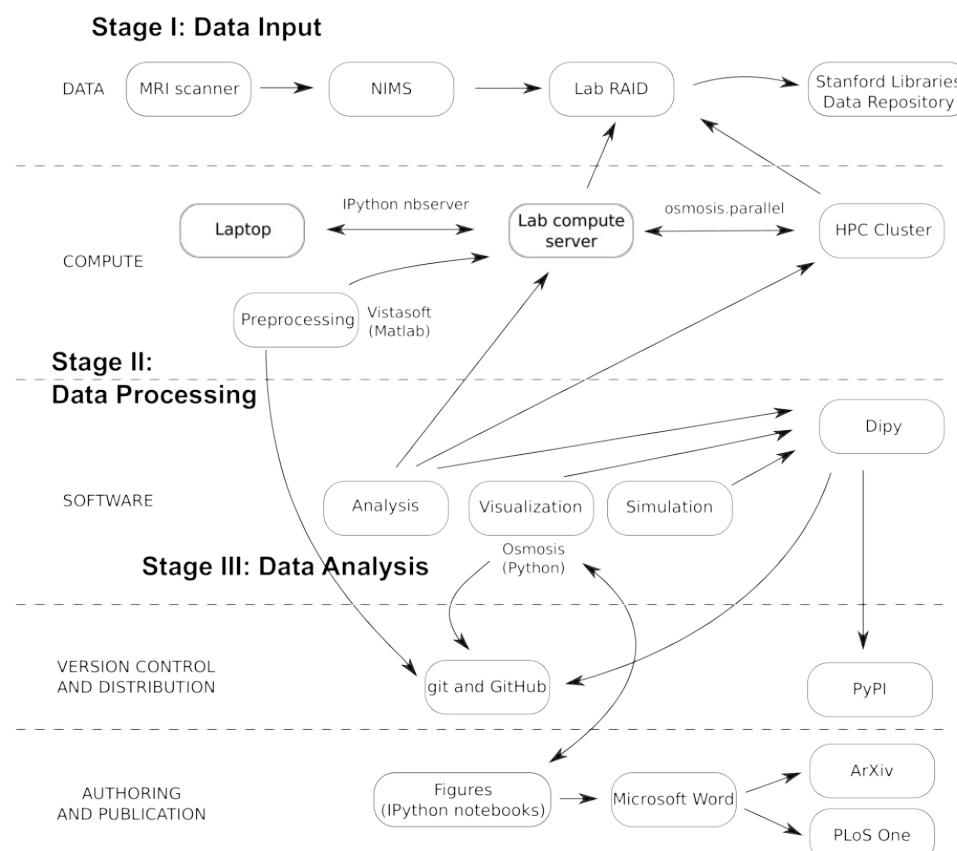
In the broadest sense, reproducibility means that I would be able to read a paper in my area of expertise and be able to run another version of the study (and experiments described within) by following the methods and protocols described within that paper. In my specific context, clear instructions would allow me to implement an experimental design/setup using identical organisms and chemical reagents. After the experiment is completed and the data are entered, I would then be able to analyze my data using models and parameters, possibly even reusing code where such methods were implemented. This would allow me to obtain results and compare them to the original.

Reproducibility in Human Neuroimaging Research: A Practical Example from the Analysis of Diffusion MRI

Ariel Rokem

My name is Ariel Rokem. I am a Data Scientist at the University of Washington eScience Institute. My research training and experience have been mostly in the field of human cognitive neuroscience. During my postdoctoral training (2011-2015) in Prof. Brian Wandell's group, at the Department of Psychology at Stanford University, I conducted studies of human brain structure and function, using quantitative MRI. A focus of the research program that I started in Brian's lab is the application of ideas from statistical learning theory to measurements of human white matter with diffusion MRI (dMRI).

Workflow



Measurements of dMRI are used as a way to assess the structure of the human brain and its connectivity *in vivo*. Many parameters of the measurement are determined by the

experimenter, and incur trade-offs between sensitivity and signal-to-noise ratio (SNR). Models of the white matter based on different measurements are commonly used to make inferences about connectivity and tissue properties, but there was no extensive study of the fits of these models to the data, and no assessment of the effects of measurement parameters on the model fits. In the study described here, we used cross-validation to evaluate two commonly-used models in a variety of measurement conditions. The work was published in [PLoS One](#)

The project started with the collection of MRI data. Six participants were scanned in different experimental conditions. The data were collected in the Stanford Center for Neurobiological Imaging (CNI). The CNI has developed the Neurobiological Image Management System (Wandell, Rokem, Perry, Schaefer, & Dougherty, 2015), which captures the data from the scanner, archives it, and exposes a web interface that allows researchers to control access to the data, and copy it into the lab's data storage, a RAID (redundant array of independent disks) system.

The data were preprocessed using standard procedures (in the sense that any practitioner of MRI would perform these steps on his or her data). This includes correction of motion artifacts, alignment to a common coordinate frame, and tissue type segmentation. These steps were performed once, at the beginning of the study. The code that performs these steps is part of the lab code distribution, [vistasoft](#), freely available through GitHub. Preprocessing also relied on freely available software from other labs.

These preprocessed data are publicly available through the Stanford Libraries Stanford Digital Repository (SDR), as two different collections: <http://purl.stanford.edu/ng782rw8378> and <http://purl.stanford.edu/rt034xr8593>. Most of the data was licensed under the Creative Commons Attribution license, and a small subset was also released under the Public Domain Dedication License, for unencumbered use in methods development.

Subsequent analysis was conducted on these preprocessed data, using a Python library: [osmosis](#). This includes implementations of methods for fitting the data, statistical analysis, simulations and visualization, as well as utility functions to handle parallel execution on an HPC (high-performance computing) cluster. The library depends on many components of the `scipy` stack, including `numpy`, `scipy`, `matplotlib`, `scikit-learn`. In addition, the code depends on components of the [neuroimaging in Python](#) libraries. Approximately 30% of the module code was covered by unit tests, with a particular emphasis on core modules and utility functions that were reused. A few end-to-end tests were implemented to track regressions. Development of the software was done openly on GitHub, and it was also released under an attribution license.

Scripts using the module code were developed using the IPython notebook. These scripts were run and edited many times, and as the project evolved a few of these were copied into a [documentation folder](#), with notebooks named "Figure1.ipynb", "Figure2.ipynb", etc, each

corresponding to a figure in the paper. In writing the paper, these figures were saved and additionally manually edited by hand to add labels and annotation, and then integrated into a Word document file, which was used to collaborate on writing with the other authors. The writing process was not versioned throughout, but several versions of the article were submitted to the arXiv preprint server, while the article underwent peer review.

Most of the computations during the development of the project were conducted on a lab multi-core compute server that was running an IPython notebook server. Thus, much of the development of the code was done on a laptop, over a web browser, connected to the server. Some procedures described in the paper would require an inordinate amount of time without the access that we had to an HPC cluster. For example, testing different settings of model regularization parameters required fitting the models hundreds of times. Data was accessible to the cluster through a mount of the lab RAID. Tasks run on the cluster were managed through a queue system (Sun Grid Engine), and a module was developed (`osmosis.parallel`) to facilitate submission of code to the cluster. These scripts could not be used as IPython `ipynb` files, and were separately invoked on the command line, but they are included as part of the code distribution, recording these steps.

The IPython notebook documenting the steps that required parallel execution includes both a 'precomputed' version (where parameters of the analysis are read from precomputed files), and 'complete' versions, which include the code that would have to be run to reproduce these results entirely on a single machine. Precomputed parameter files were not made publicly available, and would have to be recomputed to reproduce the results in these notebooks.

Though reproduction of the results in the paper could, in principle, be achieved using this library, it is not necessarily useful as a tool for others to work with, and not easy to extend beyond the models that we tested. During the work on this project, I became involved in an open-source project, which develops Python software for the analysis of dMRI data: [Dipy](#). The main ideas in `osmosis` were eventually ported into Dipy, accomodating the application programming interface (API), documentation and testing requirements of that project. Furthermore, the prediction and cross-validation API that I implemented in `Dipy` is designed to be sufficiently general to accomodate new models, and mechanisms to evaluate their performance in fitting dMRI data.

Through Dipy, the code in this project is now also distributed widely through both GitHub and the Python Package Index (PYPI), under the permissive BSD license.

Pain points

One of the main difficulties encountered was the duration of some of the calculations. Some of the models, when fit on the entire brain volume, would require many hours. In particular, using the IPython notebook as a computational environment proved to be limiting, because connection to the kernel is only reliably maintained as long as the computer running the browser is kept on and prevented from sleeping. This also made it hard to perform computations that required a long duration in the notebook. One of the ways to deal with this was the development of caching mechanisms for model fit parameters. The models would be fit using a script, and the parameters cached to file. The model instantiation in the notebook would then know how to load these parameters from file, if the file existed.

Another point of frustration was that as the code in the modules evolved, code that was stored in the notebooks became outdated, and was no longer usable. This meant that as the analysis code itself evolved, new notebooks had to be written. Furthermore, as the writing and review of the article proceeded, figures were moved around in the article, and other figures got added; Figures that had started as appendices were integrated into the body of the article, etc. Thus, it might have been better to wait until the end result was an accepted article, and only then organize the entire reproducible workflow that led to this result.

Key benefits

Though sometimes cumbersome and effortful, one of the major benefits of the process of producing a reproducible workflow is the level of confidence in the results. There is never a doubt about what code is associated with what result, because the full chain of evidence is documented in the code leading to that result.

Key tools

A specific module (`osmosis.parallel`) was developed to deal with submission of jobs to parallel execution on the HPC cluster. This module would read in a 'template' script, and then create from this template, Python script files that contained the instructions to run the fitting process with different conditions, or on different parts of the same brain. The creation of this module resulted in a highly reproducible process. Consequently reuse of elements of this module produced benefits in time-saving during the development of the analysis methods.

Questions

What does "reproducibility" mean to you?

Reproducibility is a matter of degree, not of kind. It usually depends on the availability of code and data from a scientific study, such that only a reasonable effort would be required to generate the evidence (numbers and visuals) used to support a scientific finding.

Ideally, a small number of commands at the command line would suffice, but in some complex cases, more work could be required. A reasonable amount of effort required might be rather extensive, when large amounts of data storage, or large amounts of computation are needed.

A higher standard, sometimes called 'replicability' would be to require that the same conclusions be reached if another group of researchers were to do the same experiments, and implement the same ideas in their analysis.

Reproducibility does not guarantee replicability (Leek & Peng, 2015). Some may even argue that reproducibility and replicability may sometimes be in conflict, because implementation errors can be propagated in reproduction, but not in replication (Baggerly, Morris, Edmonson, & Coombes, 2005; R. D. Peng, 2009).

Why do you think that reproducibility in your domain is important?

Human neuroscience is a field which is particularly likely to have an abundance of false findings (Ioannidis, 2005): Sample sizes are usually small, particularly in MRI, which is an expensive experimental technique. The standards of the field focus on statistical significance of effects, rather than effect sizes, which tend to be small. Though standards limiting the selection of tested relationships, and limiting the flexibility of experimental and analytic designs are starting to emerge, in practice these are not very strictly limited. Some of the aspects of the field that make it interesting and important, are also pernicious in this regard: the direct application to human health means that there is a perception of potential financial incentives. Finally, it is a burgeoning field, with many groups working on similar questions. Higher standards of reproducibility in this case would mean less false findings, because at least some of these factors would be ameliorated by a full "chain of evidence" to support every finding.

How or where did you learn about reproducibility?

Many of these practices evolved out of laziness. Early on in grad school, I learned that most analyses that are done once eventually need to be redone, and that ultimately I would have to do less work, not more, if I had a script that generated all my figures for every study that I was doing. This also evolved from being rather bad at taking notes about the work I was doing in the lab. I would need programs, and eventually IPython notebooks, just to remember what I did to get from the data to the conclusions.

A huge impact was the mentorship I got from Fernando Perez during graduate school. He was not shy about how little of the research in our field he believed to be true, and this skepticism inspired me to struggle to be more confident in my own research conclusions.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

There are several barriers to wider adoption of reproducible research practices in human neuroscience. The first is that there is very little practical cost to not being reproducible. As mentioned above, there is likely to be a large proportion of false results in the neuroscience literature, and it's more likely to be false if it's not reproducible. Since a false positive result is more likely to result in a publishable unit, there seem to be incentives in place to not be reproducible, slowing down the progress of the entire field.

What do you view as the major incentives for doing reproducible research?

The level of confidence that I have in my results is quite high. That helps me sleep well at night.

Would you recommend any specific resources for learning more about reproducibility?

There are several papers that provide guidelines for reproducibility with a specific focus on neuroimaging. Two recent examples include Gorgolewksi & Poldrack (2016) and Pernet & Poline (2015).

References

Baggerly, K. A., Morris, J. S., Edmonson, S. R., & Coombes, K. R. (2005). Signal in Noise: Evaluating Reported Reproducibility of Serum Proteomic Tests for Ovarian Cancer. *J Natl Cancer Inst*, 97, 307–309.

Gorgolewksi, K., & Poldrack, R. (2016). A practical guide for improving transparency and reproducibility in neuroimaging research. *J Natl Cancer Inst*, 14(7), e1002506.
<http://doi.org/http://dx.doi.org/10.1371/journal.pbio.1002506>

Ioannidis, J. P. A. (2005). Why Most Published Research Findings Are False. *PLoS Med*, 2(8), e124. <http://doi.org/10.1371/journal.pmed.0020124>

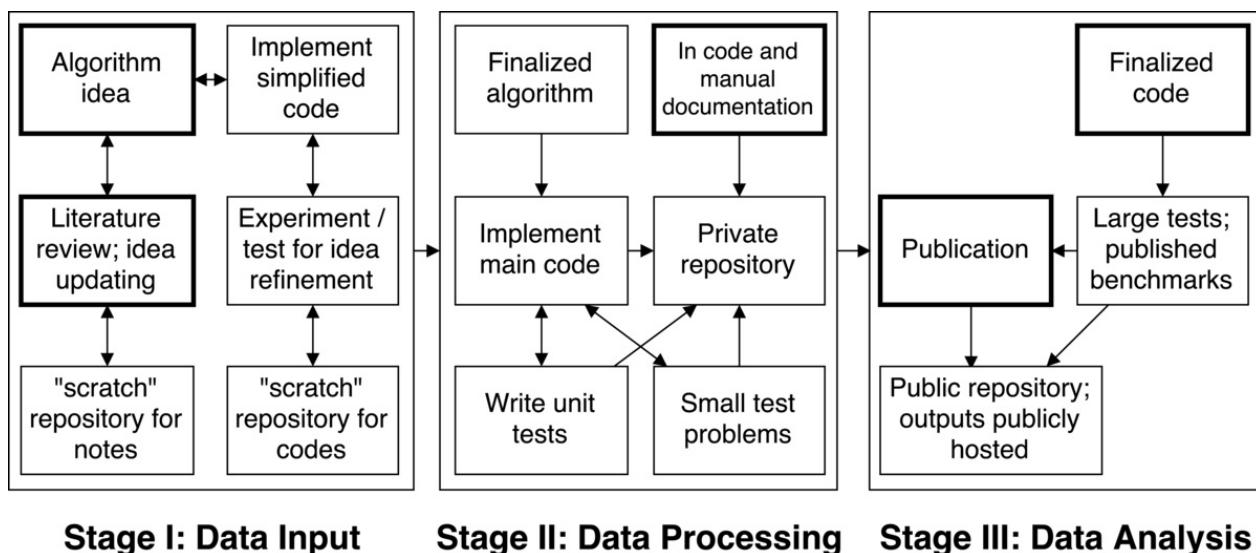
- Leek, J. T., & Peng, R. D. (2015). Opinion: Reproducible research can still be wrong: Adopting a prevention approach. *Proceedings of the National Academy of Sciences*, 112(6), 1645–1646. <http://doi.org/10.1073/pnas.1421412111>
- Peng, R. D. (2009). Reproducible research and Biostatistics. *Biostatistics*, 10, 405–408.
- Pernet, C., & Poline, J. B. (2015). Improving functional magnetic resonance imaging reproducibility. *Gigascience*, 4(15). <http://doi.org/http://dx.doi.org/10.1186/s13742-015-0055-8>
- Wandell, B. A., Rokem, A., Perry, L. M., Schaefer, G., & Dougherty, R. F. (2015). Data management to support reproducible research. *arXiv*, 1502.06900v1.

Reproducible Computational Science on High Performance Computers:

Rachel Slaybaugh

My name is Rachel Slaybaugh and I am an Assistant Professor in the Nuclear Engineering Department at the University of California, Berkeley. I study computational methods for neutron transport: numerical methods for solving the Boltzmann equation applied to neutral particle interactions. The methods I study are both deterministic (e.g. finite difference, etc.) and stochastic (Monte Carlo). I develop these algorithms for reactor design and analysis, radiation shielding, and nuclear nonproliferation applications. Much of my work has an emphasis on high performance computing. (Tagline: intersection of applied math and computational science, informed by nuclear engineering)

Workflow



I tend to think of my workflow for code development as having three fundamental steps: (1) idea generation and refinement, (2) idea implementation and testing, and (3) large scale testing and publication.

Step 1: The starting point of a new project is the development of an algorithm. This comes from a combination of reviewing literature, discussion with colleagues, familiarity with challenges in the field, and so on. As I refine an idea, I find I need to review more literature; as I research the literature, I refine the idea. The algorithm development tends to be collaborative as it is based on discussions with others, but the literature review tends to be independent. I like to write notes while reading papers in one large LaTeX document and

keep that document in a repository with all of my other notes and reviews so all of my notes on a given topic are in one place and I only have one place to look for things I have researched in the past.

Next, I implement a simplified version of the algorithm to make sure that it works at all. For example, I would implement a 0D or 1D version (as opposed to 3D) of a method quickly and simply in Python to use for testing. In this step there can be iteration between the algorithm idea and the test code, informed by additional literature review as necessary. Once satisfied with the experiments with the simple code, the algorithm is considered "final" (though of course it can be adjusted later if needed). I am not sure that this part of the workflow is reproducible in the sense that the process could be exactly replicated, but version controlling everything makes it possible to recover intermediate steps, which in some ways allows the idea refinement to be traced.

Step 2: Once there is a finalized algorithm, it gets implemented in the "real" code that has multiple developers and is written in a compiled language like C++. The repository for the code is typically private because, as mentioned, these codes are not completely open. It is often the case that only one or a very few people are working on this idea, so we make a branch and do the development there. I add unit tests to a testing framework associated with the code as I go (for example GoogleTest); the tests reside in the same repository as the main code. As the code approaches completion, I use small "system" test problems to investigate basic system functionality: does the code get the correct answer for analytical/known solutions? what does basic performance look like? etc. The small tests are also version controlled--either in the same repository as the source code or in a separate one.

Once the unit tests are deemed sufficient and, combined with the small tests, everything indicates that code is correct, I finalize documentation. Throughout development I typically use [Doxygen](#) to comment the code. Doxygen automatically generates documentation from source code comments when those comments are made using particular, simple annotations. Doxygen works for languages like C++, Python, Java, and others. Using Doxygen is useful for creating an application program interface quickly and easily. However, some extra work is often required to get the theory down and provide clearer directions for using the new algorithm. All of that is written in LaTeX for incorporation into the user and/or theory manual. The documentation LaTeX files are version controlled, often in a separate documentation directory. At this point the code will be reviewed and merged into the main code base. Once the code is finalized, the unit test and small test results should all be reproducible by the other developers--the people who have access to the developer repository.

Step 3: Once there is "finalized" code, it is time to do the real demonstration testing for publication. This involves running large test problems that demonstrate performance of the new algorithm for problems of interest as well as comparison to benchmarks to demonstrate correctness. The literature review, algorithm description, and results of the large (and sometimes small) tests will go in a final LaTeX document for journal submission. Recording of work for journal publication will also be version controlled, typically in a public GitHub repository. The idea is that, beyond the text writeup, the large test input files will be in the same repository as any scripts required to process data and generate plots, all with corresponding directions. Thus if you have access to the code and the repository with tests, scripts, and results, you can rerun all the calculations and process the data.

Pain points

There are a few pain points: An annoying one is getting the documentation right. It seems like just using Doxygen is not enough. To get something that really is user-manual quality you have to write a lot of things twice, just slightly differently. I try to reuse as much as I can, but if things are replicated there is the challenge of maintaining consistency.

A tough one is ensuring that the version of what is released in the end is actually reproducible. This requires the extra step of documenting which version of the code was used (the results should not change in the future, but it is better to have the version clearly written just in case). In principle one can figure this out from the repositories, but if everything isn't stored together that becomes more challenging. Providing directions about how to run everything and which versions of third party libraries were used is also some extra work.

A final pain point is re-implementing the algorithm from the simple case to the complex case, since the simple code is never really used for anything. However, this is a pretty small issue since the toy code didn't take long to develop.

Key benefits

The largest benefit of this approach is having confidence in the validity of the data that you publish. For me that confidence starts with implementing the methods and their tests at the same time. I think everyone should have a unit testing system; it is difficult for me to see how one could have confidence in the correctness of their software without one. I get very nervous about using code that I write if I haven't written tests to go with it.

Having an up-to-date application programming interface is also very useful. When I'm interfacing or working with a piece of code I wrote a long time ago I would not otherwise remember what it did or how to use it. It is also helpful when interfacing with parts of the program other people wrote. This extends to proper documentation. I personally can't

remember many things. I must write them down for future reference. Keeping a user and theory manual means not only that users and other developers will know what the code does and why, it means next week I will also know what the code does and why.

I also find that having little bits of experimental code, the low-dimension test pieces I write at the beginning, are useful to have on hand. This does not particularly impact reproducibility, but it is useful to have chunks of code to start with when playing around with new ideas. Similarly, having a repository with literature review notes is good for remembering past research, speeds up writing papers and documentation, and provides a place to start looking the next time.

Key tools

The key tools I use are Doxygen, git (for version control), LaTeX, and plotting and data manipulation tools (usually in Python).

Questions

What does "reproducibility" mean to you?

The first way I think of reproducibility is "can I/my lab reproduce the results in my paper exactly?" After that, "can an independent researcher, given that they have legal access to the required data and software, reproduce the results?" Nuclear engineering data and codes are often controlled, so for many projects only researchers within my field will have access to the required data and software. Fortunately, such non-open-source codes are typically available at no cost to researchers through a simple licensing process.

Why do you think that reproducibility in your domain is important?

The codes that we write are often used to investigate new nuclear systems and make long-term policy or design decisions based on the results. They are also used to study existing nuclear systems. This is important stuff; the codes need to be right and the results need to be verifiable.

How or where did you learn about reproducibility?

- Mentors: my PhD advisers valued reproducible practices and insisted that we used them
- Student groups: the Hacker Within

- Practice: taking on a project that used good practices was how I really learned many of these skills
- Community exposure: spending time with others who value reproducible practices
- Self-study: looking up things I saw people do that looked helpful

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

The biggest challenge is legal: only some people can access the codes and data required. A secondary challenge is access: some of the work I do requires high-performance computing that is not readily available to many.

What do you view as the major incentives for doing reproducible research?

Ethical mandate: I want my work to be right and for others to be able to know that it is right.

Impact: My ideas and products might then be adopted and built upon.

Are there any best practices that you'd recommend for researchers in your field?

Testing, testing, testing.

Would you recommend any specific resources for learning more about reproducibility?

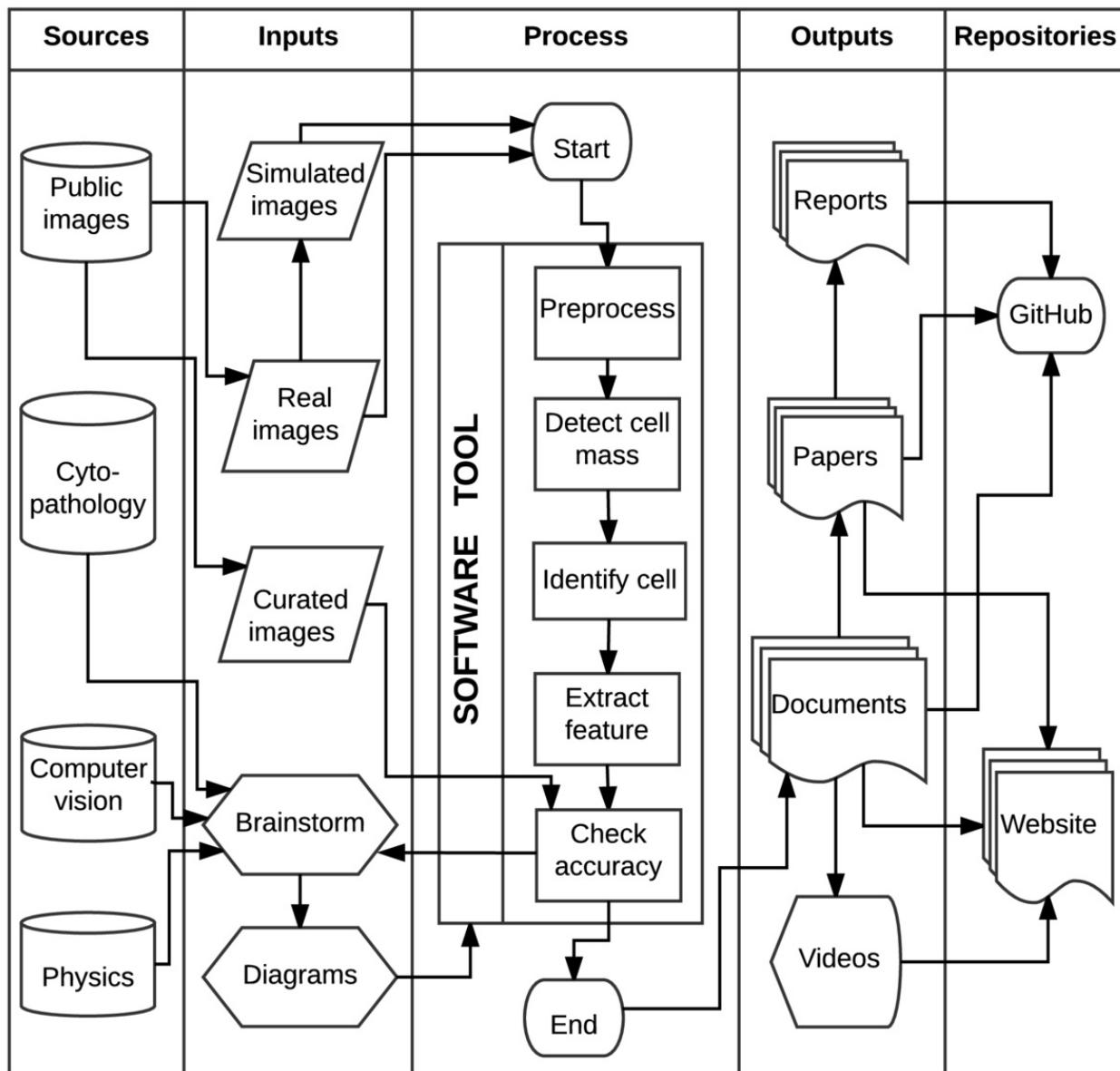
Software Carpentry; the new Scopatz-Huff book.

Detection and Classification of Cervical Cells

Daniela Ushizima

My name is Daniela Ushizima, I am currently a [Staff Scientist](#) at the Lawrence Berkeley National Laboratory and a [Data Science Fellow](#) for the Berkeley Institute for Data Science at the University of California, Berkeley. Much of my research work is in devising machine vision and pattern recognition algorithms as part of software tools for handling image-centric data, especially those arising from the [Department of Energy](#) imaging facilities. The case study I describe here illustrates the core steps in designing a machine vision algorithm to analyze a set of digital images and organize them according to the desired criteria. There are several image processing and analysis frameworks that encapsulate algorithms; this case study refers to the application of [ImageJ](#), a powerful image analysis tool.

Workflow

Stage I: Data Input Stage II: Data Processing Stage III: Data Analysis


The workflow diagram follows a data model called SIPOC, which stands for suppliers, inputs, process, outputs, and customers; these correspond to the columns of the table. Here, we adapt SIPOC diagram to better represent our use-case, hence the first column is called sources and the final column represents repositories. The proposed workflow prioritizes the compartmentalization of different processing steps of the software tool, and hides potential feedback loops that might occur.

This diagram tells the story of research investigations among doctors, cyto-pathologists, physicists, and computer scientists, aiming to design, develop and deploy algorithms for improving the analysis of biomedical images. Some of the tasks include increasing the number of image fields under scrutiny in order to speed up the cell counting and recognition, comparison among cells, quantitative description of samples, to name a few.

Historically, this use-case began with brainstorming among pathologists, physicist and software engineers on how to provide scalable computer-aided analysis to the acquired large datasets of biomedical images, containing [cervical cells](#). Communication and diagramming were fundamental sources of information to understand how to categorize types of cells and foresee limitations imposed by the datasets, such as cervical cell lineages, cell fragments, magnification and usable area within the image sample.

In order to develop analysis pathways, the team organized the datasets into three main *input* image sets: simulated, real and curated. A simulated image consists of several clipped real cells collated with different levels of overlapping, which facilitates algorithmic validation since the ground-truth is known *a priori*. A real image consists of a digital picture of a Pap smear slide, obtained at the light microscope -- these images often contain several types of cervical cells and other findings (e.g. bacteria, blood), and may be corrupted by noise and other artifacts, such as staining variations, dirt, hair, etc.

The core of this case study lies in the *process* column that illustrates the main steps in the machine vision algorithm. The preprocessing step transforms the samples into more compact and reliable representations of the image, for example, removing areas or eliminating the whole image if it is over-stained. This step includes essential image transformations, such as anisotropic filtering that preserves borders and smooths regions that compose a supposedly homogeneous image partition.

As part of the analysis, the software tool must detect the regions of the interest in each image, i.e., the cellular mass and the rest of the image. This step requires several iterations of the machine learning algorithm (statistical region merging) before it can correctly split the image into foreground and background. The next step is to separate the cellular mass into individual cells: by modeling simple biological prior knowledge, such as the relationship between nucleus and cytoplasm, the software tool is able to quickly estimate cytoplasm boundaries. After identification of the cells, feature extraction takes place, including nucleus-cytoplasm area ratio, convexity of cytoplasm contour, and other parameters that are relevant for identifying cell lineages. Finally, we use simulated and curated datasets to validate results, for example, considering sensitivity and specificity measures based on the number of pixels or the number of cells identified.

An important step of the data processing is keeping track of the *outputs*. The fourth column lists four main outputs of the system: technical reports, scientific papers, documentation about the software tool and educational material about the science problem and algorithm development. Although we omit outputs of partial results (checkpoints), they are very common and useful throughout the design of the analytic software tools.

The fifth column shows the different outputs being archived in *repositories* to enable access to the research discoveries, for example [GitHub](#) and [websites](#). In the context of this case study, it indicates the main hubs of information distribution for the project.

Pain points

The software tool design and testing have required intense communication among the team members through reports and presentations. Although part of the team used version control, much of the code is still to be made available open-source through GitHub. In addition to commit messages, which tend to be short, we have also maintained an electronic diary of activities -- these were fundamental to keep the whole team synchronized and up to speed. The painful side of such an electronic lab book has been the unstructured format of the inputs that may require extra-time to parse.

Key benefits

The most reproducible part of this project has been the development of code allied to simulated datasets. This activity improved across the team, particularly due to the participation in code competitions, which forced the whole team to organize data sources and code repositories such that reviewers could quickly reproduce the results. In addition, keeping track of advancements in a common digital lab book helped in preparing manuscripts and other technical reports.

Key tools

An important tool has been ImageJ, a Java-based image processing software program, which was originally developed by [Wayne Rasband](#) at the National Institute of Health circa 1997. Although most of the ImageJ plug-ins focus on medical imaging, this framework has been widely used in other applications, such as [material sciences](#).

Questions

What does "reproducibility" mean to you?

In the context of this case study, the work will be computationally reproducible when the software tools our team builds can also be used by the science domain experts, e.g. pathologists, who should be able to transform raw data files into quantifiable patterns, obtaining consistent results with previous/tested analyses. Because algorithmic parameters often change from a dataset to another, it can be challenging to get results with the same accuracy, given different datasets.

Why do you think that reproducibility in your domain is important?

Reproducibility is essential in quantitative microscopic because it can guarantee accurate measurements and improved quality control.

How or where did you learn about reproducibility?

Flow charts and code documentation improved grades during college, and their absence meant dire punishment: reproducibility once was a time-consuming protocol to get good grades and spend extra ribbon cartridge with my dot matrix printer. The tech-industry introduced me to version control, and [TortoiseSVN](#) and I started working together with several colleagues. Other softwares came along the way, such as [git](#) and [Atlassian](#)algorithm to enhance usability and extension of the codes. After entering BIDS, reproducibility turned into a fun activity, a conversation starter and a never-ending code improvement process. Lots of concepts came together more systematically, particularly after attending Software Carpentry classes and becoming a instructor myself.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

The major pitfalls of doing reproducible research in image analysis have been (a) the lack of domain-specific image examples that illustrate applicability of the algorithms, (b) dependency of packages that are not freely available, and (c) absence of documentation that enables understanding how/why the data transformations occurred.

What do you view as the major incentives for doing reproducible research?

The major incentives for doing reproducible research are the ability to replicate the experiments later, to fix and/or reuse code for different applications, to easily work in larger teams, and the potential for a broader impact, even with the help of collaborators you have never heard of.

Are there any best practices that you'd recommend for researchers in your field?

While there are no general rules, some tools can only help a person to reproduce work. In my opinion, they are: (a) use of version control, (b) practice of [software quality assurance](#), (c) organization of data samples and code systematically, for example using [Cookiecutter](#).

Would you recommend any specific resources for learning more about reproducibility?

Among the several options out there, [Software Carpentry](#) is certainly an important resource and the book [Making Software](#) by A. Oram and G. Wilson.

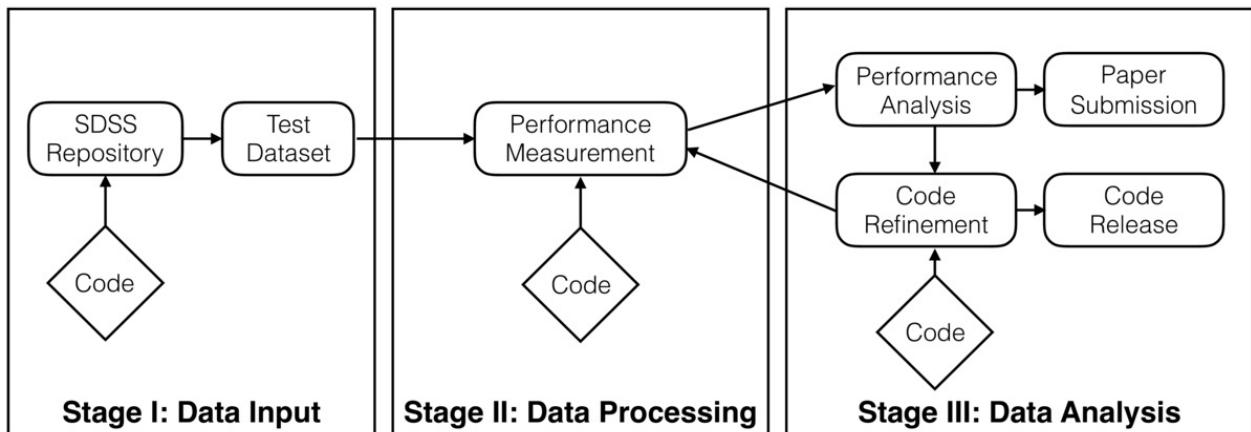
Enabling Astronomy Image Processing With Cloud Computing Using Apache Spark

Zhao Zhang

My name is Zhao Zhang, I am a joint postdoctoral researcher at AMPLab and Berkeley Institute for Data Science, University of California, Berkeley. The theme of my research is to enable data-driven science with computer systems.

This case study describes the process of building Kira, a distributed astronomy image processing pipeline in the cloud environment. The idea of the Kira project is to explore the applicability of cloud computing based software stack in supporting scientific applications. Specifically, we use the SEP (Source Extraction Python) library for domain computation. We choose Apache Spark and Hadoop to build the infrastructure of distributed processing and data storage.

Workflow



We use LaTeX and Slides to track the merit evaluation: why do we need a new system for astronomy image processing, what makes it a better system, and what lessons we can learn from this research.

We use a private GitHub repository to keep track of solutions for technical barriers such as I/O processing, Spark interaction with C program, Spark system parameter configurations and many others.

The whole system is built with multiple programming languages and tools. At the programming language level, we use Scala, Java, Python, Bash, and C. At the system level, we use Spark for task coordination, HDFS for persistent storage, and the SEP library for actual computation.

The source code of the project is kept in a public GitHub repository to make it open source.

The manuscript is being kept in a private GitHub repository since it is under review.

In system design phase, we decided to use three datasets for development, testing and performance measurements. But we end up with four datasets. A trivial dataset that contains only a few image files is used for development and testing. A small dataset (12GB) is used

for quick verification at scales. A large dataset (65GB) is used for large scale performance measurement. A fourth dataset (1TB) is used to show the data processing capacity of the Kira system as we put up the paper.

All these datasets come from the Sloan Digital Sky Survey. Some of them are from Data Release 2 while some are from Data Release 7. We choose them arbitrarily as we care more about the system capacity rather than the science in this research.

Our collaborators are: Kyle Barbary, Oliver Zahn, Saul Perlmutter are astronomers. Frank Nohaft, Evan Sparks, Michael Franklin, David Patterson are experts about Spark and cloud computing in general. Zhao Zhang has rich experience in HPC community and some experience in cloud computing as well as a bit astronomy background.

We use private GitHub repository for manuscript management and public GitHub repository for project management.

We have summaries for Team Brainstorming and Merit Evaluation phase. System Design's output is in the form of figures and is kept in GitHub repository. Solutions for Technical Barriers are kept in a private GitHub repository. Documents, Source code, system configurations as the products of coding/testing/tuning/measurements are kept in a public GitHub repository. The paper draft is kept in a private GitHub repository.

Before explaining the details of the diagram, I will first briefly review the software and systems that are used in this case study.

- FITS (Flexible Image Transport System) is a widely adopted image format in the astronomy and cosmology community. It is a fixed format with the image metadata as text and the actual image as binary format.
- SEP (Source Extraction Python) is the software that detects light source objects from images. It rewrites the SEXtractor software by exposing primitive functionalities through a library interface with both C and Python.
- Apache Spark is a popular distributed computing framework in cloud computing. It offers implicit parallelism and the lineage-based fault tolerance through the Resilient Distributed Dataset (RDD) abstraction. Spark is built using the Scala programming language which compiles a program that is executable on Java Virtual Machine (JVM).
- JNI (Java Native Interface) provides a method to call existing C libraries inside a Java/Scala program. C and Java/Scala data structures can be used to exchange information between the two runtimes.
- Amazon EC2 (Elastic Compute Cloud) is a public cloud service provided by Amazon. Users can request a number of compute nodes with various hardware and software combination.

- Amazon S3 is a data storage service provided by Amazon. Users can host their dataset on S3.
- NERSC (National Energy Research Scientific Computing Center) is a high performance computing facility operated by Lawrence Berkeley National Lab. It hosts a few supercomputers and clusters.
- SDSS (Sloan Digital Sky Survey) is a large scale sky survey, its data is publicly available online.
- Thread safety is an operating system concept that describes the concurrent execution of multiple threads safely manipulating shared data structures.

The process begins with team brainstorming of how modern computer software and hardware can accelerate the astronomy image processing pipeline. This requires a wide and also deep understanding of the state-of-the-art research and technical solution. In this research, we gather domain expertise (astronomers), cloud computing expertise and high performance computing expertise. We review the existing work and we think using cloud computing software-hardware stack can improve the overall application performance, but we have no idea by how much it can improve. The research is an exploratory process to implement the idea and quantitatively measure the improvements if there is any.

The Team Brainstorming and Merit Evaluation phase happened back and forth as we keep asking why are we building such a project. Detail questions include: What are the existing solutions? How does the new project make difference in terms of performance and usability? Who are the potential users? This procedure lasts for about two weeks, all members of the Kira project are involved in the discussion. The pros and cons of each existing solution was documented, and later used in the paper.

The System Design phase lays out the programming interface of Kira, the modules and interactions between the modules. In this phase, we also identify some technical barriers of this project. I am listing them below, feel free to contact me if this is hard to understand:

1. Kira I/O, how to make Spark read FITS images
2. Calling C library in Spark, how to make Spark work with existing C code in the SEP library
3. Setting up compilation environment, set Maven to automatically build Kira

As we progress with the code, we notice a few other technical barriers:

1. Thread safety, neither the jFITS library nor the SEP library is thread safe
2. Load imbalance, scheduler tuning for this particular workload

For each of these technical barriers, we seek solutions for them. The solutions come from three sources: colleague expertise, google, and documents. By isolating the barriers, we were able to focus on a single barrier each time and can quickly verify the solution. The resulting code is stored in GitHub, and later merged into the project. This process takes about two weeks.

The Software Coding and Testing phase takes about three weeks, we managed to integrate the SEP library through Java Native Interface with Spark, thus finally implement Kira. I implemented the code, and wrote the documents to make it convenient for myself to repeatedly run experiments. In the meantime, I prepared four datasets for performance measurements. A 24MB (4 files) dataset for sanity check, a 12GB (2,310 files) dataset for small scale test, a 65GB (111,50 files) for medium scale test and a 1TB (176,938 files) for large scale tests. The datasets were initially stored in NERSC shared file system, later I made a mirroring on EC2 S3 service, as most experiments were run on EC2 where S3 has a better transfer bandwidth to.

Performance Measurements and Performance Tuning come in pair and we go back and forth frequently. The key thing in these two steps is that we need a reasonable expected performance before the measurements. If the measurement does not match with our expectation, we need to analyze the reason and tune the system. Our methodology is like this: we started on 1 core on a single machine. We compare the Kira performance against the equivalent implementation to understand the slowdown introduced by Spark and JVM. Then we started to scale up with more cores on the same node, and observe the scaling curve. By doing that we understand the bounding factor of the performance on a single node. Later on, we scale out on multiple nodes by doubling the number of compute nodes in each step and observe the performance scaling. Since Spark hides the scalability complexity in the system, all we need to do here for different scale is to set relevant parameters in the configuration files. The code and documents are kept in GitHub, and the dataset is kept in Amazon S3 service.

With all of the scripts from Merit Evaluation, System Design, and Source code, we put together the paper. Writing the paper is a collaborative process. We used a private GitHub repository to host the paper, and using Pull Request to manage everybody's editing.

Pain points

1. (reproducible results) For the results to be reproducible, the readers should be able to tell and access the computers with the same hardware, the code base used particularly for the experiments, the dataset that was used for performance measurements.

1.1 Hardware access. Since we are using Amazon EC2 resources, the same computer hardware is mostly accessible unless Amazon upgrades the hardware. It happens every few years. A second risk is that for large scale test, the reader might contact Amazon to increase the hardware limit which Amazon uses to limit the quantity of resources each user can posses at the same time.

1.2 Code base. We maintain our code under a public GitHub repository, so it is accessible to all. However, the pain point is that the software evolves and the performance might change with the software evolution. Thus it is important that the authors should let the readers know which version of the software is related to the results that readers care about.

1.3 Dataset. The astronomy image dataset we use is Sloan Digital Sky Survey Data Release 7, which is publicly accessible. As long as the data hosting service is up running, the dataset is available. We also make a copy of the dataset we used in Amazon S3 service with public accessible permission. The pain point is that we have to pay Amazon for maintaining the 1TB dataset, and eventually we will run out of funding, so instead we have to publish the dataset file list as a text file in the code base.

Key benefits

I break this question down to two: non-usuable workflow and non-reproducible workflow.

1. Non-usuable workflow. I have seen a couple of projects in astronomy, where the authors conducted study on applying new tools to solve the old problems, but the authors failed to publish their source code along with the paper. This gives up the opportunity for people to build solutions upon their work. For Kira, we make the source code available on GitHub, so people can extend this code base for more functionalities.
2. Non-reproducible workflow. There was one experiment I read in a paper that I would like to reproduce, and design a new solution for it. However, the experiment was not reproducible due to the software version evolution. During the Kira building process, we particularly care about this issue, we documented the hardware, code base and dataset that are used for the performance measurement, so any user that follow the documented instructions should be able to reproduce the results.

Key tools

GitHub for code management, and Amazon S3 service for data hosting. We built Kira with Apache Spark which is a highly active open source project, so that we do not have the concern of the computing framework is out of maintenance if our academic funding ends.

Questions

What does "reproducibility" mean to you?

In the context of my case study, reproducibility has several levels of meanings. The very baseline is that users can compile the source code and pass the tests. Secondly, users should be able to configure the computer cluster so they can reproduce the performance in the documents. Since it is impossible to reproduce the exact performance measurement for every single run, a statistical repetition should be fine (average performance with bounded variation). Generally for computer system research that involves data, a public available data source is necessary for the performance to be reproducible.

Why do you think that reproducibility in your domain is important?

As computer system researchers, we build systems assuming people will use them. So it is important that people can follow the instructions in the documents to reproduce the state in which the system works. And it is important that users can reproduce the improvements over existing systems we describe in the paper or documents so they are more likely to adopt our systems. As paper reviewers, it is more convincing if they can reproduce the results in the paper as these are the evidence of the paper's idea.

How or where did you learn about reproducibility?

I learned the reproducible practices since the first time I submitted my homework project in college and ever since. I need to write a README file along with my code so the teaching assistant could compile and run my code to test if my solution is right. The later research experience follows the same path.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

I used to design systems on supercomputers, where many people including paper reviewers have no access to. Thus, it is impossible for the research to be reproducible.

Another major pitfall is due to software version evolving. At the time of writing the paper, some features of a piece of software was working, and the researchers measured and published the numbers. But these numbers are no longer reproducible after a few versions.

What do you view as the major incentives for doing reproducible research?

I break this down to reproducible results and reproducible research process.

4.1. (reproducible results) The systems I build are usually to facilitate scientific research. The systems either expedite the execution of computer programs or provide novel functionalities (e.g. failure diagnosis). To make sense about my research, users should be able to see the performance improvement I documented in the paper or documents. They should be able to use the novel functionalities to ease their research. So reproducibility of the systems is the key to prove the system actually works.

4.2. (reproducible research process) As a whole process, this particular research case is exploratory. We only have a conjecture about the performance before the implementation and measurement. I am not familiar with the tools I am using also (Apache Spark, Scala, Java Native Library, SEP library, Source Extractor C program). I think (not quite sure) the incentive for the reproducible research progress is helpful in my future projects. Once again, if I am facing such situation, I know where to start to tackle a complicated problem. My methodology particularly for this research is: 1) a more-or-less valid hypothesis, 2) a performance profile of the existing solution, 3) isolating the technical barriers, 4) solving the technical barriers, 5) build the new solution, 6) performance measurement and tuning.

Are there any best practices that you'd recommend for researchers in your field?

I would recommend for open source software and related publications. The authors should maintain a version of the software for readers to reproduce the results in the paper. These versions and repository should be included in the paper.

Would you recommend any specific resources for learning more about reproducibility?

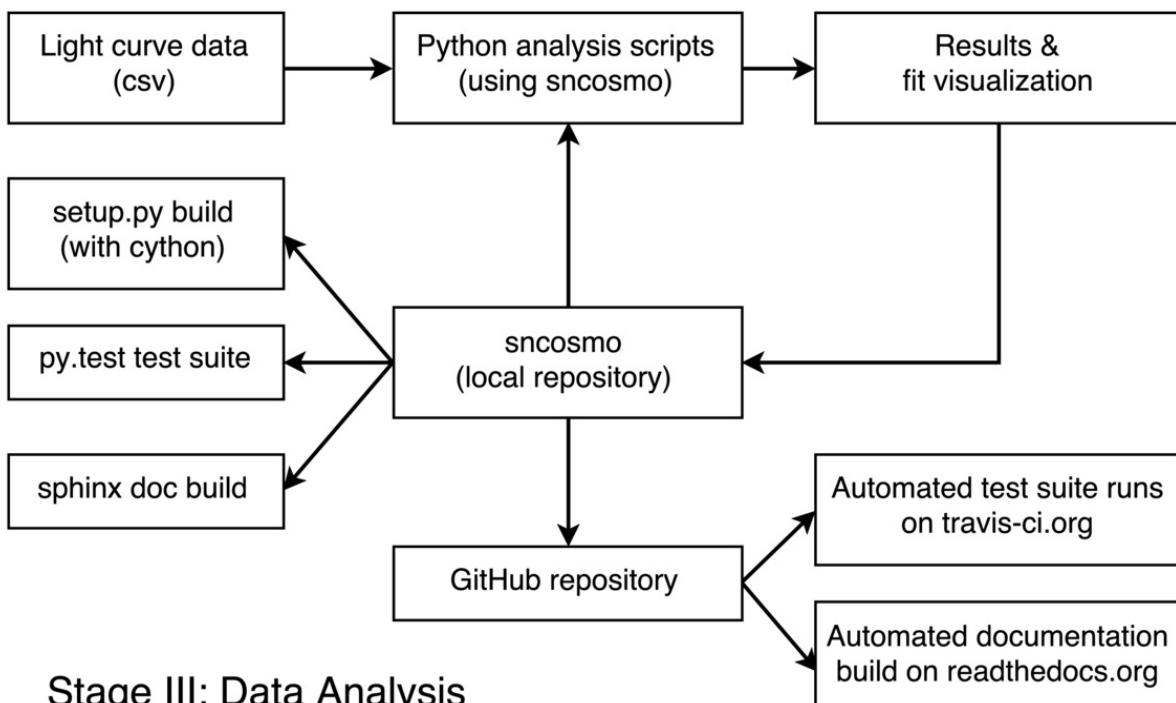
I can only think of GitHub for code management and Amazon S3 for data management right now.

Software for Analyzing Supernova Light Curve Data for Cosmology

Kyle Barbary

My name is Kyle Barbary and I am currently a postdoc in the physics department and a Data Science Fellow in the Institute for Data Science at the University of California, Berkeley. I am an observational cosmologist. More specifically, I use a particular variety of exploding stars, known as Type Ia supernovae, as markers to measure how the universe has expanded over its history. To make this measurement as precisely as possible, it is necessary to combine supernova data from many different surveys targeting different distances. The workflow I describe is about the creation of software tools used to combine and analyze that data in a uniform way.

Workflow



I will describe the development of software for analyzing supernova light curve data. A "light curve" in the parlance of my domain is simply the brightness of a supernova as a function of time. These brightness measurements are derived from images of the same patch of sky spaced in time, ideally showing the supernova growing brighter and then fainter. Analyzing these light curves is a key step in deriving final results for most supernova cosmology

studies. The software in question was originally developed for analyzing data from the Dark Energy Survey, but it can be (and has been) used for analyzing data from other surveys, as I will discuss below.

The analysis starts from reduced light curve data produced by a separate pipeline (not discussed here). A Python script reads the data, performs analysis tasks such as model fitting or parameter sampling, and saves the results or produces plots allowing the user to visualize the results. There are generally multiple scripts for performing different analyses or variations on an analysis, and these can be written by several different scientists on the project. The key aspect of the process is that all commonly useful functionality is split out into a Python *library* (SNCosmo). The top-level analysis scripts contain logic specific to the analysis and to the survey, and the SNCosmo library contains functionality applicable to a variety of surveys and analyses.

The development of the SNCosmo library itself is an iterative process where features of the library are added or refined in response to the needs of various analyses or users. Although there are official release versions of the library, several users stay up-to-date with the development version to keep this feedback loop tighter.

We use git for version control of the library and GitHub to coordinate development, where work is centered around an "SNCosmo" GitHub organization. Users who follow the development version periodically pull changes from the copy of the repository owned by the "SNCosmo" organization. We use two services in conjunction with GitHub. First, continuous integration is done with [Travis](#): every time a change is made to the GitHub repository, this service is triggered. It builds the library and runs the full suite of unit tests for multiple combinations of supported library versions. This allows the developers to catch and fix problems before they are reported by users. Second, automated documentation builds are done by [Read the Docs](#). This service builds the library and runs the documentation builder which produces a set of HTML pages (and also a PDF with the same content). This allows users to see the documentation for the latest development version immediately if needed. These two services are free for open-source projects and are widely used.

Within the repository, we use a number of standard tools: there is a `setup.py` script which can be used to build the library via `setup.py build` or to run the tests using `setup.py test`. The `py.test` package is used internally to run the tests.

Finally, at some point we make an official release version of the library. This is typically done after features have been user-tested for some time and the API is stable enough to be supported in future release versions. This is often a difficult judgement call.

Pain points

- **Feature stability:** There is a trade-off between adding some feature immediately versus waiting until it is obvious whether to include it and what the specific interface should be. In the past I've marked such features as "experimental" with a warning in the documentation that users might have to change their code in the next library release version.
- **Multiple platforms:** I develop on Linux but most users are on Mac OS X day-to-day. This hasn't been a huge problem yet, but it has produced a few headaches. Automated build services are starting to support OS X for free, so this will help.

Key benefits

The separation of common software functionality into a *library* is surprisingly unique in this subfield of supernova cosmology. It is a boon for reproducibility: published results can include the (relatively short) analysis scripts that were used, along with the version of the SNCosmo library used. The fact that the core software is a well-documented library means that readers and practitioners can more easily understand the specifics of the algorithms used.

Questions

What does "reproducibility" mean to you?

To me, reproducibility has two facets: the availability of usable software (preferably under an open-source license), and the availability of data (preferably in both raw and reduced forms). Together, these should give an outsider the ability to reproduce the results of a study from start to finish.

I separate these two aspects because each can be beneficial without the other. For example, even without releasing data, it can still be quite beneficial to release software. If released under an open-source licence, this provides a different flavor of reproducibility - the ability to reproduce an algorithm described in a paper and use and improve that algorithm in subsequent work.

As a side note, in my domain we often settle for a weaker form of full reproducibility, where a "reduced" data product and the software to analyze it is released, but not the raw data and not the software to go from raw to reduced data.

Why do you think that reproducibility in your domain is important?

Efficiency. Reproducibility makes cosmology research more efficient in the following ways:

- Reuse of code. Cosmologists are as guilty as any of reinventing the wheel, particularly when the blueprints for the wheel are not made available.
- Better understanding of algorithms spreads more rapidly. Algorithms are often explained coarsely in papers but without the detail necessary to reimplement them. Allowing the reader to directly read the code (if desired) solves this problem.
- Fewer unexplained conflicting results. Research is often held up or lead down the wrong track by conflicting results from multiple groups. Allowing different groups to reproduce each other's results will help resolve such situations more quickly.

How or where did you learn about reproducibility?

Mainly through working on the AstroPy project, which develops a community astronomy Python package. I got involved in AstroPy when it was started in 2011. Like many other large open-source projects, AstroPy is developed on GitHub and follows typical best practices such as extensive unit testing, automated documentation builds and continuous integration on multiple platforms. In short, I learned these practices by interacting with more experienced programmers also working on the project.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

In astronomy, like other fields, observers have a desire to carefully guard their hard-won data until they have eeked out every possible analysis. I'm sympathetic to this; acquiring the data often requires designing, building and deploying a new instrument or even an entire telescope. It can be a very large fraction of the work that goes into a project. The threat that someone else will download your data and use it to publish a result that you could have published is very real.

I'm less sympathetic about the reluctance to release software. Some of the reasons that I've experienced:

- perceived lack of quality
- perceived extra work to clean it up, maintain and support it
- perceived competitive advantage or that the software is an asset or bargaining chip

Even for those who do wish to release their software under an open-source license, it is often difficult to do so in a fully legal manner through "official" channels due to university or lab copyright. Often, scientists just release the software without official permission.

Finally, one technical issue with releasing data is data volume. Raw imaging data from an entire survey can be many terabytes. Making this data publicly available often requires dedicated servers and support staff.

What do you view as the major incentives for doing reproducible research?

- **Long term project efficiency:** Projects are often carried out over multiple generations of grad students and postdocs. Doing things reproducibly within a collaboration makes the transition between generations much less lossy.
- **Ability to back up claims:** It often happens that two competing research groups make the same measurement and find results that differ by a marginally significant amount. The differences can often be due to specific statistical choices that were made in the analysis. In such disputes, having reproducible research means that you can invite the competing group to inspect your analysis in detail (and hopefully be proven right!).

pyMooney: Generating a Database of Two-Tone, Mooney Images

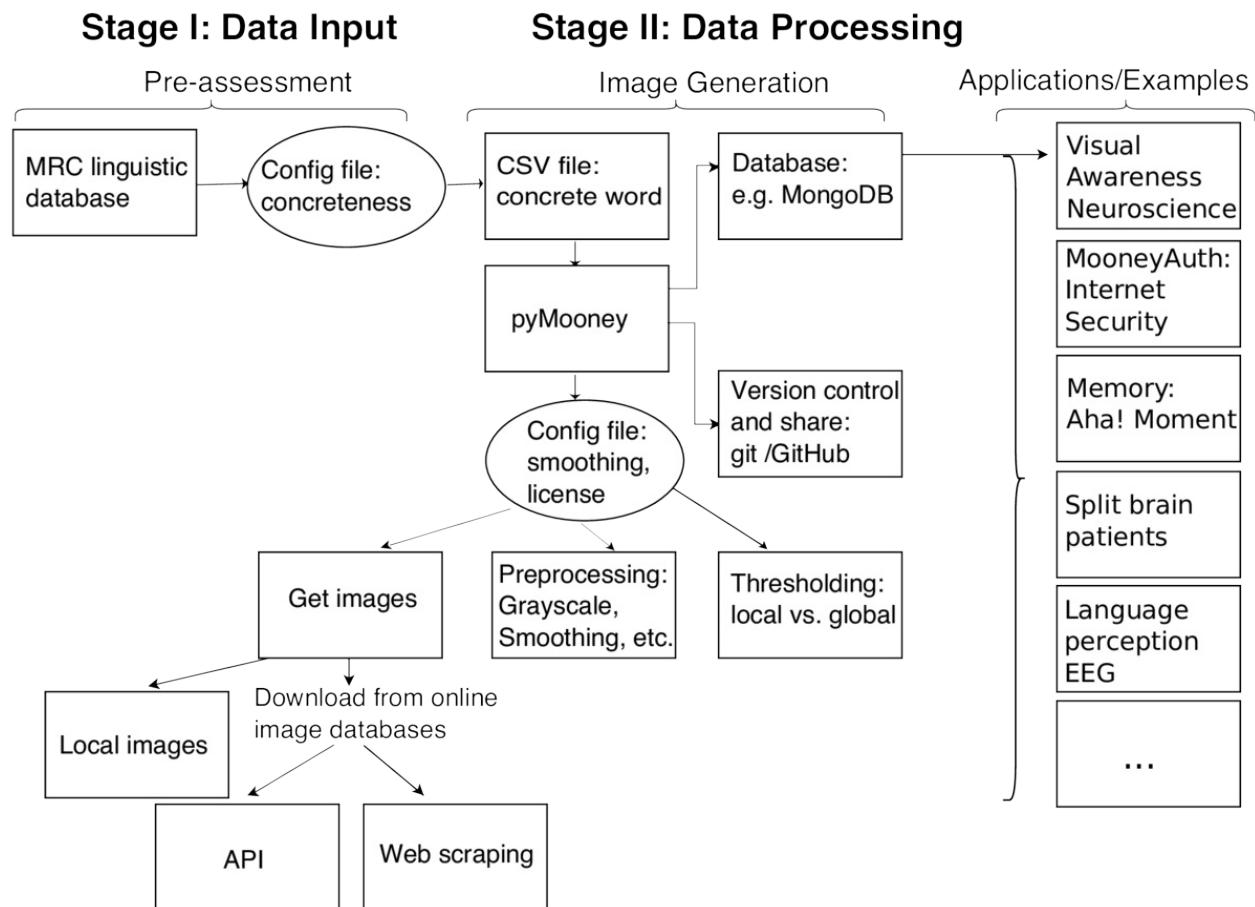
Fatma Deniz

My name is Fatma Deniz, I am a postdoctoral fellow at the [International Computer Science Institute, Helen Wills Neuroscience Institute](#) and a Data Science fellow at the [Berkeley Institute for Data Science](#).

I use functional Magnetic Resonance Imaging (fMRI) and computational modeling to investigate how the human brain represents the world perceived through different sensory modalities. The current case study will only describe one part of my usual research pipeline, i.e., stimulus generation. Every successful neuroscience experiment needs to run several pilot experiments to create the best stimulus that is suitable for one or several questions that the researcher is asking. Hence, this step is usually tedious, where different parameters are tested until a final set of stimulus is used in the experiment. Ideally, the stimulus set is broad enough that new experiments (new hypothesis) can be derived using the same stimulus set. So, it is very important to provide access to the stimulus set that was used in a study, or provide the algorithmic procedure that created the stimulus.

In this case study I will focus on an image database that I created for a specific cognitive neuroscience experiment. These images have been used in several other experiments since the original paper was published. The images that I created for the study were two-tone, Mooney images. These images are binary, black and white images, where a single hidden object is only recognizable when the original image has been shown previously to the observer, the hidden object was outlined, or after a certain time when the observer intrinsically finds hints in the image that allows recognition. The image is not recognized immediately but after some time, which makes stimulus creation for a suitable experiment very difficult. Using these images in the original experiment (F. Imamoglu, Kahnt, Koch, & Haynes, 2013) I presented that functional connectivity in the human brain is altered when the hidden object is recognized vs. when it is not recognized.

Workflow



Generating two-tone, Mooney images starts with a selection of concrete words. There is a database called [MRC Psycholinguistic Database](#) where each word has been labeled as concrete or abstract, how frequently it is used, etc. From this database I selected 967 concrete words. The necessary parameters are saved in a config file (e.g. concreteness rating and imageability rating between 550 and 700). These concrete words are saved in a CSV file. Using these concrete words as image search tags I downloaded real-world images from an online image database (e.g. Flickr, or Google images) in an automated fashion using the custom written [pyMooney](#) python package. This package is based on a python API (Flickr API) and the scikit-image library. Each project needs to have a config file that specifies whether the Mooney images are created based on images that are stored locally or whether the images should first be downloaded from an image database. If images are downloaded from an online resource licensing information needs to be set in the same config file. Using a smoothing and thresholding process (Otsu, 1979) and prescreening of images, I created a database of 330 two-tone images for the pilot image selection experiment. These images can be stored in a document oriented database (e.g. MongoDB). A database has the advantage that information such as what preprocessing steps have been applied, what license information an images has, what is the average reaction time of the images in specific experiments etc. can be stored among the images. In addition images can be searched and selected according to this information. In this pilot experiment human subjects were presented with the two-tone, Mooney image and were asked to indicate the time when they recognized the hidden object in the image with a key press. They were

further asked to label the name of the object that they think they recognized.

In our functional magnetic resonance imaging (fMRI) experiment we were interested in the question how brain activity changes when subject's recognize the hidden object versus when they not. The two-tone images do not change over the course of the presentation but a subject's perception change over time, and this moment in recognition is associated with a change in brain activity, which we wanted to capture. FMRI image acquisition is relatively slow (every 2 seconds). In addition, as fMRI scans are costly, we are limited by time. Hence, for this fMRI experiment I selected 120 Mooney images (resized to have a 400 x 400 pixel size) that were recognized within 4-10 seconds in the pilot experiment.

The code is written in Python and is available on GitHub.

Pain points

When images are downloaded from an online resource copyright issues can emerge. This can be avoided by downloading images that are licenced as Creative Common. This change is reflected on the latest version controlled pyMooney code and new images can be created with such criteria.

Key benefits

The main benefit of this part of a larger experiment is that these images are now available for further research. The images are currently used in 30 different experiments ranging from clinical set-ups, human memory experiments (Kizilirmak, Silva, Imamoglu, & Richardson-Klavehn, 2016), vision research, and latest internet security applications (Castelluccia, Duermuth, Golla, & Deniz, 2017).

Key tools

Image processing libraries are important building blocks of this particular case study. In this case the open source python library scikit-image was used. In addition online image database APIs such as FlickrAPI are essential.

Questions

What does "reproducibility" mean to you?

In the context of this study reproducibility means that given the code and a database of images, a new researcher can have access to the images and use the code that was used to create the same images or new images with the same parameters. These images can

then be used to (i) replicate the current findings, (ii) create new questions potentially based on the current findings.

Why do you think that reproducibility in your domain is important?

I think without reproducible research we waste a lot of professional time and research money. In my own domain, and in direct connection to this case study I can outline an example. When I came upon the two-tone images in the literature, I contacted several groups with a request to use their images (of course I mentioned that I will properly cite their work), who had used similar images for other behavioral or neuroscience experiments. Unfortunately, the images that these groups used were very limited in number but nevertheless, I never got a response from these groups. Hence, in order to start my experiment I had to spend almost a year to create the new stimuli.

How or where did you learn about reproducibility?

This was a self taught practice. Hence, the stages described here still has some space for improvement, which I will elaborate further below.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

In neuroscience the main pitfalls are sharing human subject data and the resistance in the field to share code or data. The majority of the community does not have an open science mind set. Researchers are afraid that someone else can conduct an experiment before they publish their results. Even if they published some results they are sometimes not willing to share the data as they think they can continue to ask new questions using the very same data.

What do you view as the major incentives for doing reproducible research?

I think reproducible research allows faster progress in a researcher's own domain and makes interdisciplinary projects possible. To see that your own research is further used not only in your own domain but also across domains is very rewarding. It opens up possibilities for new collaborations. For example, I collaborated in two new projects that recently got published by making my stimulus available (Kizilirmak et al., 2016, Castelluccia et al. (2017)).

References

- Castelluccia, C., Duermuth, M., Golla, M., & Deniz, F. (2017). Towards implicit visual memory-based authentication. In *Network and distributed system security symposium (ndss)*.
- Imamoglu, F., Kahnt, T., Koch, C., & Haynes, J.-D. (2013). Changes in functional connectivity support conscious object recognition. *Neuroimage*, 63, 1909–1917.
- Kizilirmak, J. M., Silva, J. G. G. da, Imamoglu, F., & Richardson-Klavehn, A. (2016). Generation and the subjective feeling of “aha!” are independently related to learning from insight. *Psychological Research*, 80(6), 1059–1074.
- Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Trans. Sys., Man., Cyber.*, 9(1), 62–66.

Problem-Specific Analysis of Molecular Dynamics Trajectories for Biomolecules

Konrad Hinsen

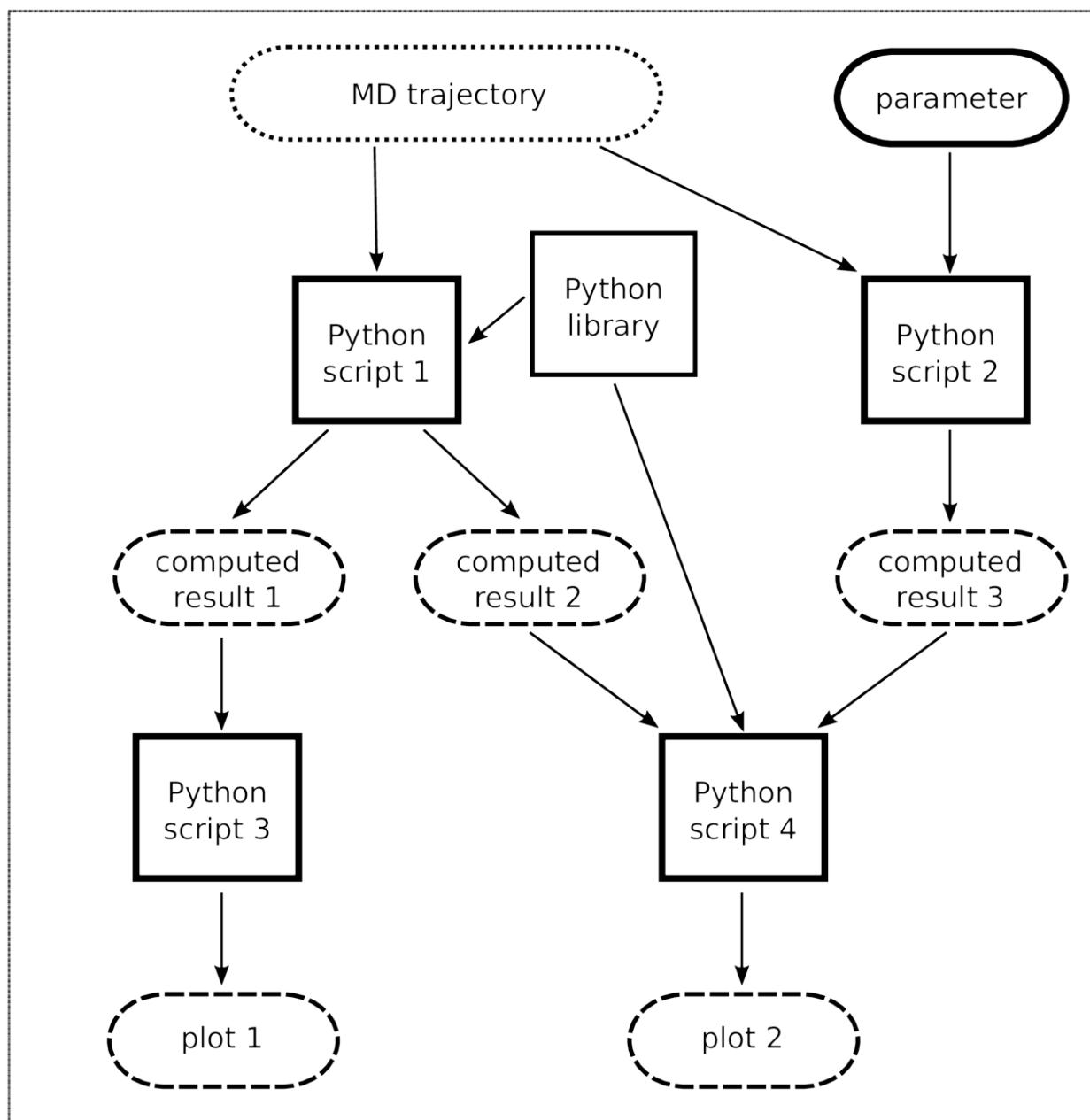
My name is [Konrad Hinsen](#), and I am a researcher at the [Centre de Biophysique Moléculaire](#) in Orléans, France. My field of research is molecular biophysics, and in particular the study of the flexibility and dynamics of proteins. All of my work is based on computational approaches, of which the most important ones are [elastic network models](#) and [Molecular Dynamics](#) (MD) simulations. Moreover, most of my work concerns the development of computational methods rather than the application of already established methods.

This case study is about the extraction of information from MD simulation trajectories, a very common type of work in my field. MD simulations themselves are relatively standard procedures, performed using one of a handful of well-known software packages. They take a few days to a few weeks on a small parallel computer with a few tens of processors, and produce a so-called trajectory file that is one to ten GB in size. Analyzing these trajectories in order to actually learn something about the system that was simulated is a separate step that is much less standardized, meaning that there is a lot of problem-specific code involved. This code is as much a result of the workflow as the plots of the computed quantities.

For reproducible and publishable workflows, there are three specific challenges in this situation:

1. The size of the trajectory files, which are difficult to publish in a citeable way, often being larger than the current upper limits of [Zenodo](#) or [figshare](#).
2. There are CPU-intensive tasks that are typically run on a parallel computing cluster in batch mode, and explorative tasks that are done interactively or near-interactively (running short scripts that take about a second) on a desktop machine. Dependency tracking across machines is not supported by most workflow management systems. It doesn't help that computing clusters often have limited network connectivity.
3. The distinction between "software packages" and "workflows" is not useful when most of the code being executed is problem-specific. A more appropriate code structure is "well-established techniques implemented in libraries", "problem-specific scripts" and at the top level "coordination of a small number of scripts". It's the last two levels that must be captured for reproducibility.

Workflow



Stage III: Data Analysis

A published example of the workflow described below can be consulted in the form of two code/data packages ([package 1](#), [package 2](#)) and the [article](#) describing the study.

The workflow diagram is actually a dataflow graph with attached workflow information. Compared to most approaches to workflow, which place the tools (workflow manager, software packages, Web services, ...) in the center of attention, the approach I describe here focuses on the data and on the way scientists interact with the data. The workflow below is not about "getting a job done" but about "developing and fine-tuning a scientific model".

The dataflow graph shows code in rectangles, and "passive" data in rounded boxes. Code consists of a small number of Python scripts, of which four are shown in the diagram. Data flows from top to bottom, as shown by the arrows, starting with the MD trajectory that is the overall input, and ending in plots showing computed quantities. The three rounded boxes labelled "computed results" are intermediate results, computed by Python scripts 1 and 2 and consumed by Python scripts 3 and 4.

From the point of view of workflow management and reproducibility, the most important distinction among the data items is "human input" (solid outline) vs. "computed data" (dotted outline). It's the human input that represents the scientific model, and thus the main output of this workflow. It consists of code (Python scripts 1 to 4) and numerical parameters (a single one in the diagram), though that distinction is rather arbitrary: every parameter could be turned into a line in a script. Computed data includes the plots that go into the journal article, but also intermediate results. In a fully reproducible workflow, the computed data need not be stored, because it can be recomputed at any time. Nevertheless, it is often preferable to store it explicitly, in particular if recomputation takes a long time. Stored computed data is also more readily available for exploration by scientists who want to gain a better understanding of the method.

The workflow consists of the iterative refinement of the models and methods. The two key tools in processing the workflow are:

- a version control system such as [git](#) for keeping track of the changes
- the [ActivePapers](#) dependency manager for coordinating the computations

Correspondingly, the state of the project consists of

- a repository under version control, which tracks the changes to the "human input" items as the project advances
- an ActivePaper file, which stores the current state of all data items and the dependency graph

There are two variants of a refinement step: adding a new script or parameter, and modifying existing scripts and parameters. The first kind, which extends the data flow graph, consists of the following user actions:

1. Write the new script.
2. Commit it to version control.
3. Check in the script to the ActivePaper.
4. Run the script via the ActivePapers dependency manager.

The second kind, which preserves the data flow graph, differs only slightly:

1. Edit scripts and parameters.
2. Commit the changes to version control.
3. Check in the modified versions to the ActivePaper.
4. Update the ActivePaper.

The fourth step recomputes all data that is affected by the changes made in step 1. The recomputation is steered by the *dependency graph*, which is obtained from the data flow graph by redirecting arrows that point into a script to point instead to the outputs of the script. The ActivePapers dependency manager computes the dependency graph automatically during the execution of the scripts. Users do not have to deal with (or even know about) either graph explicitly. They write and run scripts as they did before reproducibility became an issue. Similar approaches are used in [Sumatra](#) and [noWorkflow](#), but most workflow managers adopt the opposite strategy of letting the user construct a workflow explicitly and then execute it.

A project can be transferred from one computer to another by copying the ActivePaper file and the version control repository. For the common situation in molecular simulations that lengthy computations are off-loaded to a cluster, step 4 in the above procedure is slightly modified: The ActivePaper is sent to the cluster, the "run new script" or "update" operation is performed on the cluster, and the modified ActivePaper file is transferred back to the user's desktop machine. All the tools have a command-line interface, making it easy to use them over an ssh connection.

Method-development projects tend to be small, involving a handful of people. The pitfalls of coordinating modifications to files can easily be avoided by having a single person perform each refinement step, or even all of the refinement steps. Other participants can of course contribute ideas, and inspect the current state of the project for analysis.

At the end of the project, the ActivePaper file(s) can be published, making all of the code and data available to other researchers. The ActivePaper file contains the complete final state of the project (though not its history), meaning that anyone can continue from that state. An ActivePaper file for a new project can re-use items from already published ActivePaper files through a DOI (Digital Object Identifier), allowing other researchers to build on published computational work. The DOI can also be used for citations in journal articles.

Pain points

The main practical difficulty is that most of today's computational scientists grew up with tools and practices that are not compatible with reproducibility. This is particularly true for the field of molecular simulations, where reproducibly published studies are still rare. Working reproducibly requires adopting new tools and habits, and modifying existing software for integration with reproducible workflows. There is a permanent temptation to give up reproducibility for faster scientific progress.

The immaturity of current workflow tools for reproducible research adds another layer of cognitive overhead. In the workflow described above, this is mainly the use of separate tools for tracking history and dependencies. Today's version control systems, designed for software development rather than computational science, cannot easily be extended by the kind of dependency management required for research. On the other hand, writing new version control software integrated with dependency management represents an effort that is hard to justify at this time.

A major constraint imposed by the ActivePapers system is that all code must be written in Python and all data must be stored in HDF5 datasets. While Python is popular enough for molecular simulation to make the first constraint very acceptable, HDF5 is still a rare choice for data storage, although this is changing thanks to initiatives such as [H5MD](#).

The use of specific tools is rarely sufficient to ensure reproducibility. Tools can only take care of *replicability*, i.e. the technical aspect of tracking all computational dependencies such that a computation can be re-run identically. Reproducibility at the scientific level requires that all steps can easily be understood and verified by fellow scientists. Best practices for reaching this goal remain to be developed. One observation from the applications of the above workflow is the importance of access to intermediate results for human inspection. This suggests an overall structure of many small scripts that each do a well-defined job and communicate via explicitly stored datasets.

Key benefits

The traditional workflow of changing scripts and running the interactively in a shell is extremely prone to mistakes. The most frequent one is forgetting to re-run a script after its input data has changed because of an update to another script. Before I adopted reproducibility support tools, I regularly found myself looking at a data file and wondering which exact sequence of script executions had produced it. The question typically comes up when writing a paper. Even for today's minimal "materials & methods" sections, it is typically necessary to look up parameters and other choices in the scripts when writing the documentation, and that's often the moment when one discovers what a mess they are. This is no longer an issue when the complete final project state is available for inspection, and guaranteed to be complete and coherent by software tools.

Key tools

The key tool in my workflow is the [ActivePapers](#) toolset, which was in fact designed specifically for supporting reproducibility in the context of atomistic and molecular simulations. It supports in particular

- computations on large datasets by storing them efficiently in [HDF5](#) files with the dependency information stored as HDF5 metadata
- dependency tracking across machines by storing all datasets and their dependency graph in a single HDF5 file that can be copied easily from one machine to another

The only other reproducibility-enabling tool in the workflow is a version control system.

Questions

What does "reproducibility" mean to you?

Given that my work is 100% computational, my long-term goal is full reproducibility, starting from a specification of the simulation and ending with the plots that go into a journal article. This goal is unrealistic at the moment because the simulation software packages do not support reproducibility. One problem is the accumulation of numerical roundoff errors, which are insufficiently standardized across processors and compilers to be reproducible. Another problem is the widespread use of random number generators without user control over the random seed.

For this reason, I have been setting myself a more modest goal for this case study: reproducibility of the trajectory analysis step, using the MD simulation trajectories as input as if they were experimental data outside of my control. This is a useful compromise because the development of trajectory analysis techniques is the central scientific topic of this work.

Why do you think that reproducibility in your domain is important?

Most MD simulation studies are so complex that in the absence of reproducibility, it is impossible to be sure what was really computed. Most mistakes do not lead to a recognizably wrong result, but to a somewhat different one that could well be correct.

How or where did you learn about reproducibility?

I developed them myself, having found nothing suitable for the specific needs of molecular simulations after a careful survey of existing technology and practices.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

The main challenges are human and social. Most of my colleagues have experienced the problems that non-reproducibility creates, but few are willing to invest the extra effort to do a better job, and many remain unaware of the tools and practices for reproducibility that already exist. Journals in my field generally don't require the publication of software or data, and do not in any way encourage reproducibility. Technical challenges exist in that the most popular software packages do not support reproducibility, but the technical challenges could be met with little effort if there were sufficient motivation in the community.

What do you view as the major incentives for doing reproducible research?

- Feeling more confident about the correctness of my results.
- Being able to build safely on earlier work (by myself or others)

Are there any best practices that you'd recommend for researchers in your field?

I'd already be happy if publishing software and data became the norm in my field. It's hard to recommend any more elaborate practices before the basics become standard.

Developing an Open, Modular Simulation Framework for Nuclear Fuel Cycle Analysis

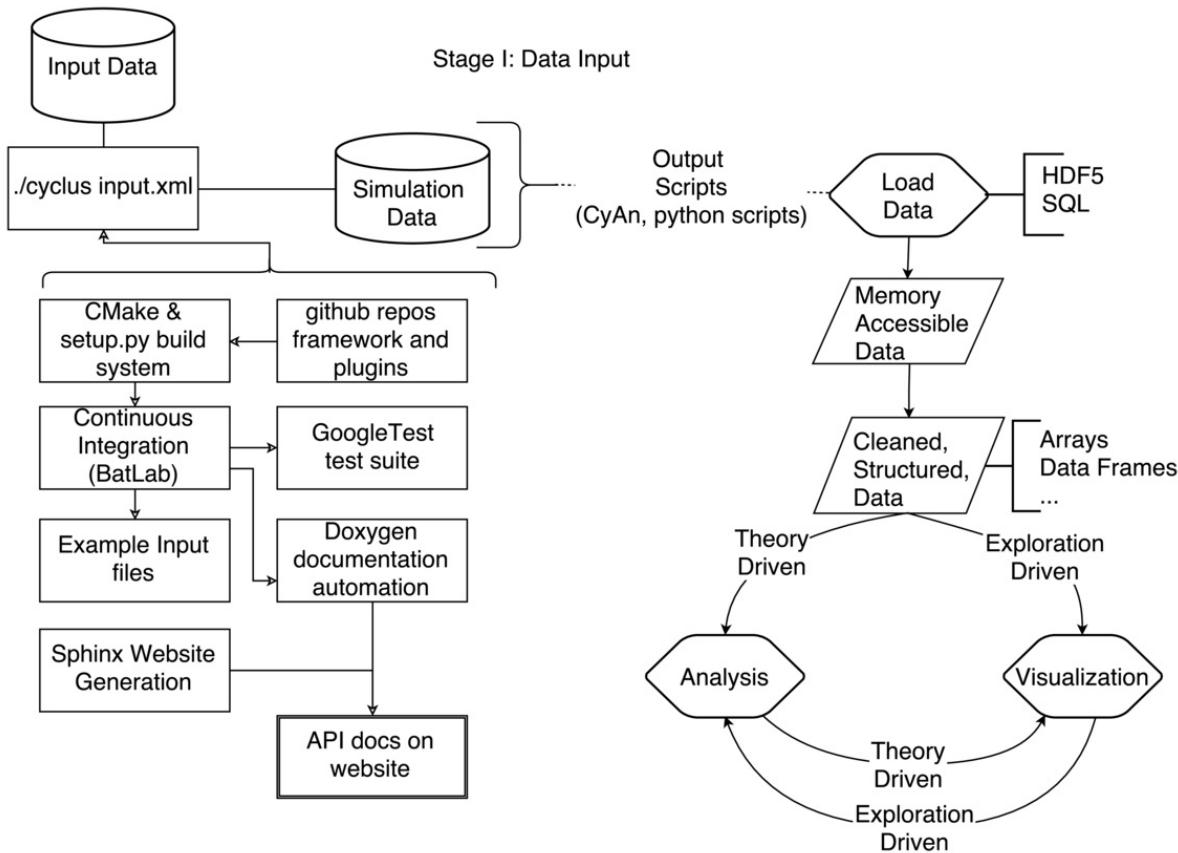
Kathryn Huff

My name is Kathryn (Katy) Huff, and I am a Nuclear Science and Security Consortium postdoctoral scholar in the Nuclear Engineering Department and a Data Science Fellow with the Berkeley Institute for Data Science. My research includes computational nuclear fuel cycle analysis and computational simulation of coupled, transient, nuclear reactor physics.

Improving the safety and sustainability of nuclear power requires improved nuclear reactor designs, fuel cycle strategies, and waste-disposal concepts. The systems are sufficiently complex that breakthrough advancements may emerge when modern data methodologies are applied to their simulation. In particular, faithful assessments of potential nuclear fuel cycles require dynamic, discrete facility, discrete-material simulations of the mining, milling, transmutation, reprocessing, and disposal of nuclear materials as well as the production of energy and movement of capital.

This case study is an overview of the workflow behind the Cyclus nuclear fuel cycle simulation framework -- a tool for exactly that kind of modeling, simulation, and analysis. The workflow described used to create a software tool that other nuclear engineers can use easily, modify quickly, and contribute to when they need to customize behavior or model a different technology.

Workflow



I and a group of geographically-dispersed researchers (graduate students and professors) collectively develop and maintain an agent-based simulation framework called Cyclus. We also develop and maintain plug-in models representing the agents in the simulation. These agents model the mining, milling, fabrication, transmutation, and disposal of nuclear material in the *nuclear fuel cycle*.

Cyclus is a C++ code base. The configuration and build system is created from a combination of Python and CMake (a crossplatform automatic makefile configuration system) and supports both Linux and MacOS operating systems. Our input validation library accepts either xml or json input files. The simulator accordingly conducts a simulation which generates an output database in either SQL or HDF5 format which can be traversed by a separately developed graphical user interface.

As we develop this software, we rely on a number of best practices to ensure reproducibility.

When a large-scale enhancement is needed, a Cyclus Enhancement Proposal (CEP) is proposed and discussed among the development team. Smaller enhancements are discussed as issues in GitHub. Once approved, the enhancement is implemented and a pull request is made in GitHub. Our automated continuous integration server (BatLab) runs the full suite of unit, integration, and regression tests. Before a proposed change is allowed into Cyclus, it must be covered by a test and all tests must pass.

Unit tests cover code units like functions and are implemented using the GoogleTest framework. Integration and regression tests are performed by running sample simulations and verifying that results match predictions or previous results. A set of standard input files are run, then the output is inspected and compared via Nose, a unit testing framework in Python.

Similarly, API changes must be documented as required by the Cyclus documentation CEP. The documentation for the current stable branch and the development branch are both provided on the Cyclus website using Doxygen and Sphinx, which are both automatic documentation systems that rely on the code comments in the C++ and Python code, respectively.

Finally, we use the Google C++ style guide to make our code as consistently formatted as possible.

When the change is made, a developer begins to conduct a particular analysis by creating an input file. That input file is provided to the Cyclus framework and validated by its input validation framework. According to the input file, a simulation is run. The ouput database that is produced contains important metadata about the simulation. It contains:

- a complete copy of the input file
- the commit hash of the current version of the Cyclus code
- commit hashes for all necessary plugins retrieved from the Cyclus ecosystem

That database, containing both data and metadata, can then be analyzed by the user. When analyzing the database, a choice is made by the user about how to interact with the data. The Cyclus development team has provided a GUI and a Go library (called CyAn) with which the database (in either SQL or HDF5 format) can be accessed and brought into memory for vizualization and analysis. Additionally, many user-developers have their own set of Python scripts that can do this stage of tasks. Given the universal nature of these database formats, most common scripting languages can be used to extract the data and metadata efficiently, so many options exist.

In summary, the research workflow in this framework has the following steps :

- If necessary, a developer proposes a change to support their analysis
- The change is made including passing tests and satisfactory documentation
- It is reviewed and pulled into the master branch
- The software is rebuilt and installed using our build system
- A simulation is defined in json or xml

- The input file is run and an HDF5 or SQL database results
- The database is analyzed with a separate GUI, python scripts, or a Go library
- A collaborative paper is created in LaTeX on GitHub
- All input files contributing to the analysis are contained in the repository holding the document

All of these steps are conducted in the context of git and GitHub.

Pain points

Build systems are painful. In particular, cross platform configuration and builds are an enormous time-sink for our research group. There are a number of reasons for this.

First, supporting C++ builds on Windows is sufficiently difficult that we abandoned supporting that platform.

Also, due to the physics-based solvers and optimization calculations in our simulations, external library dependencies are essential to Cyclus. We rely on libraries like Boost and LibXML2 to facilitate development, and we rely on libraries like Blas, Lapack, and COIN for mathematical solvers. For this reason, new developers spend a non-trivial bulk of their spin-up time building and installing the dependencies necessary to install Cyclus on their particular platforms.

Finally, our continuous integration system relies on our ability to create scripts that build, install, and test Cyclus. For this, we use a set of servers at the University of Wisconsin called the BatLab. Unfortunately, BatLab has a few problems. Because of the proprietary nature of MacOSX, it cannot run truly MacOSX instances. It runs, instead, Darwin servers that mimic the behavior of MacOSX. For this reason, idiosyncratic failures apparent in Mavericks and Yosemite but not Darwin cannot be caught before entering the code-base. Additionally, BatLab is somewhat unpredictable and inflexible. Since the behavior of BatLab undergoes a lot of churn, our continuous integration suite is sometimes rendered completely useless.

Key benefits

The [Cyclus Enhancement Proposal \(CEP\) strategy](#) was a bright workflow choice that was inspired by the analogous strategy in the Python community (PEPs). I recommend this to any research group that values strategic planning, consensus, and thoughtful development. A discussion of our workflow around these proposals can be found [here](#).

Fundamentally, a CEP is :

a design document providing information to the Cyclus community, or describing a new feature or process for Cyclus and related projects in its ecosystem. The CEP should provide a concise technical specification of the feature and a rationale for the feature.

CEPs document major new features, community discussions, and documentation of theory or design not captured by the in-code documentation. Because they are maintained alongside the website source code in a version controlled repository, provenance of the discussion surrounding their acceptance is maintained.

Key tools

We use CMake to configure and build our software. Much more human readable than the configuration files within the GNU autotools suite, CMake makes our lives easier.

The continuous integration system, though difficult to implement due to build issues, has decreased development time. It would not be possible without CMake, GoogleTest, and Nose.

Questions

What does "reproducibility" mean to you?

A reproducible research product is one that has been sufficiently documented, well-constructed, and preserved for its results to be recreated by some external researcher or group.

Why do you think that reproducibility in your domain is important?

My domain, nuclear engineering, is one where precision and accuracy are both of utmost importance to both human and environmental outcomes. Any conclusions drawn by science can only make an impact in the real world if they can meet the standards set out by the Nuclear Regulatory Commission. For this, reproducibility is paramount.

How or where did you learn about reproducibility?

I learned these practices primarily from my advisor, Paul P.H. Wilson at the University of Wisconsin, Madison. I also learned from colleagues in The Hacker Within, the Scientific Python community, and Software Carpentry.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

One major problem is export control. Making software and data open source is restricted by the US Department of Energy, in some cases.

What do you view as the major incentives for doing reproducible research?

- **Fear.** The fear of retractions due to faulty software or data can be reduced by enforcing transparent reproducible practices, which tend to reduce the likelihood of being accused of scientific fraud.
- **Surprise.** Six months after a paper is submitted, the surprise of no longer recalling your own thought process is unpleasant. To avoid it, reproducible practices can help you reproduce your present work in the future.
- **Ruthless Efficiency.** The automation inherent in reproducible workflows, makes tweaking and re-running of simulations and analysis very efficient.

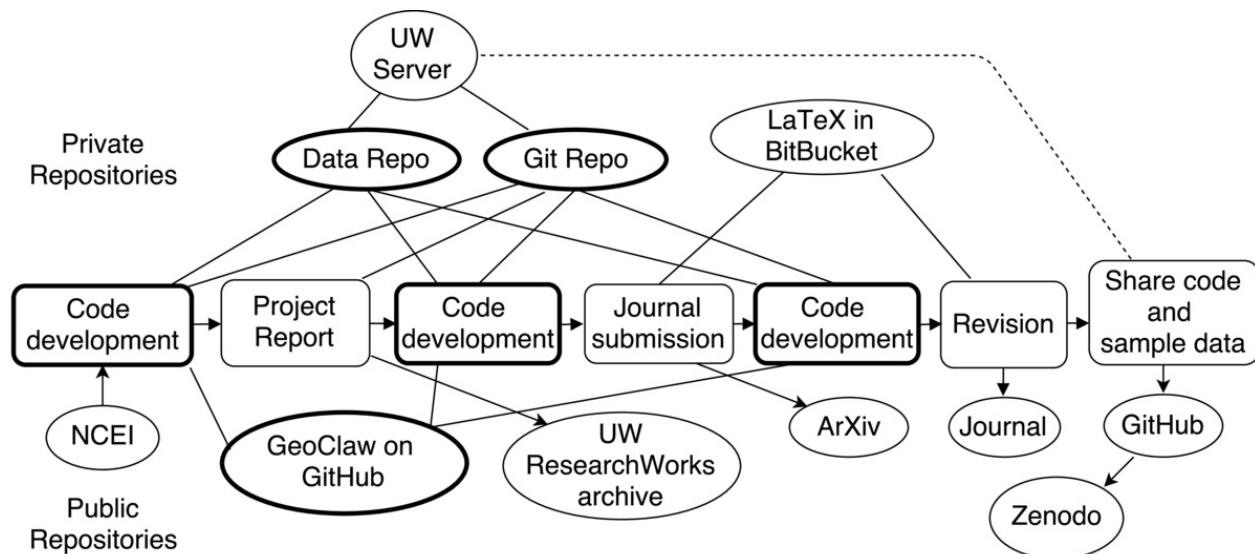
Producing a Journal Article on Probabilistic Tsunami Hazard Assessment

Randall J. LeVeque

My name is Randy LeVeque and I am a Professor of Applied Mathematics and one of the core developers of the open source [GeoClaw](#) software package for modeling tsunamis and other geophysical flows. Recently we have been using this software to study new approaches to probabilistic tsunami hazard assessment (PTHA), in which the goal is to take some probability distribution of possible future earthquakes that might cause large tsunamis and produce a probabilistic hazard map for a particular community, indicating which regions are most at risk and estimating the annual probability of flooding to a given depth at each point. This is complicated by the fact that the depth of flooding by a particular hypothetical tsunami depends on whether it arrives at low tide or high tide, and we have developed ways to incorporate this uncertainty.

The workflow I will describe relates to a [journal publication](#) on this topic. Much of the research was originally performed as part of a consulting contract funded by a private firm, as part of a broader pilot study funded by the Federal Emergency Management Administration (FEMA). We used version control for the code developed as part of this project that was initially in a private repository, along with the results of many tsunami simulations. A final project report based on this work was made available in our institutional repository, but was not published in a journal. We later improved the description of the methodology and performed additional computational experiments in the process of writing up a portion as a journal article. A variety of different private and public repositories were used in the course of this work, along with several platforms for sharing code, data, and the report and journal article.

Workflow



Stage I: Data Input

Stage II: Simulation Data Generation and Analysis

We first created a new account on the University of Washington (UW) campus computing system dedicated to this project that could be shared by the three collaborators, with sufficient storage for accumulating simulation results (and securely backed up by campus services). On this account we created a git repository that we could all access via ssh to use as our master repository for developing code, and eventually for writing the project reports. We did not use GitHub since we wanted a private repository and did not need the web features of GitHub (or Bitbucket) for this phase of the project.

This project required using some large datasets that are openly available from the [National Centers for Environmental Information \(NCEI\)](#), in particular topography and bathymetry data for running the tsunami model and tide gauge data. We downloaded and archived some of this data on the UW account, but did not put it in the git repository since these did not need to be under version control. Instead we wrote shell scripts to rsync this data to each collaborator's laptop or other computers as needed. (rsync is a utility on unix-like systems to transfer and synchronize files). These scripts were kept in the git repository. Similarly we wrote scripts to rsync simulation results from the computer where the simulation was performed back to this account, along with some metadata. The new methods being developed for tidal uncertainty were implemented in Python code used for postprocessing the simulation results. One collaborator was doing most of the simulation runs, on several different computers, while another was developing and testing the postprocessing code, so rsync'ing the necessary data between laptops via the campus account was convenient and insured that it results were archived as we went along.

The shared campus account was also used to host webpages so that the visualizations produced from each simulation could be viewed by all collaborators. These webpages were also eventually used to share results with the project sponsors and reviewers of our preliminary report.

The private git repository was also used for writing the final report in LaTeX and collecting the final figures to go into the report. The third collaborator, who was less involved in the coding, was not completely comfortable with git and so we also used Dropbox for sharing and commenting on drafts of the report, but all changes to the LaTeX were made in the git version.

The final report was made available to the public by depositing it [in the UW ResearchWorks Archive](#).

When we started working on the journal paper, we created a private Bitbucket git repository for collecting the code specific to the paper and for the LaTeX file. Bitbucket is similar to GitHub but offers free private repositories, requiring a paid account only for public repositories. By contrast, GitHub offers free public repositories, and charges for private repositories. The interfaces are similar and it is easy to transfer a git repository between them, or maintain copies on both services, so it is often convenient to use both for different purposes. The submitted preprint was also posted [on the arXiv](#), a widely-used preprint server from which it was available with open access.

The referees requested changes to the paper and some figures needed to be redone, which was easy to do since we had produced all figures with scripts in the git repository. The revised paper was developed in this same repository, along with the authors' responses to the referees.

After the revised paper was accepted by the journal, we created a new public GitHub repository for the code and small datasets needed to reproduce some of the figures in the paper that illustrated the basic methodology. This repository was also linked to Zenodo, and a GitHub release was performed that triggered automatic archiving of a zip file of all the code at that stage, and assignment of a [DOI](#).

In addition to the test problem for which we shared code, the final paper also contained some figures with results from the overall project, the probabilistic maps produced for Crescent City, CA using this methodology. Reproducing these results would require running roughly 100 tsunami simulations. We are fairly confident that we have all the code and data to reproduce these results if required, but we have not made this publicly available.

Pain points

Using rsync for large datasets worked fine once we figured out a good workflow and scripts, but is not ideal. A version control system like git that works well for large quantities of data would have been very useful.

Some data could not be shared and we also had to be careful about sharing preliminary results since emergency managers and the agencies involved are very sensitive about publicizing risk maps for specific communities before they have been properly vetted and agreed on.

Key benefits

This workflow proved to be very valuable for this long-term project in which many parts of the code and methodology were evolving. The initial project was followed by additional funding for a [Phase II](#), in which the focus was on studying current velocities rather than only flow depth. This required re-running all the tsunami simulations with a modified version of GeoClaw. Having done all the initial work via scripts archived under git, it was relatively painless to redo these runs. In the meantime other changes had also been made to the GeoClaw code, and having both our code and GeoClaw under version control was very useful when comparing results.

Questions

What does "reproducibility" mean to you?

There were two distinct aspects of reproducibility important in this work. The original development of new techniques was performed in the context of a project that went on for several years and required running many tsunami simulations with the GeoClaw code for the probabilistic studies, each of which took several hours of computing time and produced large quantities of output data. During this time frame the GeoClaw software was evolving, along with our methodologies. [GeoClaw](#) is openly developed on the GitHub site. We needed to be able to compare new results with those computed previously, and be able to identify what changed in the software or our code in between, if necessary. For this aspect the goal was not to openly share all of our work or the results (nor were we allowed to, due to the nature of the project), but we needed to be able to reproduce results ourselves if necessary and keep the code under version control to identify changes.

The other aspect is that we wanted the particular new method written up in the journal paper to be accompanied by the Python code that implements the method and a sample set of data that was used to produce some of the figures in the paper. In this context we wanted the figures to be reproducible by a reader using this code, in hopes that this would aid others in understanding the methodology and adapting the code to solve their own problems.

Why do you think that reproducibility in your domain is important?

For researchers who develop new methods and algorithms, it is often important to be able to see the details that are in the code but don't make it into the paper, both to better understand the work and to find potential errors. It also facilitates comparing different methods for the same problem.

In natural hazards modeling, the simulation results may be used by engineers or policy makers to make decisions with public safety implications. Transparency and reproducibility are important aspects of accountability.

How or where did you learn about reproducibility?

My interest in the topic came out of frustration with the publications in numerical analysis (including my own) where it was impossible to reproduce published results or fully understand the implementation of new algorithms they describe. I became proficient with git initially through involvement in open source software projects.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

- Convincing collaborators to learn and use a common set of tools is sometimes a challenge, and some researchers are more willing to share code and data than others.
- Some input data and/or results can not be shared publicly, so it may be necessary to selectively share data and perhaps have both private and public repositories.

What do you view as the major incentives for doing reproducible research?

- Ability to easily modify and build on past work.
- Ability to compare new approaches or software with past versions and determine what changes made a difference in results.
- Facilitates collaboration.

Are there any best practices that you'd recommend for researchers in your field?

- Using version control of some sort is the single most important first step.

- Make a habit of cleaning up code used to produce final results so that it's well documented and all the necessary steps are clearly laid out. Then run through them from scratch if possible to insure that it works. Even if you don't plan to share it with others, your future self will thank you.
- If you do share code and/or data, do so in an archival repository that issues a DOI, and attach a license.

Would you recommend any specific resources for learning more about reproducibility?

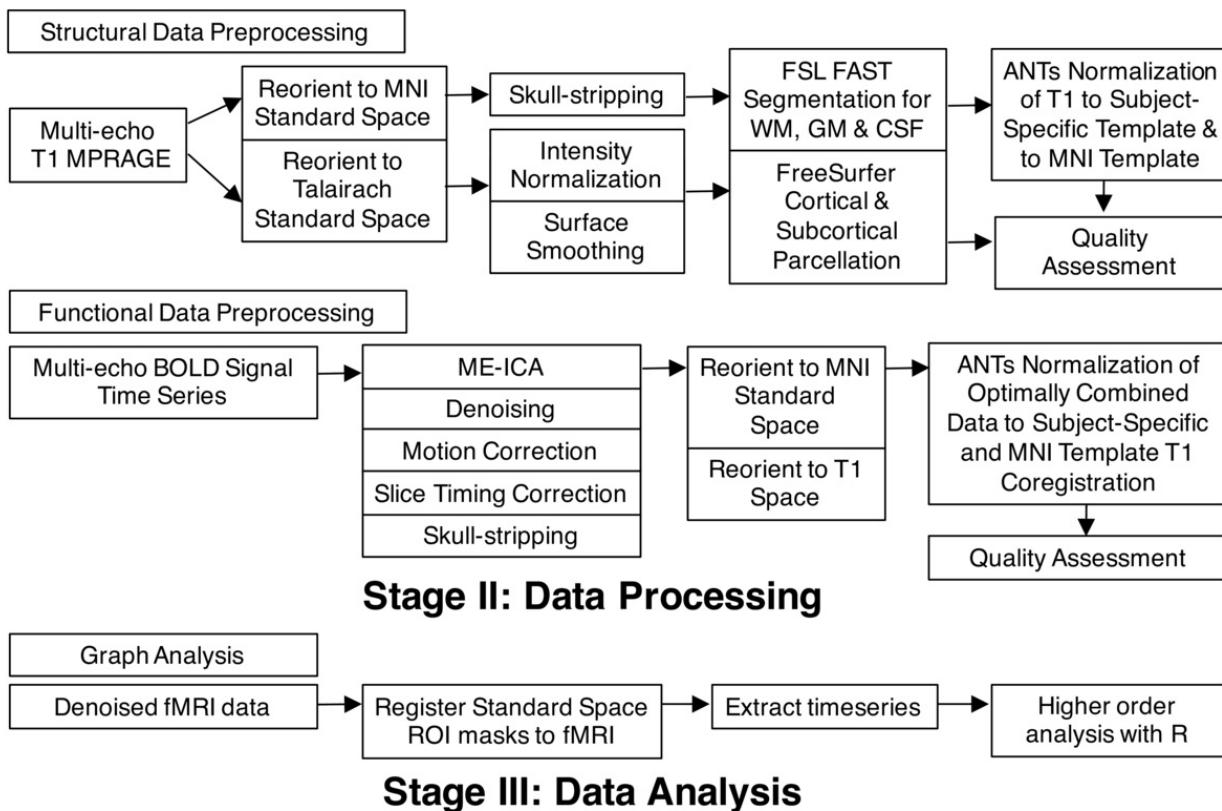
- The UW eScience Institute Reproducibility and Open Science Working Group has developed [some guidelines](#).
- The 2012 [ICERM Workshop on Reproducibility in Computational and Experimental Mathematics](#) resulted in a [final report](#) with recommendations and additional links.

A Reproducible Neuroimaging Workflow using the Automated Build Tool make

Tara Madhyastha, Natalie Koh and Mary K. Askren

We are Tara Madhyastha, Natalie Koh and Mary K. Askren, affiliated with the Integrated Brain Imaging Center (Radiology). In this project, we use functional magnetic resonance imaging (fMRI) to interrogate the function of the brain in elderly individuals to understand how physiological changes relate to mild cognitive impairment and might be predictive of dementia. Specific challenges to processing fMRI data are that the number of steps involved are complicated and can have significant impact on the results, and to achieve the best results, components from many different software packages must be combined. To do this, we need to structure our own pipelines and quality assurance. Data produced using this workflow are visualized in Bach et al. (2016).

Workflow



One of the major advances of systems neuroscience in the last two decades is the discovery that, at rest, the correlations of the blood oxygen level-dependent (BOLD) signal in cortical regions of the brain (measured non-invasively using fMRI) describe large-scale networks. These networks are altered in a variety of psychiatric and neurodegenerative disease; however, accurate measurement of networks is difficult in part because of a variety of artifacts, including subject motion.

The purpose of this workflow is to preprocess fMRI data for 54 subjects, examining the quality of the data, and generating time series of specific regions of interest. An earlier version of this pipeline (with less exhaustive quality assurance and with a different algorithm for aligning subjects to a standard template) was used to compare a traditional and an improved method for noise removal, generating time-varying correlation matrices for input to an exploratory visualization paradigm. The current preprocessing pipeline supports many primary analyses that are in progress.

Preparation of data for functional connectivity analyses involved the preprocessing of both structural and functional data using a combination of open source neuroimaging toolkits that run in the UNIX environment (Debian Wheezy). These programs include [FSL](#), [FreeSurfer](#), Advanced Normalization Tools ([ANTs](#)) and Analysis of Functional Neuroimages ([AFNI](#)).

While the details of this processing, described below, are not necessarily important to understanding this case study, the important point is that best practice processing of neuroimaging data requires multiple steps to be performed using different software packages, and a complete reporting of the details involved in these steps is crucial for reproducibility.

Structural data was processed in two ways. First, we used the high resolution structural image to align the lower-resolution functional image (anatomical-functional co-registration). We also used ANTs nonlinear registration to create a study-specific template for all structural images. Second, we used FreeSurfer to create a cortical/sub-cortical parcellation of the brain. Functional data was preprocessed using multi-echo independent components analysis, or ME-ICA (using AFNI's meica.py script; see Kundu, Inati, Evans, Luh, & Bandettini (2012)). Removal of sources of noise in fMRI data is a huge issue, and the ME-ICA uses fMRI images acquired with different parameters (echo times) to automatically classify sources of variation in the fMRI data as BOLD-related or noise, producing a denoised dataset. It also produces an "optimally combined" image that can be denoised using more traditional techniques. ME-ICA also performs the standard steps of skull-stripping (removal of non-brain tissue from the image) and correction for motion and timing of the acquisition of different slices. The optimally combined output from ME-ICA was aligned to Montreal Neurological Institute (MNI) standard space via the structural registrations and study-specific template defined earlier. Preprocessing steps are fully automated using Make (Askren et al., 2016). Make is a UNIX utility which takes an expression of workflow in the form of target files and their dependencies (a "Makefile") and creates a graph describing what work needs to be done to "make" a target file.

Quality assessment (QA) of preprocessed data involved the manual checking of reports for each study subject comprising images, animated GIFs (data concatenated temporally) and statistics that were generated at both the intermediary and final steps of the workflow. This QA was performed by Natalie Koh, and took approximately 20 minutes per subject. Aberrations and outliers in data were logged in a spreadsheet, and efforts were sometimes made to re-process data that had been flagged for poor quality. Special attention was given to making sure that the skull-stripping for both structural and functional had been performed properly, that registrations were acceptable, and that data had been adequately corrected for motion. Motion, in particular, can significantly bias estimates of time series correlations. As such, motion parameters, statistics and graphs plotting framewise displacement (FD) and delta variation signal (DVARS) over time were looked at carefully to ensure that data was acceptable for further analyses. Ultimately, one subject had to be excluded due to the mean displacement of the subject's data greatly exceeding the absolute threshold of 2mm. For this subject, motion correction using ME-ICA also failed to correct for movement and the variation in BOLD signal.

Finally, after QA, we extracted the BOLD time series from specific pre-selected regions of interest (ROIs), using the denoised fMRI data and the precomputed and checked spatial normalization to the MNI template. These time series were combined in a comma separated value file and moved to a separate computer for Tara Madhyastha to process using R scripts.

Although this processing pipeline is well set up for us to replicate our own analysis, it is less straightforward to share. Raw data is not currently online, although it will ultimately be archived as part of the Adult Changes in Thought study. All main software to execute the pipeline is available online. However, there are some minor scripts that we have written that are not available online. To fully replicate the pipeline off-site, we would need to supply these scripts and the versions of all software that we used, along with the Makefiles. It would be easiest to do this by supplying a clean copy of the working analysis directory for the subjects from our site, so that the remote site could edit pathnames to specific packages as necessary and rerun our workflow.

Documentation for Makefiles is embedded in our Makefiles either using comments (because we have relatively standardized target names and file naming conventions). We have been recently trying to adopt standards for documentation of naming conventions and help systems. Thus far, however, processing of primary targets has been relatively intuitive and this has not been as important as developing extensive QA and provenance.

Processing is not currently online because the workflows are too computationally expensive to run on a single multicore server, and we often have to trade off disk storage for processing power when deciding where and how to parallelize. We also lack programming resources to develop web-based interfaces for these pipelines. Writing and running the makefiles is less difficult than determining what they should look like, so web-based sharing of workflows has not been a priority.

Pain points

An important part of neuroimaging workflow is checking the quality of automated processing steps. However, when manual corrections are necessary it is less clear how to record them in such a way that they can be completely replicated. We currently maintain a spreadsheet with this information, but clearly corrections introduce problems with reproducibility.

Generating figures for papers can involve significant handwork. For example, to assemble a montage of brain slices that show statistically significant results might involve generating several screen dumps of different coordinates, setting the minimum and maximum manually to be consistent across images that will share a common colorbar. Once the researcher has decided upon the images and coordinates to include, this process cannot always be scripted, and the relevant parameters must be carefully recorded.

Tools such as Rmarkdown and pandoc (or Sweave, odfweave) allow fabulous integration of statistical analysis and text. However, researchers in many labs rely heavily upon Microsoft Word's "track changes" feature to edit papers, making it difficult to entirely couple paper generation and statistical analysis. Thus, there is an unnatural separation of generation of methods text and tables from assembly and editing of the final paper.

Key benefits

Our workflow is unique compared to state-of-the-art in the neuroimaging field because we use Make to describe dependencies. This ensures that only the parts of the workflow that need to be re-executed (because of a failure, or the change to an intermediate result) will be rerun. Practically, avoiding unnecessary computation time is important in neuroimaging workflows that can take many hours or days to run on small clusters of computers. Using scripts written in languages that do not inherently express dependencies (such as bash or Matlab) it is easy to introduce errors when modifying scripts to execute only uncompleted work.

There are two key advantages of Make over other neuroimaging workflow systems (such as nipype and LONI Pipeline) which also support a dependency graph model. First, Make does not require the core neuroimaging programs to be "wrapped", or surrounded by interface code, that adds development time to the process of designing a workflow and slows adoption of new versions of software programs as they become incompatible with their wrappers. Second, Make builds the dependency graph implicitly from target files, rather than requiring the graph to be drawn or programmed explicitly.

Key tools

There are two key tools that support reproducibility in our workflows. The first is Make, described above. The second is R Markdown, which we use here in combination with Make to generate complex QA reports including statistics, QA images, and graphs generated using R. In workflows not described we use R Markdown to generate data provenance reports suitable for inclusion in a methods section, combining English text describing the workflow with parameters obtained automatically that include software versions and software and scanner acquisition parameters.

Questions

What does "reproducibility" mean to you?

Reproducibility in this context means that given the specifics of the software versions that we are running, our workflow, and the source data, a scientist can obtain the same results from our data that we can.

Why do you think that reproducibility in your domain is important?

Many results in neuroimaging are highly dependent upon the methods used to process the data. It is often the case that scientists discover that some source of noise introduces artifactual findings, or that data that were previously dismissed as noise contain information. Therefore, it is vital to maintain a programmatic description of how data are processed so that findings can be replicated with the same data.

How or where did you learn about reproducibility?

The ideas behind make are from computer science classes. Most other practices described here have been developed together in IBIC based on experience.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

Cognitive neuroscience is an inherently interdisciplinary field, which means that the level of technical skill as well as core domain knowledge varies across researchers within a team. Practices encouraging reproducibility (e.g., scripted workflow) that are second nature to members of one field may be completely novel and a bit intimidating to members of another field. This can lead to slow adoption of best practices.

What do you view as the major incentives for doing reproducible research?

One of the major incentives is that it saves our future selves time trying to figure out what we did after we've forgotten.

Are there any best practices that you'd recommend for researchers in your field?

We recommend automated, scripted workflow as much as possible with minimal hand-editing to avoid human-induced bias (e.g., in boundary editing). Time invested in developing clear, consistent, and maintainable workflow is rarely misspent.

Would you recommend any specific resources for learning more about reproducibility?

The Organization for Human Brain Mapping (OHBM) recently formed a Committee on Best Practices in Data Analysis and Sharing (COBIDAS) to identify best practices of data analysis and data sharing in the brain mapping community. The committee is expected to publish a final report on these practices in the near future, and this may prove to be a useful source for individuals interested in learning about reproducibility in the context of neuroimaging research.

References

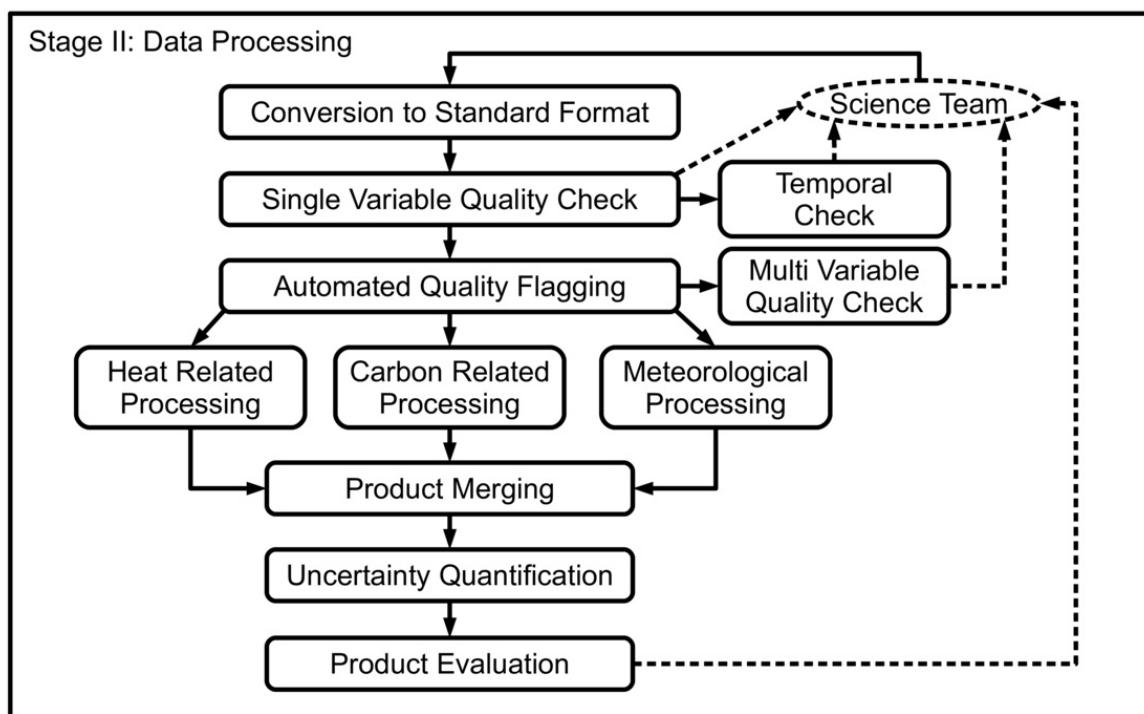
- Askren, M., McAllister-Day, T., Koh, N., Mestre, Z., Dines, J., Korman, B., ... Madhyastha, T. M. (2016). Using make for reproducible and parallel neuroimaging workflow and quality assurance. *Frontiers in Neuroinformatics*, 10(2).
- Bach, B., Shi, C., Heulot, N., Madhyastha, T., Grabowski, T., & Dragicevic, P. (2016). Time Curves: Folding Time to Visualize Patterns of Temporal Evolution in Data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1).
- Kundu, P., Inati, S., Evans, J., Luh, W.-M., & Bandettini, P. (2012). Differentiating BOLD and Non-BOLD Signals in fMRI Time Series Using Multi-Echo EPI. *Neuroimage*, 60(3), 1759–1770.

Generation of Uniform Data Products for AmeriFlux and FLUXNET

Gilberto Pastorello

I'm [Gilberto Pastorello](#), a [Research Scientist](#) at Lawrence Berkeley National Laboratory, doing research on life-cycle management of scientific data, encompassing data and metadata structures and linkages, data quality and data uncertainty quantification, and end-to-end data systems. The part of my work described here involves development of data processing pipelines and data management solutions within the environmental domain. This work is done for the [AmeriFlux](#) and [FLUXNET](#) research networks.

Workflow



Multiple Science Teams collect carbon, water, and energy fluxes from over 800 field sites across the world. Currently, more than 400 of these sites share their data with regional networks such as [AmeriFlux](#), allowing the creation of data products with a global scope for

the [FLUXNET](#) network, such as the [FLUXNET2015 dataset](#). These sites are operated independently and methods for data collection, processing, and data quality control by the Science Teams can vary significantly. Our workflow aims at processing these heterogeneous datasets to generate data products that are comparable across these sites. A general view of the steps in our workflow is shown in the figure. In this context, reproducibility is strongly related to identifying and documenting data quality control checks, parameterizations for processing, sequences of correction steps, and data filtering.

The pipeline is executed every time new data are sent to us by the Science Teams. We keep track of multiple submissions with a combination of simple incremental counters for versions and timestamps -- logs of data transfers are stored in a relational database. The frequency of data submissions can range from daily to yearly updates from a Science Team. All executions of the pipeline generate a version, with successful executions being made public after being vetted by the Science Teams.

The first few steps are related to data quality and aim at identifying serious quality issues and making data quality more uniform across sites. Specialized algorithms and visualization methods are used in these steps. Automated generation of flags and manual inspection of datasets are done first, followed by the compilation of metrics about the data and checks. Any decision to change the datasets is first shared with the Science Team and confirmed before being executed. Major issues are identified and solutions are developed in collaboration with Science Teams. A custom issue tracking system developed by us is used to keep track of interactions with Science Teams and the issues being addressed in the datasets. The information in this issue tracking system is private and accessible only to our team and the Science Team for the field site providing the data, but reports and status information from this system can be made available with a published data product. Changes to data are also documented in log files and quality flags that are added to the datasets themselves, these being made available along with data products. Changes and corrections generate a new version of the input data and the pipeline starts again from this new version.

The central portion of the workflow includes the processing steps for heat, carbon and micrometeorological variables. The parameters used to configure these executions are stored with the datasets, and specialized checks are also executed within each step. The results from most of these checks are stored as quality flags added to the datasets. Versions of the code used for these steps are also recorded in the dataset's metadata.

The product merging step reformats the data into common structures and combines quality information into quality flags at a higher level of abstraction, to simplify using the datasets. The uncertainty quantification step generates quantiles representing uncertainty intervals originating from each of the processing steps, also becoming part of the data products.

The more multiple team members understand and can generate products, it is much more likely that problems will be identified early, questions from external members of the community will be answered more easily, and funding agencies will have documented product releases (data and software), which can be combined with other types of publications in assessments of scientific impact.

Pain points

The execution of this pipeline can happen several times before acceptable results are reached, and it also includes several interactions with the Science Teams potentially resulting in changes to the input datasets. The changes can be applied by the Science Teams or by our team. With multiple iterations, the changes that are needed to make a dataset correct are often spread across many versions of the input data. Consolidating these changes is particularly difficult, especially when the changes were applied to versions leading to unsuccessful runs.

More specifically, one version related challenge is keeping versions consistent across multiple processing instances. We use mapping of versions of inputs to outputs, but with outputs influencing what a new version of the dataset looks like, these mappings are not always straightforward.

Another challenge is related to the multi-source nature of our datasets. Many of these datasets span over a decade of data collection, processing, and curation by multiple people. Since the teams are distributed and follow potentially different data collection and processing protocols, fully automated reproducibility is difficult to be achieved. A workaround has been to document the choices in collection and processing data in such cases.

Finally, combining reproducibility issues with credit issues for datasets is another challenge. Giving proper credit to data providers and data processors/curators is an essential part in large data sharing efforts. Comprehensive data sharing policies help with assigning proper credit. Multiple versions of the datasets are created over time, and close communication with the Science Teams is important to allow tracking all contributors.

Key benefits

Without the use of versioning and issue tracking coordinated between code and data, it would be unfeasible to fully document choices for the datasets, many of which affect important aspects such as data quality or uncertainty.

Key tools

Data and software versioning are certainly the main practices adopted in this case study. We are currently assigning versions to data and software in a coordinated way, to allow assessment of changes to data and code. We use a private Subversion server for code version control. We developed a custom Web-based system called FIT (Flux Issue Tracking) for tracking the interactions with Science Teams and data quality issues for datasets. FIT was later also adapted for tracking code related issues. It was implemented in Django/Python using PostgreSQL databases.

For the processing pipeline, we combine specialized code written in multiple languages. Automated quality checks and data product generation steps mostly use C implementations, with a couple of steps implemented in Python and MATLAB; visual data quality checks are implemented in MATLAB and Python; and driving code and generation of final data products is done using Python, with extensive use of the NumPy and SciPy packages.

Questions

What does "reproducibility" mean to you?

For this case study, reproducibility means being able to apply standard methods to process heterogeneous datasets to generate comparable data products. The data sources for our processing are distributed across very different ecosystems, with data acquired, processed, and quality checked by different teams. The data products we generate from these datasets need to be in the same scales and have comparable levels of quality, representativity, etc.

Are there any best practices that you'd recommend for researchers in your field?

While version control is widely adopted for software code, this is not necessarily the case for data. Consistently keeping track of versions of a dataset at a minimum helps with data changes and is well worth the extra effort. Another lesson we learned early was that issue/bug tracking ideas and tools can simplify data management activities, and can also be valuable in building the history of a dataset and generating its documentation.

Would you recommend any specific resources for learning more about reproducibility?

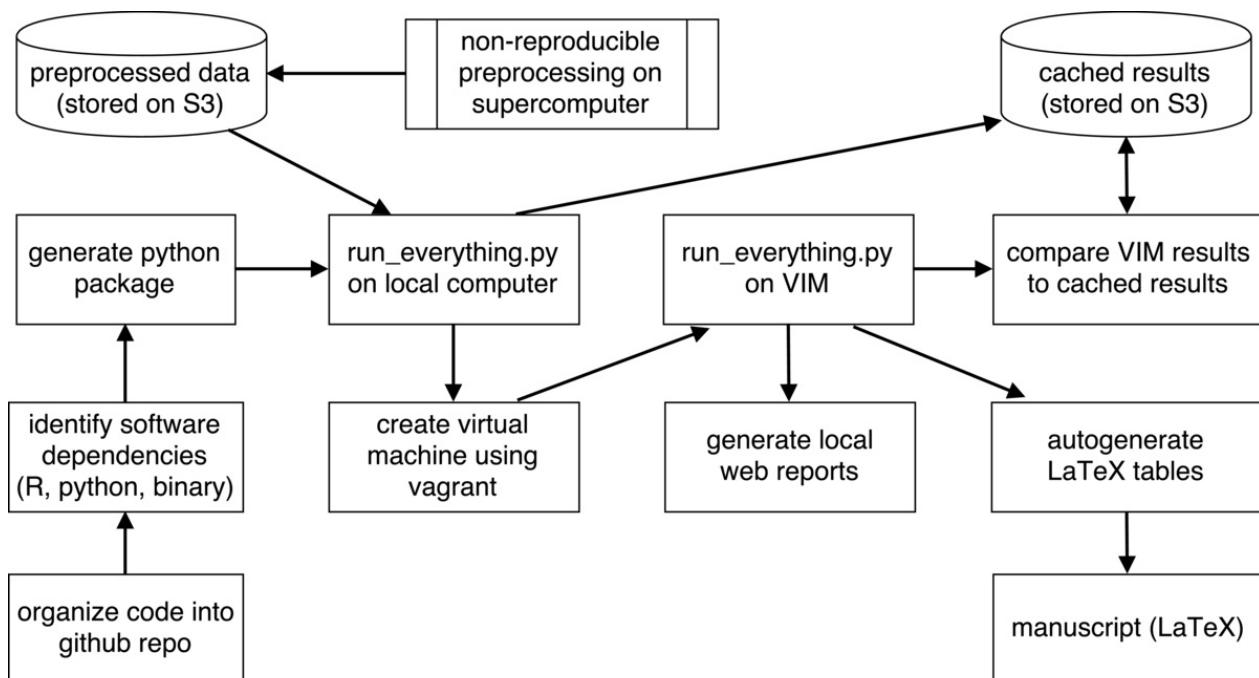
While not always featuring fully developed practices or tools, reproducibility scientific events (and their published proceedings) often showcase new and interesting ideas. Events featuring latest research in provenance include, for instance, the International Provenance and Annotation Workshop (IPAW) and the International Conference on eScience.

Developing a Reproducible Workflow for Large-scale Phenotyping

Russell Poldrack

My name is Russell Poldrack, and I am a professor in the Department of Psychology at Stanford University. My work uses neuroimaging, genomics, and behavioral studies to examine the brain systems involved in decision making and executive control. Many of our workflows use high-performance computing due to the large data and complex nature of the workflows. This particular case study focuses on analysis of a study known as the "[MyConnectome study](#)", which involved intensive data collection from a single individual over the course of 18 months, including neuroimaging, genomic, metabolomic, and behavioral data. This large heterogenous dataset raised a number of new challenges for reproducible data analysis.

Workflow



Stage III: Data Analysis

This workflow is meant to outline the analysis of a complex dataset including neuroimaging, behavioral, transcriptomic, and metabolomic data (<http://www.myconnectome.org>). The data were collected over the course of 18 months from a single individual, and will be made fully available online upon publication of the manuscript via the [OpenFMRI project](#). The data are

released under a public domain dedication, meaning that anyone can do anything they wish with the data, with no restrictions on redistribution or requirements of attribution. I chose this approach because I feel that it will provide greatest degree of utility for the data.

The data processing stream was initially built in a non-reproducible manner on a single laptop. After completing this, I became interested in generating a reproducible version of the workflow so that other researchers can exactly reproduce the analyses. A challenge of reproducible analysis in this study is that the raw data are very large (several TB including the raw genomic and neuroimaging data). These data are being made available to any who wishes to use them, but it is not possible to easily provide reproducible workflows for these operations because they require large-scale supercomputing resources. For the processes that require supercomputing (such as genome alignment for RNA-sequencing data and surface-based parcellation for MRI data), we have shared most of the code used to complete these workflows. Another complication of full reproducibility is that some of the preprocessing operations were performed by another laboratory, using code that they are not currently willing to share openly. Thus, we made the decision to focus on building an open reproducible workflow that encompasses as much as possible of the processing stream, using preprocessed data downloaded from an online archive.

The goal of this process was to create a completed automated analysis stream that requires no manual intervention. The workflow uses a number of tools, including python, R, MATLAB, and the Connectome Workbench (a domain-specific software tool for analysis of neuroimaging data). I started by generating scripts to perform each of the operations, but ultimately decided to generate a single python package to coordinate the entire workflow (available at <https://github.com/poldrack/myconnectome>). The first operation of this package is to download the preprocessed data from Amazon S3, using the boto package in python. We then perform additional processing of each of the different data types.

For the neuroimaging data, we developed a set of python functions to extract and summarize connectivity measures between different brain regions. These analyses included assessment of connectivity between regions using both standard correlation measures as well as regularized partial correlation (using the R QUIC package). Network analyses were performed using the Brain Connectivity Toolbox, and visualized using the Cytoscape software package.

For the transcriptome and metabolomic data, analyses were performed using a set of Rmarkdown scripts executed using R. These analyses included the identification of coexpression networks using the Weighted Gene Coexpression Network Analysis (WGCNA) package; eigengenes identified from these network were then used in subsequent analyses. The gene networks were annotated using DAVID. Metabolomic measures were clustered using affinity propagation, and annotated using IMPALA.

The outcomes of each of the foregoing analyses were saved and used to compute time series correlations between each of the measures across all domains (for a total of more than 20,000 statistical tests), using the R forecast package. These are then summarized in a web report generated using Rmarkdown. Currently the only test is one that compares the results of the full workflow to a set of results cached on S3. Documentation of the code is minimal.

After implementing the workflow on a single system, I then implemented it on a virtual machine in order to allow anyone anywhere to run it. I used the Vagrant software package to provision a virtual machine with all necessary requirements. Once installed, the user can run the entire workflow with a single command. In addition to running the entire data analysis workflow, the virtual machine also includes a web server that provides access to the results of all of the analyses, along with a data browser for the detailed results. This system is identical to the one exposed publicly at <http://results.myconnectome.org>. Documentation for installing and running the software is evolving.

Pain points

A number of pain points were encountered in the development of a reproducible analysis workflow. First, there were a number of processing stream operations that could not be implemented in this manner. In particular, some of the preprocessing operations required high-performance computing resources, which could not be generalized due to specifics of job submission systems. Second, there was a substantial amount of extra work necessary to generalize the code to work on an arbitrary system, primarily involving the identification and resolution of software dependencies. A third pain point involved the identification of appropriate technologies for sharing of a reproducible workflow. We have used a VM provisioned using Vagrant, but there are many other approaches that one might use. Finally, we struggled with identifying what level of user at whom we were targeting the project. We ultimately decided to make it easy to use for non-power-users, which required substantial extra work.

Key benefits

The reproducible workflow provides a number of important benefits. First, it provides a degree of detail that could not be feasibly included in the publication. Second, it increases the degree of trust amongst others in the field in the results that are presented in the publication. Third, it provides an example for others in our subfield of how to implement reproducible shared workflows.

Key tools

I have used [Vagrant](#) to allow any user to easily provision a virtual machine that includes all of the necessary dependencies to run the workflow (see <https://github.com/poldrack/myconnectome-vm>).

Questions

What does "reproducibility" mean to you?

In the context of my case study, "reproducibility" means the ability to exactly reproduce the analysis workflow that was used to obtain the results reported in a manuscript. More generally, I take the term to also encompass the consistency of results across different workflows or datasets.

Why do you think that reproducibility in your domain is important?

The workflows used for neuroimaging data are highly complex with a great degree of analytic flexibility, which raises concerns regarding the reproducibility of results. We desperately need a greater degree of transparency in order to make research in our domain more reproducible.

How or where did you learn about reproducibility?

I think I primarily learned from negative examples; that is, from seeing other researchers whose research findings rely upon code that is not openly available and data that are not shared.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

Human subjects issues and concerns about scooping are often raised here, but I think these are red herrings. The primary pitfall is that reproducible research practices make it harder to obtain splashy findings that get high-profile publications.

What do you view as the major incentives for doing reproducible research?

Increasing trust in one's research.

Are there any best practices that you'd recommend for researchers in your field?

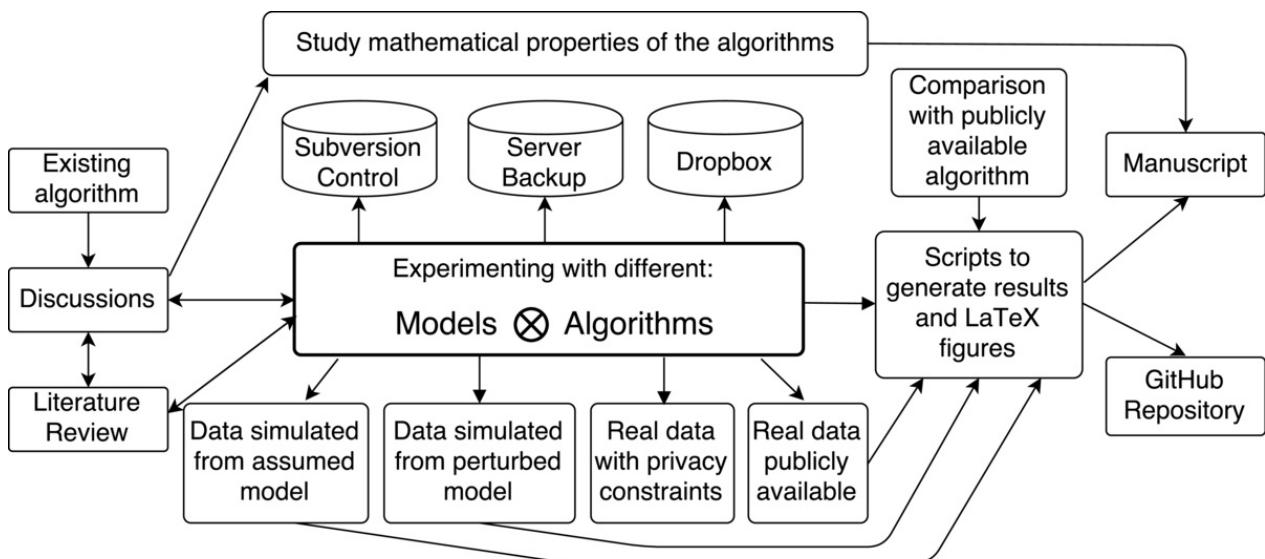
Using version control software, and open sharing of data.

Developing and Testing Stochastic Filtering Methods for Tracking Objects in Videos

Valentina Staneva

My name is Valentina Staneva and I work as a data scientist at the eScience Institute at University of Washington. I am an applied mathematician who develops methods to extract information from diverse datasets. Most of my experience is in the domain of image processing and its biomedical applications. This case study describes the workflow of a particular project whose goal was to develop and test new algorithms for tracking objects in videos which aim to preserve the original structure of the objects. This research was done while I was a graduate student at the Center for Imaging Science, at Johns Hopkins University, and was motivated by the task of tracking heart motion in cardiac images. I believe it reflects a typical experience of an applied mathematician working on biomedical imaging problems.

Workflow



Stage II: Data Processing

This work follows a typical flow for problems in my field: motivated by an existing algorithm, we aimed to extend it to processing sequences of images (as opposed to a single image). Usually the process involves experimenting with different models of the data and different

inference algorithms (intertwined with discussions with my advisor and literature reviews). The various combinations of these models and algorithms can be tested on three main types of datasets:

- a dataset simulated from the assumed model: since the dataset is coming from the "correct model", this experiment is mainly testing the performance of the inference algorithms
- a dataset simulated from a model which deviates from the assumed model (in some interpretable way): this experiment is testing the robustness of the algorithms
- a 'real' dataset: a video of a moving object; this tests the applicability of this methodology in practice

This results in exploring quite a lot of different setups and most of the research time is spent at this stage (it can take from weeks to several months to implement and test a specific formulation). The work was performed on an account on a university server, which was sequentially backed up. I also used Subversion for version control and stored my files in a Dropbox folder (which has its own version control). Luckily, I never lost a file, even when our server got hacked.

In general the evaluation of these algorithms on real data is difficult. There are no standard testing datasets, and it is hard to design ones as different image sequences describe different processes, and some algorithms perform well in some situations and poorly in others. One usually considers a range of typical tracking hurdles and checks whether a given algorithm can overcome them. Since there is no one final metric to submit this requires storing all the results from all the experiments. In the end I saved all the code (written in MATLAB), data, and experiments in a folder, which provides everything necessary to regenerate the results in our manuscript with a few simple commands. When reviewers requested an additional plot to be included in our article, I could easily obtain it from the original data. We also selected a journal which does not prevent us from posting the preprint of the article elsewhere and stored all the supplementary materials on GitHub.

The workflow also contains a parallel path in which one studies the mathematical properties of the models and algorithms. For example, we aimed to develop algorithms which preserve the topology of the tracked objects, and we proved that our framework ensures that, thus eliminating the need to test this property in multiple cases. Sometimes it is possible to guarantee the performance of algorithms even without implementing them, which makes reproducibility of mathematical research quite easy!

Pain points

1) Private data: some of the motivation for this project was driven by the need to process a specific cardiac dataset and perform statistical analysis on the obtained results. I initially tested the algorithms on this dataset, however, eventually I was not allowed to use it in my publication due to some privacy concerns. I resorted to searching for a public cardiac dataset which turned out extremely difficult to find: a website which aimed to maintain a public database of cardiac images was permanently down as the creator left the field. One good source for public biomedical datasets is [MICCAI](#) conference challenges: they contain datasets on which to assess methods for solving very specific problems. The drawback is that sometimes the datasets are not complete, as they are designed to solve a particular challenge: for example, the dataset I obtained from an image segmentation challenge did not contain ground truth relevant to the tracking problem.

2) Volume (Storage): when processing video sequences, an issue simply arises from the size of the produced outputs. If I generate many experiments (which is inevitable when performing MCMC simulation) I have to store many videos. Without a ground truth, I could not store just a small measure of mismatch instead of the whole sequence. GitHub's policy does not accept files larger than 100MB. This caused difficulty when trying to upload even only the input data to the repository. This makes it hard to keep code and data together and easily accessible.

3) Randomness: working with Monte Carlo simulations results in different outputs every time the algorithms are applied. This requires the extra step of forcing the random number generators to produce the same sequence of random numbers. This procedure is not so simple when parallelization is involved: MATLAB (and similar scripting languages) usually do not have control over the order in which the separate threads are started and that results in extra randomness in the output. Further, attempting to generate multiple random streams simultaneously results in producing identical sequences of pseudo-random numbers (the seed is based on the current CPU time) which corrupts the Monte Carlo algorithm. One solution is to generate all random sequences that would be needed in the parallel threads in advance, but this requires modification of the programs themselves. Working with random outputs also makes it hard to generate unit tests: we only have asymptotic results of what the outputs should be, so it is difficult to set the confidence intervals for the outputs even with simulated data.

4) Backup: I used Subversion for version control of this project. I wanted to use the integration with the Nautilus file manager that Subversion was providing. It turned out it was buggy (it was a new feature at that time) and not all commits were recorded through the graphical interface: quite dangerous! I learned that it is more reliable to use explicit terminal commands with version control systems.

Key benefits

One of the main advantages of this workflow is that all the code was written in one language without resorting to external libraries and toolboxes. Usually core language functionality changes much more rarely than add-on packages, which makes software better sustainable in the long run and across platforms.

Our approach of encoding certain mathematical properties into the developed algorithms also makes the research more robustly reproducible under deviations of the original set-up.

Questions

What does "reproducibility" mean to you?

"Reproducibility" has two meanings for me:

- (1) "Exactly reproducible" - when a result can be regenerated exactly as suggested given the same set of inputs and parameters. For example: a manuscript is "exactly reproducible" when one can provide some scripts and environment which with a press of a button (or an explicit set of instructions) can generate all the figures and calculations in the manuscript.
- (2) "Approximately reproducible" - when a result or similar performance can be generated with similar or different methods than the one proposed on the same or possibly slightly different data. Often in science, the goal is to test a hypothesis and the methods to achieve this do not matter, it is actually better if the same hypothesis is supported through different approaches. Further, the data on which the study was performed might never be observed again, so it is not so important to reproduce the results on these data, but it is important to produce similar results on similar data. We are interested in the robustness of the methods and the conclusions, and a better term may be "robustly reproducible".

This case study directly addresses the first type of reproducibility, but it explores also a bit of the second interpretation.

Why do you think that reproducibility in your domain is important?

I find two main reasons for the importance of reproducibility in my domain:

- Personal: working with image data is often quite involved, and one does not want to do things twice, but it is often necessary, and in that case it is better to have an automated process to repeat experiments.
- Public: there is overabundance of algorithms and studies but it is hard to use them in practice, because there is no simple way to reproduce the results (one usually needs to reimplement the algorithms or redo the studies). So if one wants their research to be

useful outside their own group, they should first ensure it is reproducible.

How or where did you learn about reproducibility?

I have been learning by myself. I believe some short reproducibility workshops would have improved my experience substantially (for example, learning about git/GitHub, virtual environments and light virtual containers).

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

Working with biomedical data one often faces privacy and storage challenges. Another problem which I did not encounter but I know is persistent in the field is the use of too many external software packages to preprocess the data: some of them are supported only by specific operating systems, or require manual operation. This makes it challenging to automate the workflow. In an attempt to improve performance on large datasets, researchers often use elaborate C++ programs which are hard to interpret and extend.

What do you view as the major incentives for doing reproducible research?

I think the incentives should be personal and based on the understanding that this would improve the workflow and this is how research should be done. Unfortunately, the time and efforts spent on creating reproducible research are not very well awarded.

Are there any best practices that you'd recommend for researchers in your field?

Be reproducible every day! It is much easier to perform reproducible research than making your research reproducible (after it was already performed).

Would you recommend any specific resources for learning more about reproducibility?

Some useful resources have been compiled by the eScience Reproducibility working group:
<http://uwescience.github.io/reproducible/>

Other resources: <https://github.com/Reproducible-Science-Curriculum>

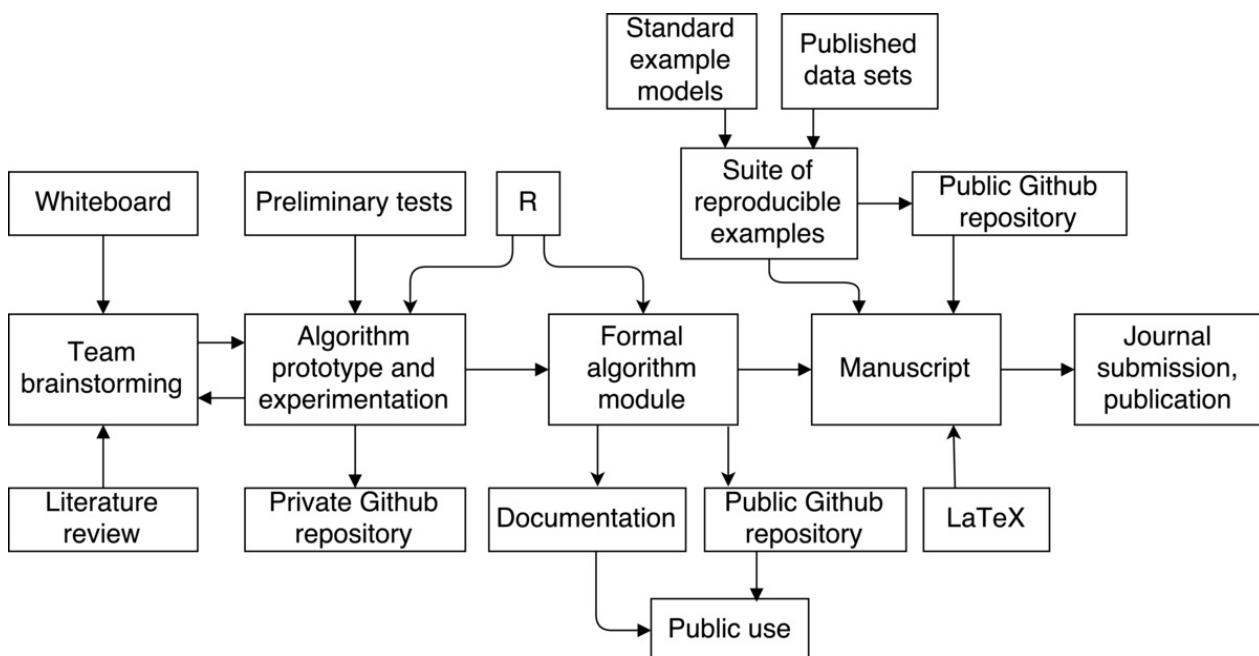
Coursera Class: <https://www.coursera.org/course/repdata>

Developing, Testing, and Deploying Efficient MCMC Algorithms for Hierarchical Models Using R

Daniel Turek

My name is Daniel Turek, and I'm an Assistant Professor in the Department of Mathematics and Statistics at Williams College. My area of research is computational statistics and algorithms, frequently with applications in ecological statistics. The workflow I describe is the process, from idea inception to publication, of creating an automated procedure to improve the sampling efficiency of Markov chain Monte Carlo (MCMC) sampling. MCMC is an accessible and commonly used statistical technique for performing inference on hierarchical model structures.

Workflow



Stage III: Data Analysis

The process begins with team brainstorming of how an automated procedure for improving MCMC efficiency could work. This is arguably the most fun part of the entire process. This involves anywhere from two to four people actually hitting the whiteboard to discuss ideas. Each of several sessions lasts a few hours. We review theory and literature between these sessions, too. This initial exploration occurs over one or two weeks.

A plausible idea is hatched, and now must be prototyped to assess effectiveness. The project lead implements the algorithm in R, since our engine for doing MCMC runs natively there. This works well for our team, since everyone is comfortable in R, and code may be shared and reviewed easily. We create a private GitHub repository where members of our team write/review/modify the algorithm. This is a private repo amongst us, since it's entirely experimental at this point, and not intended for the public. There is little (or no) documentation at this point.

Multiple iterations are possible at this stage, whereby ideas are implemented and undergo preliminary testing. Depending on the results of each iteration, we go back to the drawing board several times to figure out where the previous algorithm failed, or how it can be improved. Once again, we implement an improved version and test it using a small number of tests we've devised. This part of the process is time consuming, and potentially frustrating, as many dead-ends are reached. The path forward is not always clear. This process of revising and testing our algorithm may take three to six months.

Eventually, this process converges to a functional algorithm. All members of our team are satisfied with the results, and agree the algorithm is ready for a more professional implementation and hopefully publication.

One or two team members (those closest with the MCMC engine) do a more formal implementation of our algorithm. This implementation is added to an existing public GitHub repository, which contains the basis of the MCMC engine for public use. This step should only take a few weeks, since the algorithm is well-defined and finalized. Appropriate documentation is also written in the form of R help files, which are also added to the public repo.

The next goal is to produce a published research paper describing the algorithm and results. Towards this end, we assemble a suite of reproducible examples. These come from known, standard, existing models and published datasets, which are chosen as being either “common” or “difficult” applications of MCMC. A new public GitHub repository is created, and these example models and datasets are added in the form of R data files. Additionally, bash scripts for running our new algorithm on these examples are added, and also a helpful README file. The sole purpose of this repo is to be referenced in our manuscript, as a place containing fully automated scripts for reproducing the results presented in the manuscript.

Our reproducible examples include fixing the random number generator seed in the executable scripts, thus we can guarantee the same sampling results for each MCMC run (otherwise, a stochastic algorithm). However, the exact *timing* of each MCMC run will vary between runs and computing platforms, and hence the final measure of efficiency will vary, too. Thus, the exact results are not perfectly reproducible, but vary approximately 5% between runs.

Team members jointly contribute to drafting a manuscript describing our new algorithm, which presents the results from the suite of example models. This is jointly written by team members using LaTeX. The manuscript specifically references the repository of reproducible examples, and also explains the caveat in exact reproduction of the results — namely, that they will vary slightly from those presented, and why. The reviewers are nonetheless thrilled with the algorithm and reproducible nature of our research, and readily accept the manuscript for publication.

Pain points

The iterative process of devising and testing our algorithm is not well-documented or particularly reproducible. The only saving grace is that GitHub is used for versioning control, so in theory we could look backwards at previous work, if necessary. But in practice, the commit messages are short and not very descriptive, since everything is experimental at this point. No less, there's basically no documentation accompanying our code. It would be difficult to actually review previous versions of the algorithm or results, if it were necessary.

In addition, the fact that our set of “reproducible” examples are not perfectly reproducible is a small point of contention. We are conflicted to call these examples reproducible, since the results presented in our manuscript cannot actually be recreated. Team members agree that this appears to be unavoidable. We explain this in the manuscript, and call our results “reproducible” nonetheless.

For preparation of our manuscript, numerical results are manually typed into a tex document. Tools such as knitr and sweave exist for automating this process, which automatically incorporate numeric and graphical results directly from R into LaTeX. We opted not to use these tools to automate the interaction between R and our manuscript, since not all team members are familiar or comfortable using these tools. Preparation of the manuscript would have been simpler and less error-prone had we used these tools, which probably would have been a wise decision, but the learning curve deterred our team from doing so.

Key benefits

The most notably reproducible aspect of this project is the public repo containing input data and scripts for re-running all analyses appearing in our manuscript. This includes individual bash scripts for running each particular analysis, as well as a single “master” script which re-runs all analyses. A reviewer can easily reproduce (to within a small margin of error) all numerical results appearing in the manuscript, and researchers reading the ensuing publication have an easy path forward to using the algorithm themselves.

Questions

What does "reproducibility" mean to you?

In the context of my case study, reproducibility means that users / reviewers can re-create the results (improvements in MCMC efficiency) presented in our manuscript. However, the results will not match exactly due to small differences in algorithm runtime.

Why do you think that reproducibility in your domain is important?

Reproducibility is important so that others may verify the results given in our publication. This ensures that the results are genuine, and also gives a clear path forward for others to use our algorithm.

How or where did you learn about reproducibility?

Mostly through the use of GitHub, from colleagues at the University of California, Berkeley, and through general programming experience. No specific classes or workshops come to mind.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

In the area of computational statistics, there are not many barriers to reproducible research aside from ignorance or technical inability. However, this case study does highlight one genuine obstacle: that of performance differences between various machines and computing platforms, which will affect algorithm runtime, which factors into our measure of efficiency.

What do you view as the major incentives for doing reproducible research?

Primarily so that others may actually (and easily) verify our results, if they so choose.

Maintaining a Reproducible Database on Political Parties, Elections, and Governments

Werner Krause and Dag Tanneberg

This chapter appears only in the online appendix of the book *The Practice of Reproducible Research*. Please cite this online version of the book as: Kitzes, J., Turek, D., & Deniz, F. (Eds.). (2018). *The Practice of Reproducible Research: Case Studies and Lessons from the Data-Intensive Sciences* [Online Version]. Retrieved from <http://www.practicereproducibleresearch.org>.

Our names are Werner Krause and Dag Tanneberg. We graduated in Political Science with a focus on Comparative Politics. Currently, we are pursuing our doctoral degrees at the research department "Democracy and Democratization" located at the WZB Berlin Social Science Center, Germany.

Much of the research at our department revolves around political competition, elections, and the dynamics of democratic government. In the mid-1990s senior fellows of our department decided to set up a permanent infrastructure offering data on elections and governments to all department members in a standardized and easily accessible format. Originally, the data were used to observe and analyze the consolidation processes in the then still young Eastern and Middle European democracies. The project has since grown into a database that includes more than eighty countries around the world between 1945 and today.

The database tracks numerous aspects of political competition. For instance, we code lower house and presidential election results, government duration, cabinet size and composition as well as the ebb and flow of electoral alliances between political parties. Department members may output raw data and digitized copies of our primary sources. Alternatively, summary statistics such as turnout, the effective number of parties, measurements of disproportionality or government stability can be obtained from the database. Unfortunately, the database is not openly accessible yet. However, the plan is to go public by the end of 2017.

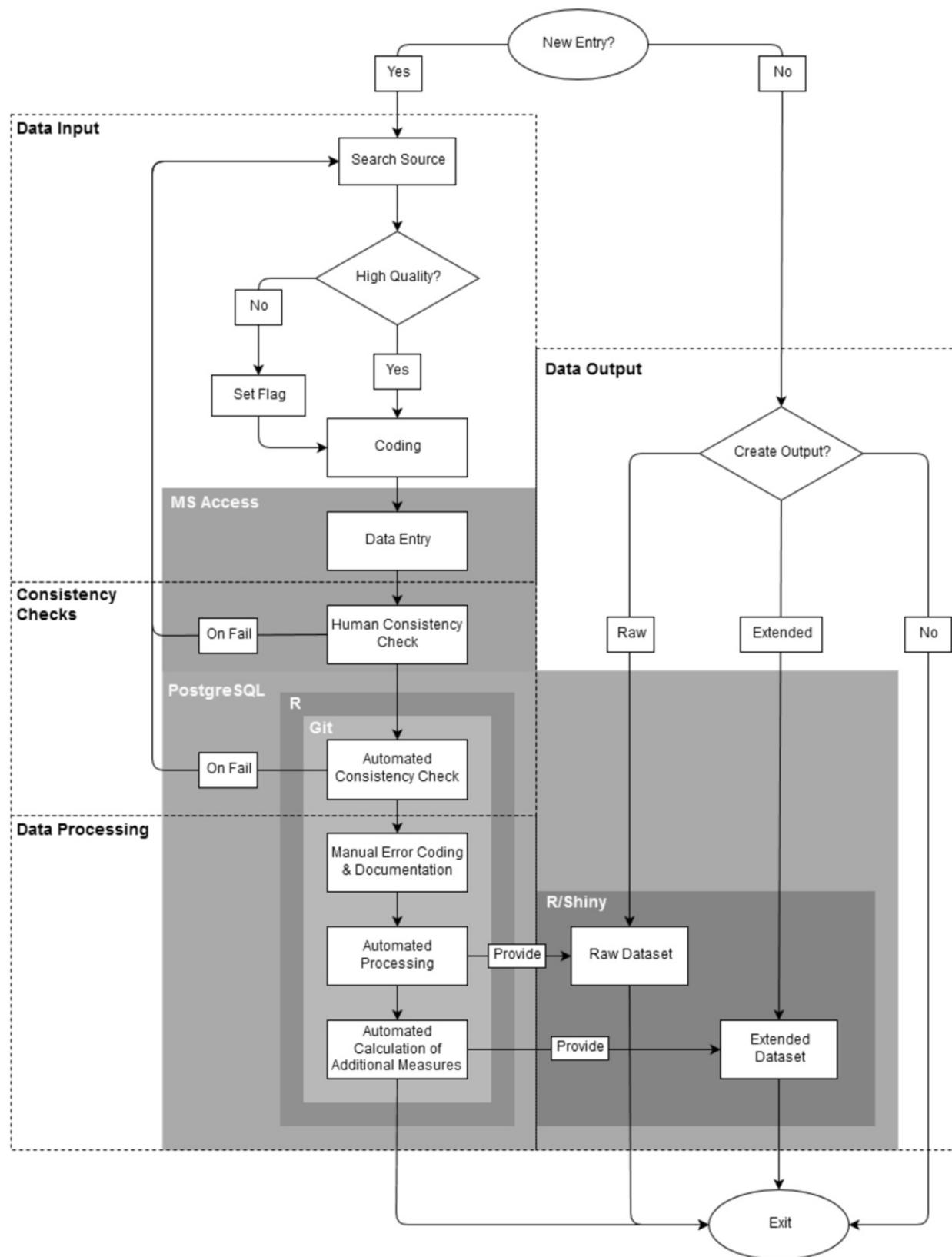
Back in 2011 and 2008 we were recruited as research assistants to continue this longstanding project. We introduced several substantive and technological innovations into the database that sum to more streamlined, less error prone coding and data management

routines. It was our goal to ensure transparency and reproducibility from the point of coding individual parties in single elections to the moment of generating summary statistics on every election covered between 1945 and today.

Broadly speaking, we define reproducibility as the responsibility to provide sufficient detail on scientific research such that others using the same data and methods will be able to replicate published results. Those may be single statistics, graphs, tables or even entire articles. Results that cannot be reproduced are neither open to critique nor revision - they are unscientific.

In the context of our database reproducibility becomes even more demanding. Since the quality of the data stored in our database affects numerous ongoing projects in and outside of our department every single piece of information should be reproducible. The challenge is thus to make the acquisition and coding of primary data as transparent to others as the standardized output we provide for secondary analyses.

Workflow



Our team consists of one research assistant, two junior researchers, and one senior fellow. The research assistant is responsible for coding and entering the data. Moreover, she performs some baseline consistency checks. The junior researchers oversee both automated consistency checks and data processing. The senior fellow supervises the project. We confront three basic challenges when collecting, coding, and processing data: a) to reduce coding errors, b) to maintain a high degree of intercoder reliability, and c) to

provide transparency on the entire decision-making process. Each is discussed in the following.

Our workflow has four separate steps. First, data have to be acquired and coded manually. Second, codings go through different human supervised and automated consistency checks. Third, the newly generated data are processed before storing them in our back-end database. Finally, information can be outputted from our database. Technically, each step is independent of all others.

First, our research assistant collects information on all upcoming elections and governments in the 82 countries covered by our database. Once a new observation is to be added to the database the assistant will compile sources on party histories, election results and/or government events. At this point it is crucial to critically evaluate the quality of a source. It must be factually correct and should offer information that is as disaggregated as possible. Both requirements are driven by the goal of correct and reliable coding. To ease the burden on the research assistant a list of high quality print and digital sources is included in a detailed codebook that accompanies the database. If high quality sources are unavailable, information from other documents will be accepted on a preliminary basis. Such entries are flagged, however, in order to update them once high quality sources become available. For example, elections will be flagged if results are available as vote and seat shares only rather than absolute numbers. Once a source has been identified it is coded manually following the guidelines of codebook.

Next, the research assistant enters the coded data into a *Microsoft Access* front end, and she saves a digital copy of the source (including all coding decisions) on a server that is accessible to all users. Access is neither free nor open, but it can be easily maintained and, more importantly, offers a user interface that makes data entry clear and easy. Via forms and reports the Access interface provides a standardized environment that reduces human error and increases intercoder reliability. Moreover, the Access interface performs basic consistency checks which enable the research assistant to evaluate the reliability of the selected sources. One such routine verifies that the sum of absolute votes equals the total number of valid votes as stated in the source. Another routine compares the total seat share of all government parties to the coded type of government. For example, minimum-wining coalitions with less than 50 per cent of the seats in parliament are immediately identified as problematic. Should any consistency check fail new sources have to be consulted in order to reach an almost error-free result.

After the data has been entered it is automatically exported to a *PostgreSQL* database. The database allows us to store the entire dataset and to put it under version control using *Git*. Changes to the database are documented on a daily basis and more complex automated consistency checks are performed at the same time. Those make use of the open source statistical software *R*. Using the *R* package *knitr* test results are saved and sent as pdf to

one of the junior researchers. Those reports document all new entries, but also all additional changes to the data, and all failed consistency checks. Consequently, the work of our research assistant can be easily monitored and potential coding errors are almost immediately spotted.

The automated reports serve as a basis for a manual classification and documentation of errors. Despite our best efforts to collect data only from high quality sources, certain inconsistencies are unavoidable. For example, sometimes we cannot identify the number of seats in parliament controlled by a coalition government. This happens when government parties competed in different electoral coalitions for which seats won by each party are not reported. *R* automatically identifies these and other cases more. Due to the wild variety of potential inconsistencies all suspicious entries are flagged and must be documented manually.

Finally, the data are processed in *R*. An *R* script joins data from several tables and generates a raw dataset which includes all entries of the database. Additional operations are run on the raw data to generate an extended dataset. This second dataset includes summary statistics such as turnout, the effective number of parties, etc.

After all these steps are finished we make both datasets accessible via [Shiny](#). This interface allows users to browse and download raw as well as processed data. It is also possible to export the entire database including all coding decisions and flags.

Pain points

Coding and documenting cases that do not fit our pre-defined coding scheme constitute one particular pain point. For example, in many countries political parties and electoral alliances do not resemble the "well-behaved" party systems of Western Europe. Frequently changing electoral alliances, electoral pacts at the local level or the implosion of entire party systems as in Italy in the mid-1990s confront us with serious difficulties. Often identifying, coding, and documenting the electoral performance of political parties on a continuous basis is daunting. Moreover, those problematic cases come in so many variations that it is almost impossible to capture them in a parsimonious set of error codes. Therefore, no explicit rule is given in the codebook, and every individual case needs to be explained separately. The final datasets contain all that information. Hence, deviations from the coding guidelines are at least made transparent to the user.

A second pain point concerns the history of the database and inter coder reliability. Often the current research assistant knows only a limited number of her predecessors. Consequently, there is little guarantee that coding decisions are made consistently across coder generations. Rather, each research assistant acquires highly individualized knowledge of coding decisions and problems which can never be communicated exhaustively between

coder generations. In other words, although an extensive codebook exists intercoder reliability is necessarily limited. As a consequence, one recurrent task is to review past codings in order to guarantee that information in our database stays consistent over time.

Key benefits

One central concern of our workflow is to make data collection and processing transparent to the user. While numerous datasets on election results, government formation, and electoral systems exist, none document the coding process down to the level of the original source. In contrast, we provide users with a codebook listing all standardized coding decisions. Individual entries that do not fit those guidelines are highlighted and explained in the database output. Moreover, we offer the user the opportunity to review our original source along with our coding decisions. There are many ways to collect and aggregate data on political parties, elections, and governments. However, only if a researcher is offered sufficient detail on the data and the decisions leading to its creation, will she be able to evaluate how idiosyncrasies of the data impact her results. Our approach combines transparency on sources, coding, and aggregation with different layers of consistency checks, error assessment, and continuous monitoring. It establishes a unique level of reproducibility in the field of Comparative Politics.

Key tools

The key tool of our workflow is the *PostgreSQL* database which allows us to efficiently store our data. In contrast to *Microsoft Access*, which we use as a user interface for data entry, *PostgreSQL* is an object-relational database management system that comes free of charge. For its compatibility with *Microsoft Access*, *Git*, and *R* it constitutes a very flexible tool. It allows to automatically produce periodic reports on changes to the database and failed consistency checks. Moreover, *PostgreSQL* enables us to access all versions of the database and the corresponding *R* scripts via the version control system *Git*. Hence, earlier versions of the database can quickly be restored allowing for the replication of data used in past analyses. *PostgreSQL* along with its compatibility with the mentioned tools enables us to ensure high levels of data quality and reproducibility.

The second tool that we want to highlight is the statistical programming package *R*. In contrast to most other software alternatives *R* is free. Although *R* has a steep learning curve, it is an excellent tool for data mining and analysis. Moreover, *R* enables us to process data and perform consistency checks automatically. Another important feature of *R* is that existing code can quickly be changed. For instance, additional summary statistics can easily be added to the datasets that we provide. Finally, *R* packages such as *knitr* and *shiny* complement our workflow. These packages allow us to create periodic reports on the dataset and to provide members of our department easy access to the data.

Questions

Why do you think that reproducibility in your domain is important?

Political scientists learn from empirical experience. If contributions to our field are not transparent enough to be reproduced, then nothing will be learned from them. However, reproducibility covers both data generation and analysis. The Garbage-In-Garbage-Out principle applies to all studies that fail on either side of the equation.

How or where did you learn about reproducibility?

Some practices we use are standard and should be taught in every introductory methods class. Others we learned from more tech savvy colleagues. Magic happened once we put the two together.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

We see two major pitfalls. First, political scientists often receive strong training in qualitative or quantitative methods, but not in basic data management. It is not unheard of that graduate students merge datasets row by row in Excel. Much would be gained if Political Science curriculae would teach key data management skills. Second, our field rewards productivity, not thoroughness. We finish one project and quickly move on, leaving procedures of data generation and analysis poorly documented. To ensure at least a minimal level of reproducibility the provision of replication packages containing raw data, data management and analysis scripts should be made mandatory.

What do you view as the major incentives for doing reproducible research?

Political scientists learn from experience. Reproducible research establishes a baseline against which to compare future analyses and thus secures scientific progress.

Are there any best practices that you'd recommend for researchers in your field?

Never change your raw data file. Stay away from the GUI. Have at least one notebook detailing the evolution of your analysis. Always comment your code or field notes.

Developing R Code for the Processing and Analysis of Optic Flow Data

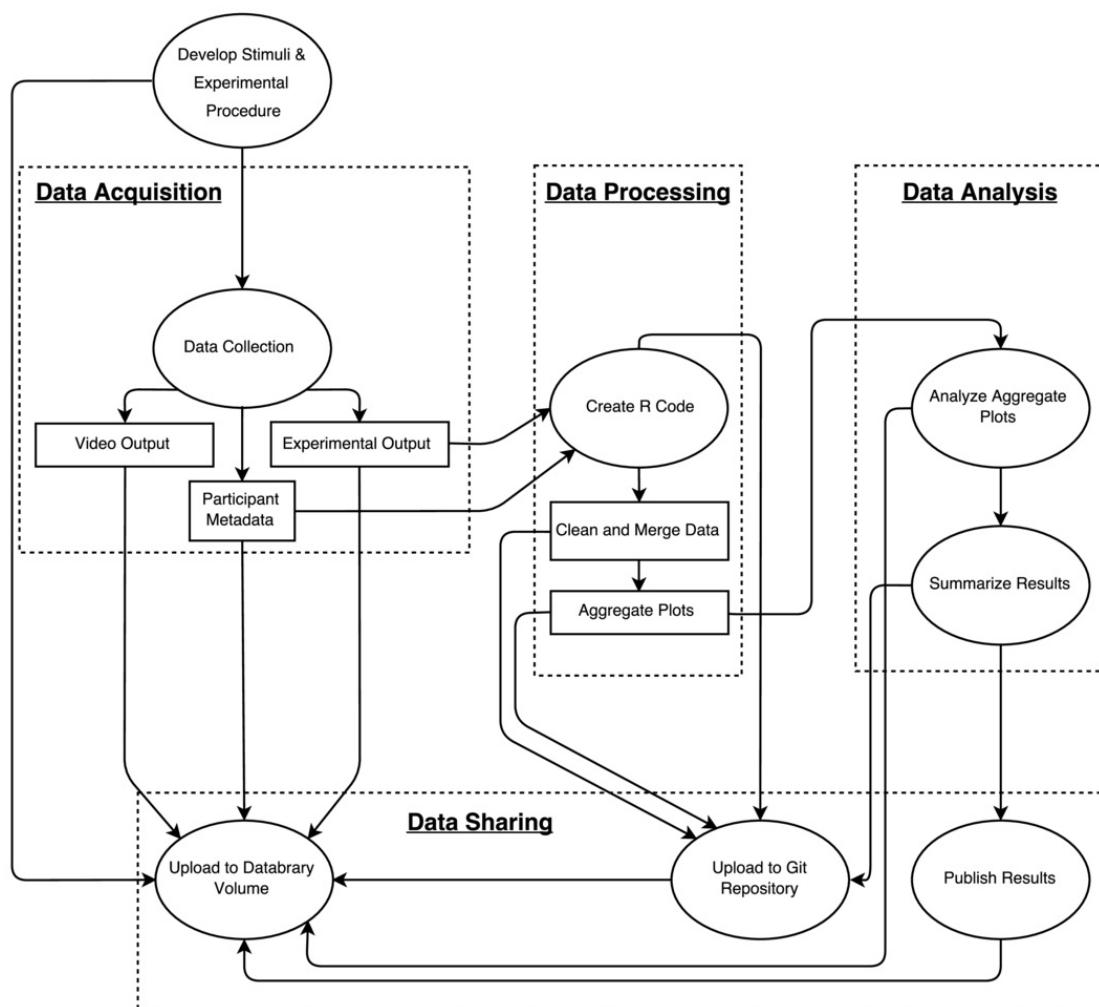
Andrea R. Seisler and Rick O. Gilmore

This chapter appears only in the online appendix of the book *The Practice of Reproducible Research*. Please cite this online version of the book as: Kitzes, J., Turek, D., & Deniz, F. (Eds.). (2018). *The Practice of Reproducible Research: Case Studies and Lessons from the Data-Intensive Sciences* [Online Version]. Retrieved from <http://www.practicereproducibleresearch.org>.

My name is Andrea Seisler and I am the lab manager for the Brain Development Laboratory run by Dr. Rick Gilmore at Penn State University. I have a background in biomedical engineering with a focus on imaging. In the last few years I have been assisting Dr. Gilmore with data collection, processing, analysis and publication of various behavioral and EEG based optic flow studies.

Reproducibility means that the processing of a dataset can be carried out multiple times by many users using the same workflow/code and get the same results. It also means that as more data is collected a few commands can be run to update the results based on the increased number of data sets.

Workflow



This study examined how sensitive child observers were to patterns of visual motion (optic flow) that differed in terms of their overall pattern and speed. The study extends one we had previously carried out with adults (<http://doi.org/10.17910/B7V88T>), and is part of a series of studies on this same theme (e.g., <http://doi.org/10.17910/B7QG6W>).

Data collection

Data were collected utilizing a script written in Matlab and the [Psychophysics Toolbox](#) to generate displays. Each display consisted of two side-by-side, time varying annular-shaped optic flow patterns consisting of small white dots moving against a black background. One side depicted random (0% coherent) motion while the other side depicted radial or translational motion. Within each trial, we varied the proportion of dots with coherent (non-random) motion. Some children saw patterns with 20, 40, 60, and 80% coherence while others saw patterns with 15, 30, 45, and 60% coherence. The participant's task was to determine which side of the screen contained coherent motion. The participant indicated their choice by pointing to the monitor. Across a set of four blocks, we also varied the speed of motion. Two blocks each were collected at 2 and 8 deg/s for a total of 4 runs.

Four separate output (CSV-format) files, one each from the 4 blocks, were generated for each participant by the Matlab script. Each CSV file included trial-by-trial information about the pattern type and coherence level of the stimulus, and reaction time and accuracy of the responses. Video was collected during participant data collection, as well. The file name consisted of the testing date (YYMMDD) concatenated with a four digit participant identifier (NNNN) the block order (1-4) and speed condition (2 or 8). This file naming scheme facilitated later processing.

During the recruitment phase, we have access to various forms of personally identifiable information (PII) about participants. This enables us to mail, email, and call participant families within our selected birthday range of participants. Most of the PII elements may **not** be shared with others and are stored only on local computers to which only our laboratory group has access.

Databrary

We use the Databrary digital library (<http://databrary.org>) to store and share data, including videos of the testing sessions. After a data collection session is complete, we create a session on Databrary's spreadsheet. We enter information about the participant (identifier code, sex, age at test, parent-reported race/ethnicity, test date, and birthdate) along with session-specific variables such as the condition, testing order, whether the session was for pilot testing or was excluded for some reason, and so forth. We upload each of the text-formatted data files and the video of the session (if available). We also ask the parent's and participant's permission to share the data with other researchers. We then record on the Databrary spreadsheet whether the parent and participant agreed to share data or not.

The combination of exact age, testing date, and date of birth are considered personally identifying under HIPAA. Databrary allows us to record all three data elements for our own record-keeping, but does not expose all three to other researchers unless participants have given permission to share data. If participants or parents decline to share data, only age at test and year of test is made visible. In some cases, we transfer data files to Penn State's Box cloud storage for analysis and removal of PII.

GitHub Repo

In addition to storing text-based data on Databrary, we store it in a GitHub repository for the project (<https://github.com/gilmore-lab/moco-3-pattern-psychophysics/tree/master/child-laminar-radial>). The GitHub repository is the home for our data cleaning and analysis code that is written in R by the lab director (Gilmore). The lab director conducts most of the analyses, and provides documentation about the analysis workflow in the comments of the relevant R functions or in the site's README file. Git's version tracking feature allows the history of changes to our analysis workflows to be carefully documented.

The data cleaning/file merging workflow consists of a series of steps. First, information about the participant ID number, speed condition, and block are extracted from the individual file names. These elements are added to an R data frame with the block-specific, trial-by-trial data. Then, the individual block data files can be merged or concatenated to create a session-level data frame for that participant. Using R's *lapply*, *Reduce*, and *merge* functions, it is possible to carry out these operations across a set of participant files to create a single data frame for subsequent analysis which is saved as a CSV file. The `analyses/import-clean-export.R` script illustrates how these steps are done. An example of the aggregate output data file created on 2016-09-15 from running this script is `analyses/aggregate-data/moco-beh-child.csv`. This file gets updated on GitHub periodically during the course of data collection. The second step is to analyze the data. We have not conducted formal analysis yet as we are still collecting data, but we have created some functions to visualize the patterns. The `analyses/plot.aggregate.R` function shows how we import the data file generated previously, summarize it, and create several illustrative plots (see `analyses/img`).

A similar workflow is utilized for multiple studies in this lab including EEG and other behavioral studies. The combination of Databrary, GitHub and R makes it easy to create a workflow for a particular type of data and reproduce it as more data are collected. This makes data sharing and analyses an ongoing process, and not something that is saved up until the end of a study. This makes writing abstracts and papers less cumbersome.

Pain points

The data transfer to Databrary can be time consuming. The output datafiles are stored locally in a folder for the CSV files and another folder for the .mp4 files. Uploading data (.mp4 and .csv) to Databrary has to happen manually as Databrary does not currently reorder files by file name. The .csv data also needs to be uploaded to Box/GitHub manually.

It's relatively easy to update the participant metadata file by exporting the data from Databrary as a .csv. From there the unnecessary columns for a particular analysis (e.g. Race, Ethnicity, Task name) can be dropped, leaving only the participant ID, test date, day age, and gender columns.

Key benefits

R contains many commands (e.g. *merge*, *lapply*, *Reduce*) which make it efficient to complete actions on multiple datasets at a time and to easily add datasets to the analyses as they are collected. We regularly 'borrow' a script used for one purpose and reuse it for a new study. By automating the data file manipulation steps, we reduce the likelihood of errors.

Key tools

Databrary allows for all of our data to be stored or referenced (e.g. GitHub, publications) in one place, and it encourages us to upload data as it is collected. This upload-as-you-go work flow is less cumbersome than post hoc data curation. Databrary volumes can be kept private to our research group until they are complete and we are ready to share the data with other researchers. Our practice is to share once we have presented our work in public or had a paper submitted for publication. Once a dataset is shared, the Databrary system creates a DOI for the dataset. This makes the dataset searchable by other researchers.

Questions

Why do you think that reproducibility in your domain is important?

Reproducibility is essential because if another researcher cannot reproduce our workflow and get the same results then the initial results may be incorrect.

How or where did you learn about reproducibility?

Self-teaching through online training and book learning.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

Ensuring that everything on GitHub can be forked and rerun by another user. We have also begun collecting videos (with permission) of our entire experimental protocols. We think that all social and behavioral scientists should do the same in order to improve the accuracy of documentation about experimental procedures.

What do you view as the major incentives for doing reproducible research?

It upholds scientific ideals.

Are there any best practices that you'd recommend for researchers in your field?

Automate as much as possible. Document, document, document. Start today.

Would you recommend any specific resources for learning more about reproducibility?

R Coding and R Markdown training: <https://www.rstudio.com/online-learning/>

All or Nothing! Public Goods Provision under Partial versus Full Decentralization in Indonesia

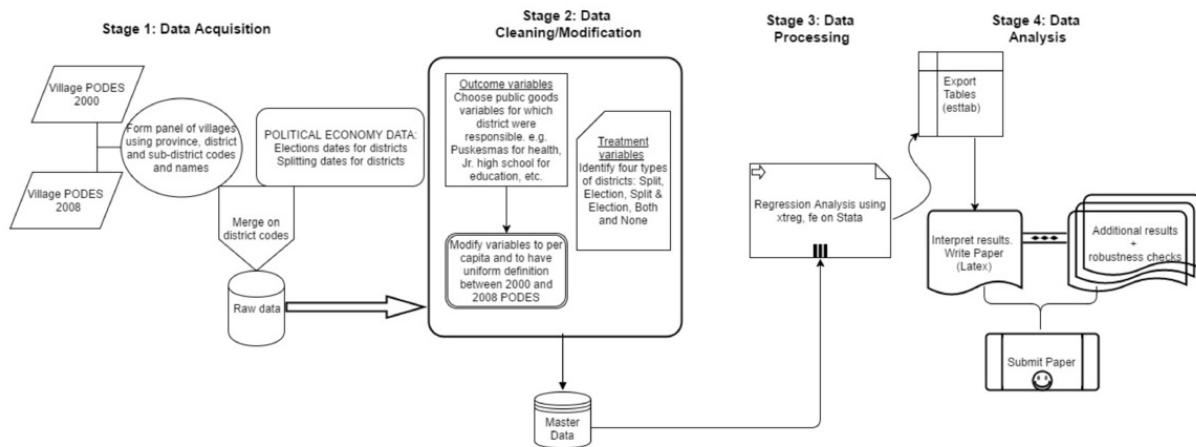
Deepak Singhania

This chapter appears only in the online appendix of the book *The Practice of Reproducible Research*. Please cite this online version of the book as: Kitzes, J., Turek, D., & Deniz, F. (Eds.). (2018). *The Practice of Reproducible Research: Case Studies and Lessons from the Data-Intensive Sciences* [Online Version]. Retrieved from <http://www.practicereproducibleresearch.org>.

I am Deepak Singhania and my area of research is in development economics and applied microeconomics. My main research also has some elements of political economics. I am a fifth year doctoral student at the University of California, Riverside. My research explores the welfare effects of complementarities between different types of decentralization. The existing literature is unclear about the direction of the effects of decentralization of governance on the provision of public goods. In my research, I argue that one of the main reasons for this ambiguity could be the misidentification of decentralization type. Theoretical literature broadly classifies decentralization types into administrative, political and fiscal decentralization. Different decentralization types and their complementarities can have strong implications for the identification of decentralization effects, and ignoring such complementarities could result in omitted variable bias. I address the identification problem by analyzing the complementarities between administrative and political decentralization under universal fiscal decentralization across Indonesian districts between 2000 and 2008. My results show that public good provision increased in the districts that faced both administrative and political decentralization compared to the districts that faced only one type of decentralization or no decentralization.

Research by definition means repeated systematic investigation which means testing and re-testing the existing findings. I see reproducibility as a way to justify the very meaning of the term research. Reproducibility is essential in my research on decentralization because I am challenging the existing findings and methods in the decentralization literature. I would like my idea of comparing different decentralization types to become a standard way of conducting a decentralization study and for this I need to ensure that my work is reproducible. I plan to make it reproducible not just within the Indonesian setting but even for using my method to test the findings in other settings.

Workflow



I have used two main types of data. The first is public good related data which is available from the Indonesian statistical agency (BPS). This data is a census of Indonesian villages and it is called PODES. I have used the 2000 and 2008 waves of PODES for short period analysis and the 2011 wave for a medium period analysis. This data is not public but it can be purchased easily from the BPS website. For reproducing my main results one can use just 2000 and 2008 waves. The second type of data is political economy related information for Indonesian districts. There are two sources for this. Data on date of district splitting is freely available from BPS, one can email them. I obtained the dates for the election of district heads from [Burgess et al. 2014](#). These dates are also available from Home Ministry of Indonesia, although I had a difficult time arranging the dates from there.

PODES is a huge dataset with over 200 variables for 60,000+ villages. One needs to be careful while using the same variable over two or three waves because the variable names and definition keeps changing. I picked those public good related variables for which districts governments were directly responsible. I got this information from some of the existing papers. Once the right set of variables are gathered and the definition is made uniform, this data is ready to be used for outcome variables. This data has various other variables that could be used as explanatory variables but I haven't used them in my paper because it is difficult to add explanatory variables other than the main treatments due to endogeneity issues between other explanatory variables and treatment variables. However, this would not change the main results of my paper.

Political economy data has district codes and names which can be used to merge them with PODES. For the treatment 'split', I assigned a dummy to the original part and the new part of the districts that were split. Similarly, the treatment of election was assigned using the dates in the data on elections. Now, the merged political economy data and PODES datasets are ready for the main analysis. In order to apply a differences-in-differences with fixed effects estimation strategy, I used the `xtreg` command in Stata [version 14 SE] to produce my main analysis results. The fixed effects were at the level of villages for which I needed a panel of villages. I discuss this panel as a pain point for reproducibility in the next section. I exported the results from Stata to .eps format so that I could produce LaTeX version of the tables. Since there were many variables under different categories, it took some iteration to produce tables that could be easy to explain.

I have commented my code extensively. I plan to make it available for each of the datasets that I have used. The process is long but straightforward. One favorable thing in the case of Indonesia, which is not true in many developing countries, is that the codes and names for districts and provinces are pretty much similar across various datasets and so it is not very difficult to match them. I would say that my codes are very carefully written and not difficult to comprehend. So it should be easy to replicate my work with my codes. Also, my specification is straightforward which will make it easier to do a similar analysis in other contexts.

Pain points

One of the main challenges in empirical research in developing countries is to work with huge datasets, at times messy ones. It was particularly difficult in the case of Indonesia because some of the data was in Bahasa language. I used Google translation to understand the variables and their meaning in PODES dataset. With a little practice it became easy to do this.

Another big problem was to convert the PODES datasets into a panel of villages. For this I used the codes and names for provinces, districts and sub-districts to match the villages. With a two weeks' time I could successfully create a panel of 94% of the villages in the baseline. I am willing to share my codes for this so that the replication of my work becomes feasible. Since it is a census of villages and I have used all the villages, with the exception of three regions for which reasons are mentioned in my paper, it should be possible to perfectly replicate my results.

Another small problem one could face is in matching codes for districts between different waves of PODES with political economy data. I used the Indonesian proliferation crosswalk available at the World Bank website under [INDO-DAPOER page](#). They have provided district codes for every year since early 1990s to recent years. I would recommend

researchers double-check the matched district codes and names (there are 434 districts in the endline) across datasets which should be a quick job using the command exact() in Excel.

I would say one should be very careful in writing codes. Just because a big dataset is available, one must not retain all the variables in the master data. Be stingy in choosing the initial variables and then keep adding more. A large part of mistakes happen due to unnecessary variables and unnecessary codes for them. Another specific piece of advice is that while re-working on an already written set of codes which you think won't need or you have a better way of doing it, never overwrite the original set of codes because you never know if the new ones are going to be fine. Copy the original ones first, paste them on a new command window and then work from there.

My next suggestion is about transferring results from Stata. Never copy and paste results. Easier tools like esttab and outreg are available to make our lives easier. There is an initial fixed cost of learning these tools but it saves a lot of time and error in a long run.

I haven't used version control yet since I got introduced to it recently. It's very useful and I intend to use it more often. I believe that I am good at archiving my old files and working the new ones in a separate folder, but we can't really trust human error. When tools like github and the Open Science Framework are available, why take risk?

Questions

Why do you think that reproducibility in your domain is important?

Many important public policy decisions are based on the results from research in my field. Most of our work involves testing the effects of something on economic welfare. If these results are wrong, due to whatever reasons, then it might affect valuable resources targeted towards improving lives. So, it is important that studies are made more and more reproducible in my field. After all the main objective of doing research is to have a positive impact on society and for such big objectives research studies must be subject to scrutiny. It is not an individual thing but it involves society-wide values.

How or where did you learn about reproducibility?

I learnt some of it from one of my committee members Professor Joseph Cummins while we were working on a collaborated paper in his class. But most of it I learnt recently at a Berkeley Initiative for Transparency in the Social Sciences (BITSS) workshop.

What do you see as the major challenges to doing reproducible research in your domain, and do you have any suggestions?

I believe the biggest pitfall in doing reproducible research in my field is that journals don't have strict requirements for publishing codes along with manuscripts and also for releasing data, if possible. If these requirements are met then even when the data are not accessible, at least the codes can be cross-checked and used in a different setting where similar data would be available. The fear of being public, with all the work, would require researchers to be more careful in with their analysis. The best way to create such environment lies at the behest of journals.

What do you view as the major incentives for doing reproducible research?

The incentives I can think of are as follows. Availability of funds for doing reproducible research would play the most important role. Another method could be to create opportunities for making a career out of doing reproducible research by appointing positions or special cells within university departments to conduct reproducible research. Lastly, it will be best to instill the values attached to reproducible research at an early stage and this could be done by including a chapter on this topic in the undergraduate course curriculum.

Are there any best practices that you'd recommend for researchers in your field?

While writing codes, always write them with the thought that you would have to publish it along with your manuscript. This way you would make sure that it's simple to understand, you have enough comments and you name your variables in sensible way.

Would you recommend any specific resources for learning more about reproducibility?

I would recommend <http://www.bitss.org>. If one can master all that is available here, it should be more than enough.

Foundations and Trends® in
Theoretical Computer Science
Vol. 9, Nos. 3–4 (2014) 211–407
© 2014 C. Dwork and A. Roth
DOI: 10.1561/0400000042



The Algorithmic Foundations of Differential Privacy

Cynthia Dwork
Microsoft Research, USA
dwork@microsoft.com

Aaron Roth
University of Pennsylvania, USA
aaroth@cis.upenn.edu

Contents

Preface	3
1 The Promise of Differential Privacy	5
1.1 Privacy-preserving data analysis	6
1.2 Bibliographic notes	10
2 Basic Terms	11
2.1 The model of computation	11
2.2 Towards defining private data analysis	12
2.3 Formalizing differential privacy	15
2.4 Bibliographic notes	26
3 Basic Techniques and Composition Theorems	28
3.1 Useful probabilistic tools	28
3.2 Randomized response	29
3.3 The laplace mechanism	30
3.4 The exponential mechanism	37
3.5 Composition theorems	41
3.6 The sparse vector technique	55
3.7 Bibliographic notes	64

4 Releasing Linear Queries with Correlated Error	66
4.1 An offline algorithm: SmallDB	70
4.2 An online mechanism: private multiplicative weights	76
4.3 Bibliographical notes	86
5 Generalizations	88
5.1 Mechanisms via α -nets	89
5.2 The iterative construction mechanism	91
5.3 Connections	109
5.4 Bibliographical notes	115
6 Boosting for Queries	117
6.1 The boosting for queries algorithm	119
6.2 Base synopsis generators	130
6.3 Bibliographical notes	139
7 When Worst-Case Sensitivity is Atypical	140
7.1 Subsample and aggregate	140
7.2 Propose-test-Release	143
7.3 Stability and privacy	150
8 Lower Bounds and Separation Results	158
8.1 Reconstruction attacks	159
8.2 Lower bounds for differential privacy	164
8.3 Bibliographic notes	170
9 Differential Privacy and Computational Complexity	172
9.1 Polynomial time curators	174
9.2 Some hard-to-Syntheticize distributions	177
9.3 Polynomial time adversaries	185
9.4 Bibliographic notes	187
10 Differential Privacy and Mechanism Design	189
10.1 Differential privacy as a solution concept	191
10.2 Differential privacy as a tool in mechanism design	193
10.3 Mechanism design for privacy aware agents	204
10.4 Bibliographical notes	213

11 Differential Privacy and Machine Learning	216
11.1 The sample complexity of differentially private machine learning	219
11.2 Differentially private online learning	222
11.3 Empirical risk minimization	227
11.4 Bibliographical notes	230
12 Additional Models	231
12.1 The local model	232
12.2 Pan-private streaming model	237
12.3 Continual observation	240
12.4 Average case error for query release	248
12.5 Bibliographical notes	252
13 Reflections	254
13.1 Toward practicing privacy	254
13.2 The differential privacy lens	258
Appendices	260
A The Gaussian Mechanism	261
A.1 Bibliographic notes	266
B Composition Theorems for (ε, δ)-DP	267
B.1 Extension of Theorem 3.16	267
Acknowledgments	269
References	270

Abstract

The problem of privacy-preserving data analysis has a long history spanning multiple disciplines. As electronic data about individuals becomes increasingly detailed, and as technology enables ever more powerful collection and curation of these data, the need increases for a robust, meaningful, and mathematically rigorous definition of privacy, together with a computationally rich class of algorithms that satisfy this definition. Differential Privacy is such a definition.

After motivating and discussing the meaning of differential privacy, the preponderance of this monograph is devoted to fundamental techniques for achieving differential privacy, and application of these techniques in creative combinations, using the query-release problem as an ongoing example. A key point is that, by rethinking the computational goal, one can often obtain far better results than would be achieved by methodically replacing each step of a non-private computation with a differentially private implementation. Despite some astonishingly powerful computational results, there are still fundamental limitations — not just on what can be achieved with differential privacy but on what can be achieved with any method that protects against a complete breakdown in privacy. Virtually all the algorithms discussed herein maintain differential privacy against adversaries of arbitrary computational power. Certain algorithms are computationally intensive, others are efficient. Computational complexity for the adversary and the algorithm are both discussed.

We then turn from fundamentals to applications other than query-release, discussing differentially private methods for mechanism design and machine learning. The vast majority of the literature on differentially private algorithms considers a single, static, database that is subject to many analyses. Differential privacy in other models, including distributed databases and computations on data streams is discussed.

Finally, we note that this work is meant as a thorough introduction to the problems and techniques of differential privacy, but is not intended to be an exhaustive survey — there is by now a vast amount of work in differential privacy, and we can cover only a small portion of it.

C. Dwork and A. Roth. *The Algorithmic Foundations of Differential Privacy*. Foundations and Trends® in Theoretical Computer Science, vol. 9, nos. 3–4, pp. 211–407, 2014.

DOI: 10.1561/0400000042.

Preface

The problem of privacy-preserving data analysis has a long history spanning multiple disciplines. As electronic data about individuals becomes increasingly detailed, and as technology enables ever more powerful collection and curation of these data, the need increases for a robust, meaningful, and mathematically rigorous definition of privacy, together with a computationally rich class of algorithms that satisfy this definition. *Differential Privacy* is such a definition.

After motivating and discussing the meaning of differential privacy, the preponderance of the book is devoted to fundamental techniques for achieving differential privacy, and application of these techniques in creative combinations (Sections 3–7), using the *query-release* problem as an ongoing example. A key point is that, by rethinking the computational goal, one can often obtain far better results than would be achieved by methodically replacing each step of a non-private computation with a differentially private implementation.

Despite some astonishingly powerful computational results, there are still fundamental limitations — not just on what can be achieved with differential privacy but on what can be achieved with *any* method that protects against a complete breakdown in privacy (Section 8).

Virtually all the algorithms discussed in this book maintain differential privacy against adversaries of arbitrary computational power. Certain algorithms are computationally intensive, others are

efficient. Computational complexity for the adversary and the algorithm are both discussed in Section 9.

In Sections 10 and 11 we turn from fundamentals to applications other than query-release, discussing differentially private methods for mechanism design and machine learning. The vast majority of the literature on differentially private algorithms considers a single, static, database that is subject to many analyses. Differential privacy in other models, including distributed databases and computations on data streams is discussed in Section 12.

Finally, we note that this book is meant as a thorough introduction to the problems and techniques of differential privacy, but is not intended to be an exhaustive survey — there is by now a vast amount of work in differential privacy, and we can cover only a small portion of it.

1

The Promise of Differential Privacy

“Differential privacy” describes a promise, made by a data holder, or *curator*, to a data subject: “You will not be affected, adversely or otherwise, by allowing your data to be used in any study or analysis, no matter what other studies, data sets, or information sources, are available.” At their best, differentially private database mechanisms can make confidential data widely available for accurate data analysis, without resorting to data clean rooms, data usage agreements, data protection plans, or restricted views. Nonetheless, data utility will eventually be consumed: the Fundamental Law of Information Recovery states that overly accurate answers to too many questions will destroy privacy in a spectacular way.¹ The goal of algorithmic research on differential privacy is to postpone this inevitability as long as possible.

Differential privacy addresses the paradox of learning nothing about an individual while learning useful information about a population. A medical database may teach us that smoking causes cancer, affecting an insurance company’s view of a smoker’s long-term medical costs. Has the smoker been harmed by the analysis? Perhaps — his insurance

¹This result, proved in Section 8.1, applies to *all* techniques for privacy-preserving data analysis, and not just to differential privacy.

premiums may rise, if the insurer knows he smokes. He may also be helped — learning of his health risks, he enters a smoking cessation program. Has the smoker’s privacy been compromised? It is certainly the case that more is known about him after the study than was known before, but was his information “leaked”? Differential privacy will take the view that it was not, with the rationale that the impact on the smoker is the same *independent of whether or not he was in the study*. It is the *conclusions reached* in the study that affect the smoker, not his presence or absence in the data set.

Differential privacy ensures that the same conclusions, for example, smoking causes cancer, will be reached, independent of whether any individual opts into or opts out of the data set. Specifically, it ensures that any sequence of outputs (responses to queries) is “essentially” equally likely to occur, independent of the presence or absence of any individual. Here, the probabilities are taken over random choices made by the privacy mechanism (something controlled by the data curator), and the term “essentially” is captured by a parameter, ε . A smaller ε will yield better privacy (and less accurate responses).

Differential privacy is a *definition*, not an algorithm. For a given computational task T and a given value of ε there will be many differentially private algorithms for achieving T in an ε -differentially private manner. Some will have better accuracy than others. When ε is small, finding a highly accurate ε -differentially private algorithm for T can be difficult, much as finding a numerically stable algorithm for a specific computational task can require effort.

1.1 Privacy-preserving data analysis

Differential privacy is a definition of privacy tailored to the problem of privacy-preserving data analysis. We briefly address some concerns with other approaches to this problem.

Data Cannot be Fully Anonymized and Remain Useful. Generally speaking, the richer the data, the more interesting and useful it is. This has led to notions of “anonymization” and “removal of personally identifiable information,” where the hope is that portions of the

data records can be suppressed and the remainder published and used for analysis. However, the richness of the data enables “naming” an individual by a sometimes surprising collection of fields, or attributes, such as the combination of zip code, date of birth, and sex, or even the names of three movies and the approximate dates on which an individual watched these movies. This “naming” capability can be used in a *linkage attack* to match “anonymized” records with non-anonymized records in a different dataset. Thus, the medical records of the governor of Massachusetts were identified by matching anonymized medical encounter data with (publicly available) voter registration records, and Netflix subscribers whose viewing histories were contained in a collection of anonymized movie records published by Netflix as training data for a competition on recommendation were identified by linkage with the Internet Movie Database (IMDb).

Differential privacy neutralizes linkage attacks: since being differentially private is a property of the data access mechanism, and is unrelated to the presence or absence of auxiliary information available to the adversary, access to the IMDb would no more permit a linkage attack to someone whose history is in the Netflix training set than to someone not in the training set.

Re-Identification of “Anonymized” Records is Not the Only Risk. Re-identification of “anonymized” data records is clearly undesirable, not only because of the re-identification *per se*, which certainly reveals membership in the data set, but also because the record may contain compromising information that, were it tied to an individual, could cause harm. A collection of medical encounter records from a specific urgent care center on a given date may list only a small number of distinct complaints or diagnoses. The additional information that a neighbor visited the facility on the date in question gives a fairly narrow range of possible diagnoses for the neighbor’s condition. The fact that it may not be possible to match a specific record to the neighbor provides minimal privacy protection to the neighbor.

Queries Over Large Sets are Not Protective. Questions about specific individuals cannot be safely answered with accuracy, and indeed one

might wish to reject them out of hand (were it computationally feasible to recognize them). Forcing queries to be over large sets is not a panacea, as shown by the following *differencing attack*. Suppose it is known that Mr. X is in a certain medical database. Taken together, the answers to the two large queries “How many people in the database have the sickle cell trait?” and “How many people, not named X, in the database have the sickle cell trait?” yield the sickle cell status of Mr. X.

Query Auditing Is Problematic. One might be tempted to *audit* the sequence of queries and responses, with the goal of interdicting any response if, in light of the history, answering the current query would compromise privacy. For example, the auditor may be on the lookout for pairs of queries that would constitute a differencing attack. There are two difficulties with this approach. First, it is possible that *refusing* to answer a query is itself disclosive. Second, query auditing can be computationally infeasible; indeed if the query language is sufficiently rich there may not even exist an algorithmic procedure for deciding if a pair of queries constitutes a differencing attack.

Summary Statistics are Not “Safe.” In some sense, the failure of summary statistics as a privacy solution concept is immediate from the differencing attack just described. Other problems with summary statistics include a variety of *reconstruction attacks* against a database in which each individual has a “secret bit” to be protected. The utility goal may be to permit, for example, questions of the form “How many people satisfying property P have secret bit value 1?” The goal of the adversary, on the other hand, is to significantly increase his chance of guessing the secret bits of individuals. The reconstruction attacks described in Section 8.1 show the difficulty of protecting against even a *linear* number of queries of this type: unless sufficient inaccuracy is introduced almost all the secret bits can be reconstructed.

A striking illustration of the risks of releasing summary statistics is in an application of a statistical technique, originally intended for confirming or refuting the presence of an individual’s DNA in a forensic mix, to ruling an individual in or out of a genome-wide association study. According to a Web site of the Human Genome Project, “Single nucleotide polymorphisms, or SNPs (pronounced “snips”), are DNA

sequence variations that occur when a single nucleotide (A,T,C, or G) in the genome sequence is altered. For example a SNP might change the DNA sequence AAGGCTAA to ATGGCTAA.” In this case we say there are two alleles: A and T. For such a SNP we can ask, given a particular reference population, what are the frequencies of each of the two possible alleles? Given the allele frequencies for SNPs in the reference population, we can examine how these frequencies may differ for a subpopulation that has a particular disease (the “case” group), looking for alleles that are associated with the disease. For this reason, genome-wide association studies may contain the allele frequencies of the case group for large numbers of SNPs. By definition, these allele frequencies are only aggregated statistics, and the (erroneous) assumption has been that, by virtue of this aggregation, they preserve privacy. However, given the genomic data of an individual, it is theoretically possible to determine if the individual is in the case group (and, therefore, has the disease). In response, the National Institutes of Health and Wellcome Trust terminated public access to aggregate frequency data from the studies they fund.

This is a challenging problem even for differential privacy, due to the large number — hundreds of thousands or even one million — of measurements involved and the relatively small number of individuals in any case group.

“*Ordinary*” Facts are Not “OK.” Revealing “ordinary” facts, such as purchasing bread, may be problematic if a data subject is followed over time. For example, consider Mr. T, who regularly buys bread, year after year, until suddenly switching to rarely buying bread. An analyst might conclude Mr. T most likely has been diagnosed with Type 2 diabetes. The analyst might be correct, or might be incorrect; either way Mr. T is harmed.

“*Just a Few.*” In some cases a particular technique may in fact provide privacy protection for “typical” members of a data set, or more generally, “most” members. In such cases one often hears the argument that the technique is adequate, as it compromises the privacy of “just a few” participants. Setting aside the concern that outliers may be precisely those people for whom privacy is most important, the “just a few”

philosophy is not intrinsically without merit: there is a social judgment, a weighing of costs and benefits, to be made. A well-articulated definition of privacy consistent with the “just a few” philosophy has yet to be developed; however, for a single data set, “just a few” privacy can be achieved by randomly selecting a subset of rows and releasing them in their entirety (Lemma 4.3, Section 4). Sampling bounds describing the quality of statistical analysis that can be carried out on random subsamples govern the number of rows to be released. Differential privacy provides an alternative when the “just a few” philosophy is rejected.

1.2 Bibliographic notes

Sweeney [81] linked voter registration records to “anonymized” medical encounter data; Narayanan and Shmatikov carried out a linkage attack against anonymized ranking data published by Netflix [65]. The work on presence in a forensic mix is due to Homer et al. [46]. The first reconstruction attacks were due to Dinur and Nissim [18].

2

Basic Terms

This section motivates and presents the formal definition of differential privacy, and enumerates some of its key properties.

2.1 The model of computation

We assume the existence of a trusted and trustworthy *curator* who holds the data of *individuals* in a database D , typically comprised of some number n of rows. The intuition is that each row contains the data of a single individual, and, still speaking intuitively, the privacy goal is to simultaneously protect every individual row while permitting statistical analysis of the database as a whole.

In the *non-interactive*, or *offline*, model the curator produces some kind of object, such as a “synthetic database,” collection of summary statistics, or “sanitized database” once and for all. After this *release* the curator plays no further role and the original data may be destroyed.

A *query* is a function to be applied to a database. The *interactive*, or *online*, model permits the data analyst to ask queries adaptively, deciding which query to pose next based on the observed responses to previous queries.

The trusted curator can be replaced by a protocol run by the set of individuals, using the cryptographic techniques for secure multi-party protocols, but for the most part we will not be appealing to cryptographic assumptions. Section 12 describes this and other models studied in the literature.

When all the queries are known in advance the non-interactive model should give the best accuracy, as it is able to correlate noise knowing the structure of the queries. In contrast, when no information about the queries is known in advance, the non-interactive model poses severe challenges, as it must provide answers to all possible queries. As we will see, to ensure privacy, or even to prevent privacy catastrophes, accuracy will necessarily deteriorate with the number of questions asked, and providing accurate answers to all possible questions will be infeasible.

A *privacy mechanism*, or simply a *mechanism*, is an algorithm that takes as input a database, a universe \mathcal{X} of data types (the set of all possible database rows), random bits, and, optionally, a set of queries, and produces an output string. The hope is that the output string can be decoded to produce relatively accurate answers to the queries, if the latter are present. If no queries are presented then we are in the non-interactive case, and the hope is that the output string can be interpreted to provide answers to future queries.

In some cases we may require that the output string be a *synthetic database*. This is a multiset drawn from the universe \mathcal{X} of possible database rows. The decoding method in this case is to carry out the query on the synthetic database and then to apply some sort of simple transformation, such as multiplying by a scaling factor, to obtain an approximation to the the true answer to the query.

2.2 Towards defining private data analysis

A natural approach to defining privacy in the context of data analysis is to require that the analyst knows no more about any individual in the data set after the analysis is completed than she knew before the analysis was begun. It is also natural to formalize this goal by

requiring that the adversary’s prior and posterior views about an individual (i.e., before and after having access to the database) shouldn’t be “too different,” or that access to the database shouldn’t change the adversary’s views about any individual “too much.” However, if the database teaches anything at all, this notion of privacy is unachievable. For example, suppose the adversary’s (incorrect) prior view is that everyone has 2 left feet. Access to the statistical database teaches that almost everyone has one left foot and one right foot. The adversary now has a very different view of whether or not any given respondent has two left feet.

Part of the appeal of before/after, or “nothing is learned,” approach to defining privacy is the intuition that if nothing is learned about an individual then the individual cannot be harmed by the analysis. However, the “smoking causes cancer” example shows this intuition to be flawed; the culprit is auxiliary information (Mr. X smokes).

The “nothing is learned” approach to defining privacy is reminiscent of semantic security for a cryptosystem. Roughly speaking, semantic security says that nothing is learned about the plaintext (the unencrypted message) from the ciphertext. That is, anything known about the plaintext after seeing the ciphertext was known before seeing the ciphertext. So if there is auxiliary information saying that the ciphertext is an encryption of either “dog” or “cat,” then the ciphertext leaks no further information about which of “dog” or “cat” has been encrypted. Formally, this is modeled by comparing the ability of the eavesdropper to guess which of “dog” and “cat” has been encrypted to the ability of a so-called *adversary simulator*, who has the auxiliary information but does not have access to the ciphertext, to guess the same thing. If for every eavesdropping adversary, and all auxiliary information (to which both the adversary and the simulator are privy), the adversary simulator has essentially the same odds of guessing as does the eavesdropper, then the system enjoys semantic security. Of course, for the system to be useful, the legitimate receiver must be able to correctly decrypt the message; otherwise semantic security can be achieved trivially.

We know that, under standard computational assumptions, semantically secure cryptosystems exist, so why can we not build semantically

secure private database mechanisms that yield answers to queries while keeping individual rows secret?

First, the analogy is not perfect: in a semantically secure cryptosystem there are three parties: the message sender (who encrypts the plaintext message), the message receiver (who decrypts the ciphertext), and the eavesdropper (who is frustrated by her inability to learn anything about the plaintext that she did not already know before it was sent). In contrast, in the setting of private data analysis there are only two parties: the curator, who runs the privacy mechanism (analogous to the sender) and the data analyst, who receives the informative responses to queries (like the message receiver) and also tries to squeeze out privacy-compromising information about individuals (like the eavesdropper). Because the legitimate receiver is the same party as the snooping adversary, the analogy to encryption is flawed: denying all information to the adversary means denying all information to the data analyst.

Second, as with an encryption scheme, we require the privacy mechanism to be useful, which means that it teaches the analyst something she did not previously know. This teaching is unavailable to an adversary simulator; that is, no simulator can “predict” what the analyst has learned. We can therefore look at the database as a weak source of random (unpredictable) bits, from which we can extract some very high quality randomness to be used as a *random pad*. This can be used in an encryption technique in which a secret message is added to a random value (the “random pad”) in order to produce a string that information-theoretically hides the secret. Only someone knowing the random pad can learn the secret; any party that knows nothing about the pad learns nothing at all about the secret, no matter his or her computational power. Given access to the database, the analyst can learn the random pad, but the adversary simulator, not given access to the database, learns nothing at all about the pad. Thus, given as auxiliary information the encryption of a secret using the random pad, the analyst can decrypt the secret, but the adversary simulator learns nothing at all about the secret. This yields a huge disparity between the ability of the adversary/analyst to learn the secret and the ability

of the adversary simulator to do the same thing, eliminating all hope of anything remotely resembling semantic security.

The obstacle in both the smoking causes cancer example and the hope for semantic security is auxiliary information. Clearly, to be meaningful, a privacy guarantee must hold even in the context of “reasonable” auxiliary knowledge, but separating reasonable from arbitrary auxiliary knowledge is problematic. For example, the analyst using a government database might be an employee at a major search engine company. What are “reasonable” assumptions about the auxiliary knowledge information available to such a person?

2.3 Formalizing differential privacy

We will begin with the technical definition of differential privacy, and then go on to interpret it. Differential privacy will provide privacy by *process*; in particular it will introduce randomness. An early example of privacy by randomized process is *randomized response*, a technique developed in the social sciences to collect statistical information about embarrassing or illegal behavior, captured by having a property P . Study participants are told to report whether or not they have property P as follows:

1. Flip a coin.
2. If **tails**, then respond truthfully.
3. If **heads**, then flip a second coin and respond “Yes” if heads and “No” if tails.

“Privacy” comes from the plausible deniability of any outcome; in particular, if having property P corresponds to engaging in illegal behavior, even a “Yes” answer is not incriminating, since this answer occurs with probability at least $1/4$ whether or not the respondent actually has property P . Accuracy comes from an understanding of the noise generation procedure (the introduction of spurious “Yes” and “No” answers from the randomization): The expected number of “Yes” answers is $1/4$ times the number of participants who do not have property P plus $3/4$ the number having property P . Thus, if p is the true fraction of

participants having property P , the expected number of “Yes” answers is $(1/4)(1-p) + (3/4)p = (1/4) + p/2$. Thus, we can estimate p as twice the fraction answering “Yes” minus $1/2$, that is, $2((1/4) + p/2) - 1/2$.

Randomization is essential; more precisely, any *non-trivial* privacy guarantee that holds regardless of all present or even future sources of auxiliary information, including other databases, studies, Web sites, on-line communities, gossip, newspapers, government statistics, and so on, requires randomization. This follows from a simple hybrid argument, which we now sketch. Suppose, for the sake of contradiction, that we have a non-trivial deterministic algorithm. Non-triviality says that there exists a query and two databases that yield different outputs under this query. Changing one row at a time we see there exists a pair of databases differing only in the value of a single row, on which the same query yields different outputs. An adversary knowing that the database is one of these two almost identical databases learns the value of the data in the unknown row.

We will therefore need to discuss the input and output space of randomized algorithms. Throughout this monograph we work with discrete probability spaces. Sometimes we will describe our algorithms as sampling from continuous distributions, but these should always be discretized to finite precision in an appropriately careful way (see Remark 2.1 below). In general, a randomized algorithm with domain A and (discrete) range B will be associated with a mapping from A to the probability simplex over B , denoted $\Delta(B)$:

Definition 2.1 (Probability Simplex). Given a discrete set B , the *probability simplex* over B , denoted $\Delta(B)$ is defined to be:

$$\Delta(B) = \left\{ x \in \mathbb{R}^{|B|} : x_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^{|B|} x_i = 1 \right\}$$

Definition 2.2 (Randomized Algorithm). A randomized algorithm \mathcal{M} with domain A and discrete range B is associated with a mapping $M : A \rightarrow \Delta(B)$. On input $a \in A$, the algorithm \mathcal{M} outputs $M(a) = b$ with probability $(M(a))_b$ for each $b \in B$. The probability space is over the coin flips of the algorithm \mathcal{M} .

We will think of databases x as being collections of records from a universe \mathcal{X} . It will often be convenient to represent databases by their histograms: $x \in \mathbb{N}^{|\mathcal{X}|}$, in which each entry x_i represents the number of elements in the database x of *type* $i \in \mathcal{X}$ (we abuse notation slightly, letting the symbol \mathbb{N} denote the set of all non-negative integers, including zero). In this representation, a natural measure of the distance between two databases x and y will be their ℓ_1 distance:

Definition 2.3 (Distance Between Databases). The ℓ_1 norm of a database x is denoted $\|x\|_1$ and is defined to be:

$$\|x\|_1 = \sum_{i=1}^{|\mathcal{X}|} |x_i|.$$

The ℓ_1 distance between two databases x and y is $\|x - y\|_1$

Note that $\|x\|_1$ is a measure of the *size* of a database x (i.e., the number of records it contains), and $\|x - y\|_1$ is a measure of how many records *differ* between x and y .

Databases may also be represented by multisets of *rows* (elements of \mathcal{X}) or even ordered lists of rows, which is a special case of a set, where the row number becomes part of the name of the element. In this case distance between databases is typically measured by the Hamming distance, i.e., the number of rows on which they differ.

However, unless otherwise noted, we will use the histogram representation described above. (Note, however, that even when the histogram notation is more mathematically convenient, in actual implementations, the multiset representation will often be much more concise).

We are now ready to formally define *differential privacy*, which intuitively will guarantee that a randomized algorithm behaves similarly on similar input databases.

Definition 2.4 (Differential Privacy). A randomized algorithm \mathcal{M} with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ε, δ) -differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$:

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\varepsilon) \Pr[\mathcal{M}(y) \in \mathcal{S}] + \delta,$$

where the probability space is over the coin flips of the mechanism \mathcal{M} . If $\delta = 0$, we say that \mathcal{M} is ε -differentially private.

Typically we are interested in values of δ that are less than the inverse of any polynomial in the size of the database. In particular, values of δ on the order of $1/\|x\|_1$ are very dangerous: they permit “preserving privacy” by publishing the complete records of a small number of database participants — precisely the “just a few” philosophy discussed in Section 1.

Even when δ is negligible, however, there are theoretical distinctions between $(\varepsilon, 0)$ - and (ε, δ) -differential privacy. Chief among these is what amounts to a switch of quantification order. $(\varepsilon, 0)$ -differential privacy ensures that, for *every* run of the mechanism $\mathcal{M}(x)$, the output observed is (almost) equally likely to be observed on *every* neighboring database, simultaneously. In contrast (ε, δ) -differential privacy says that for every pair of neighboring databases x, y , it is extremely unlikely that, *ex post facto* the observed value $\mathcal{M}(x)$ will be much more or much less likely to be generated when the database is x than when the database is y . However, given an output $\xi \sim \mathcal{M}(x)$ it may be possible to find a database y such that ξ is much more likely to be produced on y than it is when the database is x . That is, the mass of ξ in the distribution $\mathcal{M}(y)$ may be substantially larger than its mass in the distribution $\mathcal{M}(x)$.

The quantity

$$\mathcal{L}_{\mathcal{M}(x) \parallel \mathcal{M}(y)}^{(\xi)} = \ln \left(\frac{\Pr[\mathcal{M}(x) = \xi]}{\Pr[\mathcal{M}(y) = \xi]} \right)$$

is important to us; we refer to it as the *privacy loss* incurred by observing ξ . This loss might be positive (when an event is more likely under x than under y) or it might be negative (when an event is more likely under y than under x). As we will see in Lemma 3.17, (ε, δ) -differential privacy ensures that for all adjacent x, y , the absolute value of the privacy loss will be bounded by ε with probability at least $1 - \delta$. As always, the probability space is over the coins of the mechanism \mathcal{M} .

Differential privacy is immune to *post-processing*: A data analyst, without additional knowledge about the private database, cannot compute a function of the output of a private algorithm \mathcal{M} and make it

less differentially private. That is, if an algorithm protects an individual's privacy, then a data analyst cannot increase privacy loss — either under the formal definition or even in any intuitive sense — simply by sitting in a corner and *thinking about* the output of the algorithm. Formally, the composition of a data-independent mapping f with an (ϵ, δ) -differentially private algorithm \mathcal{M} is also (ϵ, δ) differentially private:

Proposition 2.1 (Post-Processing). Let $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \rightarrow R$ be a randomized algorithm that is (ϵ, δ) -differentially private. Let $f : R \rightarrow R'$ be an arbitrary randomized mapping. Then $f \circ \mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \rightarrow R'$ is (ϵ, δ) -differentially private.

Proof. We prove the proposition for a deterministic function $f : R \rightarrow R'$. The result then follows because any randomized mapping can be decomposed into a convex combination of deterministic functions, and a convex combination of differentially private mechanisms is differentially private.

Fix any pair of neighboring databases x, y with $\|x - y\|_1 \leq 1$, and fix any event $S \subseteq R'$. Let $T = \{r \in R : f(r) \in S\}$. We then have:

$$\begin{aligned} \Pr[f(\mathcal{M}(x)) \in S] &= \Pr[\mathcal{M}(x) \in T] \\ &\leq \exp(\epsilon) \Pr[\mathcal{M}(y) \in T] + \delta \\ &= \exp(\epsilon) \Pr[f(\mathcal{M}(y)) \in S] + \delta \end{aligned}$$

which was what we wanted. \square

It follows immediately from Definition 2.4 that $(\epsilon, 0)$ -differential privacy composes in a straightforward way: the composition of two $(\epsilon, 0)$ -differentially private mechanisms is $(2\epsilon, 0)$ -differentially private. More generally (Theorem 3.16), “the epsilons and the deltas add up”: the composition of k differentially private mechanisms, where the i th mechanism is (ϵ_i, δ_i) -differentially private, for $1 \leq i \leq k$, is $(\sum_i \epsilon_i, \sum_i \delta_i)$ -differentially private.

Group privacy for $(\epsilon, 0)$ -differentially private mechanisms also follows immediately from Definition 2.4, with the strength of the privacy guarantee drops linearly with the size of the group.

Theorem 2.2. Any $(\varepsilon, 0)$ -differentially private mechanism \mathcal{M} is $(k\varepsilon, 0)$ -differentially private for groups of size k . That is, for all $\|x - y\|_1 \leq k$ and all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(k\varepsilon) \Pr[\mathcal{M}(y) \in \mathcal{S}],$$

where the probability space is over the coin flips of the mechanism \mathcal{M} .

This addresses, for example, the question of privacy in surveys that include multiple family members.¹

More generally, composition and group privacy are not the same thing and the improved composition bounds in Section 3.5.2 (Theorem 3.20), which substantially improve upon the factor of k , do not — and cannot — yield the same gains for group privacy, even when $\delta = 0$.

2.3.1 What differential privacy promises

An Economic View. Differential privacy promises to protect individuals from any *additional* harm that they might face due to their data being in the private database x that they would not have faced had their data not been part of x . Although individuals may indeed face harm once the results $\mathcal{M}(x)$ of a differentially private mechanism \mathcal{M} have been released, differential privacy promises that the probability of harm was not significantly increased by their choice to participate. This is a very utilitarian definition of privacy, because when an individual is deciding whether or not to include her data in a database that will be used in a differentially private manner, it is exactly this difference that she is considering: the probability of harm given that she participates, as compared to the probability of harm given that she does not participate. She has no control over the remaining contents of the database. Given the promise of differential privacy, she is assured that she should

¹However, as the group gets larger, the privacy guarantee deteriorates, and this is what we want: clearly, if we replace an entire surveyed population, say, of cancer patients, with a completely different group of respondents, say, healthy teenagers, we *should* get different answers to queries about the fraction of respondents who regularly run three miles each day. Although something similar holds for (ε, δ) -differential privacy, the approximation term δ takes a big hit, and we only obtain $(k\varepsilon, ke^{(k-1)\varepsilon}\delta)$ -differential privacy for groups of size k .

be almost indifferent between participating and not, from the point of view of future harm. Given any incentive — from altruism to monetary reward — differential privacy may convince her to allow her data to be used. This intuition can be formalized in a utility-theoretic sense, which we here briefly sketch.

Consider an individual i who has arbitrary preferences over the set of all possible future events, which we denote by \mathcal{A} . These preferences are expressed by a utility function $u_i : \mathcal{A} \rightarrow \mathbb{R}_{\geq 0}$, and we say that individual i experiences utility $u_i(a)$ in the event that $a \in \mathcal{A}$ comes to pass. Suppose that $x \in \mathbb{N}^{|\mathcal{X}|}$ is a data-set containing individual i 's private data, and that \mathcal{M} is an ε -differentially private algorithm. Let y be a data-set that is identical to x except that it does not include the data of individual i (in particular, $\|x - y\|_1 = 1$), and let $f : \text{Range}(\mathcal{M}) \rightarrow \Delta(\mathcal{A})$ be the (arbitrary) function that determines the distribution over future events \mathcal{A} , conditioned on the output of mechanism M . By the guarantee of differential privacy, together with the resilience to arbitrary post-processing guaranteed by Proposition 2.1, we have:

$$\begin{aligned}\mathbb{E}_{a \sim f(\mathcal{M}(x))}[u_i(a)] &= \sum_{a \in \mathcal{A}} u_i(a) \cdot \Pr_{f(\mathcal{M}(x))}[a] \\ &\leq \sum_{a \in \mathcal{A}} u_i(a) \cdot \exp(\varepsilon) \Pr_{f(\mathcal{M}(y))}[a] \\ &= \exp(\varepsilon) \mathbb{E}_{a \sim f(\mathcal{M}(y))}[u_i(a)]\end{aligned}$$

Similarly,

$$\mathbb{E}_{a \sim f(\mathcal{M}(y))}[u_i(a)] \geq \exp(-\varepsilon) \mathbb{E}_{a \sim f(\mathcal{M}(x))}[u_i(a)].$$

Hence, by promising a guarantee of ε -differential privacy, a data analyst can promise an individual that his expected future utility will not be harmed by more than an $\exp(\varepsilon) \approx (1 + \varepsilon)$ factor. Note that this promise holds *independently* of the individual i 's utility function u_i , and holds *simultaneously* for multiple individuals who may have completely different utility functions.

2.3.2 What differential privacy does not promise

As we saw in the Smoking Causes Cancer example, while differential privacy is an extremely strong guarantee, it does not promise unconditional freedom from harm. Nor does it create privacy where none previously exists. More generally, differential privacy does not guarantee that what one believes to be one's secrets will remain secret. It merely ensures that one's participation in a survey will not in itself be disclosed, nor will participation lead to disclosure of any specifics that one has contributed to the survey. It is very possible that conclusions drawn from the survey may reflect statistical information about an individual. A health survey intended to discover early indicators of a particular ailment may produce strong, even conclusive results; that these conclusions hold for a given individual is not evidence of a differential privacy violation; the individual may not even have participated in the survey (again, differential privacy ensures that these conclusive results would be obtained with very similar probability whether or not the individual participated in the survey). In particular, if the survey teaches us that specific *private* attributes correlate strongly with *publicly observable* attributes, this is not a violation of differential privacy, since this same correlation would be observed with almost the same probability independent of the presence or absence of any respondent.

Qualitative Properties of Differential Privacy. Having introduced and formally defined differential privacy, we recapitulate its key desirable qualities.

1. *Protection against arbitrary risks*, moving beyond protection against re-identification.
2. *Automatic neutralization of linkage attacks*, including all those attempted with all past, present, and future datasets and other forms and sources of auxiliary information.
3. *Quantification of privacy loss*. Differential privacy is not a binary concept, and has a measure of privacy loss. This permits comparisons among different techniques: for a fixed bound on privacy loss, which technique provides better accuracy? For a fixed accuracy, which technique provides better privacy?

4. *Composition.* Perhaps most crucially, the quantification of loss also permits the analysis and control of cumulative privacy loss over multiple computations. Understanding the behavior of differentially private mechanisms under composition enables the design and analysis of complex differentially private algorithms from simpler differentially private building blocks.
5. *Group Privacy.* Differential privacy permits the analysis and control of privacy loss incurred by groups, such as families.
6. *Closure Under Post-Processing* Differential privacy is immune to post-processing: A data analyst, without additional knowledge about the private database, cannot compute a function of the output of a differentially private algorithm M and make it less differentially private. That is, a data analyst cannot increase privacy loss, either under the formal definition or even in any intuitive sense, simply by sitting in a corner and thinking about the output of the algorithm, *no matter what auxiliary information is available*.

These are the signal attributes of differential privacy. Can we prove a converse? That is, do these attributes, or some subset thereof, imply differential privacy? Can differential privacy be weakened in these respects and still be meaningful? These are open questions.

2.3.3 Final remarks on the definition

The Granularity of Privacy. Claims of differential privacy should be carefully scrutinized to ascertain the level of granularity at which privacy is being promised. Differential privacy promises that the behavior of an algorithm will be roughly unchanged even if a single entry in the database is modified. But what constitutes a single entry in the database? Consider for example a database that takes the form of a *graph*. Such a database might encode a social network: each individual $i \in [n]$ is represented by a vertex in the graph, and friendships between individuals are represented by edges.

We could consider differential privacy at a level of granularity corresponding to individuals: that is, we could require that differentially

private algorithms be insensitive to the addition or removal of any *vertex* from the graph. This gives a strong privacy guarantee, but might in fact be stronger than we need. the addition or removal of a single vertex could after all add or remove up to n edges in the graph. Depending on what it is we hope to learn from the graph, insensitivity to n edge removals might be an impossible constraint to meet.

We could on the other hand consider differential privacy at a level of granularity corresponding to edges, and ask our algorithms to be insensitive only to the addition or removal of single, or small numbers of, *edges* from the graph. This is of course a weaker guarantee, but might still be sufficient for some purposes. Informally speaking, if we promise ε -differential privacy at the level of a single edge, then no data analyst should be able to conclude anything about the existence of any subset of $1/\varepsilon$ edges in the graph. In some circumstances, large groups of social contacts might not be considered sensitive information: for example, an individual might not feel the need to hide the fact that the majority of his contacts are with individuals in his city or workplace, because where he lives and where he works are public information. On the other hand, there might be a small number of social contacts whose existence is highly sensitive (for example a prospective new employer, or an intimate friend). In this case, edge privacy should be sufficient to protect sensitive information, while still allowing a fuller analysis of the data than vertex privacy. Edge privacy will protect such an individual's sensitive information provided that he has fewer than $1/\varepsilon$ such friends.

As another example, a differentially private movie recommendation system can be designed to protect the data in the training set at the “event” level of single movies, hiding the viewing/rating of any single movie but not, say, hiding an individual’s enthusiasm for cowboy westerns or gore, or at the “user” level of an individual’s entire viewing and rating history.

All Small Epsilons Are Alike. When ε is small, $(\varepsilon, 0)$ -differential privacy asserts that for all pairs of adjacent databases x, y and all outputs o , an adversary cannot distinguish which is the true database

on the basis of observing o . When ε is small, *failing* to be $(\varepsilon, 0)$ -differentially private is not necessarily alarming — for example, the mechanism may be $(2\varepsilon, 0)$ -differentially private. The nature of the privacy guarantees with differing but small epsilons are quite similar. But what of large values for ϵ ? Failure to be $(15, 0)$ -differentially private merely says there exist neighboring databases and an output o for which the ratio of probabilities of observing o conditioned on the database being, respectively, x or y , is large. An output of o might be very unlikely (this is addressed by (ε, δ) -differential privacy); databases x and y might be terribly contrived and unlikely to occur in the “real world”; the adversary may not have the right auxiliary information to recognize that a revealing output has occurred; or may not know enough about the database(s) to determine the value of their symmetric difference. Thus, much as a weak cryptosystem may leak anything from only the least significant bit of a message to the complete decryption key, the failure to be $(\varepsilon, 0)$ - or (ε, δ) -differentially private may range from effectively meaningless privacy breaches to complete revelation of the entire database. A large epsilon is large after its own fashion.

A Few Additional Formalisms. Our privacy mechanism \mathcal{M} will often take some auxiliary parameters w as input, in addition to the database x . For example, w may specify a query q_w on the database x , or a collection \mathcal{Q}_w of queries. The mechanism $\mathcal{M}(w, x)$ might (respectively) respond with a differentially private approximation to $q_w(x)$ or to some or all of the queries in \mathcal{Q}_w . For all $\delta \geq 0$, we say that a mechanism $\mathcal{M}(\cdot, \cdot)$ satisfies (ε, δ) -differential privacy if for every w , $\mathcal{M}(w, \cdot)$ satisfies (ε, δ) -differential privacy.

Another example of a parameter that may be included in w is a *security parameter* κ to govern how small $\delta = \delta(\kappa)$ should be. That is, $\mathcal{M}(\kappa, \cdot)$ should be $(\varepsilon, \delta(\kappa))$ differentially private for all κ . Typically, and throughout this monograph, we require that δ be a negligible function in κ , i.e., $\delta = \kappa^{-\omega(1)}$. Thus, we think of δ as being cryptographically small, whereas ε is typically thought of as a moderately small constant.

In the case where the auxiliary parameter w specifies a collection $\mathcal{Q}_w = \{q : \mathcal{X}^n \rightarrow \mathbb{R}\}$ of queries, we call the mechanism \mathcal{M} a

synopsis generator. A synopsis generator outputs a (differentially private) synopsis \mathcal{A} which can be used to compute answers to all the queries in \mathcal{Q}_w . That is, we require that there exists a reconstruction procedure R such that for each input v specifying a query $q_v \in \mathcal{Q}_w$, the reconstruction procedure outputs $R(\mathcal{A}, v) \in \mathbb{R}$. Typically, we will require that with high probability \mathcal{M} produces a synopsis \mathcal{A} such that the reconstruction procedure, using \mathcal{A} , computes accurate answers. That is, for all or most (weighted by some distribution) of the queries $q_v \in \mathcal{Q}_w$, the error $|R(\mathcal{A}, v) - q_v(x)|$ will be bounded. We will occasionally abuse notation and refer to the reconstruction procedure taking as input the actual query q (rather than some representation v of it), and outputting $R(\mathcal{A}, q)$.

A special case of a synopsis is a *synthetic database*. As the name suggests, the rows of a synthetic database are of the same type as rows of the original database. An advantage to synthetic databases is that they may be analyzed using the same software that the analyst would use on the original database, obviating the need for a special reconstruction procedure R .

Remark 2.1. Considerable care must be taken when programming real-valued mechanisms, such as the Laplace mechanism, due to subtleties in the implementation of floating point numbers. Otherwise differential privacy can be destroyed, as outputs with non-zero probability on a database x , may, because of rounding, have zero probability on adjacent databases y . This is just one way in which the implementation of floating point requires scrutiny in the context of differential privacy, and it is not unique.

2.4 Bibliographic notes

The definition of differential privacy is due to Dwork et al. [23]; the precise formulation used here and in the literature first appears in [20] and is due to Dwork and McSherry. The term “differential privacy” was coined by Michael Schroeder. The impossibility of semantic security is due to Dwork and Naor [25]. Composition and group privacy for $(\varepsilon, 0)$ -differentially private mechanisms is first addressed in [23].

Composition for (ε, δ) -differential privacy was first addressed in [21] (but see the corrected proof in Appendix B, due to Dwork and Lei [22]). The vulnerability of differential privacy to inappropriate implementations of floating point numbers was observed by Mironov, who proposed a mitigation [63].

3

Basic Techniques and Composition Theorems

After reviewing a few probabilistic tools, we present the Laplace mechanism, which gives differential privacy for real (vector) valued queries. An application of this leads naturally to the exponential mechanism, which is a method for differentially private selection from a discrete set of candidate outputs. We then analyze the cumulative privacy loss incurred by composing multiple differentially private mechanisms. Finally we give a method — the sparse vector technique — for privately reporting the outcomes of a potentially very large number of computations, provided that only a few are “significant.”

In this section, we describe some of the most basic techniques in differential privacy that we will come back to use again and again. The techniques described here form the basic building blocks for all of the other algorithms that we will develop.

3.1 Useful probabilistic tools

The following concentration inequalities will frequently be useful. We state them in easy to use forms rather than in their strongest forms.

Theorem 3.1 (Additive Chernoff Bound). Let X_1, \dots, X_m be independent random variables bounded such that $0 \leq X_i \leq 1$ for all i . Let $S = \frac{1}{m} \sum_{i=1}^m X_i$ denote their mean, and let $\mu = \mathbb{E}[S]$ denote their expected mean. Then:

$$\Pr[S > \mu + \varepsilon] \leq e^{-2m\varepsilon^2}$$

$$\Pr[S < \mu - \varepsilon] \leq e^{-2m\varepsilon^2}$$

Theorem 3.2 (Multiplicative Chernoff Bound). Let X_1, \dots, X_m be independent random variables bounded such that $0 \leq X_i \leq 1$ for all i . Let $S = \frac{1}{m} \sum_{i=1}^m X_i$ denote their mean, and let $\mu = \mathbb{E}[S]$ denote their expected mean. Then:

$$\Pr[S > (1 + \varepsilon)\mu] \leq e^{-m\mu\varepsilon^2/3}$$

$$\Pr[S < (1 - \varepsilon)\mu] \leq e^{-m\mu\varepsilon^2/2}$$

When we do not have independent random variables, all is not lost. We may still apply Azuma's inequality:

Theorem 3.3 (Azuma's Inequality). Let f be a function of m random variables X_1, \dots, X_m , each X_i taking values from a set A_i such that $\mathbb{E}[f]$ is bounded. Let c_i denote the maximum effect of X_i on f — i.e., for all $a_i, a'_i \in A_i$:

$$|\mathbb{E}[f|X_1, \dots, X_{i-1}, X_i = a_i] - \mathbb{E}[f|X_1, \dots, X_{i-1}, X_i = a'_i]| \leq c_i$$

Then:

$$\Pr[f(X_1, \dots, X_m) \geq \mathbb{E}[f] + t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^m c_i^2}\right)$$

Theorem 3.4 (Stirling's Approximation). $n!$ can be approximated by $\sqrt{2n\pi}(n/e)^n$:

$$\sqrt{2n\pi}(n/e)^n e^{1/(12n+1)} < n! < \sqrt{2n\pi}(n/e)^n e^{1/(12n)}.$$

3.2 Randomized response

Let us recall the simple randomized response mechanism, described in Section 2, for evaluating the frequency of embarrassing or illegal

behaviors. Let XYZ be such an activity. Faced with the query, “Have you engaged in XYZ in the past week?” the respondent is instructed to perform the following steps:

1. Flip a coin.
2. If **tails**, then respond truthfully.
3. If **heads**, then flip a second coin and respond “Yes” if heads and “No” if tails.

The intuition behind randomized response is that it provides “plausible deniability.” For example, a response of “Yes” may have been offered because the first and second coin flips were both Heads, which occurs with probability $1/4$. In other words, *privacy is obtained by process*, there are no “good” or “bad” responses. The process by which the responses are obtained affects how they may legitimately be interpreted. As the next claim shows, randomized response is differentially private.

Claim 3.5. The version of randomized response described above is $(\ln 3, 0)$ -differentially private.

Proof. Fix a respondent. A case analysis shows that $\Pr[\text{Response} = \text{Yes} | \text{Truth} = \text{Yes}] = 3/4$. Specifically, when the truth is “Yes” the outcome will be “Yes” if the first coin comes up tails (probability $1/2$) or the first and second come up heads (probability $1/4$), while $\Pr[\text{Response} = \text{Yes} | \text{Truth} = \text{No}] = 1/4$ (first comes up heads and second comes up tails; probability $1/4$). Applying similar reasoning to the case of a “No” answer, we obtain:

$$\begin{aligned} & \frac{\Pr[\text{Response} = \text{Yes} | \text{Truth} = \text{Yes}]}{\Pr[\text{Response} = \text{Yes} | \text{Truth} = \text{No}]} \\ &= \frac{3/4}{1/4} = \frac{\Pr[\text{Response} = \text{No} | \text{Truth} = \text{No}]}{\Pr[\text{Response} = \text{No} | \text{Truth} = \text{Yes}]} = 3. \end{aligned}$$

□

3.3 The laplace mechanism

Numeric queries, functions $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^k$, are one of the most fundamental types of database queries. These queries map databases to k

real numbers. One of the important parameters that will determine just how accurately we can answer such queries is their ℓ_1 sensitivity:

Definition 3.1 (ℓ_1 -sensitivity). The ℓ_1 -sensitivity of a function $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^k$ is:

$$\Delta f = \max_{\substack{x, y \in \mathbb{N}^{|\mathcal{X}|} \\ \|x-y\|_1=1}} \|f(x) - f(y)\|_1.$$

The ℓ_1 sensitivity of a function f captures the magnitude by which a single individual's data can change the function f in the worst case, and therefore, intuitively, the uncertainty in the response that we must introduce in order to hide the participation of a single individual. Indeed, we will formalize this intuition: the sensitivity of a function gives an upper bound on how much we must perturb its output to preserve privacy. One noise distribution naturally lends itself to differential privacy.

Definition 3.2 (The Laplace Distribution). The Laplace Distribution (centered at 0) with scale b is the distribution with probability density function:

$$\text{Lap}(x|b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right).$$

The variance of this distribution is $\sigma^2 = 2b^2$. We will sometimes write $\text{Lap}(b)$ to denote the Laplace distribution with scale b , and will sometimes abuse notation and write $\text{Lap}(b)$ simply to denote a random variable $X \sim \text{Lap}(b)$.

The Laplace distribution is a symmetric version of the exponential distribution.

We will now define the *Laplace Mechanism*. As its name suggests, the Laplace mechanism will simply compute f , and perturb each coordinate with noise drawn from the Laplace distribution. The scale of the noise will be calibrated to the sensitivity of f (divided by ε).¹

¹Alternately, using Gaussian noise with variance calibrated to $\Delta f \ln(1/\delta)/\varepsilon$, one can achieve (ε, δ) -differential privacy (see Appendix A). Use of the Laplace mechanism is cleaner and the two mechanisms behave similarly under composition (Theorem 3.20).

Definition 3.3 (The Laplace Mechanism). Given any function $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^k$, the Laplace mechanism is defined as:

$$\mathcal{M}_L(x, f(\cdot), \varepsilon) = f(x) + (Y_1, \dots, Y_k)$$

where Y_i are i.i.d. random variables drawn from $\text{Lap}(\Delta f / \varepsilon)$.

Theorem 3.6. The Laplace mechanism preserves $(\varepsilon, 0)$ -differential privacy.

Proof. Let $x \in \mathbb{N}^{|\mathcal{X}|}$ and $y \in \mathbb{N}^{|\mathcal{X}|}$ be such that $\|x - y\|_1 \leq 1$, and let $f(\cdot)$ be some function $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^k$. Let p_x denote the probability density function of $\mathcal{M}_L(x, f, \varepsilon)$, and let p_y denote the probability density function of $\mathcal{M}_L(y, f, \varepsilon)$. We compare the two at some arbitrary point $z \in \mathbb{R}^k$

$$\begin{aligned} \frac{p_x(z)}{p_y(z)} &= \prod_{i=1}^k \left(\frac{\exp(-\frac{\varepsilon|f(x)_i - z_i|}{\Delta f})}{\exp(-\frac{\varepsilon|f(y)_i - z_i|}{\Delta f})} \right) \\ &= \prod_{i=1}^k \exp \left(\frac{\varepsilon(|f(y)_i - z_i| - |f(x)_i - z_i|)}{\Delta f} \right) \\ &\leq \prod_{i=1}^k \exp \left(\frac{\varepsilon|f(x)_i - f(y)_i|}{\Delta f} \right) \\ &= \exp \left(\frac{\varepsilon \cdot \|f(x) - f(y)\|_1}{\Delta f} \right) \\ &\leq \exp(\varepsilon), \end{aligned}$$

where the first inequality follows from the triangle inequality, and the last follows from the definition of sensitivity and the fact that $\|x - y\|_1 \leq 1$. That $\frac{p_x(z)}{p_y(z)} \geq \exp(-\varepsilon)$ follows by symmetry. \square

Example 3.1 (Counting Queries). Counting queries are queries of the form “How many elements in the database satisfy Property P ?”. We will return to these queries again and again, sometimes in this pure form, sometimes in fractional form (“What fraction of the elements in the databases...?”), sometimes with weights (linear queries), and sometimes in slightly more complex forms (e.g., apply $h : \mathbb{N}^{|\mathcal{X}|} \rightarrow [0, 1]$ to each element in the database and sum the results). Counting is an

extremely powerful primitive. It captures everything learnable in the statistical queries learning model, as well as many standard datamining tasks and basic statistics. Since the sensitivity of a counting query is 1 (the addition or deletion of a single individual can change a count by at most 1), it is an immediate consequence of Theorem 3.6 that $(\varepsilon, 0)$ -differential privacy can be achieved for counting queries by the addition of noise scaled to $1/\varepsilon$, that is, by adding noise drawn from $\text{Lap}(1/\varepsilon)$. The expected distortion, or error, is $1/\varepsilon$, independent of the size of the database.

A fixed but arbitrary list of m counting queries can be viewed as a vector-valued query. Absent any further information about the set of queries a worst-case bound on the sensitivity of this vector-valued query is m , as a single individual might change every count. In this case $(\varepsilon, 0)$ -differential privacy can be achieved by adding noise scaled to m/ε to the true answer to each query.

We sometimes refer to the problem of responding to large numbers of (possibly arbitrary) queries as the *query release problem*.

Example 3.2 (Histogram Queries). In the special (but common) case in which the queries are structurally disjoint we can do much better — we don't necessarily have to let the noise scale with the number of queries. An example is the *histogram query*. In this type of query the universe $\mathbb{N}^{|\mathcal{X}|}$ is partitioned into cells, and the query asks how many database elements lie in each of the cells. Because the cells are disjoint, the addition or removal of a single database element can affect the count in exactly one cell, and the difference to that cell is bounded by 1, so histogram queries have sensitivity 1 and can be answered by adding independent draws from $\text{Lap}(1/\varepsilon)$ to the true count in each cell.

To understand the accuracy of the Laplace mechanism for general queries we use the following useful fact:

Fact 3.7. If $Y \sim \text{Lap}(b)$, then:

$$\Pr[|Y| \geq t \cdot b] = \exp(-t).$$

This fact, together with a union bound, gives us a simple bound on the accuracy of the Laplace mechanism:

Theorem 3.8. Let $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^k$, and let $y = \mathcal{M}_L(x, f(\cdot), \varepsilon)$. Then $\forall \delta \in (0, 1]$:

$$\Pr \left[\|f(x) - y\|_\infty \geq \ln \left(\frac{k}{\delta} \right) \cdot \left(\frac{\Delta f}{\varepsilon} \right) \right] \leq \delta$$

Proof. We have:

$$\begin{aligned} \Pr \left[\|f(x) - y\|_\infty \geq \ln \left(\frac{k}{\delta} \right) \cdot \left(\frac{\Delta f}{\varepsilon} \right) \right] &= \Pr \left[\max_{i \in [k]} |Y_i| \geq \ln \left(\frac{k}{\delta} \right) \cdot \left(\frac{\Delta f}{\varepsilon} \right) \right] \\ &\leq k \cdot \Pr \left[|Y_i| \geq \ln \left(\frac{k}{\delta} \right) \cdot \left(\frac{\Delta f}{\varepsilon} \right) \right] \\ &= k \cdot \left(\frac{\delta}{k} \right) \\ &= \delta \end{aligned}$$

where the second to last inequality follows from the fact that each $Y_i \sim \text{Lap}(\Delta f / \varepsilon)$ and Fact 3.7. \square

Example 3.3 (First Names). Suppose we wish to calculate which first names, from a list of 10,000 potential names, were the most common among participants of the 2010 census. This question can be represented as a query $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^{10000}$. This is a histogram query, and so has sensitivity $\Delta f = 1$, since every person can only have at most one first name. Using the above theorem, we see that we can simultaneously calculate the frequency of all 10,000 names with $(1, 0)$ -differential privacy, and with probability 95%, no estimate will be off by more than an additive error of $\ln(10000/.05) \approx 12.2$. That's pretty low error for a nation of more than 300,000,000 people!

Differentially Private Selection. The task in Example 3.3 is one of *differentially private selection*: the space of outcomes is discrete and the task is to produce a “best” answer, in this case the most populous histogram cell.

Example 3.4 (Most Common Medical Condition). Suppose we wish to know which condition is (approximately) the most common in the medical histories of a set of respondents, so the set of questions is, for each condition under consideration, whether the individual has ever received a diagnosis of this condition. Since individuals can experience many conditions, the sensitivity of this set of questions can be high. Nonetheless, as we next describe, this task can be addressed using addition of $\text{Lap}(1/\varepsilon)$ noise to each of the counts (note the small scale of the noise, which is independent of the total number of conditions). Crucially, the m noisy counts themselves will *not* be released (although the “winning” count can be released at no extra privacy cost).

Report Noisy Max. Consider the following simple algorithm to determine which of m counting queries has the highest value: Add independently generated Laplace noise $\text{Lap}(1/\varepsilon)$ to each count and return the index of the largest noisy count (we ignore the possibility of a tie). Call this algorithm Report Noisy Max.

Note the “information minimization” principle at work in the Report Noisy Max algorithm: rather than releasing all the noisy counts and allowing the analyst to find the max and its index, only the index corresponding to the maximum is made public. Since the data of an individual can affect all counts, the vector of counts has high ℓ_1 -sensitivity, specifically, $\Delta f = m$, and much more noise would be needed if we wanted to release all of the counts using the Laplace mechanism.

Claim 3.9. The Report Noisy Max algorithm is $(\varepsilon, 0)$ -differentially private.

Proof. Fix $D = D' \cup \{a\}$. Let c , respectively c' , denote the vector of counts when the database is D , respectively D' . We use two properties:

1. *Monotonicity of Counts.* For all $j \in [m]$, $c_j \geq c'_j$; and
2. *Lipschitz Property.* For all $j \in [m]$, $1 + c'_j \geq c_j$.

Fix any $i \in [m]$. We will bound from above and below the ratio of the probabilities that i is selected with D and with D' .

Fix r_{-i} , a draw from $[\text{Lap}(1/\varepsilon)]^{m-1}$ used for all the noisy counts except the i th count. We will argue for each r_{-i} independently. We

use the notation $\Pr[i|\xi]$ to mean the probability that the output of the Report Noisy Max algorithm is i , conditioned on ξ .

We first argue that $\Pr[i|D, r_{-i}] \leq e^\varepsilon \Pr[i|D', r_{-i}]$. Define

$$r^* = \min_{r_i} : c_i + r_i > c_j + r_j \quad \forall j \neq i.$$

Note that, having fixed r_{-i} , i will be the output (the argmax noisy count) when the database is D if and only if $r_i \geq r^*$.

We have, for all $1 \leq j \neq i \leq m$:

$$\begin{aligned} c_i + r^* &> c_j + r_j \\ \Rightarrow (1 + c'_i) + r^* &\geq c_i + r^* > c_j + r_j \geq c'_j + r_j \\ \Rightarrow c'_i + (r^* + 1) &> c'_j + r_j. \end{aligned}$$

Thus, if $r_i \geq r^* + 1$, then the i th count will be the maximum when the database is D' and the noise vector is (r_i, r_{-i}) . The probabilities below are over the choice of $r_i \sim \text{Lap}(1/\varepsilon)$.

$$\begin{aligned} \Pr[r_i \geq 1 + r^*] &\geq e^{-\varepsilon} \Pr[r_i \geq r^*] = e^{-\varepsilon} \Pr[i|D, r_{-i}] \\ \Rightarrow \Pr[i|D', r_{-i}] &\geq \Pr[r_i \geq 1 + r^*] \geq e^{-\varepsilon} \Pr[r_i \geq r^*] = e^{-\varepsilon} \Pr[i|D, r_{-i}], \end{aligned}$$

which, after multiplying through by e^ε , yields what we wanted to show: $\Pr[i|D, r_{-i}] \leq e^\varepsilon \Pr[i|D', r_{-i}]$.

We now argue that $\Pr[i|D', r_{-i}] \leq e^\varepsilon \Pr[i|D, r_{-i}]$. Define

$$r^* = \min_{r_i} : c'_i + r_i > c'_j + r_j \quad \forall j \neq i.$$

Note that, having fixed r_{-i} , i will be the output (argmax noisy count) when the database is D' if and only if $r_i \geq r^*$.

We have, for all $1 \leq j \neq i \leq m$:

$$\begin{aligned} c'_i + r^* &> c'_j + r_j \\ \Rightarrow 1 + c'_i + r^* &> 1 + c'_j + r_j \\ \Rightarrow c'_i + (r^* + 1) &> (1 + c'_j) + r_j \\ \Rightarrow c_i + (r^* + 1) &\geq c'_i + (r^* + 1) > (1 + c'_j) + r_j \geq c_j + r_j. \end{aligned}$$

Thus, if $r_i \geq r^* + 1$, then i will be the output (the argmax noisy count) on database D with randomness (r_i, r_{-i}) . We therefore have, with probabilities taken over choice of r_i :

$$\Pr[i|D, r_{-i}] \geq \Pr[r_i \geq r^* + 1] \geq e^{-\varepsilon} \Pr[r_i \geq r^*] = e^{-\varepsilon} \Pr[i|D', r_{-i}],$$

which, after multiplying through by e^ε , yields what we wanted to show:
 $\Pr[i|D', r_{-i}] \leq e^\varepsilon \Pr[i|D, r_{-i}]$. \square

3.4 The exponential mechanism

In both the “most common name” and “most common condition” examples the “utility” of a response (name or medical condition, respectively) we estimated counts using Laplace noise and reported the noisy maximum. In both examples the utility of the response is directly related to the noise values generated; that is, the popularity of the name or condition is appropriately measured on the same scale and in the same units as the magnitude of the noise.

The *exponential mechanism* was designed for situations in which we wish to choose the “best” response but adding noise directly to the computed quantity can completely destroy its value, such as setting a price in an auction, where the goal is to maximize revenue, and adding a small amount of positive noise to the optimal price (in order to protect the privacy of a bid) could dramatically reduce the resulting revenue.

Example 3.5 (Pumpkins.). Suppose we have an abundant supply of pumpkins and four bidders: A, F, I, K , where A, F, I each bid \$1.00 and K bids \$3.01. What is the optimal price? At \$3.01 the revenue is \$3.01, at \$3.00 and at \$1.00 the revenue is \$3.00, but at \$3.02 the revenue is zero!

The exponential mechanism is the natural building block for answering queries with arbitrary utilities (and arbitrary non-numeric range), while preserving differential privacy. Given some arbitrary range \mathcal{R} , the exponential mechanism is defined with respect to some utility function $u : \mathbb{N}^{|\mathcal{X}|} \times \mathcal{R} \rightarrow \mathbb{R}$, which maps database/output pairs to utility scores. Intuitively, for a fixed database x , the user prefers that the mechanism outputs some element of \mathcal{R} with the maximum possible utility score. Note that when we talk about the sensitivity of the utility score $u : \mathbb{N}^{|\mathcal{X}|} \times \mathcal{R} \rightarrow \mathbb{R}$, we care only about the sensitivity of u with respect to its database argument; it can be arbitrarily sensitive in its

range argument:

$$\Delta u \equiv \max_{r \in \mathcal{R}} \max_{x,y: \|x-y\|_1 \leq 1} |u(x, r) - u(y, r)|.$$

The intuition behind the exponential mechanism is to output each possible $r \in \mathcal{R}$ with probability proportional to $\exp(\varepsilon u(x, r)/\Delta u)$ and so the privacy loss is approximately:

$$\ln \left(\frac{\exp(\varepsilon u(x, r)/\Delta u)}{\exp(\varepsilon u(y, r)/\Delta u)} \right) = \varepsilon[u(x, r) - u(y, r)]/\Delta u \leq \varepsilon.$$

This intuitive view overlooks some effects of a normalization term which arises when an additional person in the database causes the utilities of some elements $r \in \mathcal{R}$ to decrease and others to increase. The actual mechanism, defined next, reserves half the privacy budget for changes in the normalization term.

Definition 3.4 (The Exponential Mechanism). The exponential mechanism $\mathcal{M}_E(x, u, \mathcal{R})$ selects and outputs an element $r \in \mathcal{R}$ with probability proportional to $\exp(\frac{\varepsilon u(x, r)}{2\Delta u})$.

The exponential mechanism can define a complex distribution over a large arbitrary domain, and so it may not be possible to implement the exponential mechanism efficiently when the range of u is super-polynomially large in the natural parameters of the problem.

Returning to the pumpkin example, utility for a price p on database x is simply the profit obtained when the price is p and the demand curve is as described by x . It is important that the range of *potential* prices is independent of the actual bids. Otherwise there would exist a price with non-zero weight in one dataset and zero weight in a neighboring set, violating differential privacy.

Theorem 3.10. The exponential mechanism preserves $(\varepsilon, 0)$ -differential privacy.

Proof. For clarity, we assume the range \mathcal{R} of the exponential mechanism is finite, but this is not necessary. As in all differential privacy proofs, we consider the ratio of the probability that an instantiation

of the exponential mechanism outputs some element $r \in \mathcal{R}$ on two neighboring databases $x \in \mathbb{N}^{|\mathcal{X}|}$ and $y \in \mathbb{N}^{|\mathcal{X}|}$ (i.e., $\|x - y\|_1 \leq 1$).

$$\begin{aligned} \frac{\Pr[\mathcal{M}_E(x, u, \mathcal{R}) = r]}{\Pr[\mathcal{M}_E(y, u, \mathcal{R}) = r]} &= \frac{\left(\frac{\exp(\frac{\varepsilon u(x, r)}{2\Delta u})}{\sum_{r' \in \mathcal{R}} \exp(\frac{\varepsilon u(x, r')}{2\Delta u})} \right)}{\left(\frac{\exp(\frac{\varepsilon u(y, r)}{2\Delta u})}{\sum_{r' \in \mathcal{R}} \exp(\frac{\varepsilon u(y, r')}{2\Delta u})} \right)} \\ &= \left(\frac{\exp(\frac{\varepsilon u(x, r)}{2\Delta u})}{\exp(\frac{\varepsilon u(y, r)}{2\Delta u})} \right) \cdot \left(\frac{\sum_{r' \in \mathcal{R}} \exp(\frac{\varepsilon u(y, r')}{2\Delta u})}{\sum_{r' \in \mathcal{R}} \exp(\frac{\varepsilon u(x, r')}{2\Delta u})} \right) \\ &= \exp \left(\frac{\varepsilon(u(x, r) - u(y, r))}{2\Delta u} \right) \\ &\quad \cdot \left(\frac{\sum_{r' \in \mathcal{R}} \exp(\frac{\varepsilon u(y, r')}{2\Delta u})}{\sum_{r' \in \mathcal{R}} \exp(\frac{\varepsilon u(x, r')}{2\Delta u})} \right) \\ &\leq \exp \left(\frac{\varepsilon}{2} \right) \cdot \exp \left(\frac{\varepsilon}{2} \right) \cdot \left(\frac{\sum_{r' \in \mathcal{R}} \exp(\frac{\varepsilon u(x, r')}{2\Delta u})}{\sum_{r' \in \mathcal{R}} \exp(\frac{\varepsilon u(x, r')}{2\Delta u})} \right) \\ &= \exp(\varepsilon). \end{aligned}$$

Similarly, $\frac{\Pr[\mathcal{M}_E(y, u) = r]}{\Pr[\mathcal{M}_E(x, u) = r]} \geq \exp(-\varepsilon)$ by symmetry. \square

The exponential mechanism can often give strong utility guarantees, because it discounts outcomes exponentially quickly as their quality score falls off. For a given database x and a given utility measure $u : \mathbb{N}^{|\mathcal{X}|} \times \mathcal{R} \rightarrow \mathbb{R}$, let $\text{OPT}_u(x) = \max_{r \in \mathcal{R}} u(x, r)$ denote the maximum utility score of any element $r \in \mathcal{R}$ with respect to database x . We will bound the probability that the exponential mechanism returns a “good” element of \mathcal{R} , where good will be measured in terms of $\text{OPT}_u(x)$. The result is that it will be highly unlikely that the returned element r has a utility score that is inferior to $\text{OPT}_u(x)$ by more than an additive factor of $O((\Delta u/\varepsilon) \log |\mathcal{R}|)$.

Theorem 3.11. Fixing a database x , let $\mathcal{R}_{\text{OPT}} = \{r \in \mathcal{R} : u(x, r) = \text{OPT}_u(x)\}$ denote the set of elements in \mathcal{R} which attain utility score

$\text{OPT}_u(x)$. Then:

$$\Pr \left[u(\mathcal{M}_E(x, u, \mathcal{R})) \leq \text{OPT}_u(x) - \frac{2\Delta u}{\varepsilon} \left(\ln \left(\frac{|\mathcal{R}|}{|\mathcal{R}_{\text{OPT}}|} \right) + t \right) \right] \leq e^{-t}$$

Proof.

$$\begin{aligned} \Pr[u(\mathcal{M}_E(x, u, \mathcal{R})) \leq c] &\leq \frac{|\mathcal{R}| \exp(\varepsilon c / 2\Delta u)}{|\mathcal{R}_{\text{OPT}}| \exp(\varepsilon \text{OPT}_u(x) / 2\Delta u)} \\ &= \frac{|\mathcal{R}|}{|\mathcal{R}_{\text{OPT}}|} \exp \left(\frac{\varepsilon(c - \text{OPT}_u(x))}{2\Delta u} \right). \end{aligned}$$

The inequality follows from the observation that each $r \in \mathcal{R}$ with $u(x, r) \leq c$ has un-normalized probability mass at most $\exp(\varepsilon c / 2\Delta u)$, and hence the entire set of such “bad” elements r has total un-normalized probability mass at most $|\mathcal{R}| \exp(\varepsilon c / 2\Delta u)$. In contrast, we know that there exist at least $|\mathcal{R}_{\text{OPT}}| \geq 1$ elements with $u(x, r) = \text{OPT}_u(x)$, and hence un-normalized probability mass $\exp(\varepsilon \text{OPT}_u(x) / 2\Delta u)$, and so this is a lower bound on the normalization term.

The theorem follows from plugging in the appropriate value for c . \square

Since we always have $|\mathcal{R}_{\text{OPT}}| \geq 1$, we can more commonly make use of the following simple corollary:

Corollary 3.12. Fixing a database x , we have:

$$\Pr \left[u(\mathcal{M}_E(x, u, \mathcal{R})) \leq \text{OPT}_u(x) - \frac{2\Delta u}{\varepsilon} (\ln(|\mathcal{R}|) + t) \right] \leq e^{-t}$$

As seen in the proofs of Theorem 3.11 and Corollary 3.12, the Exponential Mechanism can be particularly easy to analyze.

Example 3.6 (Best of Two). Consider the simple question of determining which of exactly two medical conditions A and B is more common. Let the two true counts be 0 for condition A and $c > 0$ for condition B . Our notion of utility will be tied to the actual counts, so that conditions with bigger counts have higher utility and $\Delta u = 1$. Thus, the utility of A is 0 and the utility of B is c . Using the Exponential Mechanism

we can immediately apply Corollary 3.12 to see that the probability of observing (wrong) outcome A is at most $2e^{-c(\varepsilon/(2\Delta u))} = 2e^{-c\varepsilon/2}$.

Analyzing Report Noisy Max appears to be more complicated, as it requires understanding what happens in the (probability 1/4) case when the noise added to the count for A is positive and the noise added to the count for B is negative.

A function is *monotonic in the data set* if the addition of an element to the data set cannot cause the value of the function to decrease. Counting queries are monotonic; so is the revenue obtained by offering a fixed price to a collection of buyers.

Consider the *Report One-Sided Noisy Arg-Max* mechanism, which adds noise to the *utility* of each potential output drawn from the *one-sided* exponential distribution with parameter $\varepsilon/\Delta u$ in the case of a monotonic utility, or parameter $\varepsilon/2\Delta u$ for the case of a non-monotonic utility, and reports the resulting arg-max.

With this algorithm, whose privacy proof is almost identical to that of Report Noisy Max (but loses a factor of two when the utility is non-monotonic), we immediately obtain in Example 3.6 above that outcome A is exponentially in $c(\varepsilon/\Delta u) = c\varepsilon$ less likely to be selected than outcome B .

Theorem 3.13. Report One-Sided Noisy Arg-Max, when run with parameter $\varepsilon/2\Delta u$ yields the same distribution on outputs as the exponential mechanism.

3.5 Composition theorems

Now that we have several building blocks for designing differentially private algorithms, it is important to understand how we can combine them to design more sophisticated algorithms. In order to use these tools, we would like that the combination of two differentially private algorithms be differentially private itself. Indeed, as we will see, this is the case. Of course the parameters ε and δ will necessarily degrade — consider repeatedly computing the same statistic using the Laplace mechanism, scaled to give ε -differential privacy each time. The average of the answer given by each instance of the mechanism will eventually

converge to the true value of the statistic, and so we cannot avoid that the strength of our privacy guarantee will degrade with repeated use. In this section we give theorems showing how exactly the parameters ε and δ compose when differentially private subroutines are combined.

Let us first begin with an easy warm up: we will see that the independent use of an $(\varepsilon_1, 0)$ -differentially private algorithm and an $(\varepsilon_2, 0)$ -differentially private algorithm, when taken together, is $(\varepsilon_1 + \varepsilon_2, 0)$ -differentially private.

Theorem 3.14. Let $\mathcal{M}_1 : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathcal{R}_1$ be an ε_1 -differentially private algorithm, and let $\mathcal{M}_2 : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathcal{R}_2$ be an ε_2 -differentially private algorithm. Then their combination, defined to be $\mathcal{M}_{1,2} : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathcal{R}_1 \times \mathcal{R}_2$ by the mapping: $\mathcal{M}_{1,2}(x) = (\mathcal{M}_1(x), \mathcal{M}_2(x))$ is $\varepsilon_1 + \varepsilon_2$ -differentially private.

Proof. Let $x, y \in \mathbb{N}^{|\mathcal{X}|}$ be such that $\|x - y\|_1 \leq 1$. Fix any $(r_1, r_2) \in \mathcal{R}_1 \times \mathcal{R}_2$. Then:

$$\begin{aligned} \frac{\Pr[\mathcal{M}_{1,2}(x) = (r_1, r_2)]}{\Pr[\mathcal{M}_{1,2}(y) = (r_1, r_2)]} &= \frac{\Pr[\mathcal{M}_1(x) = r_1] \Pr[\mathcal{M}_2(x) = r_2]}{\Pr[\mathcal{M}_1(y) = r_1] \Pr[\mathcal{M}_2(y) = r_2]} \\ &= \left(\frac{\Pr[\mathcal{M}_1(x) = r_1]}{\Pr[\mathcal{M}_1(y) = r_1]} \right) \left(\frac{\Pr[\mathcal{M}_2(x) = r_2]}{\Pr[\mathcal{M}_2(y) = r_2]} \right) \\ &\leq \exp(\varepsilon_1) \exp(\varepsilon_2) \\ &= \exp(\varepsilon_1 + \varepsilon_2) \end{aligned}$$

By symmetry, $\frac{\Pr[\mathcal{M}_{1,2}(x) = (r_1, r_2)]}{\Pr[\mathcal{M}_{1,2}(y) = (r_1, r_2)]} \geq \exp(-(\varepsilon_1 + \varepsilon_2))$. \square

The composition theorem can be applied repeatedly to obtain the following corollary:

Corollary 3.15. Let $\mathcal{M}_i : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathcal{R}_i$ be an $(\varepsilon_i, 0)$ -differentially private algorithm for $i \in [k]$. Then if $\mathcal{M}_{[k]} : \mathbb{N}^{|\mathcal{X}|} \rightarrow \prod_{i=1}^k \mathcal{R}_i$ is defined to be $\mathcal{M}_{[k]}(x) = (\mathcal{M}_1(x), \dots, \mathcal{M}_k(x))$, then $\mathcal{M}_{[k]}$ is $(\sum_{i=1}^k \varepsilon_i, 0)$ -differentially private.

A proof of the generalization of this theorem to (ε, δ) -differential privacy appears in Appendix B:

Theorem 3.16. Let $\mathcal{M}_i : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathcal{R}_i$ be an $(\varepsilon_i, \delta_i)$ -differentially private algorithm for $i \in [k]$. Then if $\mathcal{M}_{[k]} : \mathbb{N}^{|\mathcal{X}|} \rightarrow \prod_{i=1}^k \mathcal{R}_i$ is defined to be $\mathcal{M}_{[k]}(x) = (\mathcal{M}_1(x), \dots, \mathcal{M}_k(x))$, then $\mathcal{M}_{[k]}$ is $(\sum_{i=1}^k \varepsilon_i, \sum_{i=1}^k \delta_i)$ -differentially private.

It is a strength of differential privacy that composition is “automatic,” in that the bounds obtained hold without any special effort by the database curator.

3.5.1 Composition: some technicalities

In the remainder of this section, we will prove a more sophisticated composition theorem. To this end, we will need some definitions and lemmas, rephrasing differential privacy in terms of distance measures between distributions. In the fractional quantities below, if the denominator is zero, then we define the value of the fraction to be infinite (the numerators will always be positive).

Definition 3.5 (KL-Divergence). The KL-Divergence, or Relative Entropy, between two random variables Y and Z taking values from the same domain is defined to be:

$$D(Y||Z) = \mathbb{E}_{y \sim Y} \left[\ln \frac{\Pr[Y = y]}{\Pr[Z = y]} \right].$$

It is known that $D(Y||Z) \geq 0$, with equality if and only if Y and Z are identically distributed. However, D is not symmetric, does not satisfy the triangle inequality, and can even be infinite, specifically when $\text{Supp}(Y)$ is not contained in $\text{Supp}(Z)$.

Definition 3.6 (Max Divergence). The Max Divergence between two random variables Y and Z taking values from the same domain is defined to be:

$$D_\infty(Y||Z) = \max_{S \subseteq \text{Supp}(Y)} \left[\ln \frac{\Pr[Y \in S]}{\Pr[Z \in S]} \right].$$

The δ -Approximate Max Divergence between Y and Z is defined to be:

$$D_\infty^\delta(Y||Z) = \max_{S \subseteq \text{Supp}(Y): \Pr[Y \in S] \geq \delta} \left[\ln \frac{\Pr[Y \in S] - \delta}{\Pr[Z \in S]} \right]$$

Remark 3.1. Note that a mechanism \mathcal{M} is

1. ε -differentially private if and only if on every two neighboring databases x and y , $D_\infty(\mathcal{M}(x)\|\mathcal{M}(y)) \leq \varepsilon$ and $D_\infty(\mathcal{M}(y)\|\mathcal{M}(x)) \leq \varepsilon$; and is
2. (ε, δ) -differentially private if and only if on every two neighboring databases x, y : $D_\infty^\delta(\mathcal{M}(x)\|\mathcal{M}(y)) \leq \varepsilon$ and $D_\infty^\delta(\mathcal{M}(y)\|\mathcal{M}(x)) \leq \varepsilon$.

One other distance measure that will be useful is the *statistical distance* between two random variables Y and Z , defined as

$$\Delta(Y, Z) \stackrel{\text{def}}{=} \max_S |\Pr[Y \in S] - \Pr[Z \in S]|.$$

We say that Y and Z are δ -close if $\Delta(Y, Z) \leq \delta$.

We will use the following reformulations of approximate max-divergence in terms of exact max-divergence and statistical distance:

Lemma 3.17.

1. $D_\infty^\delta(Y\|Z) \leq \varepsilon$ if and only if there exists a random variable Y' such that $\Delta(Y, Y') \leq \delta$ and $D_\infty(Y'\|Z) \leq \varepsilon$.
2. We have both $D_\infty^\delta(Y\|Z) \leq \varepsilon$ and $D_\infty^\delta(Z\|Y) \leq \varepsilon$ if and only if there exist random variables Y', Z' such that $\Delta(Y, Y') \leq \delta/(e^\varepsilon + 1)$, $\Delta(Z, Z') \leq \delta/(e^\varepsilon + 1)$, and $D_\infty(Y'\|Z') \leq \varepsilon$.

Proof. For Part 1, suppose there exists Y' δ -close to Y such that $D_\infty(Y\|Z) \leq \varepsilon$. Then for every S ,

$$\Pr[Y \in S] \leq \Pr[Y' \in S] + \delta \leq e^\varepsilon \cdot \Pr[Z \in S] + \delta,$$

and thus $D_\infty^\delta(Y\|Z) \leq \varepsilon$.

Conversely, suppose that $D_\infty^\delta(Y\|Z) \leq \varepsilon$. Let $S = \{y : \Pr[Y = y] > e^\varepsilon \cdot \Pr[Z = y]\}$. Then

$$\sum_{y \in S} (\Pr[Y = y] - e^\varepsilon \cdot \Pr[Z = y]) = \Pr[Y \in S] - e^\varepsilon \cdot \Pr[Z \in S] \leq \delta.$$

Moreover, if we let $T = \{y : \Pr[Y = y] < \Pr[Z = y]\}$, then we have

$$\begin{aligned} \sum_{y \in T} (\Pr[Z = y] - \Pr[Y = y]) &= \sum_{y \notin T} (\Pr[Y = y] - \Pr[Z = y]) \\ &\geq \sum_{y \in S} (\Pr[Y = y] - \Pr[Z = y]) \\ &\geq \sum_{y \in S} (\Pr[Y = y] - e^\varepsilon \cdot \Pr[Z = y]) / \end{aligned}$$

Thus, we can obtain Y' from Y by lowering the probabilities on S and raising the probabilities on T to satisfy:

1. For all $y \in S$, $\Pr[Y' = y] = e^\varepsilon \cdot \Pr[Z = y] < \Pr[Y = y]$.
2. For all $y \in T$, $\Pr[Y = y] \leq \Pr[Y' = y] \leq \Pr[Z = y]$.
3. For all $y \notin S \cup T$, $\Pr[Y' = y] = \Pr[Y = y] \leq e^\varepsilon \cdot \Pr[Z = y]$.

Then $D_\infty(Y' \| Z) \leq \varepsilon$ by inspection, and

$$\Delta(Y, Y') = \Pr[Y \in S] - \Pr[Y' \in S] = \Pr[Y \in S] - e^\varepsilon \cdot \Pr[Z \in S] \leq \delta.$$

We now prove Part 2. Suppose there exist random variables Y' and Z' as stated. Then, for every set S ,

$$\begin{aligned} \Pr[Y \in S] &\leq \Pr[Y' \in S] + \frac{\delta}{e^\varepsilon + 1} \\ &\leq e^\varepsilon \cdot \Pr[Z' \in S] + \frac{\delta}{e^\varepsilon + 1} \\ &\leq e^\varepsilon \cdot \left(\Pr[Z \in S] + \frac{\delta}{e^\varepsilon + 1} \right) + \frac{\delta}{e^\varepsilon + 1} \\ &= e^\varepsilon \cdot \Pr[Z \in S] + \delta. \end{aligned}$$

Thus $D_\infty^\delta(Y \| Z) \leq \varepsilon$, and by symmetry, $D_\infty^\delta(Z \| Y) \leq \varepsilon$.

Conversely, given Y and Z such that $D_\infty^\delta(Y \| Z) \leq \varepsilon$ and $D_\infty^\delta(Z \| Y) \leq \varepsilon$, we proceed similarly to Part 1. However, instead of simply decreasing the probability mass of Y on S to obtain Y' and eliminate the gap with $e^\varepsilon \cdot Z$, we also increase the probability mass of Z on S . Specifically, for every $y \in S$, we'll take

$$\begin{aligned} \Pr[Y' = y] &= e^\varepsilon \cdot \Pr[Z' = y] \\ &= \frac{e^\varepsilon}{1 + e^\varepsilon} \cdot (\Pr[Y = y] + \Pr[Z = y]) \\ &\in [e^\varepsilon \cdot \Pr[Z = y], \Pr[Y = y]]. \end{aligned}$$

This also implies that for $y \in S$, we have:

$$\begin{aligned} & \Pr[Y = y] - \Pr[Y' = y] \\ &= \Pr[Z' = y] - \Pr[Z = y] \frac{\Pr[Y = y] - e^\varepsilon \cdot \Pr[Z = y]}{e^\varepsilon + 1}, \end{aligned}$$

and thus

$$\begin{aligned} \alpha &\stackrel{\text{def}}{=} \sum_{y \in S} (\Pr[Y = y] - \Pr[Y' = y]) \\ &= \sum_{y \in S} (\Pr[Z' = y] - \Pr[Z = y]) \\ &= \frac{\Pr[Y \in S] - e^\varepsilon \cdot \Pr[Z \in S]}{e^\varepsilon + 1} \\ &\leq \frac{\delta}{e^\varepsilon + 1}. \end{aligned}$$

Similarly on the set $S' = \{y : \Pr[Z = y] > e^\varepsilon \cdot \Pr[Y = y]\}$, we can decrease the probability mass of Z and increase the probability mass of Y by a total of some $\alpha' \leq \delta/(e^\varepsilon + 1)$ so that for every $y \in S'$, we have $\Pr[Z' = y] = e^\varepsilon \cdot \Pr[Y' = y]$.

If $\alpha = \alpha'$, then we can take $\Pr[Z' = y] = \Pr[Z = y]$ and $\Pr[Y' = y] = \Pr[Y = y]$ for all $y \notin S \cup S'$, giving $D_\infty(Y \| Z) \leq \varepsilon$ and $\Delta(Y, Y') = \Delta(Z, Z') = \alpha$. If $\alpha \neq \alpha'$, say $\alpha > \alpha'$, then we need to still increase the probability mass of Y' and decrease the mass of Z' by a total of $\beta = \alpha - \alpha'$ on points outside of $S \cup S'$ in order to ensure that the probabilities sum to 1. That is, if we try to take the “mass functions” $\Pr[Y' = y]$ and $\Pr[Z' = y]$ as defined above, then while we do have the property that for every y , $\Pr[Y' = y] \leq e^\varepsilon \cdot \Pr[Z' = y]$ and $\Pr[Z' = y] \leq e^\varepsilon \cdot \Pr[Y' = y]$ we also have $\sum_y \Pr[Y' = y] = 1 - \beta$ and $\sum_y \Pr[Z' = y] = 1 + \beta$. However, this means that if we let $R = \{y : \Pr[Y' = y] < \Pr[Z' = y]\}$, then

$$\sum_{y \in R} (\Pr[Z' = y] - \Pr[Y' = y]) \geq \sum_y (\Pr[Z' = y] - \Pr[Y' = y]) = 2\beta.$$

So we can increase the probability mass of Y' on points in R by a total of β and decrease the probability mass of Z' on points in R by a total of β , while retaining the property that for all $y \in R$, $\Pr[Y' = y] \leq \Pr[Z' = y]$.

The resulting Y' and Z' have the properties we want: $D_\infty(Y', Z') \leq \varepsilon$ and $\Delta(Y, Y'), \Delta(Z, Z') \leq \alpha$. \square

Lemma 3.18. Suppose that random variables Y and Z satisfy $D_\infty(Y\|Z) \leq \varepsilon$ and $D_\infty(Z\|Y) \leq \varepsilon$. Then $D(Y\|Z) \leq \varepsilon \cdot (e^\varepsilon - 1)$.

Proof. We know that for any Y and Z it is the case that $D(Y\|Z) \geq 0$ (via the “log-sum inequality”), and so it suffices to bound $D(Y\|Z) + D(Z\|Y)$. We get:

$$\begin{aligned} D(Y\|Z) &\leq D(Y\|Z) + D(Z\|Y) \\ &= \sum_y \Pr[Y = y] \cdot \left(\ln \frac{\Pr[Y = y]}{\Pr[Z = y]} + \ln \frac{\Pr[Z = y]}{\Pr[Y = y]} \right) \\ &\quad + (\Pr[Z = y] - \Pr[Y = y]) \cdot \left(\ln \frac{\Pr[Z = y]}{\Pr[Y = y]} \right) \\ &\leq \sum_y [0 + |\Pr[Z = y] - \Pr[Y = y]| \cdot \varepsilon] \\ &= \varepsilon \cdot \sum_y [\max\{\Pr[Y = y], \Pr[Z = y]\} \\ &\quad - \min\{\Pr[Y = y], \Pr[Z = y]\}] \\ &\leq \varepsilon \cdot \sum_y [(e^\varepsilon - 1) \cdot \min\{\Pr[Y = y], \Pr[Z = y]\}] \\ &\leq \varepsilon \cdot (e^\varepsilon - 1). \end{aligned}$$

\square

Lemma 3.19 (Azuma's Inequality). Let C_1, \dots, C_k be real-valued random variables such that for every $i \in [k]$, $\Pr[|C_i| \leq \alpha] = 1$, and for

every $(c_1, \dots, c_{i-1}) \in \text{Supp}(C_1, \dots, C_{i-1})$, we have

$$\mathbb{E}[C_i | C_1 = c_1, \dots, C_{i-1} = c_{i-1}] \leq \beta.$$

Then for every $z > 0$, we have

$$\Pr \left[\sum_{i=1}^k C_i > k\beta + z\sqrt{k} \cdot \alpha \right] \leq e^{-z^2/2}.$$

3.5.2 Advanced composition

In addition to allowing the parameters to degrade more slowly, we would like our theorem to be able to handle more complicated forms of composition. However, before we begin, we must discuss what exactly we mean by composition. We would like our definitions to cover the following two interesting scenarios:

1. Repeated use of differentially private algorithms on the same database. This allows both the repeated use of the same mechanism multiple times, as well as the modular construction of differentially private algorithms from arbitrary private building blocks.
2. Repeated use of differentially private algorithms on *different* databases that may nevertheless contain information relating to the same individual. This allows us to reason about the cumulative privacy loss of a single individual whose data might be spread across multiple data sets, each of which may be used independently in a differentially private way. Since new databases are created all the time, and the adversary may actually influence the makeup of these new databases, this is a fundamentally different problem than repeatedly querying a single, fixed, database.

We want to model composition where the adversary can adaptively affect the databases being input to future mechanisms, as well as the queries to those mechanisms. Let \mathcal{F} be a family of database access mechanisms. (For example \mathcal{F} could be the set of all ε -differentially private mechanisms.) For a probabilistic adversary A , we consider two experiments, Experiment 0 and Experiment 1, defined as follows.

Experiment b for family \mathcal{F} and adversary A :

For $i = 1, \dots, k$:

1. A outputs two adjacent databases x_i^0 and x_i^1 , a mechanism $\mathcal{M}_i \in \mathcal{F}$, and parameters w_i .
2. A receives $y_i \in_R \mathcal{M}_i(w_i, x_{i,b})$.

We allow the adversary A above to be stateful throughout the experiment, and thus it may choose the databases, mechanisms, and the parameters adaptively depending on the outputs of previous mechanisms. We define A 's *view* of the experiment to be A 's coin tosses and all of the mechanism outputs (y_1, \dots, y_k) . (The x_i^j 's, \mathcal{M}_i 's, and w_i 's can all be reconstructed from these.)

For intuition, consider an adversary who always chooses x_i^0 to hold Bob's data and x_i^1 to differ only in that Bob's data are deleted. Then Experiment 0 can be thought of as the “real world,” where Bob allows his data to be used in many data releases, and Experiment 1 as an “ideal world,” where the outcomes of these data releases do not depend on Bob's data. Our definitions of privacy still require these two experiments to be “close” to each other, in the same way as required by the definitions of differential privacy. The intuitive guarantee to Bob is that the adversary “can't tell”, given the output of all k mechanisms, whether Bob's data was ever used.

Definition 3.7. We say that the family \mathcal{F} of database access mechanisms satisfies ε -differential privacy under k -fold adaptive composition if for every adversary A , we have $D_\infty(V^0 \| V^1) \leq \varepsilon$ where V^b denotes the view of A in k -fold Composition Experiment b above.

(ε, δ) -differential privacy under k -fold adaptive composition instead requires that $D_\infty^\delta(V^0 \| V^1) \leq \varepsilon$.

Theorem 3.20 (Advanced Composition). For all $\varepsilon, \delta, \delta' \geq 0$, the class of (ε, δ) -differentially private mechanisms satisfies $(\varepsilon', k\delta + \delta')$ -differential privacy under k -fold adaptive composition for:

$$\varepsilon' = \sqrt{2k \ln(1/\delta')} \varepsilon + k\varepsilon(e^\varepsilon - 1).$$

Proof. A view of the adversary A consists of a tuple of the form $v = (r, y_1, \dots, y_k)$, where r is the coin tosses of A and y_1, \dots, y_k are the outputs of the mechanisms $\mathcal{M}_1, \dots, \mathcal{M}_k$. Let

$$B = \{v : \Pr[V^0 = v] > e^{\varepsilon'} \cdot \Pr[V^1 = v]\}.$$

We will show that $\Pr[V^0 \in B] \leq \delta$, and hence for every set S , we have

$$\Pr[V^0 \in S] \leq \Pr[V^0 \in B] + \Pr[V^0 \in (S \setminus B)] \leq \delta + e^{\varepsilon'} \cdot \Pr[V^1 \in S].$$

This is equivalent to saying that $D_\infty^\delta(V^0 \| V^1) \leq \varepsilon'$.

It remains to show $\Pr[V^0 \in B] \leq \delta$. Let random variable $V^0 = (R^0, Y_1^0, \dots, Y_k^0)$ denote the view of A in Experiment 0 and $V^1 = (R^1, Y_1^1, \dots, Y_k^1)$ the view of A in Experiment 1. Then for a fixed view $v = (r, y_1, \dots, y_k)$, we have

$$\begin{aligned} & \ln \left(\frac{\Pr[V^0 = v]}{\Pr[V^1 = v]} \right) \\ &= \ln \left(\frac{\Pr[R^0 = r]}{\Pr[R^1 = r]} \cdot \prod_{i=1}^k \frac{\Pr[Y_i^0 = y_i | R^0 = r, Y_1^0 = y_1, \dots, Y_{i-1}^0 = y_{i-1}]}{\Pr[Y_i^1 = y_i | R^1 = r, Y_1^1 = y_1, \dots, Y_{i-1}^1 = y_{i-1}]} \right) \\ &= \sum_{i=1}^k \ln \left(\frac{\Pr[Y_i^0 = y_i | R^0 = r, Y_1^0 = y_1, \dots, Y_{i-1}^0 = y_{i-1}]}{\Pr[Y_i^1 = y_i | R^1 = r, Y_1^1 = y_1, \dots, Y_{i-1}^1 = y_{i-1}]} \right) \\ &\stackrel{\text{def}}{=} \sum_{i=1}^k c_i(r, y_1, \dots, y_i). \end{aligned}$$

Now for every prefix (r, y_1, \dots, y_{i-1}) we condition on $R^0 = r$, $Y_1^0 = y_1, \dots, Y_{i-1}^0 = y_{i-1}$, and analyze the expectation and maximum possible value of the random variable $c_i(R^0, Y_1^0, \dots, Y_i^0) = c_i(r, y_1, \dots, y_{i-1}, Y_i^0)$. Once the prefix is fixed, the next pair of databases x_i^0 and x_i^1 , the mechanism \mathcal{M}_i , and parameter w_i output by A are also determined (in both Experiment 0 and 1). Thus Y_i^0 is distributed according to $\mathcal{M}_i(w_i, x_i^0)$. Moreover for any value y_i , we have

$$c_i(r, y_1, \dots, y_{i-1}, y_i) = \ln \left(\frac{\Pr[\mathcal{M}_i(w_i, x_i^0) = y_i]}{\Pr[\mathcal{M}_i(w_i, x_i^1) = y_i]} \right).$$

By ε -differential privacy this is bounded by ε . We can also reason as follows:

$$\begin{aligned} |c_i(r, y_1, \dots, y_{i-1}, y_i)| \\ \leq \max\{D_\infty(\mathcal{M}_i(w_i, x_i^0) \| \mathcal{M}_i(w_i, x_i^1)), \\ D_\infty(\mathcal{M}_i(w_i, x_i^1) \| \mathcal{M}_i(w_i, x_i^0))\} \\ = \varepsilon. \end{aligned}$$

By Lemma 3.18, we have:

$$\begin{aligned} \mathbb{E}[c_i(R^0, Y_1^0, \dots, Y_i^0) | R^0 = r, Y_1^0 = y_1, \dots, Y_{i-1}^0 = y_{i-1}] \\ = D(\mathcal{M}_i(w_i, x_i^0) \| \mathcal{M}_i(w_i, x_i^1)) \\ \leq \varepsilon(e^\varepsilon - 1). \end{aligned}$$

Thus we can apply Azuma's Inequality to the random variables $C_i = c_i(R^0, Y_1^0, \dots, Y_i^0)$ with $\alpha = \varepsilon$, $\beta = \varepsilon \cdot \varepsilon_0$, and $z = \sqrt{2 \ln(1/\delta)}$, to deduce that

$$\Pr[V^0 \in B] = \Pr\left[\sum_i C_i > \varepsilon'\right] < e^{-z^2/2} = \delta,$$

as desired.

To extend the proof to composition of (ε, δ) -differentially private mechanisms, for $\delta > 0$, we use the characterization of approximate max-divergence from Lemma 3.17 (Part 2) to reduce the analysis to the same situation as in the case of $(\varepsilon, 0)$ -indistinguishable sequences. Specifically, using Lemma 3.17, Part 2 for each of the differentially private mechanisms selected by the adversary A and the triangle inequality for statistical distance, it follows that that V^0 is $k\delta$ -close to a random variable $W = (R, Z_1, \dots, Z_k)$ such that for every prefix r, y_1, \dots, y_{i-1} , if we condition on $R = R^1 = r, Z_1 = Y_1^1 = y_1, \dots, Z_{i-1} = Y_{i-1}^1 = y_{i-1}$, then it holds that $D_\infty(Z_i \| Y_i^1) \leq \varepsilon$ and $D_\infty(Y_i^1 \| Z_i) \leq \varepsilon$.

This suffices to show that $D_\infty^{\delta'}(W \| V^1) \leq \varepsilon'$. Since V^0 is $k\delta$ -close to W , Lemma 3.17, Part 1 gives $D^{\delta' + k\delta}(V^0 \| W) \leq \varepsilon'$. \square

An immediate and useful corollary tells us a safe choice of ε for each of k mechanisms if we wish to ensure $(\varepsilon', k\delta + \delta')$ -differential privacy for a given ε', δ' .

Corollary 3.21. Given target privacy parameters $0 < \varepsilon' < 1$ and $\delta' > 0$, to ensure $(\varepsilon', k\delta + \delta')$ cumulative privacy loss over k mechanisms, it suffices that each mechanism is (ε, δ) -differentially private, where

$$\varepsilon = \frac{\varepsilon'}{2\sqrt{2k \ln(1/\delta')}}.$$

Proof. Theorem 3.20 tells us the composition will be $(\varepsilon^*, k\delta + \delta')$ for all δ' , where $\varepsilon^* = \sqrt{2k \ln(1/\delta')} \cdot \varepsilon + k\varepsilon^2$. When $\varepsilon' < 1$, we have that $\varepsilon^* \leq \varepsilon'$ as desired. \square

Note that the above corollary gives a rough guide for how to set ε to get desired privacy parameters under composition. When one cares about optimizing constants (which one does when dealing with actual implementations), ε can be set more tightly by appealing directly to the composition theorem.

Example 3.7. Suppose, over the course of his lifetime, Bob is a member of $k = 10,000$ $(\varepsilon_0, 0)$ -differentially private databases. Assuming no coordination among these databases — the administrator of any given database may not even be aware of the existence of the other databases — what should be the value of ε_0 so that, over the course of his lifetime, Bob's cumulative privacy loss is bounded by $\varepsilon = 1$ with probability at least $1 - e^{-32}$? Theorem 3.20 says that, taking $\delta' = e^{-32}$ it suffices to have $\varepsilon_0 \leq 1/801$. This turns out to be essentially optimal against an arbitrary adversary, assuming no coordination among distinct differentially private databases.

So how many queries can we answer with non-trivial accuracy? On a database of size n let us say the accuracy is non-trivial if the error is of order $o(n)$. Theorem 3.20 says that for fixed values of ε and δ , it is possible to answer close to n^2 counting queries with non-trivial accuracy. Similarly, one can answer close to n queries while still having noise $o(\sqrt{n})$ — that is, noise less than the sampling error. We will see that it is possible to dramatically improve on these results, handling, in some cases, even an exponential number of queries with noise only slightly larger than \sqrt{n} , by coordinating the noise added to the individual responses. It turns out that such coordination is essential: without

coordination the bound in the advanced composition theorem is almost tight.

3.5.3 Laplace versus Gauss

An alternative to adding Laplacian noise is to add Gaussian noise. In this case, rather than scaling the noise to the ℓ_1 sensitivity Δf , we instead scale to the ℓ_2 sensitivity:

Definition 3.8 (ℓ_2 -sensitivity). The ℓ_2 -sensitivity of a function $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^k$ is:

$$\Delta_2(f) = \max_{\substack{x, y \in \mathbb{N}^{|\mathcal{X}|} \\ \|x - y\|_1=1}} \|f(x) - f(y)\|_2.$$

The *Gaussian Mechanism* with parameter b adds zero-mean Gaussian noise with variance b in each of the k coordinates. The following theorem is proved in Appendix A.

Theorem 3.22. Let $\varepsilon \in (0, 1)$ be arbitrary. For $c^2 > 2\ln(1.25/\delta)$, the Gaussian Mechanism with parameter $\sigma \geq c\Delta_2(f)/\varepsilon$ is (ε, δ) -differentially private.

Among the advantages to Gaussian noise is that the noise added for privacy is of the same type as other sources of noise; moreover, the sum of two Gaussians is a Gaussian, so the effects of the privacy mechanism on the statistical analysis may be easier to understand and correct for.

The two mechanisms yield the same cumulative loss under composition, so even though the privacy guarantee is weaker for each individual computation, the cumulative effects over many computations are comparable. Also, if δ is sufficiently (e.g., subpolynomially) small, in practice we will never experience the weakness of the guarantee.

That said, there is a theoretical disadvantage to Gaussian noise, relative to what we experience with Laplace noise. Consider Report Noisy Max (with Laplace noise) in a case in which every candidate output has the same quality score on database x as on its neighbor y . Independent of the number of candidate outputs, the mechanism yields $(\varepsilon, 0)$ -differential privacy. If instead we use Gaussian noise and report the max, and if the number of candidates is large compared to $1/\delta$,

then we will exactly select for the events with large Gaussian noise — noise that occurs with probability less than δ . When we are this far out on the tail of the Gaussian we no longer have a guarantee that the observation is within an $e^{\pm\epsilon}$ factor as likely to occur on x as on y .

3.5.4 Remarks on composition

The ability to analyze cumulative privacy loss under composition gives us a handle on what a world of differentially private databases can offer. A few observations are in order.

Weak Quantification. Assume that the adversary always chooses x_i^0 to hold Bob’s data, and x_i^1 to be the same database but with Bob’s data deleted. Theorem 3.20, with appropriate choice of parameters, tells us that an adversary — including one that knows or even selects(!) the database pairs — has little advantage in determining the value of $b \in \{0, 1\}$. This is an inherently weak quantification. We can ensure that the adversary is unlikely to distinguish reality from any given alternative, but we cannot ensure this simultaneously for all alternatives. If there are one zillion databases but Bob is a member of only 10,000 of these, then we are not simultaneously protecting Bob’s *absence* from all zillion minus ten thousand. This is analogous to the quantification in the definition of (ϵ, δ) -differential privacy, where we fix in advance a pair of adjacent databases and argue that with high probability the output will be almost equally likely with these two databases.

Humans and Ghosts. Intuitively, an $(\epsilon, 0)$ -differentially private database with a small number of bits per record is less protective than a differentially private database with the same choice of ϵ that contains our entire medical histories. So in what sense is our principle privacy measure, ϵ , telling us the same thing about databases that differ radically in the complexity and sensitivity of the data they store? The answer lies in the composition theorems. Imagine a world inhabited by two types of beings: ghosts and humans. Both types of beings behave the same, interact with others in the same way, write, study, work, laugh, love, cry, reproduce, become ill, recover, and age in the same fashion. The only difference is that ghosts have no records in

databases, while humans do. The goal of the privacy adversary is to determine whether a given 50-year old, the “target,” is a ghost or a human. Indeed, the adversary is given all 50 years to do so. The adversary does not need to remain passive, for example, she can organize clinical trials and enroll patients of her choice, she can create humans to populate databases, effectively creating the worst-case (for privacy) databases, she can expose the target to chemicals at age 25 and again at 35, and so on. She can know everything about the target that could possibly be entered into any database. She can know which databases the target would be in, were the target human. The composition theorems tell us that the privacy guarantees of each database — regardless of the data type, complexity, and sensitivity — give comparable protection for the human/ghost bit.

3.6 The sparse vector technique

The Laplace mechanism can be used to answer adaptively chosen low sensitivity queries, and we know from our composition theorems that the privacy parameter degrades proportionally to the number of queries answered (or its square root). Unfortunately, it will often happen that we have a very large number of questions to answer — too many to yield a reasonable privacy guarantee using independent perturbation techniques, even with the advanced composition theorems of Section 3.5. In some situations however, we will only care to know the identity of the queries that lie above a certain threshold. In this case, we can hope to gain over the naïve analysis by discarding the numeric answer to queries that lie significantly below the threshold, and merely reporting that they do indeed lie below the threshold. (We will be able to get the numeric values of the above-threshold queries as well, at little additional cost, if we so choose). This is similar to what we did in the Report Noisy Max mechanism in section 3.3, and indeed iterating either that algorithm or the exponential mechanism would be an option for the non-interactive, or offline, case.

In this section, we show how to analyze a method for this in the online setting. The technique is simple — add noise and report only

whether the noisy value exceeds the threshold — and our emphasis is on the analysis, showing that privacy degrades only with the number of queries which actually lie above the threshold, rather than with the total number of queries. This can be a huge savings if we know that the set of queries that lie above the threshold is much smaller than the total number of queries — that is, if the answer vector is *sparse*.

In a little more detail, we will consider a sequence of events — one for each query — which occur if a query evaluated on the database exceeds a given (known, public) threshold. Our goal will be to release a bit vector indicating, for each event, whether or not it has occurred. As each query is presented, the mechanism will compute a noisy response, compare it to the (publicly known) threshold, and, if the threshold is exceeded, reveal this fact. For technical reasons in the proof of privacy (Theorem 3.24), the algorithm works with a noisy version \hat{T} of the threshold T . While T is public the noisy version \hat{T} is not.

Rather than incurring a privacy loss for each *possible* query, the analysis below will result in a privacy cost only for the query values that are near or above the threshold.

The Setting. Let m denote the total number of sensitivity 1 queries, which may be chosen adaptively. Without loss of generality, there is a single threshold T fixed in advance (alternatively, each query can have its own threshold, but the results are unchanged). We will be adding noise to query values and comparing the results to T . A *positive* outcome means that a noisy query value exceeds the threshold. We expect a small number c of noisy values to exceed the threshold, and we are releasing only the noisy values above the threshold. The algorithm will use c in its stopping condition.

We will first analyze the case in which the algorithm halts after $c = 1$ above-threshold query, and show that this algorithm is ϵ -differentially private no matter how long the *total* sequence of queries is. We will then analyze the case of $c > 1$ by using our composition theorems, and derive bounds both for $(\epsilon, 0)$ and (ϵ, δ) -differential privacy.

We first argue that `AboveThreshold`, the algorithm specialized to the case of only one above-threshold query, is private and accurate.

Algorithm 1 Input is a private database D , an adaptively chosen stream of sensitivity 1 queries f_1, \dots , and a threshold T . Output is a stream of responses a_1, \dots

```

AboveThreshold( $D, \{f_i\}, T, \epsilon$ )
  Let  $\hat{T} = T + \text{Lap}\left(\frac{2}{\epsilon}\right)$ .
  for Each query  $i$  do
    Let  $\nu_i = \text{Lap}\left(\frac{4}{\epsilon}\right)$ 
    if  $f_i(D) + \nu_i \geq \hat{T}$  then
      Output  $a_i = \top$ .
      Halt.
    else
      Output  $a_i = \perp$ .
    end if
  end for

```

Theorem 3.23. AboveThreshold is $(\epsilon, 0)$ -differentially private.

Proof. Fix any two neighboring databases D and D' . Let A denote the random variable representing the output of **AboveThreshold**($D, \{f_i\}, T, \epsilon$) and let A' denote the random variable representing the output of **AboveThreshold**($D', \{f_i\}, T, \epsilon$). The output of the algorithm is some realization of these random variables, $a \in \{\top, \perp\}^k$ and has the form that for all $i < k$, $a_i = \perp$ and $a_k = \top$. There are two types of random variables internal to the algorithm: the noisy threshold \hat{T} and the perturbations to each of the k queries, $\{\nu_i\}_{i=1}^k$. For the following analysis, we will fix the (arbitrary) values of ν_1, \dots, ν_{k-1} and take probabilities over the randomness of ν_k and \hat{T} . Define the following quantity representing the maximum noisy value of any query f_1, \dots, f_{k-1} evaluated on D :

$$g(D) = \max_{i < k} (f_i(D) + \nu_i)$$

In the following, we will abuse notation and write $\Pr[\hat{T} = t]$ as shorthand for the pdf of \hat{T} evaluated at t (similarly for ν_k), and write $\mathbf{1}[x]$ to denote the indicator function of event x . Note that fixing the values

of ν_1, \dots, ν_{k-1} (which makes $g(D)$ a deterministic quantity), we have:

$$\begin{aligned}
\Pr_{\hat{T}, \nu_k} [A = a] &= \Pr_{\hat{T}, \nu_k} [\hat{T} > g(D) \text{ and } f_k(D) + \nu_k \geq \hat{T}] \\
&= \Pr_{\hat{T}, \nu_k} [\hat{T} \in (g(D), f_k(D) + \nu_k)] \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Pr[\nu_k = v] \\
&\quad \cdot \Pr[\hat{T} = t] \mathbf{1}[t \in (g(D), f_k(D) + v)] dv dt \\
&\doteq *
\end{aligned}$$

We now make a change of variables. Define:

$$\begin{aligned}
\hat{v} &= v + g(D) - g(D') + f_k(D') - f_k(D) \\
\hat{t} &= t + g(D) - g(D')
\end{aligned}$$

and note that for any D, D' , $|\hat{v} - v| \leq 2$ and $|\hat{t} - t| \leq 1$. This follows because each query $f_i(D)$ is 1-sensitive, and hence the quantity $g(D)$ is 1-sensitive as well. Applying this change of variables, we have:

$$\begin{aligned}
* &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Pr[\nu_k = \hat{v}] \cdot \Pr[\hat{T} = \hat{t}] \mathbf{1}[(t + g(D) - g(D')) \\
&\quad \in (g(D), f_k(D') + v + g(D) - g(D'))] dv dt \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Pr[\nu_k = \hat{v}] \cdot \Pr[\hat{T} = \hat{t}] \mathbf{1}[(t \in (g(D'), f_k(D') + v))] dv dt \\
&\leq \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp(\epsilon/2) \Pr[\nu_k = v] \\
&\quad \cdot \exp(\epsilon/2) \Pr[\hat{T} = t] \mathbf{1}[(t \in (g(D'), f_k(D') + v))] dv dt \\
&= \exp(\epsilon) \Pr_{\hat{T}, \nu_k} [\hat{T} > g(D') \text{ and } f_k(D') + \nu_k \geq \hat{T}] \\
&= \exp(\epsilon) \Pr_{\hat{T}, \nu_k} [A' = a]
\end{aligned}$$

where the inequality comes from our bounds on $|\hat{v} - v|$ and $|\hat{t} - t|$ and the form of the pdf of the Laplace distribution. \square

Definition 3.9 (Accuracy). We will say that an algorithm which outputs a stream of answers $a_1, \dots, \in \{\top, \perp\}^*$ in response to a stream of k

queries f_1, \dots, f_k is (α, β) -accurate with respect to a threshold T if except with probability at most β , the algorithm does not halt before f_k , and for all $a_i = \top$:

$$f_i(D) \geq T - \alpha$$

and for all $a_i = \perp$:

$$f_i(D) \leq T + \alpha.$$

What can go wrong in Algorithm 1? The noisy threshold \hat{T} can be very far from T , say, $|\hat{T} - T| > \alpha$. In addition a small count $f_i(D) < T - \alpha$ can have so much noise added to it that it is reported as above threshold (even when the threshold is close to correct), and a large count $f_i(D) > T + \alpha$ can be reported as below threshold. All of these happen with probability exponentially small in α . In summary, we can have a problem with the choice of the noisy threshold or we can have a problem with one or more of the individual noise values ν_i . Of course, we could have both kinds of errors, so in the analysis below we allocate $\alpha/2$ to each type.

Theorem 3.24. For any sequence of k queries f_1, \dots, f_k such that $|\{i < k : f_i(D) \geq T - \alpha\}| = 0$ (i.e. the only query close to being above threshold is possibly the last one), $\text{AboveThreshold}(D, \{f_i\}, T, \epsilon)$ is (α, β) accurate for:

$$\alpha = \frac{8(\log k + \log(2/\beta))}{\epsilon}.$$

Proof. Observe that the theorem will be proved if we can show that except with probability at most β :

$$\max_{i \in [k]} |\nu_i| + |T - \hat{T}| \leq \alpha$$

If this is the case, then for any $a_i = \top$, we have:

$$f_i(D) + \nu_i \geq \hat{T} \geq T - |T - \hat{T}|$$

or in other words:

$$f_i(D) \geq T - |T - \hat{T}| - |\nu_i| \geq T - \alpha$$

Similarly, for any $a_i = \perp$ we have:

$$f_i(D) < \hat{T} \leq T + |T - \hat{T}| + |\nu_i| \leq T + \alpha$$

We will also have that for any $i < k$: $f_i(D) < T - \alpha < T - |\nu_i| - |T - \hat{T}|$, and so: $f_i(D) + \nu_i \leq \hat{T}$, meaning $a_i = \perp$. Therefore the algorithm does not halt before k queries are answered.

We now complete the proof.

Recall that if $Y \sim \text{Lap}(b)$, then: $\Pr[|Y| \geq t \cdot b] = \exp(-t)$. Therefore we have:

$$\Pr[|T - \hat{T}| \geq \frac{\alpha}{2}] = \exp\left(-\frac{\epsilon\alpha}{4}\right)$$

Setting this quantity to be at most $\beta/2$, we find that we require $\alpha \geq \frac{4 \log(2/\beta)}{\epsilon}$

Similarly, by a union bound, we have:

$$\Pr[\max_{i \in [k]} |\nu_i| \geq \alpha/2] \leq k \cdot \exp\left(-\frac{\epsilon\alpha}{8}\right)$$

Setting this quantity to be at most $\beta/2$, we find that we require $\alpha \geq \frac{8(\log(2/\beta) + \log k)}{\epsilon}$. These two claims combine to prove the theorem. \square

We now show how to handle multiple “above threshold” queries using composition.

The Sparse algorithm can be thought of as follows: As queries come in, it makes repeated calls to AboveThreshold. Each time an above threshold query is reported, the algorithm simply restarts the remaining stream of queries on a new instantiation of AboveThreshold. It halts after it has restarted AboveThreshold c times (i.e. after c above threshold queries have appeared). Each instantiation of AboveThreshold is $(\epsilon, 0)$ -private, and so the composition theorems apply.

Theorem 3.25. Sparse is (ϵ, δ) -differentially private.

Proof. We observe that Sparse is exactly equivalent to the following procedure: We run $\text{AboveThreshold}(D, \{f_i\}, T, \epsilon')$ on our stream of queries $\{f_i\}$ setting

$$\epsilon' = \begin{cases} \frac{\epsilon}{c}, & \text{If } \delta = 0; \\ \frac{\epsilon}{\sqrt{8c \ln \frac{1}{\delta}}}, & \text{Otherwise.} \end{cases}$$

Algorithm 2 Input is a private database D , an adaptively chosen stream of sensitivity 1 queries f_1, \dots , a threshold T , and a cutoff point c . Output is a stream of answers a_1, \dots

```

Sparse( $D, \{f_i\}, T, c, \epsilon, \delta$ )
  If  $\delta = 0$  Let  $\sigma = \frac{2c}{\epsilon}$ . Else Let  $\sigma = \frac{\sqrt{32c \ln \frac{1}{\delta}}}{\epsilon}$ 
  Let  $\hat{T}_0 = T + \text{Lap}(\sigma)$ 
  Let count = 0
  for Each query  $i$  do
    Let  $\nu_i = \text{Lap}(2\sigma)$ 
    if  $f_i(D) + \nu_i \geq \hat{T}_{\text{count}}$  then
      Output  $a_i = \top$ .
    Let count = count + 1.
    Let  $\hat{T}_{\text{count}} = T + \text{Lap}(\sigma)$ 
  else
    Output  $a_i = \perp$ .
  end if
  if count  $\geq c$  then
    Halt.
  end if
end for

```

using the answers supplied by `AboveThreshold`. When `AboveThreshold` halts (after 1 above threshold query), we simply restart `Sparse($D, \{f_i\}, T, \epsilon'$)` on the remaining stream, and continue in this manner until we have restarted `AboveThreshold` c times. After the c 'th restart of `AboveThreshold` halts, we halt as well. We have already proven that `AboveThreshold($D, \{f_i\}, T, \epsilon'$)` is $(\epsilon', 0)$ differentially private. Finally, by the advanced composition theorem (Theorem 3.20), c applications of an $\epsilon' = \frac{\epsilon}{\sqrt{8c \ln \frac{1}{\delta}}}$ -differentially private algorithm is (ϵ, δ) -differentially private, and c applications of an $\epsilon' = \epsilon/c$ differentially private algorithm is $(\epsilon, 0)$ -private as desired. \square

It remains to prove accuracy for `Sparse`, by again observing that `Sparse` consists only of c calls to `AboveThreshold`. We note that if each

of these calls to AboveThreshold is $(\alpha, \beta/c)$ -accurate, then Sparse will be (α, β) -accurate.

Theorem 3.26. For any sequence of k queries f_1, \dots, f_k such that $L(T) \equiv |\{i : f_i(D) \geq T - \alpha\}| \leq c$, if $\delta > 0$, Sparse is (α, β) accurate for:

$$\alpha = \frac{(\ln k + \ln \frac{2c}{\beta}) \sqrt{512c \ln \frac{1}{\delta}}}{\epsilon}.$$

If $\delta = 0$, Sparse is (α, β) accurate for:

$$\alpha = \frac{8c(\ln k + \ln(2c/\beta))}{\epsilon}$$

Proof. We simply apply Theorem 3.24 setting β to be β/c , and ϵ to be $\frac{\epsilon}{\sqrt{8c \ln \frac{1}{\delta}}}$ and ϵ/c , depending on whether $\delta > 0$ or $\delta = 0$, respectively. \square

Finally, we give a version of Sparse that actually outputs the numeric values of the above threshold queries, which we can do with only a constant factor loss in accuracy. We call this algorithm NumericSparse, and it is simply a composition of Sparse with the Laplace mechanism. Rather than outputting a vector $a \in \{\top, \perp\}^*$, it outputs a vector $a \in (\mathbb{R} \cup \{\perp\})^*$.

We observe that NumericSparse is private:

Theorem 3.27. NumericSparse is (ϵ, δ) -differentially private.

Proof. Observe that if $\delta = 0$, $\text{NumericSparse}(D, \{f_i\}, T, c, \epsilon, 0)$ is simply the adaptive composition of $\text{Sparse}(D, \{f_i\}, T, c, \frac{8}{9}\epsilon, 0)$, together with the Laplace mechanism with privacy parameters $(\epsilon', \delta) = (\frac{1}{9}\epsilon, 0)$. If $\delta > 0$, then $\text{NumericSparse}(D, \{f_i\}, T, c, \epsilon, 0)$ is the composition of $\text{Sparse}(D, \{f_i\}, T, c, \frac{\sqrt{512}}{\sqrt{512}+1}\epsilon, \delta/2)$ together with the Laplace mechanism with privacy parameters $(\epsilon', \delta) = (\frac{1}{\sqrt{512}+1}\epsilon, \delta/2)$. Hence the privacy of NumericSparse follows from simple composition. \square

To discuss accuracy, we must define what we mean by the accuracy of a mechanism that outputs a stream $a \in (\mathbb{R} \cup \{\perp\})^*$ in response to a sequence of numeric valued queries:

Algorithm 3 Input is a private database D , an adaptively chosen stream of sensitivity 1 queries f_1, \dots , a threshold T , and a cutoff point c . Output is a stream of answers a_1, \dots

```

NumericSparse( $D, \{f_i\}, T, c, \epsilon, \delta$ )
  If  $\delta = 0$  Let  $\epsilon_1 \leftarrow \frac{8}{9}\epsilon$ ,  $\epsilon_2 \leftarrow \frac{2}{9}\epsilon$ . Else Let  $\epsilon_1 = \frac{\sqrt{512}}{\sqrt{512}+1}\epsilon$ ,  $\epsilon_2 = \frac{2}{\sqrt{512}+1}$ 
  If  $\delta = 0$  Let  $\sigma(\epsilon) = \frac{2c}{\epsilon}$ . Else Let  $\sigma(\epsilon) = \frac{\sqrt{32c \ln \frac{2}{\delta}}}{\epsilon}$ 
  Let  $\hat{T}_0 = T + \text{Lap}(\sigma(\epsilon_1))$ 
  Let count = 0
  for Each query  $i$  do
    Let  $\nu_i = \text{Lap}(2\sigma(\epsilon_1))$ 
    if  $f_i(D) + \nu_i \geq \hat{T}_{\text{count}}$  then
      Let  $v_i \leftarrow \text{Lap}(\sigma(\epsilon_2))$ 
      Output  $a_i = f_i(D) + v_i$ .
      Let count = count + 1.
      Let  $\hat{T}_{\text{count}} = T + \text{Lap}(\sigma(\epsilon_1))$ 
    else
      Output  $a_i = \perp$ .
    end if
    if count  $\geq c$  then
      Halt.
    end if
  end for

```

Definition 3.10 (Numeric Accuracy). We will say that an algorithm which outputs a stream of answers $a_1, \dots, \in (\mathbb{R} \cup \{\perp\})^*$ in response to a stream of k queries f_1, \dots, f_k is (α, β) -accurate with respect to a threshold T if except with probability at most β , the algorithm does not halt before f_k , and for all $a_i \in \mathbb{R}$:

$$|f_i(D) - a_i| \leq \alpha$$

and for all $a_i = \perp$:

$$f_i(D) \leq T + \alpha.$$

Theorem 3.28. For any sequence of k queries f_1, \dots, f_k such that $L(T) \equiv |\{i : f_i(D) \geq T - \alpha\}| \leq c$, if $\delta > 0$, NumericSparse is (α, β)

accurate for:

$$\alpha = \frac{(\ln k + \ln \frac{4c}{\beta}) \sqrt{c \ln \frac{2}{\delta}} (\sqrt{512} + 1)}{\epsilon}.$$

If $\delta = 0$, Sparse is (α, β) accurate for:

$$\alpha = \frac{9c(\ln k + \ln(4c/\beta))}{\epsilon}$$

Proof. Accuracy requires two conditions: first, that for all $a_i = \perp$: $f_i(D) \leq T + \alpha$. This holds with probability $1 - \beta/2$ by the accuracy theorem for Sparse. Next, for all $a_i \in \mathbb{R}$, it requires $|f_i(D) - a_i| \leq \alpha$. This holds for with probability $1 - \beta/2$ by the accuracy of the Laplace mechanism. \square

What did we show in the end? If we are given a sequence of queries together with a guarantee that only at most c of them have answers above $T - \alpha$, we can answer those queries that are above a given threshold T , up to error α . This accuracy is equal, up to constants and a factor of $\log k$, to the accuracy we would get, given the same privacy guarantee, if we knew the identities of these large above-threshold queries ahead of time, and answered them with the Laplace mechanism. That is, the sparse vector technique allowed us to fish out the identities of these large queries almost “for free”, paying only logarithmically for the irrelevant queries. This is the same guarantee that we could have gotten by trying to find the large queries with the exponential mechanism and then answering them with the Laplace mechanism. This algorithm, however, is trivial to run, and crucially, allows us to choose our queries adaptively.

3.7 Bibliographic notes

Randomized Response is due to Warner [84] (predating differential privacy by four decades!). The Laplace mechanism is due to Dwork et al. [23]. The exponential mechanism was invented by McSherry and Talwar [60]. Theorem 3.16 (simple composition) was claimed in [21]; the proof appearing in Appendix B is due to Dwork and Lei [22];

McSherry and Mironov obtained a similar proof. The material in Sections 3.5.1 and 3.5.2 is taken almost verbatim from Dwork et al. [32]. Prior to [32] composition was modeled informally, much as we did for the simple composition bounds. For specific mechanisms applied on a single database, there are “evolution of confidence” arguments due to Dinur, Dwork, and Nissim [18, 31], (which pre-date the definition of differential privacy) showing that the privacy parameter in k -fold composition need only deteriorate like \sqrt{k} if we are willing to tolerate a (negligible) loss in δ (for $k < 1/\varepsilon^2$). Theorem 3.20 generalizes those arguments to arbitrary differentially private mechanisms,

The claim that without coordination in the noise the bounds in the composition theorems are almost tight is due to Dwork, Naor, and Vadhan [29]. The sparse vector technique is an abstraction of a technique that was introduced, by Dwork, Naor, Reingold, Rothblum, and Vadhan [28] (indicator vectors in the proof of Lemma 4.4). It has subsequently found wide use (e.g. by Roth and Roughgarden [74], Dwork, Naor, Pitassi, and Rothblum [26], and Hardt and Rothblum [44]). In our presentation of the technique, the proof of Theorem 3.23 is due to Salil Vadhan.

4

Releasing Linear Queries with Correlated Error

One of the most fundamental primitives in private data analysis is the ability to answer numeric valued queries on a dataset. In the last section, we began to see tools that would allow us to do this by adding independently drawn noise to the query answers. In this section, we continue this study, and see that by instead adding carefully correlated noise, we can gain the ability to privately answer vastly more queries to high accuracy. Here, we see two specific mechanisms for solving this problem, which we will generalize in the next section.

In this section, we consider algorithms for solving the *query release* problem with better accuracy than we would get by simply using compositions of the Laplace mechanism. The improvements are possible because the set of queries is handled as a whole — even in the online setting! — permitting the noise on individual queries to be correlated. To immediately see that something along these lines might be possible, consider the pair of queries in the differencing attack described in Section 1: “How many people in the database have the sickle cell trait?” and “How many people, not named X, in the database have the sickle cell trait?” Suppose a mechanism answers the first question using the Laplace mechanism and then, when the second question is posed,

responds “You already know the approximate answer, because you just asked me almost the exact same question.” This coordinated response to the pair of questions incurs no more privacy loss than either question would do taken in isolation, so a (small) privacy savings has been achieved.

The query release problem is quite natural: given a class of queries \mathcal{Q} over the database, we wish to release some answer a_i for each query $f_i \in \mathcal{Q}$ such that the error $\max_i |a_i - f_i(x)|$ is as low as possible, while still preserving differential privacy.¹ Recall that for any family of low sensitivity queries, we can apply the Laplace mechanism, which adds fresh, independent, noise to the answer to each query. Unfortunately, at a fixed privacy level, for $(\epsilon, 0)$ -privacy guarantees, the magnitude of the noise that we must add with the Laplace mechanism scales with $|\mathcal{Q}|$ because this is the rate at which the sensitivity of the combined queries may grow. Similarly, for (ϵ, δ) -privacy guarantees, the noise scales with $\sqrt{|\mathcal{Q}| \ln(1/\delta)}$. For example, suppose that our class of queries \mathcal{Q} consists only of many copies of the same query: $f_i = f^*$ for all i . If we use the Laplace mechanism to release the answers, it will add independent noise, and so each a_i will be an independent random variable with mean $f^*(x)$. Clearly, in this regime, the noise rate must grow with $|\mathcal{Q}|$ since otherwise the average of the a_i will converge to the true value $f^*(x)$, which would be a privacy violation. However, in this case, because $f_i = f^*$ for all i , it would make more sense to approximate f^* only once with $a^* \approx f^*(x)$ and release $a_i = a^*$ for all i . In this case, the noise rate would not have to scale with $|\mathcal{Q}|$ at all. In this section, we aim to design algorithms that are much more accurate than the Laplace mechanism (with error that scales with $\log |\mathcal{Q}|$) by adding non-independent noise as a function of the set of queries.

Recall that our universe is $\mathcal{X} = \{\chi_1, \chi_2, \dots, \chi_{|\mathcal{X}|}\}$ and that databases are represented by histograms in $\mathbb{N}^{|\mathcal{X}|}$. A *linear query* is simply a counting query, but generalized to take values in the interval $[0, 1]$ rather than only boolean values. Specifically, a linear query f takes the

¹It is the privacy constraint that makes the problem interesting. Without this constraint, the query release problem is trivially and optimally solved by just outputting exact answers for every query.

form $f : \mathcal{X} \rightarrow [0, 1]$, and applied to a database x returns either the *sum* or *average* value of the query on the database (we will think of both, depending on which is more convenient for the analysis). When we think of linear queries as returning *average* values, we will refer to them as *normalized* linear queries, and say that they take value:

$$f(x) = \frac{1}{\|x\|_1} \sum_{i=1}^{|\mathcal{X}|} x_i \cdot f(\chi_i).$$

When we think of linear queries as returning *sum* values we will refer to them as *un-normalized* linear queries, and say that they take value:

$$f(x) = \sum_{i=1}^{|\mathcal{X}|} x_i \cdot f(\chi_i).$$

Whenever we state a bound, it should be clear from context whether we are speaking of normalized or un-normalized queries, because they take values in very different ranges. Note that normalized linear queries take values in $[0, 1]$, whereas un-normalized queries take values in $[0, \|x\|_1]$.

Note that with this definition linear queries have sensitivity $\Delta f \leq 1$. Later sections will discuss arbitrary low-sensitivity queries.

We will present two techniques, one each for the offline and online cases. Surprisingly, and wonderfully, the offline technique is an immediate application of the exponential mechanism using well-known sampling bounds from learning theory! The algorithm will simply be to apply the exponential mechanism with range equal to the set of all *small* databases y and quality function $u(x, y)$ equal to minus the maximum approximation error incurred by querying y to obtain an approximation for $f(x)$:

$$u(x, y) = -\max_{f \in \mathcal{Q}} |f(x) - f(y)|. \quad (4.1)$$

Sampling bounds (see Lemma 4.3 below) tell us that a random subset of $\ln |\mathcal{Q}|/\alpha^2$ elements of x will very likely give us a good approximation for all $f(x)$ (specifically, with additive error bounded by α), so we know it is sufficient to restrict the set of possible outputs to small databases. We don't actually care that the potential output databases are small, only that they are not too numerous: their number plays a role in the proof of

utility, which is an immediate application of the utility theorem for the exponential mechanism (Theorem 3.11). More specifically, if the total number of potential outputs is not too numerous then, in particular, the total number of low-utility outputs is not too numerous, and therefore the ratio of bad outputs to good outputs (there is at least one) is not too large.

The online mechanism, which, despite not knowing the entire set of queries in advance, will achieve the same accuracy as the offline mechanism, and will be a direct application of the sparse vector technique. As a result, privacy will be immediate, but utility will require a proof. The key will be to argue that, even for a very large set of counting queries, few queries are “significant”; that is, significant queries will be sparse. As with the sparse vector algorithms, we can scale noise according to the number of significant queries, with little dependence on the total number of queries.

Before we go on and present the mechanisms, we will give just one example of a useful class of linear queries.

Example 4.1. Suppose that elements of the database are represented by d *boolean* features. For example, the first feature may represent whether or not the individual is male or female, the second feature may represent whether or not they are a college graduate, the third feature may represent whether or not they are US citizens, etc. That is, our data universe is $\mathcal{X} = \{0, 1\}^d$. Given a subset of these attributes $S \subseteq \{1, \dots, d\}$, we might like to know how many people in the dataset have these attributes. (e.g., “What fraction of the dataset consists of male college graduates with a family history of lung cancer?”). This naturally defines a query called a *monotone conjunction query*, parameterized by a subset of attributes S and defined as $f_S(z) = \prod_{i \in S} z_i$, for $z \in \mathcal{X}$. The class of *all* such queries is simply $\mathcal{Q} = \{f_S : S \subseteq \{1, \dots, d\}\}$, and has size $|\mathcal{Q}| = 2^d$. A collection of answers to conjunctions is sometimes called a *contingency* or *marginal* table, and is a common method of releasing statistical information about a dataset. Often times, we may not be interested in the answers to *all* conjunctions, but rather just those that ask about subsets of features S of size $|S| = k$ for some fixed k . This class of queries $\mathcal{Q}_k = \{f_S : S \subseteq \{1, \dots, d\}, |S| = k\}$ has size $\binom{d}{k}$.

This large and useful class of queries is just one example of the sorts of queries that can be accurately answered by the algorithms given in this section. (Note that if we wish to also allow (non-monotone) conjunctions which ask about *negated* attributes, we can do that as well — simply double the feature space from d to $2d$, and set $z_{d+i} = 1 - z_i$ for all $i \in \{1, \dots, d\}$.)

4.1 An offline algorithm: SmallDB

In this section, we give an algorithm based on the idea of sampling a small database using the exponential mechanism. What we will show is that, for counting queries, it suffices to consider databases that are small: their size will only be a function of the query class, and our desired approximation accuracy α , and crucially *not* on $\|x\|_1$, the size of the private database. This is important because it will allow us to simultaneously guarantee, for all sufficiently large databases, that there is at least *one* database in the range of the exponential mechanism that well approximates x on queries in \mathcal{Q} , and that there are not *too many* databases in the range to dissipate the probability mass placed on this “good” database.

Algorithm 4 The Small Database Mechanism

SmallDB($x, \mathcal{Q}, \varepsilon, \alpha$)

Let $\mathcal{R} \leftarrow \{y \in \mathbb{N}^{|\mathcal{X}|} : \|y\|_1 = \frac{\log |\mathcal{Q}|}{\alpha^2}\}$

Let $u : \mathbb{N}^{|\mathcal{X}|} \times \mathcal{R} \rightarrow \mathbb{R}$ be defined to be:

$$u(x, y) = -\max_{f \in \mathcal{Q}} |f(x) - f(y)|$$

Sample And Output $y \in \mathcal{R}$ with the exponential mechanism
 $\mathcal{M}_E(x, u, \mathcal{R})$

We first observe that the Small Database mechanism preserves ε -differential privacy.

Proposition 4.1. The Small Database mechanism is $(\varepsilon, 0)$ differentially private.

Proof. The Small Database mechanism is simply an instantiation of the exponential mechanism. Therefore, privacy follows from Theorem 3.10. \square

We may similarly call on our analysis of the exponential mechanism to understand the utility guarantees of the Small Database mechanism. But first, we must justify our choice of range $\mathcal{R} = \{y \in \mathbb{N}^{|\mathcal{X}|} : \|y\|_1 = \frac{\log |\mathcal{Q}|}{\alpha^2}\}$, the set of all databases of size $\log |\mathcal{Q}|/\alpha^2$.

Theorem 4.2. For any finite class of linear queries \mathcal{Q} , if $\mathcal{R} = \{y \in \mathbb{N}^{|\mathcal{X}|} : \|y\|_1 = \frac{\log |\mathcal{Q}|}{\alpha^2}\}$ then for all $x \in \mathbb{N}^{|\mathcal{X}|}$, there exists a $y \in \mathcal{R}$ such that:

$$\max_{f \in \mathcal{Q}} |f(x) - f(y)| \leq \alpha$$

In other words, we will show that for any collection of linear queries \mathcal{Q} and for any database x , there is a “small” database y of size $\|y\|_1 = \frac{\log |\mathcal{Q}|}{\alpha^2}$ that approximately encodes the answers to every query in \mathcal{Q} , up to error α .

Lemma 4.3 (Sampling Bounds). For any $x \in \mathbb{N}^{|\mathcal{X}|}$ and for any collection of linear queries \mathcal{Q} , there exists a database y of size

$$\|y\|_1 = \frac{\log |\mathcal{Q}|}{\alpha^2}$$

such that:

$$\max_{f \in \mathcal{Q}} |f(x) - f(y)| \leq \alpha$$

Proof. Let $m = \frac{\log |\mathcal{Q}|}{\alpha^2}$. We will construct a database y by taking m uniformly random samples from the elements of x . Specifically, for $i \in \{1, \dots, m\}$, let X_i be a random variable taking value $\chi_j \in \mathcal{X}$ with probability $x_j/\|x\|_1$, and let y be the database containing elements X_1, \dots, X_m . Now fix any $f \in \mathcal{Q}$ and consider the quantity $f(y)$. We have:

$$f(y) = \frac{1}{\|y\|_1} \sum_{i=1}^{|\mathcal{X}|} y_i \cdot f(\chi_i) = \frac{1}{m} \sum_{i=1}^m f(X_i).$$

We note that each term $f(X_i)$ of the sum is a bounded random variable taking values $0 \leq f(X_i) \leq 1$ with expectation

$$\mathbb{E}[f(X_i)] = \sum_{j=1}^{|\mathcal{X}|} \frac{x_j}{\|x\|_1} f(\chi_j) = f(x),$$

and that the expectation of $f(y)$ is:

$$\mathbb{E}[f(y)] = \frac{1}{m} \sum_{i=1}^m \mathbb{E}[f(X_i)] = f(x).$$

Therefore, we can apply the Chernoff bound stated in Theorem 3.1 which gives:

$$\Pr [|f(y) - f(x)| > \alpha] \leq 2e^{-2m\alpha^2}.$$

Taking a union bound over all of the linear queries $f \in \mathcal{Q}$, we get:

$$\Pr \left[\max_{f \in \mathcal{Q}} |f(y) - f(x)| > \alpha \right] \leq 2|\mathcal{Q}|e^{-2m\alpha^2}.$$

Plugging in $m = \frac{\log |\mathcal{Q}|}{\alpha^2}$ makes the right hand side smaller than 1 (so long as $|\mathcal{Q}| > 2$), proving that there exists a database of size m satisfying the stated bound, which completes the proof of the lemma. \square

The proof of Theorem 4.2 simply follows from the observation that \mathcal{R} contains *all* databases of size $\frac{\log |\mathcal{Q}|}{\alpha^2}$.

Proposition 4.4. Let \mathcal{Q} be any class of linear queries. Let y be the database output by $\text{SmallDB}(x, \mathcal{Q}, \varepsilon, \alpha)$. Then with probability $1 - \beta$:

$$\max_{f \in \mathcal{Q}} |f(x) - f(y)| \leq \alpha + \frac{2 \left(\frac{\log |\mathcal{X}| \log |\mathcal{Q}|}{\alpha^2} + \log \left(\frac{1}{\beta} \right) \right)}{\varepsilon \|x\|_1}.$$

Proof. Applying the utility bounds for the exponential mechanism (Theorem 3.11) with $\Delta u = \frac{1}{\|x\|_1}$ and $\text{OPT}_q(D) \leq \alpha$ (which follows from Theorem 4.2), we find:

$$\Pr \left[\max_{f \in \mathcal{Q}} |f(x) - f(y)| \geq \alpha + \frac{2}{\varepsilon \|x\|_1} (\log(|\mathcal{R}|) + t) \right] \leq e^{-t}.$$

We complete the proof by (1) noting that \mathcal{R} , which is the set of all databases of size at most $\log |\mathcal{Q}|/\alpha^2$, satisfies $|\mathcal{R}| \leq |\mathcal{X}|^{\log |\mathcal{Q}|/\alpha^2}$ and (2) by setting $t = \log \left(\frac{1}{\beta} \right)$. \square

Finally, we may now state the utility theorem for SmallDB.

Theorem 4.5. By the appropriate choice of α , letting y be the database output by $\text{SmallDB}(x, \mathcal{Q}, \varepsilon, \frac{\alpha}{2})$, we can ensure that with probability $1 - \beta$:

$$\max_{f \in \mathcal{Q}} |f(x) - f(y)| \leq \left(\frac{16 \log |\mathcal{X}| \log |\mathcal{Q}| + 4 \log \left(\frac{1}{\beta}\right)}{\varepsilon \|x\|_1} \right)^{1/3}. \quad (4.2)$$

Equivalently, for any database x with

$$\|x\|_1 \geq \frac{16 \log |\mathcal{X}| \log |\mathcal{Q}| + 4 \log \left(\frac{1}{\beta}\right)}{\varepsilon \alpha^3} \quad (4.3)$$

with probability $1 - \beta$: $\max_{f \in \mathcal{Q}} |f(x) - f(y)| \leq \alpha$.

Proof. By Theorem 4.2, we get:

$$\max_{f \in \mathcal{Q}} |f(x) - f(y)| \leq \frac{\alpha}{2} + \frac{2 \left(\frac{4 \log |\mathcal{X}| \log |\mathcal{Q}|}{\alpha^2} + \log \left(\frac{1}{\beta}\right) \right)}{\varepsilon \|x\|_1}.$$

Setting this quantity to be at most α and solving for $\|x\|_1$ yields (4.3). Solving for α yields (4.4). \square

Note that this theorem states that for fixed α and ε , even with $\delta = 0$, it is possible to answer almost *exponentially* many queries in the size of the database.² This is in contrast to the Laplace mechanism, when we use it directly to answer linear queries, which can only answer *linearly* many.

Note also that in this discussion, it has been most convenient to think about normalized queries. However, we can get the corresponding bounds for unnormalized queries simply by multiplying by $\|x\|_1$:

Theorem 4.6 (Accuracy theorem for un-normalized queries). By the appropriate choice of α , letting y be the database output by

²Specifically, solving for k we find that the mechanism can answer k queries for:

$$k \leq \exp \left(O \left(\frac{\alpha^3 \varepsilon \|x\|_1}{\log |\mathcal{X}|} \right) \right).$$

$\text{SmallDB}(x, \mathcal{Q}, \varepsilon, \frac{\alpha}{2})$, we can ensure that with probability $1 - \beta$:

$$\max_{f \in \mathcal{Q}} |f(x) - f(y)| \leq \|x\|_1^{2/3} \left(\frac{16 \log |\mathcal{X}| \log |\mathcal{Q}| + 4 \log \left(\frac{1}{\beta}\right)}{\varepsilon} \right)^{1/3}. \quad (4.4)$$

More Refined Bounds. We proved that *every* set of linear queries \mathcal{Q} has a collection of databases of size at most $|\mathcal{X}|^{\log |\mathcal{Q}|/\alpha^2}$ that well-approximates every database x with respect to \mathcal{Q} with error at most α . This is often an over-estimate however, since it completely ignores the structure of the queries. For example, if \mathcal{Q} simply contains the same query repeated over and over again, each time in a different guise, then there is no reason that the size of the range of the exponential mechanism should grow with $|\mathcal{Q}|$. Similarly, there may even be classes of queries \mathcal{Q} that have *infinite* cardinality, but nevertheless are well approximated by small databases. For example, queries that correspond to asking whether a point lies within a given interval on the real line form an infinitely large class \mathcal{Q} , since there are uncountably many intervals on the real line. Nevertheless, this class of queries exhibits very simple structure that causes it to be well approximated by small databases. By considering more refined structure of our query classes, we will be able to give bounds for differentially private mechanisms which improve over the simple sampling bounds (Lemma 4.3) and can be non-trivial even for doubly exponentially large classes of queries.³ We will not fully develop these bounds here, but will instead state several results for the simpler class of *counting queries*. Recall that a counting query $f : \mathcal{X} \rightarrow \{0, 1\}$ maps database points to boolean values, rather than any value in the interval $[0, 1]$ as linear queries do.

Definition 4.1 (Shattering). A class of counting queries \mathcal{Q} *shatters* a collection of points $S \subseteq \mathcal{X}$ if for every $T \subseteq S$, there exists an $f \in \mathcal{Q}$ such that $\{x \in S : f(x) = 1\} = T$. That is, \mathcal{Q} shatters S if for every one of the $2^{|S|}$ subsets T of S , there is some function in \mathcal{Q} that labels exactly

³In fact, our complexity measure for a class of queries can be finite even for *infinite* classes of queries, but here we are dealing with queries over a finite universe, so there do not exist infinitely many distinct queries.

those elements as positive, and does not label any of the elements in $S \setminus T$ as positive.

Note that for \mathcal{Q} to shatter S it must be the case that $|\mathcal{Q}| \geq 2^{|S|}$ since \mathcal{Q} must contain a function f for each subset $T \subseteq S$. We can now define our complexity measure for counting queries.

Definition 4.2 (Vapnik–Chervonenkis (VC) Dimension). A collection of counting queries \mathcal{Q} has VC-dimension d if there exists some set $S \subseteq \mathcal{X}$ of cardinality $|S| = d$ such that \mathcal{Q} shatters S , and \mathcal{Q} does not shatter any set of cardinality $d+1$. We can denote this quantity by $\text{VC-DIM}(\mathcal{Q})$.

Consider again the class of 1-dimensional intervals on the range $[0, \infty]$ defined over the domain $\mathcal{X} = \mathbb{R}$. The function $f_{a,b}$ corresponding to the interval $[a, b]$ is defined such that $f_{a,b}(x) = 1$ if and only if $x \in [a, b]$. This is an infinite class of queries, but its VC-dimension is 2. For any pair of distinct points $x < y$, there is an interval that contains neither point ($a, b < x$), an interval that contains both points ($a < x < y < b$), and an interval that contains each of the points but not the other ($a < x < b < y$ and $x < a < y < b$). However, for any 3 distinct points $x < y < z$, there is no interval $[a, b]$ such that $f_{a,b}[x] = f_{a,b}[z] = 1$ but $f_{a,b}[y] = 0$.

We observe that the VC-dimension of a finite concept class can never be too large.

Lemma 4.7. For any finite class \mathcal{Q} , $\text{VC-DIM}(\mathcal{Q}) \leq \log |\mathcal{Q}|$.

Proof. If $\text{VC-DIM}(\mathcal{Q}) = d$ then \mathcal{Q} shatters some set of items $S \subseteq \mathcal{X}$ of cardinality $|S| = d$. But by the definition of shattering, since S has 2^d distinct subsets, \mathcal{Q} must have at least 2^d distinct functions in it. \square

It will turn out that we can essentially replace the term $\log |\mathcal{Q}|$ with the term $\text{VC-DIM}(\mathcal{Q})$ in our bounds for the SmallDB mechanism. By the previous lemma, this is can only be an improvement for finite classes \mathcal{Q} .

Theorem 4.8. For any finite class of linear queries \mathcal{Q} , if $\mathcal{R} = \{y \in \mathbb{N}^{|\mathcal{X}|} : \|y\| \in O\left(\frac{\text{VC-DIM}(\mathcal{Q})}{\alpha^2}\right)\}$ then for all $x \in \mathbb{N}^{|\mathcal{X}|}$, there exists a $y \in \mathcal{R}$

such that:

$$\max_{f \in \mathcal{Q}} |f(x) - f(y)| \leq \alpha$$

As a result of this theorem, we get the analogue of Theorem 4.5 with VC-dimension as our measure of query class complexity:

Theorem 4.9. Let y be the database output by $\text{SmallDB}(x, \mathcal{Q}, \varepsilon, \frac{\alpha}{2})$. Then with probability $1 - \beta$:

$$\max_{f \in \mathcal{Q}} |f(x) - f(y)| \leq O \left(\left(\frac{\log |\mathcal{X}| \text{VC-DIM}(\mathcal{Q}) + \log \left(\frac{1}{\beta} \right)}{\varepsilon \|x\|_1} \right)^{1/3} \right)$$

Equivalently, for any database x with

$$\|x\|_1 \geq O \left(\frac{\log |\mathcal{X}| \text{VC-DIM}(\mathcal{Q}) + \log \left(\frac{1}{\beta} \right)}{\varepsilon \alpha^3} \right)$$

with probability $1 - \beta$: $\max_{f \in \mathcal{Q}} |f(x) - f(y)| \leq \alpha$.

An analogous (although more cumbersome) measure of query complexity, the “Fat Shattering Dimension,” defines the complexity of a class of linear queries, as opposed to simply counting queries. The Fat Shattering Dimension controls the size of the smallest “ α -net” (Definition 5.2 in Section 5) for a class of linear queries \mathcal{Q} as VC-dimension does for counting queries. This measure can similarly be used to give more refined bounds for mechanisms designed to privately release linear queries.

4.2 An online mechanism: private multiplicative weights

We will now give a mechanism for answering queries that arrive online and may be interactively chosen. The algorithm will be a simple combination of the sparse vector algorithm (which can answer threshold queries adaptively), and the exponentiated gradient descent algorithm for learning linear predictors online.

This latter algorithm is also known as Hedge or more generally the multiplicative weights technique. The idea is the following: When we

view the database $D \in \mathbb{N}^{|\mathcal{X}|}$ as a histogram and are interested only in linear queries (i.e., linear functions of this histogram), then we can view the problem of answering linear queries as the problem of learning the linear function D that defines the query answers $\langle D, q \rangle$, given a query $q \in [0, 1]^{|\mathcal{X}|}$. If the learning algorithm only needs to access the data using privacy-preserving queries, then rather than having a privacy cost that grows with the number of queries we would like to answer, we can have a privacy cost that grows only with the number of queries the learning algorithm needs to make. The “multiplicative weights” algorithm which we present next is a classical example of such a learning algorithm: it can learn any linear predictor by making only a small number of queries. It maintains at all times a current “hypothesis predictor,” and accesses the data only by requiring examples of queries on which its hypothesis predictor differs from the (true) private database by a large amount. Its guarantee is that it will always learn the target linear function up to small error, given only a small number of such examples. How can we find these examples? The sparse vector algorithm that we saw in the previous section allows us to do this on the fly, while paying for only those examples that have high error on the current multiplicative weights hypothesis. As queries come in, we ask whether the true answer to the query differs substantially from the answer to the query on the current multiplicative weights hypothesis. Note that this is a threshold query of the type handled by the sparse vector technique. If the answer is “no” — i.e., the difference, or error, is “below threshold,” — then we can respond to the query using the publicly known hypothesis predictor, and have no further privacy loss. If the answer is “yes,” meaning that the currently known hypothesis predictor gives rise to an error that is above threshold, then we have found an example appropriate to update our learning algorithm. Because “above threshold” answers correspond exactly to queries needed to update our learning algorithm, the total privacy cost depends only on the learning rate of the algorithm, and not on the total number of queries that we answer.

First we give the multiplicative weights update rule and prove a theorem about its convergence in the language of answering linear queries.

It will be convenient to think of databases x as being probability distributions over the data universe \mathcal{X} . That is, letting $\Delta([\mathcal{X}])$ denote the set of probability distributions over the set $[\mathcal{X}]$, we have $x \in \Delta([\mathcal{X}])$. Note that we can always scale a database to have this property without changing the normalized value of any linear query.

Algorithm 5 The Multiplicative Weights (MW) Update Rule. It is instantiated with a parameter $\eta \leq 1$. In the following analysis, we will take $\eta = \alpha/2$, where α is the parameter specifying our target accuracy.

MW(x^t, f_t, v_t):

```

if  $v_t < f_t(x^t)$  then
    Let  $r_t = f_t$ 
else
    Let  $r_t = 1 - f_t$ 
    (i.e., for all  $\chi_i$ ,  $r_t(\chi_i) = 1 - f_t[\chi_i]$ )
end if
```

Update: For all $i \in [\mathcal{X}]$ Let

$$\hat{x}_i^{t+1} = \exp(-\eta r_t[i]) \cdot x_i^t$$

$$x_i^{t+1} = \frac{\hat{x}_i^{t+1}}{\sum_{j=1}^{|\mathcal{X}|} \hat{x}_j^{t+1}}$$

Output x^{t+1} .

Theorem 4.10. Fix a class of linear queries \mathcal{Q} and a database $x \in \Delta([\mathcal{X}])$, and let $x^1 \in \Delta([\mathcal{X}])$ describe the uniform distribution over \mathcal{X} : $x_i^1 = 1/|\mathcal{X}|$ for all i . Now consider a maximal length sequence of databases x^t for $t \in \{2, \dots, L\}$ generated by setting $x^{t+1} = \text{MW}(x^t, f_t, v_t)$ as described in Algorithm 5, where for each t , $f_t \in \mathcal{Q}$ and $v_t \in \mathbb{R}$ are such that:

1. $|f_t(x) - f_t(x^t)| > \alpha$, and
2. $|f_t(x) - v_t| < \alpha$.

Then it must be that:

$$L \leq 1 + \frac{4 \log |\mathcal{X}|}{\alpha^2}.$$

Note that if we prove this theorem, we will have proven that for the last database x^{L+1} in the sequence it must be that for all $f \in \mathcal{Q}$: $|f(x) - f(x^{L+1})| \leq \alpha$, as otherwise it would be possible to extend the sequence, contradicting maximality. In other words, given *distinguishing queries* f^t , the multiplicative weights update rule learns the private database x with respect to any class of linear queries \mathcal{Q} , up to some tolerance α , in only a small number (L) of steps. We will use this theorem as follows. The Private Online Multiplicative Weights algorithm, described (twice!) below, will at all times t have a *public* approximation x^t to the database x . Given an input query f , the algorithm will compute a noisy approximation to the difference $|f(x) - f(x^t)|$. If the (noisy) difference is large, the algorithm will provide a noisy approximation $f(x) + \lambda_t$ to the true answer $f(x)$, where λ_t is drawn from some appropriately chosen Laplace distribution, and the Multiplicative Weights Update Rule will be invoked with parameters $(x^t, f, f(x) + \lambda_t)$. If the update rule is invoked only when the difference $|f(x) - f(x^t)|$ is truly large (Theorem 4.10, condition 1), and if the approximations $f(x) + \lambda_t$ are sufficiently accurate (Theorem 4.10, condition 2), then we can apply the theorem to conclude that updates are not so numerous (because L is not so large) *and* the resulting x^{L+1} gives accurate answers to all queries in \mathcal{Q} (because no distinguishing query remains).

Theorem 4.10 is proved by keeping track of a potential function Ψ measuring the similarity between the hypothesis database x^t at time t , and the true database D . We will show:

1. The potential function does not start out too large.
2. The potential function decreases by a significant amount at each update round.
3. The potential function is always non-negative.

Together, these 3 facts will force us to conclude that there cannot be too many update rounds.

Let us now begin the analysis for the proof of the convergence theorem.

Proof. We must show that any sequence $\{(x^t, f_t, v_t)\}_{t=1,\dots,L}$ with the property that $|f_t(x^t) - f_t(x)| > \alpha$ and $|v_t - f_t(x)| < \alpha$ cannot have $L > \frac{4\log|\mathcal{X}|}{\alpha^2}$.

We define our potential function as follows. Recall that we here view the database as a probability distribution — i.e., we assume $\|x\|_1 = 1$. Of course this does not require actually modifying the real database. The potential function that we use is the relative entropy, or KL divergence, between x and x^t (when viewed as probability distributions):

$$\Psi_t \stackrel{\text{def}}{=} KL(x\|x^t) = \sum_{i=1}^{|\mathcal{X}|} x[i] \log \left(\frac{x[i]}{x^t[i]} \right).$$

We begin with a simple fact:

Proposition 4.11. For all t : $\Psi_t \geq 0$, and $\Psi_1 \leq \log|\mathcal{X}|$.

Proof. Relative entropy (KL-Divergence) is always a non-negative quantity, by the log-sum inequality, which states that if a_1, \dots, a_n and b_1, \dots, b_n are non-negative numbers, then

$$\sum_i a_i \log \frac{a_i}{b_i} \geq \left(\sum_i a_i \right) \frac{\sum_i a_i}{\sum_i b_i}.$$

To see that $\Psi_1 \leq \log|\mathcal{X}|$, recall that $x^1[i] = 1/|\mathcal{X}|$ for all i , and so $\Psi_1 = \sum_{i=1}^{|\mathcal{X}|} x[i] \log(|\mathcal{X}|x[i])$. Noting that x is a probability distribution, we see that this quantity is maximized when $x[1] = 1$ and $x[i] = 0$ for all $i > 1$, giving $\Psi_1 = \log|\mathcal{X}|$. \square

We will now argue that at each step, the potential function drops by at least $\alpha^2/4$. Because the potential begins at $\log|\mathcal{X}|$, and must always be non-negative, we therefore know that there can be at most $L \leq 4\log|\mathcal{X}|/\alpha^2$ steps in the database update sequence. To begin, let us see exactly how much the potential drops at each step:

Lemma 4.12.

$$\Psi_t - \Psi_{t+1} \geq \eta \left(\langle r_t, x^t \rangle - \langle r_t, x \rangle \right) - \eta^2$$

Proof. Recall that $\sum_{i=1}^{|\mathcal{X}|} x[i] = 1$.

$$\begin{aligned}
\Psi_t - \Psi_{t+1} &= \sum_{i=1}^{|\mathcal{X}|} x[i] \log \left(\frac{x[i]}{x_i^t} \right) - \sum_{i=1}^{|\mathcal{X}|} x[i] \log \left(\frac{x[i]}{x_i^{t+1}} \right) \\
&= \sum_{i=1}^{|\mathcal{X}|} x[i] \log \left(\frac{x_i^{t+1}}{x_i^t} \right) \\
&= \sum_{i=1}^{|\mathcal{X}|} x[i] \log \left(\frac{\hat{x}_i^{t+1} / \sum_i \hat{x}_i^{t+1}}{x_i^t} \right) \\
&= \sum_{i=1}^{|\mathcal{X}|} x[i] \left[\log \left(\frac{x_i^t \exp(-\eta r_t[i]))}{x_i^t} \right) \right. \\
&\quad \left. - \log \left(\sum_{j=1}^{|\mathcal{X}|} \exp(-\eta r_t[j]) x_j^t \right) \right] \\
&= - \left(\sum_{i=1}^{|\mathcal{X}|} x[i] \eta r_t[i] \right) - \log \left(\sum_{i=1}^{|\mathcal{X}|} \exp(-\eta r_t[i]) x_i^t \right) \\
&= -\eta \langle r_t, x \rangle - \log \left(\sum_{j=1}^{|\mathcal{X}|} \exp(-\eta r_t[j]) x_j^t \right) \\
&\geq -\eta \langle r_t, x \rangle - \log \left(\sum_{j=1}^{|\mathcal{X}|} x_j^t (1 + \eta^2 - \eta r_t[j]) \right) \\
&= -\eta \langle r_t, x \rangle - \log \left(1 + \eta^2 - \eta \langle r_t, x^t \rangle \right) \\
&\geq \eta \left(\langle r_t, x^t \rangle - \langle r_t, x \rangle \right) - \eta^2.
\end{aligned}$$

The first inequality follows from the fact that:

$$\exp(-\eta r_t[j]) \leq 1 - \eta r_t[j] + \eta^2 (r_t[j])^2 \leq 1 - \eta r_t[j] + \eta^2.$$

The second inequality follows from the fact that $\log(1 + y) \leq y$ for $y > -1$. \square

The rest of the proof now follows easily. By the conditions of the database/query sequence (described in the hypothesis for Theorem 4.10 above), for every t ,

1. $|f_t(x) - f_t(x^t)| \geq \alpha$ and
2. $|v_t - f_t(x)| < \alpha$.

Thus, $f_t(x) < f_t(x^t)$ if and only if $v_t < f_t(x^t)$. In particular, $r_t = f_t$ if $f_t(x^t) - f_t(x) \geq \alpha$, and $r_t = 1 - f_t$ if $f_t(x) - f_t(x^t) \geq \alpha$. Therefore, by Lemma 4.12 and the choice of $\eta = \alpha/2$ as described in the Update Rule,

$$\Psi_t - \Psi_{t+1} \geq \frac{\alpha}{2} (\langle r_t, x^t \rangle - \langle r_t, x \rangle) - \frac{\alpha^2}{4} \geq \frac{\alpha}{2} (\alpha) - \frac{\alpha^2}{4} = \frac{\alpha^2}{4}.$$

Finally we know:

$$0 \leq \Psi_L \leq \Psi_0 - L \cdot \frac{\alpha^2}{4} \leq \log |\mathcal{X}| - L \frac{\alpha^2}{4}.$$

Solving, we find: $L \leq \frac{4\log|\mathcal{X}|}{\alpha^2}$. This completes the proof. \square

We can now combine the Multiplicative Weights Update Rule with the NumericSparse algorithm to give an interactive query release mechanism. For $(\epsilon, 0)$ privacy, we essentially (with somewhat worse constants) recover the bound for SmallDB. For (ϵ, δ) -differential privacy, we obtain better bounds, by virtue of being able to use the composition theorem. The queries to NumericSparse are asking whether the magnitude of the error given by estimating $f_i(x)$ by applying f_i to the current approximation x^t to x is above an appropriately chosen threshold T , that is, they are asking if $|f(x) - f(x^t)|$ is large. For technical reasons this is done by asking about $f(x) - f(x^t)$ (without the absolute value) and about $f(x^t) - f(x)$. Recall that the NumericSparse algorithm responds with either \perp or some (positive) value exceeding T . We use the mnemonic E for the responses to emphasize that the query is asking about an error.

Theorem 4.13. The Online Multiplicative Weights Mechanism (via NumericSparse) is $(\epsilon, 0)$ -differentially private.

Algorithm 6 The Online Multiplicative Weights Mechanism (via NumericSparse) takes as input a private database x , privacy parameters ϵ, δ , accuracy parameters α and β , and a stream of linear queries $\{f_i\}$ that may be chosen adaptively from a class of queries \mathcal{Q} . It outputs a stream of answers $\{a_i\}$.

OnlineMW via NumericSparse ($x, \{f_i\}, \epsilon, \delta, \alpha, \beta$)

```

Let  $c \leftarrow \frac{4 \log |\mathcal{X}|}{\alpha^2}$ ,
if  $\delta = 0$  then
    Let  $T \leftarrow \frac{18c(\log(2|\mathcal{Q}|) + \log(4c/\beta))}{\epsilon \|x\|_1}$ 
else
    Let  $T \leftarrow \frac{(2+32\sqrt{2})\sqrt{c \log \frac{2}{\delta}}(\log k + \log \frac{4c}{\beta})}{\epsilon \|x\|_1}$ 
end if
```

Initialize NumericSparse($x, \{f'_i\}, T, c, \epsilon, \delta$) with a stream of queries $\{f'_i\}$, outputting a stream of answers E_i .

Let $t \leftarrow 0$, and let $x^0 \in \Delta([\mathcal{X}])$ satisfy $x_i^0 = 1/|\mathcal{X}|$ for all $i \in [\mathcal{X}]$.

for each query f_i **do**

Let $f'_{2i-1}(\cdot) = f_i(\cdot) - f_i(x^t)$.

Let $f'_{2i}(\cdot) = f_i(x^t) - f_i(\cdot)$

if $E_{2i-1} = \perp$ and $E_{2i} = \perp$ **then**

Let $a_i = f_i(x^t)$

else

if $E_{2i-1} \in \mathbb{R}$ **then**

Let $a_i = f_i(x^t) + E_{2i-1}$

else

Let $a_i = f_i(x^t) - E_{2i}$

end if

Let $x^{t+1} = MW(x^t, f_i, a_i)$

Let $t \leftarrow t + 1$.

end if

end for

Proof. This follows directly from the privacy analysis of NumericSparse, because the OnlineMW algorithm accesses the database only through NumericSparse. \square

Speaking informally, the proof of utility for the Online Multiplicative Weights Mechanism (via NumericSparse) uses the utility theorem for the NumericSparse (Theorem 3.28) to conclude that, with high probability, the Multiplicative Weights Update Rule is only invoked when the query f_t is truly a distinguishing query, meaning, $|f_i(x) - f_t(x^t)|$ is “large,” and the released noisy approximations to $f_i(x)$ are “accurate.” Under this assumption, we can apply the convergence theorem (Theorem 4.10) to conclude that the total number of updates is small and therefore the algorithm can answer all queries in \mathcal{Q} .

Theorem 4.14. For $\delta = 0$, with probability at least $1 - \beta$, for all queries f_i , the Online Multiplicative Weights Mechanism (via NumericSparse) returns an answer a_i such that $|f_i(x) - a_i| \leq 3\alpha$ for any α such that:

$$\alpha \geq \frac{32 \log |\mathcal{X}| \left(\log(|\mathcal{Q}|) + \log \left(\frac{32 \log |\mathcal{X}|}{\alpha^2 \beta} \right) \right)}{\epsilon \alpha^2 \|x\|_1}$$

Proof. Recall that, by Theorem 3.28, given k queries and a maximum number c of above-threshold queries, NumericSparse is (α, β) -accurate for any α such that:

$$\alpha \geq \frac{9c(\log k + \log(4c/\beta))}{\epsilon}.$$

In our case $c = 4 \log |\mathcal{X}| / \alpha^2$ and $k = 2|\mathcal{Q}|$, and we have been normalizing, which reduces α by a factor of $\|x\|_1$. With this in mind, we can take

$$\alpha = \frac{32 \log |\mathcal{X}| \left(\log(|\mathcal{Q}|) + \log \left(\frac{32 \log |\mathcal{X}|}{\alpha^2 \beta} \right) \right)}{\epsilon \alpha^2 \|x\|_1}$$

and note that with this value we get $T = 2\alpha$ for the case $\delta = 0$.

Assume we are in this high $(1 - \beta)$ probability case. Then for all i such that f_i triggers an update, $|f_i(x) - f_i(x^t)| \geq T - \alpha = \alpha$ (Theorem 4.10, condition 1). Thus, f_i, a_i form a valid pair of query/value updates as required in the hypothesis of Theorem 4.10 and so, by that theorem, there can be at most $c = \frac{4 \log |\mathcal{X}|}{\alpha^2}$ such update steps.

In addition, still by the accuracy properties of the Sparse Vector algorithm,

1. at most one of E_{2i-1}, E_{2i} will have value \perp ;

2. for all i such that no update is triggered ($a_i = f_i(x^t)$) we have $|f_i(x) - f_i(x^t)| \leq T + \alpha = 3\alpha$; and
3. for all i such that an update is triggered we have $|f_i(x) - a_i| \leq \alpha$ (Theorem 4.10, condition 2). \square

Optimizing the above expression for α and removing the normalization factor, we find that the OnlineMW mechanism can answer each linear query to accuracy 3α except with probability β for:

$$\alpha = \|x\|_1^{2/3} \left(\frac{36 \log |\mathcal{X}| \left(\log(|\mathcal{Q}|) + \log \left(\frac{32 \log |\mathcal{X}|^{1/3} \|x\|_1^{2/3}}{\beta} \right) \right)}{\epsilon} \right)^{1/3}$$

which is comparable to the SmallDB mechanism.

By repeating the same argument, but instead using the utility theorem for the (ϵ, δ) -private version of Sparse Vector (Theorem 3.28), we obtain the following theorem.

Theorem 4.15. For $\delta > 0$, with probability at least $1 - \beta$, for all queries f_i , OnlineMW returns an answer a_i such that $|f_i(x) - a_i| \leq 3\alpha$ for any α such that:

$$\alpha \geq \frac{(2 + 32\sqrt{2}) \cdot \sqrt{\log |\mathcal{X}| \log \frac{2}{\delta}} \left(\log |\mathcal{Q}| + \log \left(\frac{32 \log |\mathcal{X}|}{\alpha^2 \beta} \right) \right)}{\alpha \epsilon \|x\|_1}$$

Again optimizing the above expression for α and removing the normalization factor, we find that the OnlineMW mechanism can answer each linear query to accuracy 3α except with probability β , for:

$$\alpha = \|x\|_1^{1/2} \left(\frac{(2 + 32\sqrt{2}) \cdot \sqrt{\log |\mathcal{X}| \log \frac{2}{\delta}} \left(\log |\mathcal{Q}| + \log \left(\frac{32 \|x\|_1}{\beta} \right) \right)}{\epsilon} \right)^{1/2}$$

which gives better accuracy (as a function of $\|x\|_1$) than the SmallDB mechanism. Intuitively, the greater accuracy comes from the iterative nature of the mechanism, which allows us to take advantage of our composition theorems for (ϵ, δ) -privacy. The SmallDB mechanism runs

in just a single shot, and so there is no opportunity to take advantage of composition.

The accuracy of the private multiplicative weights algorithm has dependencies on several parameters, which are worth further discussion. In the end, the algorithm answers queries using the *sparse vector technique* paired with a *learning algorithm for linear functions*. As we proved in the last section, the sparse vector technique introduces error that scales like $O(c \log k / (\epsilon \|x\|_1))$ when a total of k sensitivity $1/\|x\|_1$ queries are made, and at most c of them can have “above threshold” answers, for any threshold T . Recall that these error terms arise because the privacy analysis for the sparse vector algorithm allows us to “pay” only for the above threshold queries, and therefore can add noise $O(c / (\epsilon \|x\|_1))$ to each query. On the other hand, since we end up adding independent Laplace noise with scale $\Omega(c / (\epsilon \|x\|_1))$ to k queries in total, we expect that the maximum error over all k queries is larger by a $\log k$ factor. But what is c , and what queries should we ask? The multiplicative weights learning algorithm gives us a query strategy and a guarantee that no more than $c = O(\log |\mathcal{X}| / \alpha^2)$ queries will be above a threshold of $T = O(\alpha)$, for any α . (The queries we ask are always: “How much does the real answer differ from the predicted answer of the current multiplicative weights hypothesis.” The answers to these questions both give us the true answers to the queries, as well as instructions how to update the learning algorithm appropriately when a query is above threshold.) Together, this leads us to set the threshold to be $O(\alpha)$, where α is the expression that satisfies: $\alpha = O(\log |\mathcal{X}| \log k / (\epsilon \|x\|_1 \alpha^2))$. This minimizes the two sources of error: error from the sparse vector technique, and error from failing to update the multiplicative weights hypothesis.

4.3 Bibliographical notes

The offline query release mechanism given in this section is from Blum et al. [8], which gave bounds in terms of the VC-Dimension of the query class (Theorem 4.9). The generalization to fat shattering dimension is given in [72].

The online query release mechanism given in this section is from Hardt and Rothblum [44]. This mechanism uses the classic multiplicative weights update method, for which Arora, Hazan and Kale give an excellent survey [1]. Slightly improved bounds for the private multiplicative weights mechanism were given by Gupta et al. [39], and the analysis here follows the presentation from [39].

5

Generalizations

In this section we generalize the query release algorithms of the previous section. As a result, we get bounds for arbitrary low sensitivity queries (not just linear queries), as well as new bounds for linear queries. These generalizations also shed some light on a connection between query release and machine learning.

The SmallDB offline query release mechanism in Section 4 is a special case of what we call the *net mechanism*. We saw that both mechanisms in that section yield *synthetic databases*, which provide a convenient means for approximating the value of any query in \mathcal{Q} on the private database: just evaluate the query on the synthetic database and take the result as the noisy answer. More generally, a mechanism can produce a *data structure* of arbitrary form, that, together with a fixed, public, algorithm (independent of the database) provides a method for approximating the values of queries.

The Net mechanism is a straightforward generalization of the SmallDB mechanism: First, fix, independent of the actual database, an α -net of data structures such that evaluation of any query in \mathcal{Q} using the released data structure gives a good (within an additive α error) estimate of the value of the query on the private database. Next, apply

the exponential mechanism to choose an element of this net, where the quality function minimizes the maximum error, over the queries in \mathcal{Q} , for the elements of the net.

We also generalize the online multiplicative weights algorithm so that we can instantiate it with any other *online learning algorithm* for learning a database with respect to a set of queries. We note that such a mechanism can be run either online, or offline, where the set of queries to be asked to the “online” mechanism is instead selected using a “private distinguisher,” which identifies queries on which the current hypothesis of the learner differs substantially from the real database. These are queries that would have yielded an update step in the online algorithm. A “distinguisher” turns out to be equivalent to an agnostic learning algorithm, which sheds light on a source of hardness for efficient query release mechanisms.

In the following sections, we will discuss *data structures* for classes of queries \mathcal{Q} .

Definition 5.1. A data structure D drawn from some class of data structures \mathcal{D} for a class of queries \mathcal{Q} is implicitly endowed with an evaluation function $\text{Eval} : \mathcal{D} \times \mathcal{Q} \rightarrow \mathbb{R}$ with which we can evaluate any query in \mathcal{Q} on D . However, to avoid being encumbered by notation, we will write simply $f(D)$ to denote $\text{Eval}(D, f)$ when the meaning is clear from context.

5.1 Mechanisms via α -nets

Given a collection of queries \mathcal{Q} , we define an α -net as follows:

Definition 5.2 (α -net). An α -net of data structures with respect to a class of queries \mathcal{Q} is a set $\mathcal{N} \subset \mathbb{N}^{|\mathcal{X}|}$ such that for all $x \in \mathbb{N}^{|\mathcal{X}|}$, there exists an element of the α -net $y \in \mathcal{N}$ such that:

$$\max_{f \in \mathcal{Q}} |f(x) - f(y)| \leq \alpha .$$

We write $\mathcal{N}_\alpha(\mathcal{Q})$ to denote an α -net of minimum cardinality among the set of all α -nets for \mathcal{Q} .

That is, for every possible database x , there exists a member of the α -net that “looks like” x with respect to all queries in \mathcal{Q} , up to an error tolerance of α .

Small α -nets will be useful for us, because when paired with the exponential mechanism, they will lead directly to mechanisms for answering queries with high accuracy. Given a class of functions \mathcal{Q} , we will define an instantiation of the exponential mechanism known as the *Net* mechanism. We first observe that the Net mechanism preserves ε -differential privacy.

Algorithm 7 The Net Mechanism

NetMechanism($x, \mathcal{Q}, \varepsilon, \alpha$)

Let $\mathcal{R} \leftarrow \mathcal{N}_\alpha(\mathcal{Q})$

Let $q : \mathbb{N}^{|\mathcal{X}|} \times \mathcal{R} \rightarrow \mathbb{R}$ be defined to be:

$$q(x, y) = -\max_{f \in \mathcal{Q}} |f(x) - f(y)|$$

Sample And Output $y \in \mathcal{R}$ with the exponential mechanism
 $\mathcal{M}_E(x, q, \mathcal{R})$

Proposition 5.1. The Net mechanism is $(\varepsilon, 0)$ differentially private.

Proof. The Net mechanism is simply an instantiation of the exponential mechanism. Therefore, privacy follows from Theorem 3.10. \square

We may similarly call on our analysis of the exponential mechanism to begin understanding the utility guarantees of the Net mechanism:

Proposition 5.2. Let \mathcal{Q} be any class of sensitivity $1/\|x\|_1$ queries. Let y be the database output by $\text{NetMechanism}(x, \mathcal{Q}, \varepsilon, \alpha)$. Then with probability $1 - \beta$:

$$\max_{f \in \mathcal{Q}} |f(x) - f(y)| \leq \alpha + \frac{2 \left(\log(|\mathcal{N}_\alpha(\mathcal{Q})|) + \log\left(\frac{1}{\beta}\right) \right)}{\varepsilon \|x\|_1}.$$

Proof. By applying Theorem 3.11 and noting that $S(q) = \frac{1}{\|x\|_1}$, and that $\text{OPT}_q(D) \leq \alpha$ by the definition of an α -net, we find:

$$\Pr \left[\max_{f \in \mathcal{Q}} |f(x) - f(y)| \geq \alpha + \frac{2}{\varepsilon \|x\|_1} (\log(|\mathcal{N}_\alpha(\mathcal{Q})|) + t) \right] \leq e^{-t}.$$

Plugging in $t = \log\left(\frac{1}{\beta}\right)$ completes the proof. \square

We can therefore see that an upper bound on $|\mathcal{N}_\alpha(\mathcal{Q})|$ for a collection of functions \mathcal{Q} immediately gives an upper bound on the accuracy that a differentially private mechanism can provide simultaneously for *all* functions in the class \mathcal{Q} .

This is exactly what we did in Section 4.1, where we saw that the key quantity is the VC-dimension of \mathcal{Q} , when \mathcal{Q} is a class of linear queries.

5.2 The iterative construction mechanism

In this section, we derive an offline generalization of the private multiplicative weights algorithm, which can be instantiated with any properly defined learning algorithm. Informally, a database update algorithm maintains a sequence of data structures D^1, D^2, \dots that give increasingly good approximations to the input database x (in a sense that depends on the database update algorithm). Moreover, these mechanisms produce the next data structure in the sequence by considering only one query f that *distinguishes* the real database in the sense that $f(D^t)$ differs significantly from $f(x)$. The algorithm in this section shows that, up to small factors, solving the query-release problem in a differentially private manner is equivalent to solving the simpler *learning* or *distinguishing* problem in a differentially private manner: given a private distinguishing algorithm and a non-private database update algorithm, we get a corresponding private release algorithm. We can plug in the exponential mechanism as a canonical private distinguisher, and the multiplicative weights algorithm as a generic database update algorithm for the general linear query setting, but more efficient distinguishers are possible in special cases.

Syntactically, we will consider functions of the form $U : \mathcal{D} \times \mathcal{Q} \times \mathbb{R} \rightarrow \mathcal{D}$, where \mathcal{D} represents a class of data structures on which queries in \mathcal{Q} can be evaluated. The inputs to U are a data structure in \mathcal{D} , which represents the current data structure D^t ; a query f , which represents the distinguishing query, and may be restricted to a certain set \mathcal{Q} ; and also a real number, which estimates $f(x)$. Formally, we define a *database update sequence*, to capture the sequence of inputs to U used to generate the database sequence D^1, D^2, \dots

Definition 5.3 (Database Update Sequence). Let $x \in \mathbb{N}^{|\mathcal{X}|}$ be any database and let $\{(D^t, f_t, v_t)\}_{t=1, \dots, L} \in (\mathcal{D} \times \mathcal{Q} \times \mathbb{R})^L$ be a sequence of tuples. We say the sequence is a $(U, x, \mathcal{Q}, \alpha, T)$ -*database update sequence* if it satisfies the following properties:

1. $D^1 = U(\perp, \cdot, \cdot)$,
2. for every $t = 1, 2, \dots, L$, $|f_t(x) - f_t(D^t)| \geq \alpha$,
3. for every $t = 1, 2, \dots, L$, $|f_t(x) - v_t| < \alpha$,
4. and for every $t = 1, 2, \dots, L - 1$, $D^{t+1} = U(D^t, f_t, v_t)$.

We note that for all of the database update algorithms we consider, the approximate answer v_t is used only to determine the *sign* of $f_t(x) - f_t(D^t)$, which is the motivation for requiring that the estimate of $f_t(x)$ (v_t) have error smaller than α . The main measure of efficiency we're interested in from a database update algorithm is the maximum number of updates we need to perform before the database D^t approximates x well with respect to the queries in \mathcal{Q} . To this end we define a database update algorithm as follows:

Definition 5.4 (Database Update Algorithm). Let $U : \mathcal{D} \times \mathcal{Q} \times \mathbb{R} \rightarrow \mathcal{D}$ be an update rule and let $T : \mathbb{R} \rightarrow \mathbb{R}$ be a function. We say U is a $T(\alpha)$ -*database update algorithm for query class \mathcal{Q}* if for every database $x \in \mathbb{N}^{|\mathcal{X}|}$, every $(U, x, \mathcal{Q}, \alpha, L)$ -database update sequence satisfies $L \leq T(\alpha)$.

Note that the definition of a $T(\alpha)$ -database update algorithm implies that if U is a $T(\alpha)$ -database update algorithm, then given any maximal $(U, x, \mathcal{Q}, \alpha, U)$ -database update sequence, the final database D^L must satisfy $\max_{f \in \mathcal{Q}} |f(x) - f(D^L)| \leq \alpha$ or else there would exist

another query satisfying property 2 of Definition 5.3, and thus there would exist a $(U, x, \mathcal{Q}, \alpha, L + 1)$ -database update sequence, contradicting maximality. That is, the goal of a $T(\alpha)$ database update rule is to generate a maximal database update sequence, and the final data structure in a maximal database update sequence necessarily encodes the approximate answers to every query $f \in \mathcal{Q}$.

Now that we have defined database update algorithms, we can remark that what we really proved in Theorem 4.10 was that the Multiplicative Weights algorithm is a $T(\alpha)$ -database update algorithm for $T(\alpha) = 4 \log |\mathcal{X}|/\alpha^2$.

Before we go on, let us build some intuition for what a database update algorithm is. A $T(\alpha)$ -database update algorithm begins with some initial guess D^1 about what the true database x looks like. Because this guess is not based on any information, it is quite likely that D^1 and x bear little resemblance, and that there is some $f \in \mathcal{Q}$ that is able to distinguish between these two databases by at least α : that is, that $f(x)$ and $f(D^1)$ differ in value by at least α . What a database update algorithm does is to update its hypothesis D^t given evidence that its current hypothesis D^{t-1} is incorrect: at each stage, it takes as input some query in \mathcal{Q} which distinguishes its current hypothesis from the true database, and then it outputs a new hypothesis. The parameter $T(\alpha)$ is an upper bound on the number of times that the database update algorithm will have to update its hypothesis: it is a promise that after at most $T(\alpha)$ distinguishing queries have been provided, the algorithm will finally have produced a hypothesis that looks like the true database with respect to \mathcal{Q} , at least up to error α .¹ For a database update algorithm, smaller bounds $T(\alpha)$ are more desirable.

Database Update Algorithms and Online Learning Algorithms: We remark that database update algorithms are essentially *online learning*

¹Imagine that the database update algorithm is attempting to sculpt x out of a block of clay. Initially, its sculpture D^1 bears no resemblance to the true database: it is simply a block of clay. However, a helpful distinguisher points out to the sculptor places in which the clay juts out much farther than the true target database: the sculptor dutifully pats down those bumps. If the distinguisher always finds large protrusions, of magnitude at least α , the sculpture will be finished soon, and the distinguisher's time will not be wasted!

algorithms in the *mistake bound model*. In the setting of online learning, unlabeled examples arrive in some arbitrary order, and the learning algorithm must attempt to label them.

Background from Learning Theory. In the *mistake bound model of learning*, labeled examples $(x_i, y_i) \in \mathcal{X} \times \{0, 1\}$ arrive one at a time, in a potentially adversarial order. At time i , the learning algorithm A observes x_i , and must make a prediction \hat{y}_i about the label for x_i . It then sees the true label y_i , and is said to *make a mistake* if its prediction was wrong: i.e., if $y_i \neq \hat{y}_i$. A learning algorithm A for a class of functions C is said to have a mistake bound of M , if for all $f \in C$, and for all adversarially selected sequences of examples $(x_1, f(x_1)), \dots, (x_i, f(x_i)), \dots$, A never makes more than M mistakes. Without loss of generality, we can think of such a learning algorithm as maintaining some hypothesis $\hat{f} : \mathcal{X} \rightarrow \{0, 1\}$ at all times, and updating it only when it makes a mistake. The adversary in this model is quite powerful — it can choose the sequence of labeled examples adaptively, knowing the current hypothesis of the learning algorithm, and its entire history of predictions. Hence, learning algorithms that have finite mistake bounds can be useful in extremely general settings.

It is not hard to see that mistake bounded online learning algorithms always exist for finite classes of functions C . Consider, for example, the *halving algorithm*. The halving algorithm initially maintains a set S of functions from C consistent with the examples that it has seen so far: Initially $S = C$. Whenever a new unlabeled example arrives, it predicts according to the majority vote of its consistent hypotheses: that is, it predicts label 1 whenever $|\{f \in S : f(x_i) = 1\}| \geq |S|/2$. Whenever it makes a mistake on an example x_i , it updates S by removing any inconsistent function: $S \leftarrow \{f \in S : f(x_i) = y_i\}$. Note that whenever it makes a mistake, the size of S is cut in half! So long as all examples are labeled by *some* function $f \in C$, there is at least one function $f \in C$ that is never removed from S . Hence, the halving algorithm has a mistake bound of $\log |C|$.

Generalizing beyond boolean labels, we can view database update algorithms as online learning algorithms in the mistake bound model:

here, examples that arrive are the queries (which may come in adversarial order). The labels are the approximate values of the queries when evaluated on the database. The database update algorithm hypothesis D^t makes a *mistake* on query f if $|f(D^t) - f(x)| \geq \alpha$, in which case we learn the label of f (that is, v_t) and allow the database update algorithm to update the hypothesis. Saying that an algorithm U is a $T(\alpha)$ -database update algorithm is akin to saying that it has a mistake bound of $T(\alpha)$: no adversarially chosen sequence of queries can ever cause it to make more than $T(\alpha)$ -mistakes. Indeed, the database update algorithms that we will see are taken from the online learning literature. The multiplicative weights mechanism is based on an online learning algorithm known as *Hedge*, which we have already discussed. The Median Mechanism (later in this section) is based on the *Halving Algorithm*, and the Perceptron algorithm is based (coincidentally) on an algorithm known as *Perceptron*. We won't discuss Perceptron here, but it operates by making *additive* updates, rather than the multiplicative updates used by multiplicative weights.

A database update algorithm for a class \mathcal{Q} will be useful together with a corresponding *distinguisher*, whose job is to output a function that behaves differently on the true database x and the hypothesis D^t , that is, to point out a mistake.

Definition 5.5 $((F(\varepsilon), \gamma)$ -Private Distinguisher). Let \mathcal{Q} be a set of queries, let $\gamma \geq 0$ and let $F(\varepsilon) : \mathbb{R} \rightarrow \mathbb{R}$ be a function. An algorithm $\text{Distinguish}_\varepsilon : \mathbb{N}^{|\mathcal{X}|} \times \mathcal{D} \rightarrow \mathcal{Q}$ is an $(F(\varepsilon), \gamma)$ -Private Distinguisher for \mathcal{Q} if for every setting of the privacy parameter ε , on every pair of inputs $x \in \mathbb{N}^{|\mathcal{X}|}$, $D \in \mathcal{D}$ it is $(\varepsilon, 0)$ -differentially private with respect to x and it outputs an $f^* \in \mathcal{Q}$ such that $|f^*(x) - f^*(D)| \geq \max_{f \in \mathcal{Q}} |f(x) - f(D)| - F(\varepsilon)$ with probability at least $1 - \gamma$.

Remark 5.1. In machine learning, the goal is to find a function $f : \mathcal{X} \rightarrow \{0, 1\}$ from a class of functions \mathcal{Q} that *best labels* a collection of labeled examples $(x_1, y_1), \dots, (x_m, y_m) \in \mathcal{X} \times \{0, 1\}$. (Examples $(x, 0)$ are known as *negative examples*, and examples $(x, 1)$ are known as *positive examples*). Each example x_i has a *true label* y_i , and a function f *correctly labels* x_i if $f(x_i) = y_i$. An *agnostic learning algorithm* for a class \mathcal{Q} is an algorithm that can find the function in \mathcal{Q} that labels

all of the data points approximately as well as the best function in \mathcal{Q} , even if no function in \mathcal{Q} can perfectly label them. Note that equivalently, an agnostic learning algorithm is one that maximizes the number of positive examples labeled 1 minus the number of negative examples labeled 1. Phrased in this way, we can see that a *distinguisher* as defined above is just an agnostic learning algorithm: just imagine that x contains all of the “positive” examples, and that y contains all of the “negative examples.” (Note that it is ok if x and y are not disjoint — in the learning problem, the same example can occur with both a positive and a negative label, since agnostic learning does not require that any function perfectly label every example.) Finally, note also that for classes of linear queries \mathcal{Q} , a distinguisher is simply an optimization algorithm. Because for linear queries f , $f(x) - f(y) = f(x - y)$, a distinguisher simply seeks to find $\arg \max_{f \in \mathcal{Q}} |f(x - y)|$.

Note that, *a priori*, a differentially private distinguisher is a weaker object than a differentially private release algorithm: A distinguisher merely finds a query in a set \mathcal{Q} with the approximately largest value, whereas a release algorithm must find the answer to every query in \mathcal{Q} . In the algorithm that follows, however, we reduce release to optimization.

We will first analyze the IC algorithm, and then instantiate it with a specific distinguisher and database update algorithm. What follows is a formal analysis, but the intuition for the mechanism is simple: we simply run the iterative database construction algorithm to construct a hypothesis that approximately matches x with respect to the queries \mathcal{Q} . If at each round our distinguisher succeeds in finding a query that has high discrepancy between the hypothesis database and the true database, then our database update algorithm will output a database that is β -accurate with respect to \mathcal{Q} . If the distinguisher ever fails to find such a query, then it must be that there are no such queries, and our database update algorithm has already learned an accurate hypothesis with respect to the queries of interest! This requires at most T iterations, and so we access the data only $2T$ times using $(\varepsilon_0, 0)$ -differentially private methods (running the given distinguisher, and then checking its answer with the Laplace mechanism). Privacy will therefore follow from our composition theorems.

Algorithm 8 The Iterative Construction (IC) Mechanism. It takes as input a parameter ε_0 , an $(F(\varepsilon_0), \gamma)$ -Private Distinguisher Distinguish for \mathcal{Q} , together with an $T(\alpha)$ -iterative database update algorithm U for \mathcal{Q} .

IC(x, α, ε_0 , Distinguish, U):

```

Let  $D^0 = U(\perp, \cdot, \cdot)$ .
for  $t = 1$  to  $T(\alpha/2)$  do
  Let  $f^{(t)} = \text{Distinguish}(x, D^{t-1})$ 
  Let  $\hat{v}^{(t)} = f^{(t)}(x) + \text{Lap}\left(\frac{1}{\|x\|_1 \varepsilon_0}\right)$ 
  if  $|\hat{v}^{(t)} - f^{(t)}(D^{t-1})| < 3\alpha/4$  then
    Output  $y = D^{t-1}$ .
  else
    Let  $D^t = U(D^{t-1}, f^{(t)}, \hat{v}^{(t)})$ .
  end if
end for
Output  $y = D^{T(\alpha/2)}$ .
```

The analysis of this algorithm just involves checking the technical details of a simple intuition. Privacy will follow because the algorithm is just the composition of $2T(\alpha)$ steps, each of which is $(\varepsilon_0, 0)$ -differentially private. Accuracy follows because we are always outputting the last database in a maximal database update sequence. If the algorithm has not yet formed a maximal Database Update Sequence, then the distinguishing algorithm will find a distinguishing query to add another step to the sequence.

Theorem 5.3. The IC algorithm is $(\varepsilon, 0)$ -differentially private for $\varepsilon_0 \leq \varepsilon/2T(\alpha/2)$. The IC algorithm is (ε, δ) -differentially private for $\varepsilon_0 \leq \frac{\varepsilon}{4\sqrt{T(\alpha/2)\log(1/\delta)}}$.

Proof. The algorithm runs at most $2T(\alpha/2)$ compositions of ε_0 -differentially private algorithms. Recall from Theorem 3.20 that ε_0 differentially private algorithms are $2k\varepsilon_0$ differentially private under $2k$ -fold composition, and are (ε', δ) private for $\varepsilon' = \sqrt{4k\ln(1/\delta')}\varepsilon_0 + 2k\varepsilon_0(e^{\varepsilon_0} - 1)$. Plugging in the stated values for ε_0 proves the claim. \square

Theorem 5.4. Given an $(F(\varepsilon), \gamma)$ -private distinguisher, a parameter ε_0 , and a $T(\alpha)$ -Database Update Algorithm, with probability at least $1 - \beta$, the IC algorithm returns a database y such that: $\max_{f \in \mathcal{Q}} |f(x) - f(y)| \leq \alpha$ for any α such that where:

$$\alpha \geq \max \left[\frac{8 \log(2T(\alpha/2)/\beta)}{\varepsilon_0 \|x\|_1}, 8F(\varepsilon_0) \right]$$

so long as $\gamma \leq \beta/(2T(\alpha/2))$.

Proof. The analysis is straightforward.

Recall that if $Y_i \sim \text{Lap}(1/(\varepsilon\|x\|_1))$, we have: $\Pr[|Y_i| \geq t/(\varepsilon\|x\|_1)] = \exp(-t)$. By a union bound, if $Y_1, \dots, Y_k \sim \text{Lap}(1/(\varepsilon\|x\|_1))$, then $\Pr[\max_i |Y_i| \geq t/(\varepsilon\|x\|_1)] \leq k \exp(-t)$. Therefore, because we make at most $T(\alpha/2)$ draws from $\text{Lap}(1/(\varepsilon_0\|x\|_1))$, except with probability at most $\beta/2$, for all t :

$$|\hat{v}^{(t)} - f^{(t)}(x)| \leq \frac{1}{\varepsilon_0 \|x\|_1} \log \frac{2T(\alpha/2)}{\beta} \leq \frac{\alpha}{8}.$$

Note that by assumption, $\gamma \leq \beta/(2T(\alpha/2))$, so we also have that except with probability $\beta/2$:

$$\begin{aligned} |f^{(t)}(x) - f^{(t)}(D^{t-1})| &\geq \max_{f \in \mathcal{Q}} |f(x) - f(D^{t-1})| - F(\varepsilon_0) \\ &\geq \max_{f \in \mathcal{Q}} |f(x) - f(D^{t-1})| - \frac{\alpha}{8}. \end{aligned}$$

For the rest of the argument, we will condition on both of these events occurring, which is the case except with probability β .

There are two cases. Either a data structure $D' = D^{T(\alpha/2)}$ is output, or data structure $D' = D^t$ for $t < T(\alpha/2)$ is output. First, suppose $D' = D^{T(\alpha/2)}$. Since for all $t < T(\alpha/2)$ it must have been the case that $|\hat{v}^{(t)} - f^{(t)}(D^{t-1})| \geq 3\alpha/4$ and by our conditioning, $|\hat{v}^{(t)} - f^{(t)}(x)| \leq \frac{\alpha}{8}$, we know for all t : $|f^{(t)}(x) - f^{(t)}(D^{t-1})| \geq \alpha/2$. Therefore, the sequence $(D^t, f^{(t)}, \hat{v}^{(t)})$, formed a maximal $(U, x, \mathcal{Q}, \alpha/2, T(\alpha/2))$ -Database Update Sequence (recall Definition 5.3), and we have that $\max_{f \in \mathcal{Q}} |f(x) - f(x')| \leq \alpha/2$ as desired.

Next, suppose $D' = D^{t-1}$ for $t < T(\alpha/2)$. Then it must have been the case that for t , $|\hat{v}^{(t)} - f^{(t)}(D^{t-1})| < 3\alpha/4$. By our conditioning, in

this case it must be that $|f^{(t)}(x) - f^{(t)}(D^{t-1})| < \frac{7\alpha}{8}$, and that therefore by the properties of an $(F(\varepsilon_0), \gamma)$ -distinguisher:

$$\max_{f \in \mathcal{Q}} |f(x) - f(D')| < \frac{7\alpha}{8} + F(\varepsilon_0) \leq \alpha$$

as desired. \square

Note that we can use the exponential mechanism as a private distinguisher: take the domain to be \mathcal{Q} , and let the quality score be: $q(D, f) = |f(D) - f(D^t)|$, which has sensitivity $1/\|x\|_1$. Applying the exponential mechanism utility theorem, we get:

Theorem 5.5. The exponential mechanism is an $(F(\varepsilon), \gamma)$ distinguisher for:

$$F(\varepsilon) = \frac{2}{\|x\|_1 \varepsilon} \left(\log \frac{|\mathcal{Q}|}{\gamma} \right).$$

Therefore, using the exponential mechanism as a distinguisher, Theorem 5.4 gives:

Theorem 5.6. Given a $T(\alpha)$ -Database Update Algorithm and a parameter ε_0 together with the exponential mechanism distinguisher, with probability at least $1 - \beta$, the IC algorithm returns a database y such that: $\max_{f \in \mathcal{Q}} |f(x) - f(y)| \leq \alpha$ where:

$$\alpha \leq \max \left[\frac{8 \log(2T(\alpha/2)/\beta)}{\varepsilon_0 \|x\|_1}, \frac{16}{\|x\|_1 \varepsilon_0} \left(\log \frac{|\mathcal{Q}|}{\gamma} \right) \right]$$

so long as $\gamma \leq \beta/(2T(\alpha/2))$.

Plugging in our values of ε_0 :

Theorem 5.7. Given a $T(\alpha)$ -Database Update Algorithm, together with the exponential mechanism distinguisher, the IC mechanism is ε -differentially private and with probability at least $1 - \beta$, the IC algorithm returns a database y such that: $\max_{f \in \mathcal{Q}} |f(x) - f(y)| \leq \alpha$ where:

$$\alpha \leq \frac{8T(\alpha/2)}{\|x\|_1 \varepsilon} \left(\log \frac{|\mathcal{Q}|}{\gamma} \right)$$

and (ε, δ) -differentially private for:

$$\alpha \leq \frac{16\sqrt{T(\alpha/2) \log(1/\delta)}}{\|x\|_1 \varepsilon} \left(\log \frac{|\mathcal{Q}|}{\gamma} \right)$$

so long as $\gamma \leq \beta/(2T(\alpha/2))$.

Note that in the language of this section, what we proved in Theorem 4.10 was exactly that the multiplicative weights algorithm is a $T(\alpha)$ -Database Update Algorithm for $T(\alpha) = \frac{4 \log |\mathcal{X}|}{\alpha^2}$. Plugging this bound into Theorem 5.7 recovers the bound we got for the online multiplicative weights algorithm. Note that now, however, we can plug in other database update algorithms as well.

5.2.1 Applications: other database update algorithms

Here we give several other database update algorithms. The first works directly from α -nets, and therefore can get non-trivial bounds even for nonlinear queries (unlike multiplicative weights, which only works for linear queries). The second is another database update algorithm for linear queries, but with bounds incomparable to multiplicative weights. (In general, it will yield improved bounds when the dataset has size close to the size of the data universe, whereas multiplicative weights will give better bounds when the dataset is much smaller than the data universe.)

We first discuss the median mechanism, which takes advantage of α -nets. The median mechanism does not operate on databases, but instead on median data structures:

Definition 5.6 (Median Data Structure). A median data structure \mathbf{D} is a collection of databases: $\mathbf{D} \subset \mathbb{N}^{|\mathcal{X}|}$. Any query f can be evaluated on a median data structure as follows: $f(\mathbf{D}) = \text{Median}(\{f(x) : x \in \mathbf{D}\})$.

In words, a median data structure is just a set of databases. To evaluate a query on it, we just evaluate the query on every database in the set, and then return the median value. Note that the answers given by the median data structure need not be consistent with *any* database! However, it will have the useful property that whenever it makes an

error, it will rule out at least half of the data sets in its collection as being inconsistent with the true data set.

The median mechanism is then very simple:

Algorithm 9 The Median Mechanism (MM) Update Rule. It inputs and outputs a median data structure. It is instantiated with an α -net $\mathcal{N}_\alpha(\mathcal{Q})$ for a query class \mathcal{Q} , and its initial state is $\mathbf{D} = \mathcal{N}_\alpha(\mathcal{Q})$

$MM_{\alpha, \mathcal{Q}}(\mathbf{D}^t, f_t, v_t)$:

```

if  $\mathbf{D}^t = \perp$  then
    Output  $\mathbf{D}^0 \leftarrow \mathcal{N}_\alpha(\mathcal{Q})$ .
end if
if  $v_t < f_t(\mathbf{D}^t)$  then
    Output  $\mathbf{D}^{t+1} \leftarrow \mathbf{D}^t \setminus \{x \in \mathbf{D} : f_t(x) \geq f_t(\mathbf{D}^t)\}$ .
else
    Output  $\mathbf{D}^{t+1} \leftarrow \mathbf{D}^t \setminus \{x \in \mathbf{D} : f_t(x) \leq f_t(\mathbf{D}^t)\}$ .
end if

```

The intuition for the median mechanism is as follows. It maintains a set of databases that are consistent with the answers to the distinguishing queries it has seen so far. Whenever it receives a query and answer that differ substantially from the real database, it updates itself to remove all of the databases that are inconsistent with the new information. Because it always chooses its answer as the median database among the set of consistent databases it is maintaining, every update step removes at least half of the consistent databases! Moreover, because the set of databases that it chooses initially is an α -net with respect to \mathcal{Q} , there is always some database that is never removed, because it remains consistent on all queries. This limits how many update rounds the mechanism can perform. How does the median mechanism do?

Theorem 5.8. For any class of queries \mathcal{Q} , The Median Mechanism is a $T(\alpha)$ -database update algorithm for $T(\alpha) = \log |\mathcal{N}_\alpha(\mathcal{Q})|$.

Proof. We must show that any sequence $\{(D^t, f_t, v_t)\}_{t=1,\dots,L}$ with the property that $|f^t(\mathbf{D}^t) - f^t(x)| > \alpha$ and $|v_t - f^t(x)| < \alpha$ cannot have $L > \log |\mathcal{N}_\alpha(\mathcal{Q})|$. First observe that because $\mathbf{D}^0 = \mathcal{N}_\alpha(\mathcal{Q})$ is an α -net

for \mathcal{Q} , by definition there is at least one y such that $y \in \mathbf{D}^t$ for all t (Recall that the update rule is only invoked on queries with error at least α . Since there is guaranteed to be a database y that has error less than α on all queries, it is never removed by an update step). Thus, we can always answer queries with \mathbf{D}^t , and for all t , $|\mathbf{D}^t| \geq 1$. Next observe that for each t , $|\mathbf{D}^t| \leq |\mathbf{D}^{t-1}|/2$. This is because each update step removes at least half of the elements: all of the elements at least as large as, or at most as large as the median element in \mathbf{D}^t with respect to query f_t . Therefore, after L update steps, $|\mathbf{D}^L| \leq 1/2^L \cdot |\mathcal{N}_\alpha(\mathcal{Q})|$. Setting $L > \log |\mathcal{N}_\alpha(\mathcal{Q})|$ gives $|\mathbf{D}^L| < 1$, a contradiction. \square

Remark 5.2. For classes of linear queries \mathcal{Q} , we may refer to the upper bound on $\mathcal{N}_\alpha(\mathcal{Q})$ given in Theorem 4.2 to see that the Median Mechanism is a $T(\alpha)$ -database update algorithm for $T(\alpha) = \log |\mathcal{Q}| \log |\mathcal{X}|/\alpha^2$. This is worse than the bound we gave for the Multiplicative Weights algorithm by a factor of $\log |\mathcal{Q}|$. On the other hand, nothing about the Median Mechanism algorithm is specific to linear queries — it works just as well for any class of queries that admits a small net. We can take advantage of this fact for nonlinear low sensitivity queries.

Note that if we want a mechanism which promises (ε, δ) -privacy for $\delta > 0$, we do not even need a particularly small net. In fact, the trivial net that simply includes every database of size $\|x\|_1$ will be sufficient:

Theorem 5.9. For every class of queries \mathcal{Q} and every $\alpha \geq 0$, there is an α -net for databases of size $\|x\|_1 = n$ of size $\mathcal{N}_\alpha(\mathcal{Q}) \leq |\mathcal{X}|^n$.

Proof. We can simply let $\mathcal{N}_\alpha(\mathcal{Q})$ be the set of all $|\mathcal{X}|^n$ databases y of size $\|y\|_1 = n$. Then, for every x such that $\|x\|_1 = n$, we have $x \in \mathcal{N}_\alpha(\mathcal{Q})$, and so clearly: $\min_{y \in \mathcal{N}_\alpha(\mathcal{Q})} \max_{f \in \mathcal{Q}} |f(x) - f(y)| = 0$. \square

We can use this fact to get query release algorithms for *arbitrary* low sensitivity queries, not just linear queries. Applying Theorem 5.7 to the above bound, we find:

Theorem 5.10. Using the median mechanism, together with the exponential mechanism distinguisher, the IC mechanism is (ε, δ) -differentially private and with probability at least $1 - \beta$, the IC algorithm returns a database y such that: $\max_{f \in \mathcal{Q}} |f(x) - f(y)| \leq \alpha$ where:

$$\alpha \leq \frac{16\sqrt{\log |\mathcal{X}| \log \frac{1}{\delta} \log \left(\frac{2|\mathcal{Q}|n \log |\mathcal{X}|}{\beta} \right)}}{\sqrt{n}\varepsilon},$$

where \mathcal{Q} can be *any* family of sensitivity $1/n$ queries, not necessarily linear.

Proof. This follows simply by combining Theorems 5.8 and 5.9 to find that the Median Mechanism is a $T(\alpha)$ -Database Update Algorithm for $T(\alpha) = n \log |\mathcal{X}|$ for databases of size $\|x\|_1 = n$ for every $\alpha > 0$ and every class of queries \mathcal{Q} . Plugging this into Theorem 5.7 gives the desired bound. \square

Note that this bound is almost as good as we were able to achieve for the special case of linear queries in Theorem 4.15! However, unlike in the case of linear queries, because arbitrary queries may not have α -nets which are significantly smaller than the trivial net used here, we are not able to get nontrivial accuracy guarantees if we want $(\varepsilon, 0)$ -differential privacy.

The next database update algorithm we present is again for linear queries, but achieves incomparable bounds to those of the multiplicative weights database update algorithm. It is based on the *Perceptron* algorithm from online learning (just as multiplicative weights is derived from the *hedge* algorithm from online learning). Since the algorithm is for linear queries, we treat each query $f_t \in \mathcal{Q}$ as being a vector $f_t \in [0, 1]^{|\mathcal{X}|}$. Note that rather than doing a multiplicative update,

Algorithm 10 The Perceptron update rule

Perceptron _{α, \mathcal{Q}} (x^t, f_t, v_t):

If: $x^t = \perp$ **then:** output $x^{t+1} = 0^{|\mathcal{X}|}$
Else if: $f_t(x^t) > v_t$ **then:** output $x^{t+1} = x^t - \frac{\alpha}{|\mathcal{X}|} \cdot f_t$
Else if: $f_t(x^t) \leq v_t$ **then:** output $x^{t+1} = x^t + \frac{\alpha}{|\mathcal{X}|} \cdot f_t$

as in the MW database update algorithm, here we do an additive update. In the analysis, we will see that this database update algorithm has an exponentially worse dependence (as compared to multiplicative weights) on the size of the universe, but a superior dependence on the size of the database. Thus, it will achieve better performance for databases that are large compared to the size of the data universe, and worse performance for databases that are small compared to the size of the data universe.

Theorem 5.11. Perceptron is a $T(\alpha)$ -database update algorithm for:

$$T(\alpha) = \left(\frac{\|x\|_2}{\|x\|_1} \right)^2 \cdot \frac{|\mathcal{X}|}{\alpha^2}.$$

Proof. Unlike for multiplicative weights, it will be more convenient to analyze the Perceptron algorithm without normalizing the database to be a probability distribution, and then prove that it is a $T(\alpha')$ database update algorithm for $T(\alpha') = \frac{\|x\|_2^2 |\mathcal{X}|}{\alpha'^2}$. Plugging in $\alpha' = \alpha \|x\|_1$ will then complete the proof. Recall that since each query f_t is linear, we can view $f_t \in [0, 1]^{|\mathcal{X}|}$ as a vector with the evaluation of $f_t(x)$ being equal to $\langle f_t, x \rangle$.

We must show that any sequence $\{(x^t, f_t, v_t)\}_{t=1,\dots,L}$ with the property that $|f_t(x^t) - f_t(x)| > \alpha'$ and $|v_t - f_t(x)| < \alpha'$ cannot have $L > \frac{\|x\|_2^2 |\mathcal{X}|}{\alpha'^2}$.

We use a potential argument to show that for every $t = 1, 2, \dots, L$, x^{t+1} is significantly closer to x than x^t . Specifically, our potential function is the L_2^2 norm of the database $x - x^t$, defined as

$$\|x\|_2^2 = \sum_{i \in \mathcal{X}} x(i)^2.$$

Observe that $\|x - x^1\|_2^2 = \|x\|_2^2$ since $x^1 = 0$, and $\|x\|_2^2 \geq 0$. Thus it suffices to show that in every step, the potential decreases by $\alpha'^2 / |\mathcal{X}|$. We analyze the case where $f_t(x^t) > v_t$, the analysis for the opposite case will be similar. Let $R^t = x^t - x$. Observe that in this case we have

$$f_t(R^t) = f_t(x^t) - f_t(x) \geq \alpha'.$$

Now we can analyze the drop in potential.

$$\begin{aligned}
\|R^t\|_2^2 - \|R^{t+1}\|_2^2 &= \|R^t\|_2^2 - \|R^t - (\alpha'/|\mathcal{X}|) \cdot f_t\|_2^2 \\
&= \sum_{i \in \mathcal{X}} ((R^t(i))^2 - (R^t(i) - (\alpha'/|\mathcal{X}|) \cdot f_t(i))^2) \\
&= \sum_{i \in \mathcal{X}} \left(\frac{2\alpha'}{|\mathcal{X}|} \cdot R^t(i)f_t(i) - \frac{\alpha'^2}{|\mathcal{X}|^2} f_t(i)^2 \right) \\
&= \frac{2\alpha'}{|\mathcal{X}|} f_t(R^t) - \frac{\alpha'^2}{|\mathcal{X}|^2} \sum_{i \in \mathcal{X}} f_t(i)^2 \\
&\geq \frac{2\alpha'}{|\mathcal{X}|} f_t(R^t) - \frac{\alpha'^2}{|\mathcal{X}|^2} |\mathcal{X}| \\
&\geq \frac{2\alpha'^2}{|\mathcal{X}|} - \frac{\alpha'^2}{|\mathcal{X}|} = \frac{\alpha'^2}{|\mathcal{X}|}.
\end{aligned}$$

This bounds the number of steps by $\|x\|_2^2 |\mathcal{X}| / \alpha'^2$, and completes the proof. \square

We may now plug this bound into Theorem 5.7 to obtain the following bound on the iterative construction mechanism:

Theorem 5.12. Using the perceptron database update algorithm, together with the exponential mechanism distinguisher, the IC mechanism is (ε, δ) -differentially private and with probability at least $1 - \beta$, the IC algorithm returns a database y such that: $\max_{f \in \mathcal{Q}} |f(x) - f(y)| \leq \alpha$ where:

$$\alpha \leq \frac{4\sqrt{4\sqrt{\|x\|_2}} (4|\mathcal{X}| \ln(1/\delta))^{1/4} \sqrt{\frac{\log(2|\mathcal{Q}||\mathcal{X}|\cdot\|x\|_2^2)}{\beta}}}{\sqrt{\epsilon}\|x\|_1},$$

where \mathcal{Q} is a class of linear queries.

If the database x represents the edge set of a graph, for example, we will have $x_i \in [0, 1]$ for all i , and so:

$$\frac{\sqrt{\|x\|_2}}{\|x\|_1} \leq \left(\frac{1}{\|x\|_1} \right)^{3/4}.$$

Therefore, the perceptron database update algorithm will outperform the multiplicative weights database update algorithm on dense graphs.

5.2.2 Iterative construction mechanisms and online algorithms

In this section, we generalize the iterative construction framework to the online setting by using the NumericSparse algorithm. The online multiplicative weights algorithm which saw in the last chapter is an instantiation of this approach. One way of viewing the online algorithm is that the NumericSparse algorithm is serving as the private distinguisher in the IC framework, but that the “hard work” of distinguishing is being foisted upon the unsuspecting user. That is: if the user asks a query that does not serve as a good distinguishing query, this is a good case. We cannot use the database update algorithm to update our hypothesis, but we don’t need to! By definition, the current hypothesis is a good approximation to the private database with respect to this query. On the other hand, if the user asks a query for which our current hypothesis is not a good approximation to the true database, then by definition the user has found a good distinguishing query, and we are again in a good case — we can run the database update algorithm to update our hypothesis!

The idea of this algorithm is very simple. We will use a database update algorithm to publicly maintain a hypothesis database. Every time a query arrives, we will classify it as either a hard query, or an easy query. An easy query is one for which the answer given by the hypothesis database is approximately correct, and no update step is needed: if we know that a given query is easy, we can simply compute its answer on the publicly known hypothesis database rather than on the private database, and incur no privacy loss. If we know that a query is hard, we can compute and release its answer using the Laplace mechanism, and update our hypothesis using the database update algorithm. This way, our total privacy loss is not proportional to the number of queries asked, but instead proportional to the number of *hard* queries asked. Because the database update algorithm guarantees that there will not need to be many update steps, we can be guaranteed that the total privacy loss will be small.

Theorem 5.13. OnlineIC is (ε, δ) -differentially private.

Algorithm 11 The Online Iterative Construction Mechanism parameterized by a $T(\alpha)$ -database update algorithm U . It takes as input a private database x , privacy parameters ε, δ , accuracy parameters α and β , and a stream of queries $\{f_i\}$ that may be chosen adaptively from a class of queries \mathcal{Q} . It outputs a stream of answers $\{a_i\}$.

OnlineIC $_U(x, \{f_i\}, \varepsilon, \delta, \alpha, \beta)$

```

Let  $c \leftarrow T(\alpha)$ ,
if  $\delta = 0$  then
    Let  $T \leftarrow \frac{18c(\log(2|\mathcal{Q}|) + \log(4c/\beta))}{\epsilon\|x\|_1}$ 
else
    Let  $T \leftarrow \frac{(2+32\sqrt{2})\sqrt{c \log \frac{2}{\delta}}(\log k + \log \frac{4c}{\beta})}{\epsilon\|x\|_1}$ 
end if
Initialize NumericSparse( $x, \{f'_i\}, T, c, \varepsilon, \delta$ ) with a stream of queries
 $\{f'_i\}$ , outputting a stream of answers  $a'_i$ .
Let  $t \leftarrow 0$ ,  $D^0 \in x$  be such that  $D_i^0 = 1/|\mathcal{X}|$  for all  $i \in [\mathcal{X}]$ .
for each query  $f_i$  do
    Let  $f'_{2i-1}(\cdot) = f_i(\cdot) - f_i(D^t)$ .
    Let  $f'_{2i}(\cdot) = f_i(D^t) - f_i(\cdot)$ 
    if  $a'_{2i-1} = \perp$  and  $a'_{2i} = \perp$  then
        Let  $a_i = f_i(D^t)$ 
    else
        if  $a'_{2i-1} \in \mathbb{R}$  then
            Let  $a_i = f_i(D^t) + a'_{2i-1}$ 
        else
            Let  $a_i = f_i(D^t) - a'_{2i}$ 
        end if
        Let  $D^{t+1} = U(D^t, f_i, a_i)$ 
        Let  $t \leftarrow t + 1$ .
    end if
end for

```

Proof. This follows directly from the privacy analysis of NumericSparse, because the OnlineIC algorithm accesses the database only through NumericSparse. \square

Theorem 5.14. For $\delta = 0$, With probability at least $1 - \beta$, for all queries f_i , OnlineIC returns an answer a_i such that $|f_i(x) - a_i| \leq 3\alpha$ for any α such that:

$$\alpha \geq \frac{9T(\alpha)(\log(2|\mathcal{Q}|) + \log(4T(\alpha)/\beta))}{\epsilon\|x\|_1}.$$

Proof. Recall that by Theorem 3.28 that given k queries and a maximum number of above-threshold queries of c , Sparse Vector is (α, β) -accurate for:

$$\alpha = \frac{9c(\log k + \log(4c/\beta))}{\epsilon\|x\|_1}.$$

Here, we have $c = T(\alpha)$ and $k = 2|\mathcal{Q}|$. Note that we have set the threshold $T = 2\alpha$ in the algorithm. First let us assume that the sparse vector algorithm does not halt prematurely. In this case, by the utility theorem, except with probability at most β , we have for all i such that $a_i = f_i(D^t)$: $|f_i(D) - f_i(D^t)| \leq T + \alpha = 3\alpha$, as we wanted. Additionally, for all i such that $a_i = a'_{2i-1}$ or $a_i = a'_{2i}$, we have $|f_i(D) - a'_i| \leq \alpha$.

Note that we also have for all i such that $a_i = a'_{2i-1}$ or $a_i = a'_{2i}$: $|f_i(D) - f_i(D')| \geq T - \alpha = \alpha$, since $T = 2\alpha$. Therefore, f_i, a_i form a valid step in a database update sequence. Therefore, there can be at most $c = T(\alpha)$ such update steps, and so the Sparse vector algorithm does not halt prematurely. \square

Similarly, we can prove a corresponding bound for (ε, δ) -privacy.

Theorem 5.15. For $\delta > 0$, With probability at least $1 - \beta$, for all queries f_i , OnlineIC returns an answer a_i such that $|f_i(x) - a_i| \leq 3\alpha$ for any α such that:

$$\alpha \geq \frac{(\sqrt{512} + 1)(\ln(2|\mathcal{Q}|) + \ln \frac{4T(\alpha)}{\beta})\sqrt{T(\alpha) \ln \frac{2}{\delta}}}{\epsilon\|x\|_1}$$

We can recover the bounds we proved for online multiplicative weights by recalling that the MW database update algorithm is a $T(\alpha)$ -database update algorithm for $T(\alpha) = \frac{4 \log |\mathcal{X}|}{\alpha^2}$. More generally, we have that *any* algorithm in the iterative construction framework can be converted into an algorithm which works in the interactive setting without loss in accuracy. (i.e., we could equally well plug in

the median mechanism database update algorithm or the Perceptron database update algorithm, or any other). Tantalizingly, this means that (at least in the iterative construction framework), there is no gap in the accuracy achievable in the online vs. the offline query release models, despite the fact that the online model seems like it should be more difficult.

5.3 Connections

5.3.1 Iterative construction mechanism and α -nets

The Iterative Construction mechanism is implemented differently than the Net mechanism, but at its heart, its analysis is still based on the existence of small α -nets for the queries C . This connection is explicit for the median mechanism, which is parameterized by a net, but it holds for all database update algorithms. Note that the database output by the iterative database construction algorithm is entirely determined by the at most T functions $f_1, \dots, f_T \in \mathcal{Q}$ fed into it, as selected by the distinguisher while the algorithm is running. Each of these functions can be indexed by at most $\log |\mathcal{Q}|$ bits, and so every database output by the mechanism can be described using only $T \log |\mathcal{Q}|$ bits. In other words, the IC algorithm itself describes an α -net for \mathcal{Q} of size at most $\mathcal{N}_\alpha(\mathcal{Q}) \leq |\mathcal{Q}|^T$. To obtain error α using the Multiplicative Weights algorithm as an iterative database constructor, it suffices by Theorem 4.10 to take $T = 4 \log |\mathcal{X}|/\alpha^2$, which gives us $\mathcal{N}_\alpha(\mathcal{Q}) \leq |\mathcal{Q}|^{4 \log |\mathcal{X}|/\alpha^2} = |\mathcal{X}|^{4 \log |\mathcal{Q}|/\alpha^2}$. Note that up to the factor of 4 in the exponent, this is exactly the bound we gave using a different α -net in Theorem 4.2! There, we constructed an α -net by considering all collections of $\log |\mathcal{Q}|/\alpha^2$ data points, each of which could be indexed by $\log |\mathcal{X}|$ bits. Here, we considered all collections of $\log |\mathcal{X}|/\alpha^2$ functions in \mathcal{Q} , each of which could be indexed by $\log |\mathcal{Q}|$ bits. Both ways, we got α -nets of the same size! Indeed, we could just as well run the Net mechanism using the α -net defined by the IC mechanism, to obtain the same utility bounds. In some sense, one net is the “dual” of the other: one is constructed of databases, the other is constructed of queries, yet both nets are of the same size. We will see the same phenomenon in the

“boosting for queries” algorithm in the next section — it too answers a large number of linear queries using a data structure that is entirely determined by a small “net” of queries.

5.3.2 Agnostic learning

One way of viewing what the IC mechanism is doing is that it is reducing the seemingly (information theoretically) more difficult problem of *query release* to the easier problem of *query distinguishing* or *learning*. Recall that the distinguishing problem is to find the query $f \in \mathcal{Q}$ which varies the most between two databases x and y . Recall that in *learning*, the learner is given a collection of labeled examples $(x_1, y_1), \dots, (x_m, y_m) \in \mathcal{X} \times \{0, 1\}$, where $y_i \in \{0, 1\}$ is the *label* of x_i . If we view x as representing the *positive examples* in some large data set, and y as representing the *negative examples* in the same data set, then we can see that the problem of distinguishing is exactly the problem of *agnostic learning*. That is, a distinguisher finds the query that best labels the positive examples, even when there is no query in the class that is guaranteed to perfectly label them (Note that in this setting, the same example can appear with both a positive and a negative label — so the reduction still makes sense even when x and y are not disjoint). Intuitively, learning should be an information-theoretically easier problem than query release. The query release problem requires that we release the approximate value of every query f in some class \mathcal{Q} , evaluated on the database. In contrast, the agnostic learning problem asks only that we return the evaluation and identity of a single query: the query that best labels the dataset. It is clear that information theoretically, the learning problem is no harder than the query release problem. If we can solve the query release problem on databases x and y , then we can solve the distinguishing problem without any further access to the true private dataset, merely by checking the approximate evaluations of every query $f \in \mathcal{Q}$ on x and y that are made available to us with our query release algorithm. What we have shown in this section is that the reverse is true as well: given access to a private distinguishing or agnostic learning algorithm, we can solve the query release problem by making a small (i.e., only $\log |\mathcal{X}|/\alpha^2$) number of calls to the

private distinguishing algorithm, *with no further access to the private dataset.*

What are the implications of this? It tells us that up to small factors, the information complexity of agnostic learning is equal to the information complexity of query release. Computationally, the reduction is only as efficient as our database update algorithm, which, depending on our setting and algorithm, may or may not be efficient. But it tells us that any sort of information theoretic bound we may prove for the one problem can be ported over to the other problem, and vice versa. For example, most of the algorithms that we have seen (and most of the algorithms that we know about!) ultimately access the dataset by making linear queries via the Laplace mechanism. It turns out that any such algorithm can be seen as operating within the so-called *statistical query* model of data access, defined by Kearns in the context of machine learning. But agnostic learning is very hard in the statistical query model: even ignoring computational considerations, there is no algorithm which can make only a polynomial number of queries to the dataset and agnostically learn conjunctions to subconstant error. For query release this means that, *in the statistical query model*, there is no algorithm for *releasing* conjunctions (i.e., contingency tables) that runs in time polynomial in $1/\alpha$, where α is the desired accuracy level. If there is a privacy preserving query release algorithm with this run-time guarantee, it must operate outside of the SQ model, and therefore look very different from the currently known algorithms.

Because privacy guarantees compose linearly, this also tells us that (up to the possible factor of $\log|\mathcal{X}|/\alpha^2$) we should not expect to be able to privately learn to significantly higher accuracy than we can privately perform query release, and vice versa: an accurate algorithm for the one problem automatically gives us an accurate algorithm for the other.

5.3.3 A game theoretic view of query release

In this section, we take a brief sojourn into game theory to interpret some of the query release algorithms we have (and will see). Let us consider an interaction between two adversarial players, Alice and Bob.

Alice has some set of actions she might take, \mathcal{A} , and Bob has a set of actions \mathcal{B} . The game is played as follows: simultaneously, Alice picks some action $a \in \mathcal{A}$ (possibly at random), and Bob picks some action $b \in \mathcal{B}$ (possibly at random). Alice experiences a cost $c(a, b) \in [-1, 1]$. Alice wishes to play so as to minimize this cost, and since he is adversarial, Bob wishes to play so as to *maximize* this cost. This is what is called a *zero sum game*.

So how should Alice play? First, we consider an easier question. Suppose we handicap Alice and require that she announce her randomized strategy to Bob before she play it, and allow Bob to respond optimally using this information? If Alice announces that she will draw some action $a \in \mathcal{A}$ according to a probability distribution \mathcal{D}_A , then Bob will respond optimally so as to maximize Alice's expected cost. That is, Bob will play:

$$b^* = \arg \max_{b \in \mathcal{B}} \mathbb{E}_{a \sim \mathcal{D}_A} [c(a, b)].$$

Hence, once Alice announces her strategy, she knows what her cost will be, since Bob will be able to respond optimally. Therefore, Alice will wish to play a distribution over actions which *minimizes her cost once Bob responds*. That is, Alice will wish to play the distribution \mathcal{D}_A defined as:

$$\mathcal{D}_A = \arg \min_{\mathcal{D} \in \Delta \mathcal{A}} \max_{b \in \mathcal{B}} \mathbb{E}_{a \sim \mathcal{D}} [c(a, b)].$$

If she plays \mathcal{D}_A (and Bob responds optimally), Alice will experience the lowest possible cost that she can guarantee, with the handicap that she must announce her strategy ahead of time. Such a strategy for Alice is called a *min-max* strategy. Let us call the cost that Alice achieves when playing a min-max strategy Alice's *value* for the game, denoted v^A :

$$v^A = \min_{\mathcal{D} \in \Delta \mathcal{A}} \max_{b \in \mathcal{B}} \mathbb{E}_{a \sim \mathcal{D}} [c(a, b)].$$

We can similarly ask what Bob should play if we instead place *him* at the disadvantage and force him to announce his strategy first to Alice. If he does this, he will play the distribution \mathcal{D}_B over actions $b \in \mathcal{B}$ that *maximizes* Alice's expected cost when Alice responds optimally. We call such a strategy \mathcal{D}_B for Bob a *max-min* strategy. We can define

Bob's value for the game, v^B , as the maximum cost he can ensure by any strategy he might announce:

$$v^B = \max_{\mathcal{D} \in \Delta \mathcal{B}} \min_{a \in \mathcal{A}} \mathbb{E}_{b \sim \mathcal{D}} [c(a, b)].$$

Clearly, $v^B \leq v^A$, since announcing one's strategy is only a handicap.

One of the foundational results of game theory is Von-Neumann's min-max Theorem, which states that in any zero sum game, $v^A = v^B$.² In other words, there is no disadvantage to "going first" in a zero sum game, and if players play optimally, we can predict exactly Alice's cost: it will be $v^A = v^B \equiv v$, which we refer to as the value of the game.

Definition 5.7. In a zero sum game defined by action sets \mathcal{A}, \mathcal{B} and a cost function $c : \mathcal{A} \times \mathcal{B} \rightarrow [-1, 1]$, let v be the value of the game. An α -approximate min-max strategy is a distribution \mathcal{D}_A such that:

$$\max_{b \in \mathcal{B}} \mathbb{E}_{a \sim \mathcal{D}_A} [c(a, b)] \leq v + \alpha$$

Similarly, an α -approximate max-min strategy is a distribution \mathcal{D}_B such that:

$$\min_{a \in \mathcal{A}} \mathbb{E}_{b \sim \mathcal{D}_B} [c(a, b)] \geq v - \alpha$$

If \mathcal{D}_A and \mathcal{D}_B are both α -approximate min-max and max-min strategies respectively, then we say that the pair $(\mathcal{D}_A, \mathcal{D}_B)$ is an α -approximate Nash equilibrium of the zero sum game.

So how does this relate to query release?

Consider a particular zero sum-game tailored to the problem of releasing a set of linear queries \mathcal{Q} over a data universe \mathcal{X} . First, assume without loss of generality that for every $f \in \mathcal{Q}$, there is a query $\hat{f} \in \mathcal{Q}$ such that $\hat{f} = 1 - f$ (i.e., for each $\chi \in \mathcal{X}$, $\hat{f}(\chi) = 1 - f(\chi)$). Define Alice's action set to be $\mathcal{A} = \mathcal{X}$ and define Bob's action set to be $\mathcal{B} = \mathcal{Q}$. We will refer to Alice as the *database player*, and to Bob as the *query player*. Finally, fixing a true private database x normalized to be a probability distribution (i.e., $\|x\|_1 = 1$), define the cost function $c : \mathcal{A} \times \mathcal{B} \rightarrow [-1, 1]$

²Von Neumann is quoted as saying "As far as I can see, there could be no theory of games ... without that theorem ... I thought there was nothing worth publishing until the Minimax Theorem was proved" [10].

to be: $c(\chi, f) = f(\chi) - f(x)$. Let us call this game the “Query Release Game.”

We begin with a simple observation:

Proposition 5.16. The value of the query release game is $v = 0$.

Proof. We first show that $v^A = v \leq 0$. Consider what happens if we let the database player’s strategy correspond to the true database: $\mathcal{D}_A = x$. Then we have:

$$\begin{aligned} v^A &\leq \max_{f \in \mathcal{B}} \mathbb{E}_{\chi \sim \mathcal{D}_A}[c(\chi, f)] \\ &= \max_{f \in \mathcal{B}} \sum_{i=1}^{|\mathcal{X}|} f(\chi_i) \cdot x_i - f(x) \\ &= f(x) - f(x) \\ &= 0. \end{aligned}$$

Next we observe that $v = v^B \geq 0$. For point of contradiction, assume that $v < 0$. In other words, that there exists a distribution \mathcal{D}_A such that *for all* $f \in \mathcal{Q}$

$$\mathbb{E}_{\chi \sim \mathcal{D}_A} c(\chi, f) < 0.$$

Here, we simply note that by definition, if $\mathbb{E}_{\chi \sim \mathcal{D}_A} c(\chi, f) = c < 0$ then $\mathbb{E}_{\chi \sim \mathcal{D}_A} c(\chi, \hat{f}) = -c > 0$, which is a contradiction since $\hat{f} \in \mathcal{Q}$. \square

What we have established implies that for any distribution \mathcal{D}_A that is an α -approximate min-max strategy for the database player, we have that for all queries $f \in \mathcal{Q}$: $|\mathbb{E}_{\chi \sim \mathcal{D}_A} f(\chi) - f(x)| \leq \alpha$. In other words, the distribution \mathcal{D}_A can be viewed as a synthetic database that answers every query in \mathcal{Q} with α -accuracy.

How about for nonlinear queries? We can repeat the same argument above if we change the query release game slightly. Rather than letting the database player have strategies corresponding to universe elements $\chi \in \mathcal{X}$, we let the database player have strategies corresponding to *databases* themselves! Then, $c(f, y) = |f(x) - f(y)|$. Its not hard to see that this game still has value 0 and that α -approximate min-max strategies correspond to synthetic data which give α -accurate answers to queries in \mathcal{Q} .

So how do we compute approximate min-max strategies in zero sum games? There are many ways! It is well known that if Alice plays the game repeatedly, updating her distribution on actions using an online-learning algorithm with a no-regret guarantee (defined in Section 11.2), and Bob responds at each round with an approximately-cost-maximizing response, then Alice’s distribution will quickly converge to an approximate min-max strategy. Multiplicative weights is such an algorithm, and one way of understanding the multiplicative weights mechanism is as a strategy for Alice to play in the query release game defined in this section. (The private distinguisher is playing the role of Bob here, picking at each round the query that corresponds to approximately maximizing Alice’s cost). The median mechanism is another such algorithm, for the game in which Alice’s strategies correspond to databases, rather than universe elements, and so is also computing an approximate min-max solution to the query release game.

However, there are other ways to compute approximate equilibria as well! For example, *Bob*, the query player, could play the game using a no-regret learning algorithm (such as multiplicative weights), and *Alice* could repeatedly respond at each round with an approximately-cost-minimizing database! In this case, the *average* over the databases that Alice plays over the course of this experiment will converge to an approximate min-max solution as well. This is exactly what is being done in Section 6, in which the private base-sanitizer plays the role of Alice, at each round playing an approximately cost-minimizing database given Bob’s distribution over queries.

In fact, a third way of computing an approximate equilibrium of a zero-sum game is to have *both* Alice and Bob play according to no-regret learning algorithms. We won’t cover this approach here, but this approach has applications in guaranteeing privacy not just to the database, but also to the set of queries being asked, and to privately solving certain types of linear programs.

5.4 Bibliographical notes

The Iterative Construction Mechanism abstraction (together with the perception based database update algorithm) was formalized by

Gupta et al. [39], generalizing the median mechanism of Roth and Roughgarden [74] (initially presented as an online algorithm), the online private multiplicative weights mechanism of Hardt and Rothblum [44], and its offline variant of Gupta et al. [38]; see also Hardt et al. [41]. All these algorithm can be seen to be instantiations. The connection between query release and agnostic learning was observed in [38]. The observation that the median mechanism, when analyzed using the composition theorems of Dwork et al. [32] for (ε, δ) privacy, can be used to answer arbitrary low sensitivity queries is due to Hardt and Rothblum. The game theoretic view of query release, along with its applications to analyst privacy, is due to Hsu, Roth, and Ullman [48].

6

Boosting for Queries

In the previous sections, we have focused on the problem of private query release in which we insist on bounding the worst-case error over all queries. Would our problem be easier if we instead asked only for low error on average, given some distribution over the queries? In this section, we see that the answer is no: given a mechanism which is able to solve the query release problem with low average error given any distribution on queries, we can “boost” it into a mechanism which solves the query release problem to worst-case error. This both sheds light on the difficulty of private query release, and gives us a new tool for designing private query release algorithms.

Boosting is a general and widely used method for improving the accuracy of learning algorithms. Given a set of labeled training examples

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\},$$

where each x_i is drawn from an underlying distribution \mathcal{D} on a universe \mathcal{U} , and each $y_i \in \{+1, -1\}$, a learning algorithm produces a hypothesis $h : \mathcal{U} \rightarrow \{+1, -1\}$. Ideally, h will not just “describe” the labeling on the given samples, but will also *generalize*, providing a reasonably accurate method of classifying other elements drawn from the underlying

distribution. The goal of boosting is to convert a weak *base* learner, which produces a hypothesis that may do just a little better than random guessing, into a strong learner, which yields a very accurate predictor for samples drawn according to \mathcal{D} . Many boosting algorithms share the following basic structure. First, an initial (typically uniform) probability distribution is imposed on the sample set. Computation then proceeds in rounds. In each round t :

1. The base learner is run on the current distribution, denoted \mathcal{D}_t , producing a classification hypothesis h_t ; and
2. The hypotheses h_1, \dots, h_t are used to re-weight the samples, defining a new distribution \mathcal{D}_{t+1} .

The process halts either after a predetermined number of rounds or when an appropriate combining of the hypotheses is determined to be sufficiently accurate. Thus, given a base learner, the design decisions for a boosting algorithm are (1) are how to modify the probability distribution from one round to the next, and (2) how to combine the hypotheses $\{h_t\}_{t=1,\dots,T}$ to form a final output hypothesis.

In this section we will use boosting on queries — that is, for the purposes of the boosting algorithm the universe \mathcal{U} is a set of queries \mathcal{Q} — to obtain an offline algorithm for answering large numbers of arbitrary low-sensitivity queries. This algorithm requires less space than the median mechanism, and, depending on the base learner, is potentially more time efficient as well.

The algorithm revolves around a somewhat magical fact (Lemma 6.5): if we can find a synopsis that provides accurate answers on a few selected queries, then in fact this synopsis provides accurate answers on *most* queries! We apply this fact to the base learner, which samples from a distribution on \mathcal{Q} and produces as output a “weak” synopsis that yields “good” answers for a majority of the weight in \mathcal{Q} , boosting, in a differentially private fashion, to obtain a synopsis that is good for all of \mathcal{Q} .

Although the boosting is performed over the queries, the privacy is still for the rows of the database. The privacy challenge in boosting for queries comes from the fact that each row in the database affects the

answers to all the queries. This will manifest in the reweighting of the queries: adjacent databases could cause radically different reweightings, which will be observable in the generated h_t that, collectively, will form the synopsis.

The running time of the boosting procedure depends quasi-linearly on the number $|\mathcal{Q}|$ of queries and on the running time of the base synopsis generator, independent of the data universe size $|\mathcal{X}|$. This yields a new avenue for constructing efficient and accurate privacy-preserving mechanisms, analogous to the approach enabled by boosting in the machine learning literature: an algorithm designer can tackle the (potentially much easier) task of constructing a weak privacy-preserving base synopsis generator, and automatically obtain a stronger mechanism.

6.1 The boosting for queries algorithm

We will use the *row representation* for databases, outlined in Section 2, where we think of the database as a multiset of rows, or elements of \mathcal{X} . Fix a database size n , a data universe \mathcal{X} , and a query set $\mathcal{Q} = \{q : \mathcal{X}^* \rightarrow \mathbb{R}\}$ of real-valued queries of sensitivity at most ρ .

We assume the existence of a *base synopsis generator* (in Section 6.2 we will see how to construct these). The property we will need of the base generator, formulated next, is that, for any distribution \mathcal{D} on the query set \mathcal{Q} , the output of base generator can be used for computing accurate answers for a *large fraction* of the queries, where the “large fraction” is defined in terms of the weights given by \mathcal{D} . The base generator is parameterized by k , the number of queries to be sampled; λ , an accuracy requirement for its outputs; η , a measurement of “large” describing what we mean by a large fraction of the queries, and β , a failure probability.

Definition 6.1 $((k, \lambda, \eta, \beta)$ -base synopsis generator). For a fixed database size n , data universe \mathcal{X} and query set \mathcal{Q} , consider a synopsis generator \mathcal{M} , that samples k queries independently from a distribution \mathcal{D} on \mathcal{Q} and outputs a synopsis. We say that \mathcal{M} is a $((k, \lambda, \eta, \beta)$ -base synopsis generator if for any distribution \mathcal{D} on \mathcal{Q} , with all but β probability

over the coin flips of \mathcal{M} , the synopsis \mathcal{S} that \mathcal{M} outputs is λ -accurate for a $(1/2 + \eta)$ -fraction of the mass of \mathcal{Q} as weighted by \mathcal{D} :

$$\Pr_{q \sim \mathcal{D}} [|q(\mathcal{S}) - q(x)| \leq \lambda] \geq 1/2 + \eta. \quad (6.1)$$

The query-boosting algorithm can be used for any class of queries and any differentially private base synopsis generator. The running time is inherited from the base synopsis generator. The booster invests additional time that is quasi-linear in $|\mathcal{Q}|$, and in particular its running time does not depend directly on the size of the data universe.

To specify the boosting algorithm we will need to specify a stopping condition, an aggregation mechanism, and an algorithm for updating the current distribution on \mathcal{Q} .

Stopping Condition. We will run the algorithm for a fixed number T of rounds — this will be our stopping condition. T will be selected so as to ensure sufficient accuracy (with very high probability); as we will see, $\log |\mathcal{Q}|/\eta^2$ rounds will suffice.

Updating the Distribution. Although the distributions are never directly revealed in the outputs, the base synopses $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_T$ are revealed, and each \mathcal{A}_i can in principle leak information about the queries chosen, from \mathcal{D}_i , in constructing \mathcal{A}_i . We therefore need to constrain the max-divergence between the probability distributions obtained on neighboring databases. This is technically challenging because, given \mathcal{A}_i , the database is very heavily involved in constructing \mathcal{D}_{i+1} .

The initial distribution, \mathcal{D}_1 , will be uniform over \mathcal{Q} . A standard method for updating \mathcal{D}_t is to increase the weight of poorly handled elements, in our case, queries for which $|q(x) - q(\mathcal{A}_t)| > \lambda$, by a fixed factor, say, e , and decrease the weight of well-handled elements by the same factor. (The weights are then normalized so as to sum to 1.) To get a feel for the difficulty, let $x = y \cup \{\xi\}$, and suppose that all queries q are handled well by \mathcal{A}_t when the database is y , but the addition of ξ causes this to fail for, say, a $1/10$ fraction of the queries; that is, $|q(y) - q(\mathcal{A}_t)| \leq \lambda$ for all queries q , but $|q(x) - q(\mathcal{A}_t)| > \lambda$ for some $|\mathcal{Q}|/10$ queries. Note that, since \mathcal{A}_t “does well” on $9/10$ of the queries even

when the database is x , it could be returned from the base sanitizer no matter which of x, y is the true data set. Our concern is with the effects of the updating: when the database is y all queries are well handled and there is no reweighting (after normalization), but when the database is x there is a reweighting: one tenth of the queries have their weights increased, the remaining nine tenths have their weights decreased. This difference in reweighting may be detected in the next iteration via \mathcal{A}_{t+1} , which is observable, and which will be built from samples drawn from rather different distributions depending on whether the database is x or y .

For example, suppose we start from the uniform distribution \mathcal{D}_1 . Then $\mathcal{D}_2^{(y)} = \mathcal{D}_1^{(y)}$, where by $\mathcal{D}_i^{(z)}$ we mean the distribution at round i when the database is z . This is because the weight of every query is decreased by a factor of e , which disappears in the normalization. So each $q \in \mathcal{Q}$ is assigned weight $1/|\mathcal{Q}|$ in $\mathcal{D}_2^{(y)}$. In contrast, when the database is x the “unhappy” queries have normalized weight

$$\frac{\frac{e}{|\mathcal{Q}|}}{\frac{9}{10} \frac{1}{|\mathcal{Q}|} \frac{1}{e} + \frac{1}{10} \frac{e}{|\mathcal{Q}|}}.$$

Consider any such unhappy query q . The ratio $\mathcal{D}_2^{(x)}(q)/\mathcal{D}_2^{(y)}(q)$ is given by

$$\begin{aligned} \frac{\mathcal{D}_2^{(x)}(q)}{\mathcal{D}_2^{(y)}(q)} &= \frac{\frac{e}{|\mathcal{Q}|}}{\frac{9}{10} \frac{1}{|\mathcal{Q}|} \frac{1}{e} + \frac{1}{10} \frac{e}{|\mathcal{Q}|}} \\ &= \frac{10}{1 + \frac{9}{e^2}} \stackrel{\text{def}}{=} F \approx 4.5085. \end{aligned}$$

Now, $\ln F \approx 1.506$, and even though the choice of queries used in round 2 by the base generator are not explicitly made public, they may be detectable from the resulting \mathcal{A}_2 , which is made public. Thus, there is a potential privacy loss of up to 1.506 per query (of course, we expect cancellations; we are simply trying to explain the source of the difficulty). This is partially addressed by ensuring that the number of samples used by the base generator is relatively small, although we still have the problem that, over multiple iterations, the distributions \mathcal{D}_t may evolve very differently even on neighboring databases.

The solution will be to attenuate the re-weighting procedure. Instead of always using a fixed ratio either for increasing the weight (when the answer is “accurate”) or decreasing it (when it is not), we set separate thresholds for “accuracy” (λ) and “inaccuracy” ($\lambda + \mu$, for an appropriately chosen μ that scales with the *bit size* of the output of the base generator; see Lemma 6.5 below). Queries for which the error is below or above these thresholds have their weight decreased or increased, respectively, by a factor of e . For queries whose error lies between these two thresholds, we scale the natural logarithm of the weight change linearly: $1 - 2(|q(x) - q(A_t)| - \lambda)/\mu$, so queries with errors of magnitude exceeding $\lambda + \mu/2$ increase in weight, and those with errors of magnitude less than $\lambda + \mu/2$ decrease in weight.

The attenuated scaling reduces the effect of any individual on the re-weighting of any query. This is because an individual can only affect the true answer to a query — and thus also the accuracy of the base synopsis generator’s output $q(A_t)$ — by a small amount, and the attenuation divides this amount by a parameter μ which will be chosen to compensate for the kT samples chosen (total) from the T distributions obtained over the course of the execution of the boosting algorithm. This helps to ensure privacy. Intuitively, we view each of these kT samples as a “mini-mechanism.” We first bound the privacy loss of sampling at any round (Claim 6.4) and then bound the cumulative loss via the composition theorem.

The larger the gap (μ) between the thresholds for “accurate” and “inaccurate,” the smaller the effect of each individual on a query’s weight can be. This means that larger gaps are better for privacy. For accuracy, however, large gaps are bad. If the inaccuracy threshold is large, we can only guarantee that queries for which the base synopsis generator is very inaccurate have their weight substantially increased during re-weighting. This degrades the accuracy guarantee of the boosting algorithm: the errors are roughly equal to the “inaccuracy” threshold ($\lambda + \mu$).

Aggregation. For $t \in [T]$ we will run the base generator to obtain a synopsis A_t . The synopses will be aggregated by taking the median: given A_1, \dots, A_T , the quantity $q(x)$ is estimated by taking the T

approximate values for $q(x)$ computed using each of the \mathcal{A}_i , and then computing their median. With this aggregation method we can show accuracy for query q by arguing that a majority of the \mathcal{A}_i , $1 \leq i \leq T$ provide $\lambda + \mu$ accuracy (or better) for q . This implies that the median value of the T approximations to $q(x)$ will be within $\lambda + \mu$ of the true value.

Notation.

1. Throughout the algorithm's operation, we keep track of several variables (explicitly or implicitly). Variables indexed by $q \in \mathcal{Q}$ hold information pertaining to query q in the query set. Variables indexed by $t \in [T]$, usually computed in round t , will be used to construct the distribution \mathcal{D}_{t+1} used for sampling in time period $t + 1$.
2. For a predicate P we use $[[P]]$ to denote 1 if the predicate is true and 0 if it is false.
3. There is a final tuning parameter α used in the algorithm. It will be chosen (see Corollary 6.3 below) to have value

$$\alpha = \alpha(\eta) = (1/2) \ln \left(\frac{1+2\eta}{1-2\eta} \right).$$

The algorithm appears in Figure 6.1. The quantity $u_{t,q}$ in Step 2(2b) is the new, un-normalized, weight of the query. For the moment, let us set $\alpha = 1$ (just so that we can ignore any α factors). Letting $a_{j,q}$ be the natural logarithm of the weight change in round j , $1 \leq j \leq t$, the new weight is given by:

$$u_{t,q} \leftarrow \exp \left(- \sum_{j=1}^t a_{j,q} \right).$$

Thus, at the end of the previous step the un-normalized weight was $u_{t-1,q} = \exp(-\sum_{j=1}^{t-1} a_{j,q})$ and the update corresponds to multiplication by $e^{-a_{j,t}}$. When the sum $\sum_{j=1}^t a_{j,q}$ is large, the weight is small. Every time a synopsis gives a very good approximation to $q(x)$, we add 1 to this sum; if the approximation is only moderately good (between λ and

Boosting for Queries($k, \lambda, \eta, \rho, \mu, T$)
Given: database $x \in \mathcal{X}^n$, query set \mathcal{Q} , where each $q \in \mathcal{Q}$ is a function $q : X^n \rightarrow \mathbb{R}$ with sensitivity at most ρ .
Initialize \mathcal{D}_1 to be the uniform distribution over \mathcal{Q} .
For $t = 1, \dots, T$:

1. Sample a sequence $S_t \subseteq \mathcal{Q}$ of k samples chosen independently and at random from \mathcal{D}_t .
Run the base synopsis generator to compute a synopsis $\mathcal{A}_t : \mathcal{Q} \rightarrow \mathbb{R}$ that is w.h.p. accurate for at least $1/2 + \eta$ of the mass of \mathcal{D}_t .
2. Reweight the queries. For each $q \in \mathcal{Q}$:
 - (a) If \mathcal{A}_t is λ -accurate, then $a_{t,q} \leftarrow 1$
If \mathcal{A}_t is $(\lambda + \mu)$ -inaccurate, then $a_{t,q} \leftarrow -1$
Otherwise, let $d_{q,t} = |q(x) - \mathcal{A}_t(q)|$ be the error of \mathcal{A}_t (between λ and $\lambda + \mu$) on q :

$$a_{t,q} \leftarrow 1 - 2(d_{q,t} - \lambda)/\mu$$
 - (b) $u_{t,q} \leftarrow \exp(-\alpha \cdot \sum_{j=1}^t a_{j,q})$, where $\alpha = (1/2) \ln((1+2\eta)/(1-2\eta))$.
3. Renormalize:

$$Z_t \leftarrow \sum_{q \in \mathcal{Q}} u_{t,q}$$

$$D_{t+1}[q] = u_{t,q}/Z_t$$

Output the final answer data structure $\mathcal{A} = (\mathcal{A}_1, \dots, \mathcal{A}_T)$. For $q \in \mathcal{Q}$:

$$\mathcal{A}(q) = \text{median}\{\mathcal{A}_1(q), \dots, \mathcal{A}_T(q)\}$$

Figure 6.1: Boosting for queries.

$\lambda + \mu/2$), we add a positive amount, but less than 1. Conversely, when the synopsis is very bad (worse than $\lambda + \mu$ accuracy), we subtract 1; when it is barely acceptable (between $\lambda + \mu/2$ and $\lambda + \mu$), we subtract a smaller amount.

In the theorem below we see an inverse relationship between privacy loss due to sampling, captured by $\varepsilon_{\text{sample}}$, and the gap μ between the thresholds for accurate and inaccurate.

Theorem 6.1. Let \mathcal{Q} be a query family with sensitivity at most ρ . For an appropriate setting of parameters, and with $T = \log |\mathcal{Q}|/\eta^2$ rounds, the algorithm of Figure 6.1 is an accurate and differentially private query-boosting algorithm:

1. When instantiated with a $(k, \lambda, \eta, \beta)$ -base synopsis generator, the output of the boosting algorithm gives $(\lambda + \mu)$ -accurate answers to all the queries in \mathcal{Q} with probability at least $1 - T\beta$, where

$$\mu \in O(((\log^{3/2} |\mathcal{Q}|) \sqrt{k} \sqrt{\log(1/\beta)\rho}) / (\varepsilon_{\text{sample}} \cdot \eta^3)). \quad (6.2)$$

2. If the base synopsis generator is $(\varepsilon_{\text{base}}, \delta_{\text{base}})$ -differentially private, then the boosting algorithm is $(\varepsilon_{\text{sample}} + T \cdot \varepsilon_{\text{base}}, \delta_{\text{sample}} + T \delta_{\text{base}})$ -differentially private.

Allowing the constant η to be swallowed up into the big-O notation, and taking $\rho = 1$ for simplicity, we get $\mu = O(((\log^{3/2} |Q|)\sqrt{k})/\sqrt{\log(1/\beta)})/\varepsilon_{\text{sample}}$. Thus we see that reducing the number k of input queries needed by the base sanitizer improves the quality of the output. Similarly, from the full statement of the theorem, we see that improving the generalization power of the base sanitizer, which corresponds to having a larger value of η (a bigger “strong majority”), also improves the accuracy.

Proof of Theorem 6.1. We first prove accuracy, then privacy.

We introduce the notation $a_{t,q}^-$ and $a_{t,q}^+$, satisfying

1. $a_{t,q}^-, a_{t,q}^+ \in \{-1, 1\}$; and
2. $a_{t,q}^- \leq a_{t,q} \leq a_{t,q}^+$.

Recall that a larger $a_{t,q}$ indicates a higher quality of the approximation of the synopsis \mathcal{A}_t for $q(x)$.

1. $a_{t,q}^-$ is 1 if \mathcal{A}_t is λ -accurate on q , and -1 otherwise. To check that $a_{t,q}^- \leq a_{t,q}$, note that if $a_{t,q}^- = 1$ then \mathcal{A}_t is λ -accurate for q , and so by definition $a_{t,q} = 1$ as well. If instead we have $a_{t,q}^- = -1$ then since we always have $a_{t,q} \in [-1, 1]$, we are done.

We will use the $a_{t,q}^-$ to lower bound a measure of the quality of the output of the base generator. By the promise of the base generator, \mathcal{A}_t is λ -accurate for at least a $1/2 + \eta$ fraction of the mass of \mathcal{D}_t . Thus,

$$r_t \triangleq \sum_{q \in \mathcal{Q}} \mathcal{D}_t[q] \cdot a_{t,q}^- \geq (1/2 + \eta) - (1/2 - \eta) = 2\eta. \quad (6.3)$$

2. $a_{t,q}^+$ is -1 if \mathcal{A}_t is $(\lambda + \mu)$ -inaccurate for q , and 1 otherwise. To check that $a_{t,q} \leq a_{t,q}^+$, note that if $a_{t,q}^+ = -1$ then \mathcal{A}_t is $(\lambda + \mu)$ -inaccurate for q , so by definition $a_{t,q} = -1$ as well. If instead $a_{t,q}^+ = 1$ then since we always have $a_{t,q} \in [-1, 1]$, we are done.

Thus $a_{t,q}^+$ is positive if and only if \mathcal{A}_t is at least minimally adequately accurate for q . We will use the $a_{t,q}^+$ to prove accuracy

of the aggregation. When we sum the values $a_{t,q}^+$, we get a positive number if and only if the majority of the \mathcal{A}_t are providing passable — that is, within $\lambda + \mu$ — approximations to $q(x)$. In this case the median value will be within $\lambda + \mu$.

Lemma 6.2. After T rounds of boosting, with all but $T\beta$ probability, the answers to all but an $\exp(-\eta^2 T)$ -fraction of the queries are $(\lambda + \mu)$ -accurate.

Proof. In the last round of boosting, we have:

$$\mathcal{D}_{T+1}[q] = \frac{u_{T,q}}{Z_T}. \quad (6.4)$$

Since $a_{t,q} \leq a_{t,q}^+$ we have:

$$u_{T,q}^+ \triangleq e^{-\alpha \sum_{t=1}^T a_{t,q}^+} \leq e^{-\alpha \sum_{t=1}^T a_{t,q}} = u_{T,q}. \quad (6.5)$$

(The superscript “+” reminds us that this unweighted value was computed using the terms $a_{t,q}^+$.) Note that we always have $u_{T,q}^+ \geq 0$. Combining Equations (6.4) and (6.5), for all $q \in \mathcal{Q}$:

$$\mathcal{D}_{T+1}[q] \geq \frac{u_{T,q}^+}{Z_T}. \quad (6.6)$$

Recalling that $[[P]]$ denotes the boolean variable that has value 1 if and only if the predicate P is true, we turn to examining the value $[[\mathcal{A} \text{ is } (\lambda + \mu)\text{-inaccurate for } q]]$. If this predicate is 1, then it must be the case that the majority of $\{\mathcal{A}_j\}_{j=1}^T$ are $(\lambda + \mu)$ -inaccurate, as otherwise their median would be $(\lambda + \mu)$ -accurate.

From our discussion of the significance of the sign of $\sum_{t=1}^T a_{t,q}^+$, we have:

$$\begin{aligned} \mathcal{A} \text{ is } (\lambda + \mu)\text{-inaccurate for } q &\Rightarrow \sum_{t=1}^T a_{t,q}^+ \leq 0 \\ &\Leftrightarrow e^{-\alpha \sum_{t=1}^T a_{t,q}^+} \geq 1 \\ &\Leftrightarrow u_{T,q}^+ \geq 1 \end{aligned}$$

Since $u_{T,q}^+ \geq 0$, We conclude that:

$$[[\mathcal{A} \text{ is } (\lambda + \mu)\text{-inaccurate for } q]] \leq u_{T,q}^+$$

Using this together with Equation (6.6) yields:

$$\begin{aligned} \frac{1}{|\mathcal{Q}|} \cdot \sum_{q \in \mathcal{Q}} [[\mathcal{A} \text{ is } (\lambda + \mu)\text{-inaccurate for } q]] &\leq \frac{1}{|\mathcal{Q}|} \cdot \sum_{q \in \mathcal{Q}} u_{T,q}^+ \\ &\leq \frac{1}{|\mathcal{Q}|} \cdot \sum_{q \in \mathcal{Q}} \mathcal{D}_{T+1}[q] \cdot Z_T \\ &= \frac{Z_T}{|\mathcal{Q}|}. \end{aligned}$$

Thus the following claim completes the proof:

Claim 6.3. In round t of boosting, with all but $t\beta$ probability:

$$Z_t \leq \exp(-\eta^2 \cdot t) \cdot |\mathcal{Q}|$$

Proof. By definition of a base synopsis generator, with all but β probability, the synopsis generated is λ -accurate for at least a $(1/2 + \eta)$ -fraction of the mass of the distribution \mathcal{D}_t . Recall that $a_{t,q}^- \in \{-1, 1\}$ is 1 if and only if \mathcal{A}_t is λ -accurate on q , and that $a_{t,q}^- \leq a_{t,q}$ and recall further the quantity $r_t \triangleq \sum_{q \in \mathcal{Q}} \mathcal{D}_t[q] \cdot a_{t,q}^-$ defined in Equation (6.3). As discussed above, r_t measures the “success” of the base synopsis generator in round t , where by “success” we mean the stricter notion of λ -accuracy. As summarized in Equation (6.3), if a $(1/2 + \eta)$ -fraction of the mass of \mathcal{D}_t is computed with λ -accuracy, then $r_t \geq 2\eta$. Now observe also that for $t \in [T]$, assuming the base sanitizer did not fail in round t :

$$\begin{aligned} Z_t &= \sum_{q \in \mathcal{Q}} u_{t,q} \\ &= \sum_{q \in \mathcal{Q}} u_{t-1,q} \cdot e^{-\alpha \cdot a_{t,q}} \\ &= \sum_{q \in \mathcal{Q}} Z_{t-1} \cdot \mathcal{D}_t[q] \cdot e^{-\alpha \cdot a_{t,q}} \\ &\leq \sum_{q \in \mathcal{Q}} Z_{t-1} \cdot \mathcal{D}_t[q] \cdot e^{-\alpha \cdot a_{t,q}^-} \\ &= Z_{t-1} \cdot \sum_{q \in \mathcal{Q}} \mathcal{D}_t[q] \cdot \left(\left(\frac{1 + a_{t,q}^-}{2} \right) \cdot e^{-\alpha} + \left(\frac{1 - a_{t,q}^-}{2} \right) \cdot e^\alpha \right) \\ &\quad \text{(case analysis)} \end{aligned}$$

$$\begin{aligned}
&= \frac{Z_{t-1}}{2} [(e^\alpha + e^{-\alpha}) + r_t(e^{-\alpha} - e^\alpha)] \\
&\leq \frac{Z_{t-1}}{2} [(e^\alpha + e^{-\alpha}) + 2\eta(e^{-\alpha} - e^\alpha)] \quad (r_t \geq 2\eta \text{ and } (e^{-\alpha} - e^\alpha) \leq 0)
\end{aligned}$$

By simple calculus we see that $(e^\alpha + e^{-\alpha}) + 2\eta(e^{-\alpha} - e^\alpha)$ is minimized when

$$\alpha = (1/2) \ln \left(\frac{1+2\eta}{1-2\eta} \right).$$

Plugging this into the recurrence, we get

$$Z_t \leq (\sqrt{1-4\eta^2})^t |\mathcal{Q}| \leq \exp(-2\eta^2 t) |\mathcal{Q}|.$$

□

This completes the proof of Lemma 6.2. □

The lemma implies that accuracy for *all* queries simultaneously can be achieved by setting

$$T > \frac{\ln |\mathcal{Q}|}{\eta^2}.$$

Privacy. We will show that the entire sequence $(S_1, \mathcal{A}_1, \dots, S_T, \mathcal{A}_T)$ can be output while preserving differential privacy. Note that this is stronger than we need — we do not actually output the sets S_1, \dots, S_T . By our adaptive composition theorems, the privacy of each \mathcal{A}_i will be guaranteed by the privacy guarantees of the base synopsis generator, together with the fact that S_{i-1} was computed in a differentially private way. Therefore, it suffices to prove that given that $(S_1, \mathcal{A}_1, \dots, S_i, \mathcal{A}_i)$ is differentially private, S_{i+1} is as well. We can then combine the privacy parameters using our composition theorems to compute a final guarantee.

Lemma 6.4. Let $\varepsilon^* = \frac{4\alpha T \rho}{\mu}$. For all $i \in [T]$, once $(S_1, \mathcal{A}_1, \dots, S_i, \mathcal{A}_i)$ is fixed, the computation of each element of S_{i+1} is $(\varepsilon^*, 0)$ -differentially private.

Proof. Fixing $\mathcal{A}_1, \dots, \mathcal{A}_i$, for every $j \leq i$, the quantity $d_{q,j}$ has sensitivity ρ , since $\mathcal{A}_j(q)$ is database independent (because \mathcal{A}_j is fixed), and

every $q \in \mathcal{Q}$ has sensitivity bounded by ρ . Therefore, for every $j \leq i$, $a_{j,q}$ is $2\rho/\mu$ sensitive by construction, and so

$$g_i(q) \stackrel{\text{def}}{=} \sum_{j=1}^i a_{j,q}$$

has sensitivity at most $2i\rho/\mu \leq 2T\rho/\mu$. Then $\Delta g_i \stackrel{\text{def}}{=} 2T\rho/\mu$ is an upper bound on the sensitivity of g_i .

To argue privacy, we will show that the selection of queries for S_{i+1} is an instance of the exponential mechanism. Think of $-g_i(q)$ as the utility of a query q during the selection process at round $i + 1$. The exponential mechanism says that to achieve $(\varepsilon^*, 0)$ -differential privacy we should choose q with probability proportional to

$$\exp\left(-g_i(q)\frac{\varepsilon^*}{2\Delta g_i}\right).$$

Since $\varepsilon^*/2\Delta g_i = \alpha$ and the algorithm selects q with probability proportional to $e^{-\alpha g_i(q)}$, we see that this is exactly what the algorithm does! \square

We bound the privacy loss of releasing the S_i s by treating each selection of a query as a “mini-mechanism” that, over the course of T rounds of boosting, is invoked kT times. By Lemma 6.4 each mini-mechanism is $(4\alpha T\rho/\mu, 0)$ -differentially private. By Theorem 3.20, for all $\beta > 0$ the composition of kT mechanisms, each of which is $(\alpha 4T\rho/\mu, 0)$ -differentially private, is $(\varepsilon_{\text{sample}}, \delta_{\text{sample}})$ -differentially private, where

$$\varepsilon_{\text{sample}} \stackrel{\text{def}}{=} \sqrt{2kT \log(1/\delta_{\text{sample}})}(\alpha 4T\rho/\mu) + kT \left(\frac{\alpha 4T\rho}{\mu}\right)^2. \quad (6.7)$$

Our total privacy loss comes from the composition of T calls to the base sanitizer and the cumulative loss from the kT samples. We conclude that the boosting algorithm in its entirety is: $(\varepsilon_{\text{boost}}, \delta_{\text{boost}})$ -differentially private, where

$$\begin{aligned} \varepsilon_{\text{boost}} &= T\varepsilon_{\text{base}} + \varepsilon_{\text{sample}} \\ \delta_{\text{boost}} &= T\delta_{\text{base}} + \delta_{\text{sample}} \end{aligned}$$

To get the parameters claimed in the statement of the theorem, we can take:

$$\mu \in O((T^{3/2} \sqrt{k} \sqrt{\log(1/\beta)} \alpha \rho) / \varepsilon_{\text{sample}}). \quad (6.8)$$

□

6.2 Base synopsis generators

Algorithm SmallDB (Section 4) is based on the insight that a small randomly selected subset of database rows provides good answers to large sets of fractional counting queries. The base synopsis generators described in the current section have an analogous insight: a small synopsis that gives good approximations to the answers to a small subset of queries also yields good approximations to most queries. Both of these are instances of *generalization bounds*. In the remainder of this section we first prove a generalization bound and then use it to construct differentially private base synopsis generators.

6.2.1 A generalization bound

We have a distribution \mathcal{D} over a large set \mathcal{Q} of queries to be approximated. The lemma below says that a sufficiently small synopsis that gives sufficiently good approximations to the answers of a *randomly selected* subset $S \subset \mathcal{Q}$ of queries, sampled according to the distribution \mathcal{D} on \mathcal{Q} , will, with high probability over the choice of S , also give good approximations to the answers to *most* queries in \mathcal{Q} (that is, to most of the mass of \mathcal{Q} , weighted by \mathcal{D}). Of course, to make any sense the synopsis must include a method of providing an answer to all queries in \mathcal{Q} , not just the subset $S \subseteq \mathcal{Q}$ received as input. Our particular generators, described in Sections 6.2.2 and Theorem 6.6 will produce synthetic databases; to answer any query one can simply apply the query to the synthetic database, but the lemma will be stated in full generality.

Let $R(y, q)$ denote the answer given by the synopsis y (when used as input for the reconstruction procedure) on query q . A synopsis y λ -*fits* a database x w.r.t a set S of queries if $\max_{q \in S} |R(y, q) - q(x)| \leq \lambda$. Let $|y|$

denote the number of bits needed to represent y . Since our synopses will be synthetic databases, $|y| = N \log_2 |\mathcal{X}|$ for some appropriately chosen number N of universe elements. The generalization bound shows that if y λ -fits x with respect to a large enough (larger than $|y|$) randomly chosen set S of queries sampled from a distribution \mathcal{D} , then with high probability y λ -fits x for *most* of the mass of \mathcal{D} .

Lemma 6.5. Let \mathcal{D} be an arbitrary distribution on a query set $\mathcal{Q} = \{q : \mathcal{X}^* \rightarrow \mathbb{R}\}$. For all $m \in \mathcal{N}$, $\gamma \in (0, 1)$, $\eta \in [0, 1/2]$, let $a = 2(\log(1/\gamma) + m)/(m(1 - 2\eta))$. Then with probability at least $1 - \gamma$ over the choice of $S \sim \mathcal{D}^{a \cdot m}$, every synopsis y of size at most m bits that λ -fits x with respect to the query set S , also λ -fits x with respect to at least a $(1/2 + \eta)$ -fraction of \mathcal{D} .

Before proving the lemma we observe that a is a compression factor: we are squeezing the answers to am queries into an m -bit output, so larger a corresponds to more compression. Typically, this means better generalization, and indeed we see that if a is larger then, keeping m and γ fixed, we would be able to have larger η . The lemma also says that, for any given output size m , the number of queries needed as input to obtain an output that does well on a majority $(1/2 + \eta)$ fraction of \mathcal{D} is only $O(\log(1/\gamma) + m)$. This is interesting because a smaller number of queries k needed by the base generator leads, via the privacy loss $\varepsilon_{\text{sample}}$ due to sampling of kT queries and its inverse relationship to the slackness μ (Equation 6.7), to improved accuracy of the output of the boosting algorithm.

Proof of Lemma 6.5. Fix a set of queries $S \subset \mathcal{Q}$ chosen independently according to $\mathcal{D}^{a \cdot m}$. Examine an arbitrary m -bit synopsis y . Note that y is described by an m -bit string. Let us say y is *bad* if $|R(y, q) - q(x)| > \lambda$ for at least a $(\log(1/\gamma) + m)/(a \cdot m)$ fraction of \mathcal{D} , meaning that $\Pr_{q \sim \mathcal{D}}[|R(y, q) - q(x)| > \lambda] \geq (\log(1/\gamma) + m)/(a \cdot m)$.

In other words, y is bad if there exists a set $Q_y \subset \mathcal{Q}$ of fractional weight at least $(\log(1/\gamma) + m)/(a \cdot m)$ such that $|R(y, q) - q(x)| > \lambda$ for $q \in Q_y$. For such a y , what is the probability that y gives λ -accurate answers for *every* $q \in S$? This is exactly the probability that none of

the queries in S is in Q_y , or

$$(1 - (\log(1/\gamma) + m)/(a \cdot m))^{a \cdot m} \leq e^{-(\log(1/\gamma) + m)} \leq \gamma \cdot 2^{-m}$$

Taking a union bound over all 2^m possible choices for y , the probability that there exists an m -bit synopsis y that is accurate on all the queries in S but inaccurate on a set of fractional weight $(\log(1/\beta) + m)/(a \cdot m)$ is at most γ . Letting $k = am = |S|$ we see that it is sufficient to have

$$a > \frac{2(\log(1/\gamma) + m)}{m \cdot (1 - 2\eta)}. \quad (6.9)$$

□

This simple lemma is extremely powerful. It tells us that when constructing a base generator at round t , we only need to worry about ensuring good answers for the small set of random queries sampled from \mathcal{D}_t ; doing well for most of \mathcal{D}_t will happen automatically!

6.2.2 The base generator

Our first generator works by brute force. After sampling a set S of k queries independently according to a distribution \mathcal{D} , the base generator will produce noisy answers for all queries in S via the Laplace mechanism. Then, making no further use of the actual database, the algorithm searches for *any* database of size n for which these noisy answers are sufficiently close, and outputs this database. Privacy will be immediate because everything after the k invocations of the Laplace mechanism is in post-processing. Thus the only source of privacy loss is the cumulative loss from these k invocations of the Laplace mechanism, which we know how to analyze via the composition theorem. Utility will follow from the utility of the Laplace mechanism — which says that we are unlikely to have “very large” error on even one query — coupled with the fact that the true database x is an n -element database that fits these noisy responses.¹

¹This argument assumes the size n of the database is known. Alternatively we can include a noisy query of the form “How many rows are in the database?” and exhaustively search all databases of size close to the response to this query.

Theorem 6.6 (Base Synopsis Generator for Arbitrary Queries). For any data universe \mathcal{X} , database size n , and class $\mathcal{Q} : \{\mathcal{X}^* \rightarrow \mathbb{R}\}$ of queries of sensitivity at most ρ , for any $\varepsilon_{\text{base}}, \delta_{\text{base}} > 0$, there exists an $(\varepsilon_{\text{base}}, \delta_{\text{base}})$ -differentially private $(k, \lambda, \eta = 1/3, \beta)$ -base synopsis generator for \mathcal{Q} , where $k = am > 6(m + \log(2/\beta)) = 6(n \log |\mathcal{X}| + \log(2/\beta))$ and $\lambda > 2b(\log k + \log(2/\beta))$, where $b = \rho\sqrt{am \log(1/\delta_{\text{base}})}/\varepsilon_{\text{base}}$.

The running time of the generator is

$$|\mathcal{X}|^n \cdot \text{poly}(n, \log(1/\beta), \log(1/\varepsilon_{\text{base}}), \log(1/\delta_{\text{base}})).$$

Proof. We first describe the base generator at a high level, then determine the values for k and λ . The synopsis y produced by the base generator will be a synthetic database of size n . Thus $m = |y| = n \cdot \log |\mathcal{X}|$. The generator begins by choosing a set S of k queries, sampled independently according to \mathcal{D} . It computes a noisy answer for each query $q \in S$ using the Laplace mechanism, adding to each true answer an independent draw from $\text{Lap}(b)$ for an appropriate b to be determined later. Let $\{\widehat{q(x)}\}_{q \in \mathcal{Q}}$ be the collection of noisy answers. The generator enumerates over all $|\mathcal{X}|^n$ databases of size n , and outputs the lexicographically first database y such that for every $q \in S$ we have $|q(y) - \widehat{q(x)}| \leq \lambda/2$. If no such database is found, it outputs \perp instead, and we say it *fails*. Note that if $|\widehat{q(x)} - q(x)| < \lambda/2$ and $|q(y) - \widehat{q(x)}| < \lambda/2$, then $|q(y) - q(x)| < \lambda$.

There are two potential sources of failure for our particular generator. One possibility is that y fails to generalize, or is *bad* as defined in the proof of Lemma 6.5. A second possibility is that one of the samples from the Laplace distribution is of excessively large magnitude, which might cause the generator to fail. We will choose our parameters so as to bound the probability of each of these events individually by at most $\beta/2$.

Substituting $\eta = 1/3$ and $m = n \log |\mathcal{X}|$ into Equation 6.9 shows that taking $a > 6(1 + \log(2/\beta)/m)$ suffices in order for the probability of failure due to the choice of S to be bounded by $\beta/2$. Thus, taking $k = am > 6(m + \log(2/\beta)) = 6(n \log |\mathcal{X}| + \log(2/\beta))$ suffices.

We have k queries of sensitivity at most ρ . Using the Laplace mechanism with parameter $b = 2\sqrt{2k \log(1/\delta_{\text{base}})}\rho/\varepsilon_{\text{base}}$, ensures that each query incurs privacy loss at most $\varepsilon_{\text{base}}/\sqrt{2k \ln(1/\delta_{\text{base}})}$, which by

Corollary 3.21 ensures that the entire procedure will be $(\varepsilon_{\text{base}}, \delta_{\text{base}})$ -differentially private.

We will choose λ so that the probability that any draw from $\text{Lap}(b)$ has magnitude exceeding $\lambda/2$ is at most $\beta/2$. Conditioned on the event that all k draws have magnitude at most λ we know that the input database itself will λ -fit our noisy answers, so the procedure will not fail.

Recall that the concentration properties of the Laplace distribution ensure that with probability at least $1 - e^{-t}$ a draw from $\text{Lap}(b)$ will have magnitude bounded by tb . Setting $\lambda/2 = tb$, the probability that a given draw will have magnitude exceeding $\lambda/2$ is bounded by $e^{-t} = e^{-\lambda/2b}$. To ensure that none of the k draws has magnitude exceeding $\lambda/2$ it suffices, by a union bound, to have

$$\begin{aligned} ke^{-\lambda/2b} &< \beta/2 \\ \Leftrightarrow e^{\lambda/2b} &> k \frac{2}{\beta} \\ \Leftrightarrow \lambda/2 &> b(\log k + \log(2/\beta)) \\ \Leftrightarrow \lambda &> 2b(\log k + \log(2/\beta)). \end{aligned}$$

□

The Special Case of Linear Queries. For the special case of linear queries it is possible to avoid the brute force search for a small database. The technique requires time that is polynomial in $(|\mathcal{Q}|, |\mathcal{X}|, n, \log(1/\beta))$. We will focus on the case of counting queries and sketch the construction.

As in the case of the base generator for arbitrary queries, the base generator begins by selecting a set S of $k = am$ queries according to \mathcal{D} and computing noisy answers using Laplace noise. The generator for linear queries then runs a *syntheticizer* on S which, roughly speaking, transforms any synopsis giving good approximations to *any* set R of queries into a synthetic database yielding approximations of similar quality on the set R . The input to the syntheticizer will be the noisy values for the queries in S , that is, $R = S$. (Recall that when we modify the size of the database we always think in terms of the fractional version of the counting queries: “What fraction of the database rows satisfies property P ?”)

The resulting database may be quite large, meaning it may have many rows. The base generator then subsamples only $n' = (\log k \log(1/\beta))/\alpha^2$ of the rows of the synthetic database, creating a smaller synthetic database that with probability at least $1 - \beta$ has α -accuracy with respect to the answers given by the large synthetic database. This yields an $m = ((\log k \log(1/\beta))/\alpha^2) \log |\mathcal{X}|$ -bit synopsis that, by the generalization lemma, with probability $(1 - \log(1/\beta))$ over the choice of the k queries, answers well on a $(1/2 + \eta)$ fraction of \mathcal{Q} (as weighted by \mathcal{D}).

As in the case of the base generator for arbitrary queries, we require $k = am > 6 \log(1/\beta) + 6m$. Taking $\alpha^2 = (\log |\mathcal{Q}|)/n$ we get that

$$\begin{aligned} k &> 6 \log(1/\beta) + 6 \frac{\log k \log(1/\beta) \log |\mathcal{X}|}{\alpha^2} \\ &= 6 \log(1/\beta) + 6n \log k \log(1/\beta) \frac{\log |\mathcal{X}|}{\log |\mathcal{Q}|}. \end{aligned}$$

The syntheticizer is nontrivial. Its properties are summarized by the following theorem.

Theorem 6.7. Let \mathcal{X} be a data universe, \mathcal{Q} a set of fractional counting queries, and A an (ε, δ) -differentially private synopsis generator with utility $(\alpha, \beta, 0)$ and arbitrary output. Then there exists a syntheticizer A' that is (ε, δ) -differentially private and has utility $(3\alpha, \beta, 0)$. A' outputs a (potentially large) synthetic database. Its running time is polynomial in the running time of A and $(|\mathcal{X}|, |\mathcal{Q}|, 1/\alpha, \log(1/\beta))$.

In our case, A is the Laplace mechanism, and the synopsis is simply the set of noisy answers. The composition theorem says that for A to be $(\varepsilon_{\text{base}}, \delta_{\text{base}})$ -differentially private the parameter to the Laplace mechanism should be $\rho / (\varepsilon_{\text{base}} / \sqrt{2k \log(1/\delta_{\text{base}})})$. For fractional counting queries the sensitivity is $\rho = 1/n$.

Thus, when we apply the Theorem we will have an α of order $(\sqrt{k \log(1/\beta)} / \varepsilon_{\text{base}})\rho$. Here, ρ is the sensitivity. For counting queries it is 1, but we will shift to fractional counting queries, so $\rho = 1/n$.

Proof Sketch for Theorem 6.7. Run A to get (differentially private) (fractional) counts on all the queries in R . We will then use linear programming to find a low-weight fractional database that approximates

these fractional counts, as explained below. Finally, we transform this fractional database into a standard synthetic database by rounding the fractional counts.

The output of A yields a fractional count for each query $q \in \mathcal{Q}$. The input database x is never accessed again and so A' is (ε, δ) -differentially private. Let v be the resulting vector of counts, i.e., v_q is the fractional count that A 's output gives on query q . With probability $1 - \beta$, all of the entries in v are α -accurate.

A “fractional” database z that approximates these counts is obtained as follows. Recall the histogram representation of a database, where for each element in the universe \mathcal{X} the histogram contains the number of instances of this element in the database. Now, for every $i \in \mathcal{X}$, we introduce a variable $a_i \geq 0$ that will “count” the (fractional) number of occurrences of i in the fractional database z . We will impose the constraint

$$\sum_{i \in \mathcal{X}} a_i = 1.$$

We represent the count of query q in z as the sum of the count of items i that satisfy q :

$$\sum_{i \in \mathcal{X} \text{ s.t. } q(i)=1} a_i$$

We want all of these counts to be within a an additive α accuracy of the respective counts in v_q . Writing this as a linear inequality we get:

$$(v_q - \alpha) \sum_{i \in \mathcal{X}} a_i \leq \sum_{i \in \mathcal{X} \text{ s.t. } q(i)=1} a_i \leq (v_q + \alpha) \sum_{i \in \mathcal{X}} a_i.$$

When the counts are all α -accurate with respect to the counts in v_c , it is also the case that (with probability $1 - \beta$) they are all 2α -accurate with respect to the true counts on the original database x .

We write a linear program with two such constraints for each query (a total of $2|\mathcal{Q}|$ constraints). A' tries to find a fractional solution to this linear program. To see that such a solution exists, observe that the database x itself is α -close to the vector of counts v , and so there *exists* a solution to the linear program (in fact even an integer solution), and hence A' will find *some* fractional solution.

We conclude that A' can generate a fractional database with $(2\alpha, \beta, 0)$ -utility, but we really want a synthetic (integer) database. To transform the fractional database into an integer one, we round down each a_i , for $i \in \mathcal{X}$, to the closest multiple of $\alpha/|\mathcal{X}|$, this changes each fractional count by at most a $\alpha/|\mathcal{X}|$ additive factor, and so the rounded counts have $(3\alpha, \beta, 0)$ utility. Now we can treat the rounded fractional database (which has total weight 1), as an integer synthetic database of (polynomial) size at most $|\mathcal{X}|/\alpha$. \square

Recall that in our application of Theorem 6.7 we defined A to be the mechanism that adds Laplace noise with parameter $\rho/(\varepsilon_{\text{base}}/\sqrt{2k \log(1/\delta_{\text{base}})})$. We have k draws, so by taking

$$\alpha' = \rho \sqrt{2k \log(1/\delta_{\text{base}})} (\log k + \log(1/\beta))$$

we have that A is $(\alpha', \beta, 0)$ -accurate. For the base generator we chose error $\alpha^2 = (\log |\mathcal{Q}|)/n$. If the output of the syntheticizer is too large, we subsample

$$n' = \frac{\log |\mathcal{Q}| \log(1/\beta)}{\alpha^2} = \frac{\log k \log(1/\beta)}{\alpha^2}$$

rows. With probability $1 - \beta$ the resulting database maintains $O(\rho \sqrt{(\log |\mathcal{Q}|)/n} + (\sqrt{2k \log(1/\delta_{\text{base}}})/\varepsilon_{\text{base}}})(\log k + \log(1/\beta))$ -accuracy on all of the concepts simultaneously.

Finally, the base generator can fail if the choice of queries $S \in \mathcal{D}^k$ does not lead to good generalization. With the parameters we have chosen this occurs with probability at most β , leading to a total failure probability of the entire generator of 3β .

Theorem 6.8 (Base Generator for Fractional Linear Queries). For any data universe \mathcal{X} , database size n , and class $\mathcal{Q} : \{\mathcal{X}^n \rightarrow \mathbb{R}\}$ of fractional linear queries (with sensitivity at most $1/n$), for any $\varepsilon_{\text{base}}, \delta_{\text{base}} > 0$, there exists an $(\varepsilon_{\text{base}}, \delta_{\text{base}})$ -differentially private $(k, \lambda, 1/3, 3\beta)$ -base synopsis generator for \mathcal{Q} , where

$$k = O\left(\frac{n \log(|\mathcal{X}|) \log(1/\beta)}{\log |\mathcal{Q}|}\right)$$

$$\lambda = O\left(\frac{\log(1/\beta)}{\sqrt{n}} \left(\sqrt{\log |\mathcal{Q}|} + \sqrt{\frac{\log |\mathcal{X}|}{\log |\mathcal{Q}|}} \cdot \frac{1}{\varepsilon_{\text{base}}}\right)\right).$$

The running time of the base generator is $\text{poly}(|\mathcal{X}|, n, \log(1/\beta), \log(1/\varepsilon_{\text{base}}))$.

The sampling bound used here is the same as that used in the construction of the SmallIDB mechanism, but with different parameters. Here we are using these bounds for a base generator in a complicated boosting algorithm with a very small query set; there we are using them for a single-shot generation of a synthetic database with an enormous query set.

6.2.3 Assembling the ingredients

The total error comes from the choice of μ (see Equation 6.2) and λ , the accuracy parameter for the based generator.

Let us recall Theorem 6.1:

Theorem 6.9 (Theorem 6.1). Let \mathcal{Q} be a query family with sensitivity at most ρ . For an appropriate setting of parameters, and with $T = \log |\mathcal{Q}|/\eta^2$ rounds, the algorithm of Figure 6.1 is an accurate and differentially private query-boosting algorithm:

- When instantiated with a $(k, \lambda, \eta, \beta)$ -base synopsis generator, the output of the boosting algorithm gives $(\lambda + \mu)$ -accurate answers to *all* the queries in \mathcal{Q} with probability at least $1 - T\beta$, where

$$\mu \in O(((\log^{3/2} |\mathcal{Q}|)\sqrt{k}\sqrt{\log(1/\beta)}\rho)/(\varepsilon_{\text{sample}} \cdot \eta^3)). \quad (6.10)$$

- If the base synopsis generator is $(\varepsilon_{\text{base}}, \delta_{\text{base}})$ -differentially private, then the boosting algorithm is $((\varepsilon_{\text{sample}} + T \cdot \varepsilon_{\text{base}}), T(\beta + \delta_{\text{base}}))$ -differentially private.

By Equation 6.7,

$$\varepsilon_{\text{sample}} \stackrel{\text{def}}{=} \sqrt{2kT \log(1/\beta)}(\alpha 4T\rho/\mu) + kT \left(\frac{\alpha 4T\rho}{\mu}\right)^2,$$

where $\alpha = (1/2)(\ln(1 + 2\eta)(1 - 2\eta)) \in O(1)$. We always have $T = (\log |\mathcal{Q}|)/\eta^2$, so substituting in this value into the above equation we see that the bound

$$\mu \in O(((\log^{3/2} |\mathcal{Q}|)\sqrt{k}\sqrt{\log(1/\beta)}\rho)/(\varepsilon_{\text{sample}} \cdot \eta^3))$$

in the statement of the theorem is acceptable.

For the case of arbitrary queries, with η a constant, we have

$$\lambda \in O\left(\frac{\rho}{\varepsilon_{\text{base}}}(\sqrt{n \log |\mathcal{X}| \log(1/\delta_{\text{base}})}(\log(n \log |\mathcal{X}|) + \log(2/\beta)))\right).$$

Now, $\varepsilon_{\text{boost}} = T\varepsilon_{\text{base}} + \varepsilon_{\text{sample}}$. Set these two terms equal, so $T\varepsilon_{\text{base}} = \varepsilon_{\text{boost}}/2 = \varepsilon_{\text{sample}}$, whence we can replace the $1/\varepsilon_{\text{base}}$ term with $2T/\varepsilon_{\text{boost}} = (\log |\mathcal{Q}|/\eta^2)/2\varepsilon_{\text{boost}}$. Now our terms for λ and μ have similar denominators, since η is constant. We may therefore conclude that the total error is bounded by:

$$\lambda + \mu \in \tilde{O}\left(\frac{\sqrt{n \log |\mathcal{X}|} \rho \log^{3/2} |\mathcal{Q}| (\log(1/\beta))^{3/2}}{\varepsilon_{\text{boost}}}\right).$$

With similar reasoning, for the case of fractional counting queries we get

$$\lambda + \mu \in \tilde{O}\left(\frac{\sqrt{\log |\mathcal{X}|} \log |\mathcal{Q}| \log(1/\beta)^{3/2}}{\varepsilon_{\text{boost}} \sqrt{n}}\right).$$

To convert to a bound for ordinary, non-fractional, counting queries we multiply by n to obtain

$$\lambda + \mu \in \tilde{O}\left(\frac{\sqrt{n \log |\mathcal{X}|} \log |\mathcal{Q}| \log(1/\beta)^{3/2}}{\varepsilon_{\text{boost}}}\right).$$

6.3 Bibliographical notes

The boosting algorithm (Figure 6.1) is a variant of AdaBoost algorithm of Schapire and Singer [78]. See Schapire [77] for an excellent survey of boosting, and the textbook “Boosting” by Freund and Schapire [79] for a thorough treatment. The private boosting algorithm covered in this section is due to Dwork et al. [32], which also contains the base generator for linear queries. This base generator, in turn, relies on the syntheticizer of Dwork et al. [28]. In particular, Theorem 6.7 comes from [28]. Dwork, Rothblum, and Vadhan also addressed differentially private boosting in the usual sense.

7

When Worst-Case Sensitivity is Atypical

In this section, we briefly describe two general techniques, both enjoying unconditional privacy guarantees, that can often make life easier for the data analyst, especially when dealing with a function that has arbitrary, or difficult to analyze, worst-case sensitivity. These algorithms are most useful in computing functions that, for some exogenous reason, the analyst has reason to believe are “usually” insensitive in practice.

7.1 Subsample and aggregate

The Subsample and Aggregate technique yields a method for “forcing” the computation of a function $f(x)$ to be insensitive, even for an *arbitrary* function f . Proving privacy will be trivial. Accuracy depends on properties of the function f and the specific data set x ; in particular, if $f(x)$ can be accurately estimated, with high probability, on $f(S)$, where S is a random subset of the elements in x , then accuracy should be good. Many maximum likelihood statistical estimators enjoy this property on “typical” data sets — this is why these estimators are employed in practice.

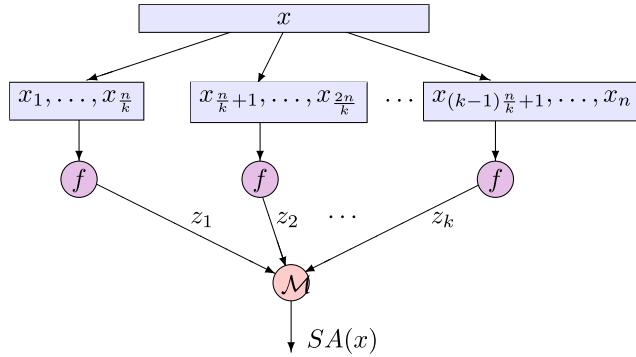


Figure 7.1: Subsample and Aggregate with a generic differentially private aggregation algorithm \mathcal{M} .

In Subsample and Aggregate, the n rows of the database x are randomly partitioned into m blocks B_1, \dots, B_m , each of size n/m . The function f is computed *exactly, without noise*, independently on each block. The intermediate outcomes $f(B_1), \dots, f(B_m)$ are then combined via a differentially private aggregation mechanism — typical examples include standard aggregations, such as the α -trimmed mean,¹ the Winsorized mean,² and the median, but there are no restrictions — and then adding Laplace noise scaled to the sensitivity of the aggregation function in question; see Figure 7.1.

The key observation in Subsample and Aggregate is that any single element can affect at most one block, and therefore the value of just a single $f(B_i)$. Thus, changing the data of any individual can change at most a single input to the aggregation function. Even if f is arbitrary, the analyst chooses the aggregation function, and so is free to choose one that is insensitive, *provided that choice is independent of the database!* Privacy is therefore immediate: For any $\delta \geq 0$ and any function f , if the aggregation mechanism \mathcal{M} is (ϵ, δ) -differentially private

¹The α -trimmed mean is the mean after the top and bottom α fraction of the inputs have been discarded.

²The Winsorized mean is similar to the α -trimmed mean except that, rather than being discarded, the top and bottom α fraction are replaced with the most extreme remaining values.

then so is the Subsample and Aggregate technique when instantiated with f and \mathcal{M} .³

Utility is a different story, and it is frustratingly difficult to argue even for the case in which data are plentiful and large random subsets are very likely to give similar results. For example, the data may be labeled training points in high dimensional space and the function is logistic regression, which produces a vector v and labels a point p with $+1$ if and only if $p \cdot v \geq T$ for some (say, fixed) threshold T . Intuitively, if the samples are sufficiently plentiful and typical then all blocks should yield similar vectors v . The difficulty comes in getting a good bound on the worst-case sensitivity of the aggregation function — we may need to use the size of the range as a fallback. Nonetheless, some nice applications are known, especially in the realm of statistical estimators, where, for example, it can be shown that, under the assumption of “generic normality,” privacy can be achieved at *no* additional cost in statistical efficiency (roughly, accuracy as the number of samples grows). We do not define generic normality here, but note that estimators fitting these assumptions include the maximum likelihood estimator for “nice” parametric families of distributions such as gaussians, and maximum-likelihood estimators for linear regression and logistic regression.

Suppose the function f has a *discrete* range of cardinality m , say, $[m]$. In this case Subsample and Aggregate will need to aggregate a set of b elements drawn from $[m]$, and we can use Report Noisy Arg-Max to find the most popular outcome. This approach to aggregation requires $b \geq \log m$ to obtain meaningful results even when the intermediate outcomes are unanimous. We will see an alternative below with no such requirement.

Example 7.1 (Choosing a Model). Much work in statistics and machine learning addresses the problem of *model selection*: Given a data set and a discrete collection of “models,” each of which is a family of probability distributions, the goal is to determine the model that best “fits”

³The choice of aggregation function can even depend on the database, but the selection must be made in a differentially private fashion. The privacy cost is then the cost of composing the choice operation with the aggregation function.

the data. For example, given a set of labeled d -dimensional data, the collection of models might be all subsets of at most $s \ll d$ features, and the goal is to find the set of features that best permits prediction of the labels. The function f might be choosing the best model from the given set of m models, a process known as *model fitting*, via an arbitrary learning algorithm. Aggregation to find the most popular value could be done via Report Noisy Max, which also yields an estimate of its popularity.

Example 7.2 (Significant Features). This is a special case of model fitting. The data are a collection of points in \mathbb{R}^d and the function is the very popular LASSO, which yields as output a list $L \in [d]^s$ of at most $s \ll d$ significant features. We can aggregate the output in two ways: feature by feature — equivalent to running d executions of Subsample and Aggregate, one for each feature, each with a range of size 2 — or on the set as a whole, in which case the cardinality of the range is $\binom{d}{s}$.

7.2 Propose-test-Release

At this point one might ask: what is the meaning of the aggregation if there is not substantial agreement among the blocks? More generally, for any reasonably large-scale statistical analysis in real life, we expect the results to be fairly stable, independent of the presence or absence of any single individual. Indeed, this is the entire intuition behind the significance of a statistic and underlying the utility of differential privacy. We can even go further, and argue that if a statistic is not stable, we should have no interest in computing it. Often, our database will in fact be a sample from a larger population, and our true goal is not to compute the value of the statistic on the database itself, but rather estimate it for the underlying population. Implicitly, therefore, when computing a statistic we are already assuming that the statistic is stable under subsampling!

Everything we have seen so far has provided privacy even on very “idiosyncratic” datasets, for which “typically” stable algorithms may be highly unstable. In this section we introduce a methodology, Propose-Test-Release, which is motivated by the philosophy that if there is

insufficient stability then the analysis can be abandoned because the results are not in fact meaningful. That is, the methodology allows the analyst to check that, *on the given dataset*, the function satisfies some “robustness” or “stability” criterion and, if it does not, to halt the analysis.

The goal of our first application of Propose-Test-Release is to come up with a variant of the Laplace mechanism that adds noise scaled to something strictly smaller than the sensitivity of a function. This leads to the notion of *local sensitivity*, which is defined for a (function, database) pair, say, (f, x) . Quite simply, the local sensitivity of f with respect to x is the amount by which the $f(y)$ can differ from $f(x)$ for any y adjacent to x .

Definition 7.1 (Local Sensitivity). The local sensitivity of a function $f : \mathcal{X}^n \rightarrow \mathbb{R}^k$ with respect to a database x is:

$$\max_{y \text{ adjacent to } x} \|f(x) - f(y)\|_1.$$

The Propose-Test-Release approach is to first *propose* a bound, say b , on local sensitivity — typically the data analyst has some idea of what this should be — and then run a differentially private *test* to ensure that the database is “far” from any database for which this bound fails to hold. If the test is passed, then the sensitivity is assumed to be bounded by b , and a differentially private mechanism such as, for example, the Laplace mechanism with parameter b/ϵ , is used to *release* the (slightly) noisy response to the query.

Note that we can view this approach as a two-party algorithm where one party plays an honest data analyst and the other is the Laplace mechanism. There is an interplay between the honest analyst and the mechanism in which the algorithm asks for an estimate of the sensitivity and then “instructs” the mechanism to use this estimated sensitivity in responding to subsequent queries. Why does it need to be so complicated? Why can’t the mechanism simply add noise scaled to the local sensitivity without playing this private estimation game? The reason is that the local sensitivity may *itself* be sensitive. This fact, combined with some auxiliary information about the database, can lead to privacy problems: the adversary may know that the database is one of x ,

which has very low local sensitivity for the computation in question, and a neighboring y , for which the function has very high local sensitivity. In this case the adversary may be able to guess rather accurately which of x and y is the true database. For example, if $f(x) = f(y) = s$ and the response is far from s , then the adversary would guess y .

This is captured by the math of differential privacy. There are neighboring instances of the median function which have the same median, say, m , but arbitrarily large gaps in the local sensitivity. Suppose the response R to the median query is computed via the Laplace mechanism with noise scaled to the local sensitivity. When the database is x the probability mass is close to m , because the sensitivity is small, but when the database is y the mass is far flung, because the sensitivity is large. As an extreme case, suppose the local sensitivity on x is exactly zero, for example, $\mathcal{X} = \{0, 10^6\}$, n is even, and x , which has size $n + 1$, contains $1 + n/2$ zeros. Then the median of x is zero and the local sensitivity of the median, when the database is x , is 0. In contrast, the neighboring database y has size n , contains $n/2$ zeros, has median zero (we have defined median to break ties in favor of the smaller value), and the local sensitivity of the median, when the database is y , is 10^6 . On x all the mass of the Laplace mechanism (with parameter $0/\varepsilon = 0$) is concentrated on the single point 0; but on y the probability distribution has standard deviation $\sqrt{2} \cdot 10^6$. This destroys all hope of differential privacy.

To test that the database is “far” from one with local sensitivity greater than the proposed bound b , we may pose the query: “What is the distance of the true database to the closest one with local sensitivity exceeding b ?” Distance to a fixed set of databases is a (global) sensitivity 1 query, so this test can be run in a differentially private fashion by adding noise $\text{Lap}(1/\varepsilon)$ to the true answer. To err on the side of privacy, the algorithm can compare this noisy distance to a conservative threshold — one that is only negligibly likely to be exceeded due to a freak event of very large magnitude Laplace noise. For example, if the threshold used is, say, $\ln^2 n$, the probability of a false positive (passing the test when the local sensitivity in fact exceeds b) is at most $O(n^{-\varepsilon \ln n})$, by the properties of the Laplace distribution. Because of the negligible probability of a false positive, the technique cannot yield $(\varepsilon, 0)$ -differential privacy for any ε .

To apply this methodology to consensus on blocks, as in our discussion of Subsample and Aggregate, view the intermediate results $f(B_1), \dots, f(B_m)$ as a data set and consider some measure of the concentration of these values. Intuitively, if the values are tightly concentrated then we have consensus among the blocks. Of course, we still need to find the correct notion of concentration, one that is meaningful and that has a differentially private instantiation. In a later section we will define and weave together two notions of stability that seem relevant to Subsample and Aggregate: insensitivity (to the removal or addition of a few data points) and stability under subsampling, capturing the notion that a subsample should yield similar results to the full data set.

7.2.1 Example: the scale of a dataset

Given a dataset, a natural question to ask is, “What is the scale, or dispersion, of the dataset?” This is a different question from *data location*, which might be captured by the median or the mean. The data scale is more often captured by the variance or an interquartile range. We will focus on the *interquartile range (IQR)*, a well-known robust estimator for the scale of the data. We begin with some rough intuition. Suppose the data are *i.i.d.* samples drawn from a distribution with cumulative distribution function F . Then $\text{IQR}(F)$, defined as $F^{-1}(3/4) - F^{-1}(1/4)$, is a constant, depending only on F . It might be very large, or very tiny, but either way, if the density of F is sufficiently high at the two quartiles, then, given enough samples from F , the empirical (that is, sample) interquartile distance should be close to $\text{IQR}(F)$.

Our Propose-Test-Release algorithm for the interquartile distance first tests how many database points need to be changed to obtain a data set with a “sufficiently different” interquartile distance. Only if the (noisy) reply is “sufficiently large” will the algorithm release an approximation to the interquartile range of the dataset.

The definition of “sufficiently different” is multiplicative, as an additive notion for difference of scale makes no sense — what would be the right scale for the additive amount? The algorithm therefore works with the logarithm of the scale, which leads to a multiplicative noise

on the IQR. To see this, suppose that, as in what might be the typical case, the sample interquartile distance cannot change by a factor of 2 by modifying a single point. Then the logarithm (base 2) of the sample interquartile has local sensitivity bounded by 1. This lets us privately release an approximation to *the logarithm of* the sample interquartile range by adding to this value a random draw from $\text{Lap}(1/\varepsilon)$.

Let $\text{IQR}(x)$ denote the sample interquartile range when the data set is x . The algorithm is (implicitly) *proposing* to add noise drawn from $\text{Lap}(1/\varepsilon)$ to the value $\log_2(\text{IQR}(x))$. To *test* whether this magnitude of noise is sufficient for differential privacy, we discretize \mathbb{R} into disjoint bins $\{[k \ln 2, (k+1) \ln 2)\}_{k \in \mathbf{Z}}$ and ask how many data points must be modified in order to obtain a new database, the logarithm (base 2) of whose interquartile range is in a different bin than that of $\log_2(\text{IQR}(x))$. If the answer is at least two then the local sensitivity (of the logarithm of the interquartile range) is bounded by the bin width. We now give more details.

To understand the choice of bin size, we write

$$\log_2(\text{IQR}(x)) = \frac{\ln \text{IQR}(x)}{\ln 2} = \frac{c \ln 2}{\ln 2},$$

whence we find that looking at $\ln(\text{IQR}(x))$ on the scale of $\ln 2$ is equivalent to looking at $\log_2(\text{IQR}(x))$ on the scale of 1. Thus we have scaled bins which are intervals whose endpoints are a pair of adjacent integers: $B_k = [k, k + 1)$, $k \in \mathbf{Z}$, and we let $k_1 = \lfloor \log_2(\text{IQR}(x)) \rfloor$, so $\log_2(\text{IQR}(x)) \in [k_1, k_1 + 1)$ and we say informally that the logarithm of the IQR is in bin k_1 . Consider the following testing query:

Q₀ : How many data points need to change in order to get a new database z such that $\log_2(\text{IQR}(z)) \notin B_{k_1}$?

Let $A_0(x)$ be the true answer to **Q₀** when the database is x . If $A_0(x) \geq 2$, then neighbors y of x satisfy $|\log_2(\text{IQR}(y)) - \log_2(\text{IQR}(x))| \leq 1$. That is, they are close to each other. This is not equivalent to being in the same interval in the discretization: $\log_2(\text{IQR}(x))$ may lie close to one of the endpoints of the interval $[k_1, k_1 + 1)$ and $\log_2(\text{IQR}(y))$ may lie just on the other side of the endpoint. Letting $R_0 = A_0(x) + \text{Lap}(1/\varepsilon)$, a small R_0 , even when the

draw from the Laplace distribution has small magnitude, might not actually indicate high sensitivity of the interquartile range. To cope with the case that the local sensitivity is very small, but $\log_2(\text{IQR}(x))$ is very close to the boundary, we consider a second discretization $\{B_k^{(2)} = [k-0.5, k+0.5]\}_{k \in \mathbf{Z}}$. We denote the two discretizations by $B^{(1)}$ and $B^{(2)}$ respectively. The value $\log_2(\text{IQR}(x))$ — indeed, any value — cannot be close to a boundary in both discretizations. The test is passed if R_0 is large in at least one discretization.

The **Scale** algorithm (Algorithm 12) below for computing database scale assumes that n , the size of the database, is known, and the distance query (“How far to a database whose interquartile range has sensitivity exceeding b ?”) is asking how many points must be *moved* to reach a database with high sensitivity of the IQR. We can avoid this assumption by having the algorithm first ask the (sensitivity 1) query: “How many data points are in x ?”. We remark that, for technical reasons, to cope with the case $\text{IQR}(x) = 0$, we define $\log 0 = -\infty$, $[-\infty] = -\infty$, and let $[-\infty, -\infty) = \{-\infty\}$.

Algorithm 12 The **Scale** Algorithm (releasing the interquartile range)

Require: dataset: $x \in \mathcal{X}^*$, privacy parameters: $\epsilon, \delta > 0$

```

1: for the  $j$ th discretization ( $j = 1, 2$ ) do
2:   Compute  $R_0(x) = A_0(x) + z_0$ , where  $z_0 \in_R \text{Lap}(1/\varepsilon)$ .
3:   if  $R_0 \leq 1 + \ln(1/\delta)$  then
4:     Let  $s^{(j)} = \perp$ .
5:   else
6:     Let  $s^{(j)} = \text{IQR}(x) \times 2^{z_s^{(j)}}$ , where  $z_s^{(j)} \sim \text{Lap}(1/\varepsilon)$ .
7:   end if
8: end for
9: if  $s^{(1)} \neq \perp$  then
10:  Return  $s^{(1)}$ .
11: else
12:  Return  $s^{(2)}$ .
13: end if
```

Note that the algorithm is efficient: let $x_{(1)}, x_{(2)}, \dots, x_{(n)}$ denote the n database points *after sorting*, and let $x(m)$ denote the median, so $m = \lfloor (n+1)/2 \rfloor$. Then the local sensitivity of the median is $\max\{x(m) - x(m-1), x(m+1) - x(m)\}$ and, more importantly, one can compute $A_0(x)$ by considering $O(n)$ sliding intervals with width 2^{k_1} and 2^{k_1+1} , each having one endpoint in x . The computational cost for each interval is constant.

We will not prove convergence bounds for this algorithm because, for the sake of simplicity, we have used a base for the logarithm that is far from optimal (a better base is $1 + 1/\ln n$). We briefly outline the steps in the proof of privacy.

Theorem 7.1. Algorithm **Scale** (Algorithm 12) is $(4\varepsilon, \delta)$ -differentially private.

Proof. (Sketch.) Letting s be shorthand for the result obtained with a single discretization, and defining $\mathcal{D}_0 = \{x : A_0(x) \geq 2\}$, the proof shows:

1. The worst-case sensitivity of query \mathbf{Q}_0 is at most 1.
2. Neighboring databases are almost equally likely to result in \perp :
For all neighboring database x, y :

$$\Pr[s = \perp | x] \leq e^\varepsilon \Pr[s = \perp | y].$$

3. Databases not in \mathcal{D}_0 are unlikely to pass the test:

$$\forall x \notin \mathcal{D}_0 : \Pr[s \neq \perp | x] \leq \frac{\delta}{2}.$$

4. $\forall C \in \mathbb{R}^+, x \in \mathcal{D}_0$ and all neighbors y of x :

$$\Pr[s \in C | x] \leq e^{2\varepsilon} \Pr[s \in C | y].$$

Thus, we get $(2\varepsilon, \delta/2)$ -differential privacy for each discretization. Applying Theorem 3.16 (Appendix B), which says that “the epsilons and the deltas add up,” yields $(4\varepsilon, \delta)$ -differential privacy. \square

7.3 Stability and privacy

7.3.1 Two notions of stability

We begin by making a distinction between the two notions of stability intertwined in this section: stability under subsampling, which yields similar results under random subsamples of the data, and perturbation stability, or low local sensitivity, for a given dataset. In this section we will define and make use of extreme versions of both of these.

- *Subsampling stability:* We say f is q -subsampling stable on x if $f(\hat{x}) = f(x)$ with probability at least $3/4$ when \hat{x} is a random subsample from x which includes each entry independently with probability q . We will use this notion in Algorithm $\mathcal{A}_{\text{samp}}$, a variant of Sample and Aggregate.
- *Perturbation Stability:* We say that f is *stable* on x if f takes the value $f(x)$ on all of the neighbors of x (and *unstable* otherwise). In other words, f is stable on x if the local sensitivity of f on x is zero. We will use this notion (implemented in Algorithm $\mathcal{A}_{\text{dist}}$ below) for the aggregation step of $\mathcal{A}_{\text{samp}}$.

At the heart of Algorithm $\mathcal{A}_{\text{samp}}$ is a relaxed version of perturbation stability, where instead of requiring that the value be unchanged on neighboring databases — a notion that makes sense for arbitrary ranges, including arbitrary discrete ranges — we required only that the value be “close” on neighboring databases — a notion that requires a metric on the range.

Functions f with arbitrary ranges, and in particular the problem of aggregating outputs in Subsample and Aggregate, motivate the next algorithm, $\mathcal{A}_{\text{dist}}$. On input f, x , $\mathcal{A}_{\text{dist}}$ outputs $f(x)$ with high probability if x is at distance at least $\frac{2\log(1/\delta)}{\varepsilon}$ from the nearest *unstable* data set. The algorithm is conceptually trivial: compute the distance to the nearest unstable data set, add Laplace noise $\text{Lap}(1/\varepsilon)$, and check that this noisy distance is at least $\frac{2\log(1/\delta)}{\varepsilon}$. If so, release $f(x)$, otherwise output \perp . We now make this a little more formal.

We begin by defining a quantitative measure of perturbation stability.

Definition 7.2. A function $f : \mathcal{X}^* \rightarrow \mathcal{R}$ is *k-stable* on input x if adding or removing any k elements from x does not change the value of f , that is, $f(x) = f(y)$ for all y such that $|x \Delta y| \leq k$. We say f is *stable* on x if it is (at least) 1-stable on x , and *unstable* otherwise.

Definition 7.3. The *distance to instability* of a data set $x \in \mathcal{X}^*$ with respect to a function f is the number of elements that must be added to or removed from x to reach a data set that is not stable under f .

Note that f is *k*-stable on x if and only if the distance of x to instability is at least k .

Algorithm $\mathcal{A}_{\text{dist}}$, an instantiation of Propose-Test-Release for discrete-valued functions g , appears in Figure 13.

Algorithm 13 $\mathcal{A}_{\text{dist}}$ (releasing $g(x)$ based on distance to instability)

Require: dataset: $x \in \mathcal{X}^*$, privacy parameters: $\epsilon, \delta > 0$, function $g : \mathcal{X}^* \rightarrow \mathbb{R}$

```

1:  $d \leftarrow$  distance from  $x$  to nearest unstable instance
2:  $\hat{d} \leftarrow d + \text{Lap}(1/\epsilon)$ 
3: if  $\hat{d} > \frac{\log(1/\delta)}{\epsilon}$  then
4:   Output  $g(x)$ 
5: else
6:   Output  $\perp$ 
7: end if

```

The proof of the following proposition is immediate from the properties of the Laplace distribution.

Proposition 7.2. For every function g :

1. $\mathcal{A}_{\text{dist}}$ is (ϵ, δ) -differentially private.
2. For all $\beta > 0$: if g is $\frac{\ln(1/\delta) + \ln(1/\beta)}{\epsilon}$ -stable on x , then $\mathcal{A}_{\text{dist}}(x) = g(x)$ with probability at least $1 - \beta$, where the probability space is the coin flips of $\mathcal{A}_{\text{dist}}$.

This distance-based result is the best possible, in the following sense: if there are two data sets x and y for which $\mathcal{A}_{\text{dist}}$ outputs different

values $g(x)$ and $g(y)$, respectively, with at least constant probability, then the distance from x to y must be $\Omega(\log(1/\delta)/\varepsilon)$.

Distance to instability can be difficult to compute, or even to lower bound, so this is not in general a practical solution. Two examples where distance to instability turns out to be easy to bound are the median and the mode (most frequently occurring value).

$\mathcal{A}_{\text{dist}}$ may also be unsatisfactory if the function, say f , is not stable on the specific datasets of interest. For example, suppose f is not stable because of the presence of a few outliers in x . Instances of the average behave this way, although for this function there are well known robust alternatives such as the winsorized mean, the trimmed mean, and the median. By what about for general functions f ? Is there a method of “forcing” an arbitrary f to be stable on a database x ?

This will be the goal of $\mathcal{A}_{\text{samp}}$, a variant of Subsample and Aggregate that outputs $f(x)$ with high probability (over its own random choices) whenever f is subsampling stable on x .

7.3.2 Algorithm $\mathcal{A}_{\text{samp}}$

In $\mathcal{A}_{\text{samp}}$, the blocks B_1, \dots, B_m are chosen *with replacement*, so that each block has the same distribution as the inputs (although now an element of x may appear in multiple blocks). We will call these subsampled datasets $\hat{x}_1, \dots, \hat{x}_m$. The intermediate outputs $z = \{f(\hat{x}_1), \dots, f(\hat{x}_m)\}$ are then aggregated via $\mathcal{A}_{\text{dist}}$ with function $g = \text{mode}$. The distance measure used to estimate the stability of the mode on z is a scaled version of the difference between the popularity of the mode and that of the second most frequent value. Algorithm $\mathcal{A}_{\text{samp}}$, appears in Figure 14. Its running time is dominated by running f about $1/q^2$ times; hence it is efficient whenever f is.

The key property of Algorithm $\mathcal{A}_{\text{samp}}$ is that, on input f, x , it outputs $f(x)$ with high probability, over its own random choices, whenever f is q -subsampling stable on x for $q = \frac{\varepsilon}{64\log(1/\delta)}$. This result has an important statistical interpretation. Recall the discussion of model selection from Example 7.1. Given a collection of models, the *sample complexity* of model selection is the number of samples from a distribution in one of the models necessary to select the correct model

with probability at least 2/3. The result says that *differentially private* model selection increases the sample complexity of (non-private) model selection by a problem-independent (and range-independent) factor of $O(\log(1/\delta)/\varepsilon)$.

Algorithm 14 $\mathcal{A}_{\text{samp}}$: Bootstrapping for Subsampling-Stable f

Require: dataset: x , function $f : \mathcal{X}^* \rightarrow \mathbb{R}$, privacy parameters $\epsilon, \delta > 0$.

- 1: $q \leftarrow \frac{\epsilon}{64 \ln(1/\delta)}$, $m \leftarrow \frac{\log(n/\delta)}{q^2}$.
- 2: Subsample m data sets $\hat{x}_1, \dots, \hat{x}_m$ from x , where \hat{x}_i includes each position of x independently with probability q .
- 3: **if** some element of x appears in more than $2mq$ sets \hat{x}_i **then**
- 4: Halt and output \perp .
- 5: **else**
- 6: $z \leftarrow \{f(\hat{x}_1), \dots, f(\hat{x}_m)\}$.
- 7: For each $r \in \mathbb{R}$, let $\text{count}(r) = \#\{i : f(\hat{x}_i) = r\}$.
- 8: Let $\text{count}_{(i)}$ denote the i th largest count, $i = 1, 2$.
- 9: $d \leftarrow (\text{count}_{(1)} - \text{count}_{(2)})/(4mq) - 1$
- 10: **Comment** Now run $\mathcal{A}_{\text{dist}}(g, z)$ using d to estimate distance to instability:
- 11: $\hat{d} \leftarrow d + \text{Lap}(\frac{1}{\epsilon})$.
- 12: **if** $\hat{d} > \ln(1/\delta)/\varepsilon$ **then**
- 13: Output $g(z) = \text{mode}(z)$.
- 14: **else**
- 15: Output \perp .
- 16: **end if**
- 17: **end if**

Theorem 7.3.

1. Algorithm $\mathcal{A}_{\text{samp}}$ is (ε, δ) -differentially private.
2. If f is q -subsampling stable on input x where $q = \frac{\varepsilon}{64 \ln(1/\delta)}$, then algorithm $\mathcal{A}_{\text{samp}}(x)$ outputs $f(x)$ with probability at least $1 - 3\delta$.
3. If f can be computed in time $T(n)$ on inputs of length n , then $\mathcal{A}_{\text{samp}}$ runs in expected time $O(\frac{\log n}{q^2})(T(qn) + n)$.

Note that the utility statement here is an input-by-input guarantee; f need not be q -subsampling stable on all inputs. Importantly, there is no dependence on the size of the range \mathcal{R} . In the context of model selection, this means that one can efficiently satisfy differential privacy with a modest blowup in sample complexity (about $\log(1/\delta)/\varepsilon$) whenever there is a particular model that gets selected with reasonable probability.

The proof of privacy comes from the insensitivity of the computation of d , the privacy of the Propose-Test-Release technique, and the privacy of Subsample and Aggregate, modified slightly to allow for the fact that this algorithm performs sampling with replacement and thus the aggregator has higher sensitivity, since any individual might affect up to $2mq$ blocks. The main observation for analyzing the utility of this approach is that the stability of the *mode* is a function of the difference between the frequency of the mode and that of the next most popular element. The next lemma says that if f is subsampling stable on x , then x is far from unstable with respect to the mode $g(z) = g(f(\hat{x}_1), \dots, f(\hat{x}_m))$ (but not necessarily with respect to f), and moreover one can estimate the distance to instability of x *efficiently* and privately.

Lemma 7.4. Fix $q \in (0, 1)$. Given $f : \mathcal{X}^* \rightarrow \mathcal{R}$, let $\hat{f} : \mathcal{X}^* \rightarrow \mathcal{R}$ be the function $\hat{f} = \text{mode}(f(\hat{x}_1), \dots, f(\hat{x}_m))$ where each \hat{x}_i includes each element of x independently with probability q and $m = \ln(n/\delta)/q^2$. Let $d(z) = (\text{count}_{(1)} - \text{count}_{(2)})/(4mq) - 1$; that is, given a “database” z of values, $d(z) + 1$ is a scaled difference between the number of occurrences of the two most popular values. Fix a data set x . Let E be the event that no position of x is included in more than $2mq$ of the subsets \hat{x}_i . Then, when $q \leq \varepsilon/64 \ln(1/\delta)$ we have:

1. E occurs with probability at least $1 - \delta$.
2. Conditioned on E , d lower bounds the stability of \hat{f} on x , and d has global sensitivity 1.
3. If f is q -subsampling stable on x , then with probability at least $1 - \delta$ over the choice of subsamples, we have $\hat{f}(x) = f(x)$, and, conditioned on this event, the final test will be passed with

probability at least $1 - \delta$, where the probability is over the draw from $\text{Lap}(1/\varepsilon)$.

The events in Parts 2 and 3 occur simultaneously with probability at least $1 - 2\delta$.

Proof. Part 1 follows from the Chernoff bound. To prove Part 2, notice that, conditioned on the event E , adding or removing one entry in the original data set changes any of the counts $\text{count}_{(r)}$ by at most $2mq$. Therefore, $\text{count}_{(1)} - \text{count}_{(2)}$ changes by at most $4mq$. This in turn means that $d(f(\hat{x}_1), \dots, f(\hat{x}_m))$ changes by at most one for any x and hence has global sensitivity of one. This also implies that d lower bounds the stability of \hat{f} on x .

We now turn to part 3. We want to argue two facts:

1. If f is q -subsampling stable on x , then there is likely to be a large gap between the counts of the two most popular bins. Specifically, we want to show that with high probability $\text{count}_{(1)} - \text{count}_{(2)} \geq m/4$. Note that if the most popular bin has count at least $5m/8$ then the second most popular bin can have count at most $3m/8$, with a difference of $m/4$. By definition of subsampling stability the most popular bin has an expected count of at least $3m/4$ and hence, by the Chernoff bound, taking $\alpha = 1/8$, has probability at most $e^{-2m\alpha^2} = e^{-m/32}$ of having a count less than $5m/8$. (All the probabilities are over the subsampling.)
2. When the gap between the counts of the two most popular bins is large, then the algorithm is unlikely to fail; that is, the test is likely to succeed. The worry is that the draw from $\text{Lap}(\frac{1}{\varepsilon})$ will be negative and have large absolute value, so that \hat{d} falls below the threshold $(\ln(1/\delta)/\varepsilon)$ even when d is large. To ensure this happens with probability at most δ it suffices that $d > 2 \ln(1/\delta)/\varepsilon$. By definition, $d = (\text{count}_{(1)} - \text{count}_{(2)})/(4mq) - 1$, and, assuming we are in the high probability case just described, this implies

$$d \geq \frac{m/4}{4mq} - 1 = \frac{1}{16q} - 1$$

so it is enough to have

$$\frac{1}{16q} > 2 \ln(1/\delta)/\varepsilon.$$

Taking $q \leq \varepsilon/64 \ln(1/\delta)$ suffices.

Finally, note that with these values of q and m we have $e^{-m/32} < \delta$. \square

Example 7.3. [The Raw Data Problem] Suppose we have an analyst whom we can trust to follow instructions and only publish information obtained according to these instructions. Better yet, suppose we have b such analysts, and we can trust them not to communicate among themselves. The analysts do not need to be identical, but they do need to be considering a common set of *options*. For example, these options might be different statistics in a fixed set S of possible statistics, and in this first step the analyst's goal is to choose, for eventual publication, the most significant statistic in S . Later, the chosen statistic will be recomputed in a differentially private fashion, and the result can be published.

As described the procedure is not private at all: the *choice* of statistic made in the first step may depend on the data of a single individual! Nonetheless, we can use the Subsample-and-Aggregate framework to carry out the first step, with the i th analyst receiving a subsample of the data points and applying to this smaller database the function f_i to obtain an option. The options are then aggregated as in algorithm $\mathcal{A}_{\text{samp}}$; if there is a clear winner this is overwhelmingly likely to be the selected statistic. This was *chosen* in a differentially private manner, and in the second step it will be *computed* with differential privacy.

Bibliographic Notes

Subsample and Aggregate was invented by Nissim, Raskhodnikova, and Smith [68], who were the first to define and exploit low local sensitivity. Propose-Test-Release is due to Dwork and Lei [22], as is the algorithm for releasing the interquartile range. The discussion of stability and privacy, and Algorithm $\mathcal{A}_{\text{samp}}$ which blends these two techniques, is due to Smith and Thakurta [80]. This paper demonstrates the power of

$\mathcal{A}_{\text{samp}}$ by analyzing the subsampling stability conditions of the famous LASSO algorithm and showing that differential privacy can be obtained “for free,” via (a generalization of $\mathcal{A}_{\text{samp}}$), precisely under the (fixed data as well as distributional) conditions for which LASSO is known to have good explanatory power.

8

Lower Bounds and Separation Results

In this section, we investigate various lower bounds and tradeoffs:

1. How *inaccurate* must responses be in order not to completely destroy any reasonable notion of privacy?
2. How does the answer to the previous question depend on the number of queries?
3. Can we separate $(\varepsilon, 0)$ -differential privacy from (ε, δ) -differential privacy in terms of the accuracy each permits?
4. Is there an intrinsic difference between what can be achieved for linear queries and for arbitrary low-sensitivity queries while maintaining $(\varepsilon, 0)$ -differential privacy?

A different flavor of separation result distinguishes the computational complexity of generating a *data structure* handling all the queries in a given class from that of generating a *synthetic database* that achieves the same goal. We postpone a discussion of this result to Section 9.

8.1 Reconstruction attacks

We argued in Section 1 that any non-trivial mechanism must be randomized. It follows that, at least for some database, query, and choice of random bits, the response produced by the mechanism is not perfectly accurate. The question of how *inaccurate* answers must be in order to protect privacy makes sense in all computational models: interactive, non-interactive, and the models discussed in Section 12.

For the lower bounds on distortion, we assume for simplicity that the database consists of a single — but very sensitive — bit per person, so we can think of the database as an n -bit Boolean vector $d = (d_1, \dots, d_n)$. This is an abstraction of a setting in which the database rows are quite complex, for example, they may be medical records, but the attacker is interested in one specific field, such as the presence or absence of the sickle cell trait. The abstracted attack consists of issuing a string of queries, each described by a subset S of the database rows. The query is asking how many 1’s are in the selected rows. Representing the query as the n -bit characteristic vector \mathbf{S} of the set S , with 1s in all the positions corresponding to rows in S and 0s everywhere else, the true answer to the query is the inner product $A(S) = \sum_{i=1}^n d_i \mathbf{S}_i$.

Fix an arbitrary privacy mechanism. We will let $r(S)$ denote the response to the query S . This may be obtained explicitly, say, if the mechanism is interactive and the query S is issued, or if the mechanism is given all the queries in advance and produces a list of answers, or implicitly, which occurs if the mechanism produces a synopsis from which the analysts extracts $r(S)$. Note that $r(S)$ may depend on random choices made by the mechanism and the history of queries. Let $E(S, r(S))$ denote the *error*, also called *noise* or *distortion*, of the response $r(S)$, so $E(S, r(S)) = |A(S) - r(S)|$.

The question we want to ask is, “How much noise is needed in order to preserve privacy?” Differential privacy is a specific privacy guarantee, but one might also consider weaker notions, so rather than guaranteeing privacy the modest goal in the lower bound arguments will simply be to prevent privacy catastrophes.

Definition 8.1. A mechanism is *blatantly non-private* if an adversary can construct a candidate database c that agrees with the real database d in all but $o(n)$ entries, i.e., $\|c - d\|_0 \in o(n)$.

In other words, a mechanism is blatantly non-private if it permits a reconstruction attack that allows the adversary to correctly guess the secret bit of all but $o(n)$ members of the database. (There is no requirement that the adversary know on which answers it is correct.)

Theorem 8.1. Let \mathcal{M} be a mechanism with distortion of magnitude bounded by E . Then there exists an adversary that can reconstruct the database to within $4E$ positions.

An easy consequence of the theorem is that a privacy mechanism adding noise with magnitude always bounded by, say, $n/401$, permits an adversary to correctly reconstruct 99% of the entries.

Proof. Let d be the true database. The adversary attacks in two phases:

1. **Estimate the number of 1s in all possible sets:** Query \mathcal{M} on all subsets $S \subseteq [n]$.
2. **Rule out “distant” databases:** For every candidate database $c \in \{0, 1\}^n$, if $\exists S \subseteq [n]$ such that $|\sum_{i \in S} c_i - \mathcal{M}(S)| > E$, then rule out c . If c is not ruled out, then output c and halt.

Since $\mathcal{M}(S)$ never errs by more than E , the real database will not be ruled out, so this simple (but inefficient!) algorithm will output *some* candidate database c . We will argue that the number of positions in which c and d differ is at most $4 \cdot E$.

Let I_0 be the indices in which $d_i = 0$, that is, $I_0 = \{i \mid d_i = 0\}$. Similarly, define $I_1 = \{i \mid d_i = 1\}$. Since c was not ruled out, $|\mathcal{M}(I_0) - \sum_{i \in I_0} c_i| \leq E$. However, by assumption $|\mathcal{M}(I_0) - \sum_{i \in I_0} d_i| \leq E$. It follows from the triangle inequality that c and d differ in at most $2E$ positions in I_0 ; the same argument shows that they differ in at most $2E$ positions in I_1 . Thus, c and d agree on all but at most $4E$ positions. \square

What if we consider more realistic bounds on the number of queries? We think of \sqrt{n} as an interesting threshold on noise, for the following reason: if the database contains n people drawn uniformly at random

from a population of size $N \gg n$, and the fraction of the population satisfying a given condition is p , then we expect the number of rows in the database satisfying the property to be roughly $np \pm \Theta(\sqrt{n})$, by the properties of the binomial distribution. That is, the sampling error is on the order of \sqrt{n} . We would like that the noise introduced for privacy is smaller than the sampling error, ideally $o(\sqrt{n})$. The next result investigates the feasibility of such small error when the number of queries is linear in n . The result is negative.

Ignoring computational complexity, to see why there might exist a query-efficient attack we modify the problem slightly, looking at databases $d \in \{-1, 1\}^n$ and query vectors $v \in \{-1, 1\}^n$. The true answer is again defined to be $d \cdot v$, and the response is a noisy version of the true answer. Now, consider a candidate database c that is far from d , say, $\|c - d\|_0 \in \Omega(n)$. For a random $v \in_R \{-1, 1\}^n$, with constant probability we have $(c - d) \cdot v \in \Omega(\sqrt{n})$. To see this, fix $x \in \{-1, 1\}^n$ and choose $v \in_R \{-1, 1\}^n$. Then $x \cdot v$ is a sum of independent random variables $x_i v_i \in \{-1, 1\}$, which has expectation 0 and variance n , and is distributed according to a scaled and shifted binomial distribution. For the same reason, if c and d differ in at least αn rows, and v is chosen at random, then $(c - d) \cdot v$ is binomially distributed with mean 0 and variance at least αn . Thus, we expect $c \cdot v$ and $d \cdot v$ to differ by at least $\alpha\sqrt{n}$ with constant probability, by the properties of the binomial distribution. Note that we are using the *anti*-concentration property of the distribution, rather than the usual appeal to concentration.

This opens an attack for ruling out c when the noise is constrained to be $o(\sqrt{n})$: compute the difference between $c \cdot v$ and the noisy response $r(v)$. If the magnitude of this difference exceeds \sqrt{n} — which will occur with constant probability over the choice of v — then rule out c . The next theorem formalizes this argument and further shows that the attack is resilient even to a large fraction of completely arbitrary responses: Using a linear number of ± 1 questions, an attacker can reconstruct almost the whole database if the curator is constrained to answer at least $\frac{1}{2} + \eta$ of the questions within an absolute error of $o(\sqrt{n})$.

Theorem 8.2. For any $\eta > 0$ and any function $\alpha = \alpha(n)$, there is a constant b and an attack using $bn \pm 1$ questions that reconstructs a

database that agrees with the real database in all but at most $(\frac{2\alpha}{\eta})^2$ entries, if the curator answers at least $\frac{1}{2} + \eta$ of the questions within an absolute error of α .

Proof. We begin with a simple lemma.

Lemma 8.3. Let $Y = \sum_{i=1}^k X_i$ where each X_i is a ± 2 independent Bernoulli random variable with mean zero. Then for any y and any $\ell \in \mathbb{N}$, $\Pr[Y \in [2y, 2(y + \ell)]] \leq \frac{\ell+1}{\sqrt{k}}$.

Proof. Note that Y is always even and that $\Pr[Y = 2y] = \binom{k}{(k+y)/2} (\frac{1}{2})^k$. This expression is at most $\binom{k}{\lceil k/2 \rceil} (\frac{1}{2})^k$. Using Stirling's approximation, which says that $n!$ can be approximated by $\sqrt{2n\pi}(n/e)^n$, this is bounded by $\sqrt{\frac{2}{\pi k}}$. The claim follows by a union bound over the $\ell+1$ possible values for Y in $[2y, 2(y + \ell)]$. \square

The adversary's attack is to choose bn random vectors $v \in \{-1, 1\}^n$, obtain responses (y_1, \dots, y_{bn}) , and then output any database c such that $|y_i - (Ac)_i| \leq \alpha$ for at least $\frac{1}{2} + \eta$ of the indices i , where A is the $bn \times n$ matrix whose rows are the random query vectors v .

Let the true database be d and let c be the reconstructed database. By assumption on the behavior of the mechanism, $|(Ad)_i - y_i| \leq \alpha$ for a $1/2 + \eta$ fraction of $i \in [bn]$. Since c was not ruled out, we also have that $|(Ac)_i - y_i| \leq \alpha$ for a $1/2 + \eta$ fraction of $i \in [bn]$. Since any two such sets of indices agree on at least a 2η fraction of $i \in [bn]$, we have from the triangle inequality that for at least $2\eta bn$ values of i , $|(c - d)_i| \leq 2\alpha$.

We wish to argue that c agrees with d in all but $(\frac{2\alpha}{\eta})^2$ entries. We will show that if the reconstructed c is far from d , disagreeing on at least $(2\alpha/\eta)^2$ entries, the probability that a randomly chosen A will satisfy $|(A(c-d))_i| \leq 2\alpha$ for at least $2\eta bn$ values of i will be extremely small — so small that, for a random A , it is extremely unlikely that there even exists a c far from d that is not eliminated by the queries in A .

Assume the vector $z = (c - d) \in \{-2, 0, 2\}^n$ has Hamming weight at least $(\frac{2\alpha}{\eta})^2$, so c is far from d . We have argued that, since c is produced by the attacker, $|(Az)_i| \leq 2\alpha$ for at least $2\eta bn$ values of i . We shall call such a z *bad with respect to A*. We will show that, with high probability over the choice of A , no z is bad with respect to A .

For any i , $v_i z$ is the sum of at least $(\frac{2\alpha}{\eta})^2 \pm 2$ random values. Letting $k = (2\alpha/\eta)^2$ and $\ell = 2\alpha$, we have by Lemma 8.3 that the probability that $v_i z$ lies in an interval of size 4α is at most η , so the expected number of queries for which $|v_i z| \leq 2\alpha$ is at most $\eta b n$. Chernoff bounds now imply that the probability that this number exceeds $2\eta b n$ is at most $\exp(-\frac{\eta b n}{4})$. Thus the probability of a particular $z = c - d$ being bad with respect to A is at most $\exp(-\frac{\eta b n}{4})$.

Taking a union bound over the atmost 3^n possible z s, we get that with probability at least $1 - \exp(-n(\frac{\eta b}{4} - \ln 3))$, no bad z exists. Taking $b > 4 \ln 3 / \eta$, the probability that such a bad z exists is exponentially small in n . \square

Preventing blatant non-privacy is a very low bar for a privacy mechanism, so if differential privacy is meaningful then lower bounds for preventing blatant non-privacy will also apply to any mechanism ensuring differential privacy. Although for the most part we ignore computational issues in this monograph, there is also the question of the efficiency of the attack. Suppose we were able to prove that (perhaps under some computational assumption) there exist low-distortion mechanisms that are “hard” to break; for example, mechanisms for which producing a candidate database c close to the original database is hard? Then, although a low-distortion mechanism might fail to be differentially private in theory, it could conceivably provide privacy against bounded adversaries. Unfortunately, this is not the case. In particular, when the noise is always in $o(\sqrt{n})$, there is an efficient attack using exactly n fixed queries; moreover, there is even a computationally efficient attack requiring a linear number of queries in which a 0.239 fraction may be answered with wild noise.

In the case of “internet scale” data sets, obtaining responses to n queries is infeasible, as n is extremely large, say, $n \geq 10^8$. What happens if the curator permits only a sublinear number of questions? This inquiry led to the first algorithmic results in (what has evolved to be) (ε, δ) -differential privacy, in which it was shown how to maintain privacy against a sublinear number of counting queries by adding binomial noise of order $o(\sqrt{n})$ — less than the sampling error! — to each true answer. Using the tools of differential privacy we can do this either

using either (1) the Gaussian mechanism or (2) the Laplace mechanism and advanced composition.

8.2 Lower bounds for differential privacy

The results of the previous section yielded lower bounds on distortion needed to ensure any reasonable notion of privacy. In contrast, the result in this section is specific to differential privacy. Although some of the details in the proof are quite technical, the main idea is elegant: suppose (somehow) the adversary has narrowed down the set of possible databases to a relatively small set S of 2^s vectors, where the L_1 distance between each pair of vectors is some large number Δ . Suppose further that we can find a k -dimensional query F , 1-Lipschitz in each of its output coordinates, with the property that the true answers to the query look very different (in L_∞ norm) on the different vectors in our set; for example, the distance on any two elements in the set may be $\Omega(k)$. It is helpful to think geometrically about the “answer space” \mathbb{R}^k . Each element x in the set S gives rise to a vector $F(x)$ in answer space. The actual response will be a perturbation of this point in answer space. Then a volume-based pigeon hole argument (in answer space) shows that, if with even moderate probability the (noisy) responses are “reasonably” close to the true answers, then ϵ cannot be very small.

This stems from the fact that for $(\epsilon, 0)$ -differentially private mechanisms \mathcal{M} , for *arbitrarily different* databases x, y , any response in the support of $\mathcal{M}(x)$ is also in the support of $\mathcal{M}(y)$. Taken together with the construction of an appropriate collection of vectors and a (contrived, non-counting) query, the result yields a lower bound on distortion that is linear k/ϵ . The argument appeals to Theorem 2.2, which discusses group privacy. In our case the group in question corresponds to the indices contributing to the (L_1) distance between a pair of vectors in S .

8.2.1 Lower bound by packing arguments

We begin with an observation which says, intuitively, that if the “likely” response regions, when the query is F , are disjoint, then we can bound

ϵ from below, showing that privacy can't be too good. When $\|F(x_i) - F(x_j)\|_\infty$ is large, this says that to get very good privacy, even when restricted to databases that differ in many places, we must get very erroneous responses on some coordinate of F .

The argument uses the histogram representation of databases. In the sequel, $d = |\mathcal{X}|$ denotes the size of the universe from which database elements are drawn.

Lemma 8.4. Assume the existence of a set $S = \{x_1, \dots, x_{2^s}\}$, where each $x_i \in \mathbb{N}^d$, such that for $i \neq j$, $\|x_i - x_j\|_1 \leq \Delta$. Further, let $F : \mathbb{N}^d \rightarrow \mathbb{R}^k$ be a k -dimensional query. For $1 \leq i \leq 2^s$, let B_i denote a region in \mathbb{R}^k , the answer space, and assume that the B_i are mutually disjoint. If \mathcal{M} is an $(\epsilon, 0)$ -differentially private mechanism for F such that, $\forall 1 \leq i \leq 2^s$, $\Pr[\mathcal{M}(x_i) \in B_i] \geq 1/2$, then $\epsilon \geq \frac{\ln(2)(s-1)}{\Delta}$.

Proof. By assumption $\Pr[\mathcal{M}(x_j) \in B_j] \geq 2^{-1}$. Since the regions B_1, \dots, B_{2^s} are disjoint, $\exists j \neq i \in [2^s]$ such that $\Pr[\mathcal{M}(x_i) \in B_j] \leq 2^{-s}$. That is, for at least one of the $2^s - 1$ regions B_j , the probability that $\mathcal{M}(x_i)$ is mapped to this B_j is at most 2^{-s} . Combining this with differential privacy, we have

$$\frac{2^{-1}}{2^{-s}} \leq \frac{\Pr[\mathcal{M}(B_j|x_j)]}{\Pr[\mathcal{M}(B_j|x_i)]} \leq \exp(\epsilon\Delta). \quad \square$$

Corollary 8.5. Let $S = \{x_1, \dots, x_{2^s}\}$ be as in Lemma 8.4, and assume that for any $i \neq j$, $\|F(x_i) - F(x_j)\|_\infty \geq \eta$. Let B_i denote the L_∞ ball in \mathbb{R}^k of radius $\eta/2$ centered at x_i . Let \mathcal{M} be any ϵ -differentially private mechanism for F satisfying

$$\forall 1 \leq i \leq 2^s : \Pr[\mathcal{M}(x_i) \in B_i] \geq 1/2.$$

Then $\epsilon \geq \frac{(\ln 2)(s-1)}{\Delta}$.

Proof. The regions B_1, \dots, B_{2^s} are disjoint, so the conditions of Lemma 8.4 are satisfied. The corollary follows by applying the lemma and taking logarithms. \square

In Theorem 8.8 below we will look at queries F that are simply k independently and randomly generated (nonlinear!) queries. For

suitable S and F (we will work to find these) the corollary says that if with probability at least $1/2$ *all* responses simultaneously have small error, then privacy can't be too good. In other words,

Claim 8.6 (Informal Restatement of Corollary 8.5). To obtain $(\varepsilon, 0)$ -differential privacy for $\varepsilon \leq \frac{\ln(2)(s-1)}{\Delta}$, the mechanism must add noise with L_∞ norm greater than $\eta/2$ with probability exceeding $1/2$.

As a warm-up exercise, we prove an easier theorem that requires a large data universe.

Theorem 8.7. Let $\mathcal{X} = \{0, 1\}^k$. Let $\mathcal{M} : \mathcal{X}^n \rightarrow \mathbb{R}^k$ be an $(\varepsilon, 0)$ -differentially private mechanism such that for every database $x \in \mathcal{X}^n$ with probability at least $1/2$ $\mathcal{M}(x)$ outputs all of the 1-way marginals of x with error smaller than $n/2$. That is, for each $j \in [k]$, the j th component of $\mathcal{M}(x)$ should approximately equal the number of rows of x whose j th bit is 1, up to an error smaller than $n/2$. Then $n \in \Omega(k/\varepsilon)$.

Note that this bound is tight to within a constant factor, by the simple composition theorem, and that it separates $(\varepsilon, 0)$ -differential privacy from (ε, δ) -differential privacy, for $\delta \in 2^{-o(n)}$, since, by the advanced composition theorem (Theorem 3.20), Laplace noise with parameter $b = \sqrt{k \ln(1/\delta)}/\varepsilon$ suffices for the former, in contrast to $\Omega(k/\varepsilon)$ needed for the latter. Taking $k \in \Theta(n)$ and, say, $\delta = 2^{-\log^2 n}$, yields the separation.

Proof. For every string $w \in \{0, 1\}^k$, consider the database x_w consisting of n identical rows, all of which equal w . Let $B_w \in \mathbb{R}^k$ consist of all tuples of numbers that provide answers to the 1-way marginals on x with error less than $n/2$. That is,

$$B_w = \{(a_1, \dots, a_k)\} \in \mathbb{R}^k : \forall i \in [k] |a_i - nw_i| < n/2\}.$$

Put differently, B_w is the open ℓ_∞ ball of radius $n/2$ around $nw \in \{0, n\}^k$. Notice that the sets B_w are mutually disjoint.

If M is an accurate mechanism for answering 1-way marginals, then for every w the probability of landing in B_w when the database is x_w should be at least $1/2$: $\Pr[\mathcal{M}(x_w) \in B_w] \geq 1/2$. Thus, setting $\Delta = n$ and $s = k$ in Corollary 8.5 we have $\varepsilon \geq \frac{\ln(2)(s-1)}{\Delta}$. \square

Theorem 8.8. For any $k, d, n \in \mathbb{N}$ and $\varepsilon \in (0, 1/40]$, where $n \geq \min\{k/\varepsilon, d/\varepsilon\}$, there is a query $F : \mathbb{N}^d \rightarrow \mathbb{R}^k$ with per-coordinate sensitivity at most 1 such that any $(\varepsilon, 0)$ -differentially private mechanism adds noise of L_∞ norm $\Omega(\min\{k/\varepsilon, d/\varepsilon\})$ with probability at least 1/2 on some databases of weight at most n .

Note that $d = |\mathcal{X}|$ need not be large here, in contrast to the requirement in Theorem 8.7.

Proof. Let $\ell = \min\{k, d\}$. Using error-correcting codes we can construct a set $S = \{x_1, \dots, x_{2^s}\}$, where $s = \ell/400$, such that each $x_i \in \mathbb{N}^d$ and in addition

1. $\forall i : \|x_i\|_1 \leq w = \ell/(1280\varepsilon)$
2. $\forall i \neq j, \|x_i - x_j\|_1 \geq w/10$

We do not give details here, but we note that the databases in S are of size at most $w < n$, and so $\|x_i - x_j\|_1 \leq 2w$. Taking $\Delta = 2w$ the set S satisfies the conditions of Corollary 8.5. The remainder of our effort is to obtain the queries F to which we will apply Corollary 8.5. Given $S = \{x_1, \dots, x_{2^s}\}$, where each $x_i \in \mathbb{N}^d$, the first step is to define a mapping from the space of histograms to vectors in \mathbb{R}^{2^s} , $\mathcal{L}_S : \mathbb{N}^d \rightarrow \mathbb{R}^{2^s}$. Intuitively (and imprecisely!), given a histogram x , the mapping lists, for each $x_i \in S$, the L_1 distance from x to x_i . More precisely, letting w be an upper bound on the weight of any x_i in our collection we define the mapping as follows.

- For every $x_i \in S$, there is a coordinate i in the mapping.
- The i th coordinate of $\mathcal{L}_S(x)$ is $\max\{w/30 - \|x_i - z\|_1, 0\}$.

Claim 8.9. If x_1, \dots, x_{2^s} satisfy the conditions

1. $\forall i \|x_i\|_1 \leq w$; and
2. $\forall i \neq j \|x_i - x_j\|_1 \geq w/10$

then the map \mathcal{L}_S is 1-Lipschitz; in particular, if $\|z_1 - z_2\|_1 = 1$, then $\|\mathcal{L}_S(z_1) - \mathcal{L}_S(z_2)\|_1 \leq 1$, assuming $w \geq 31$.

Proof. Since we assume $w \geq 31$ we have that if $z \in \mathbb{N}^d$ is close to some $x_i \in S$, meaning $w/30 > \|x_i - z\|_1$, then z cannot be close to any other $x_j \in S$, and the same is true for all $\|z' - z\|_1 \leq 1$. Thus, for any z_1, z_2 such that $\|z_1 - z_2\| \leq 1$, if A denotes the set of coordinates where at least one of $\mathcal{L}_S(z_1)$ or $\mathcal{L}_S(z_2)$ is non-zero, then A is either empty or is a singleton set. Given this, the statement in the claim is immediate from the fact that the mapping corresponding to any particular coordinate is clearly 1-Lipschitz. \square

We can finally describe the queries F . Corresponding to any $r \in \{-1, 1\}^{2^s}$, we define $f_r : \mathbb{N}^d \rightarrow \mathbb{R}$, as

$$f_r(x) = \sum_{i=1}^d \mathcal{L}_S(x)_i \cdot r_i,$$

which is simply the inner product $\mathcal{L}_S \cdot r$. F will be a random map $F : \mathbb{N}^d \rightarrow \mathbb{R}^k$: Pick $r_1, \dots, r_k \in \{-1, 1\}^{2^s}$ independently and uniformly at random and define

$$F(x) = (f_{r_1}(x), \dots, f_{r_k}(x)).$$

That is, $F(x)$ is simply the result of the inner product of $\mathcal{L}_S(x)$ with k randomly chosen ± 1 vectors.

Note that for any $x \in S$ $\mathcal{L}_S(x)$ has one coordinate with value $w/30$ (and the others are all zero), so $\forall r_i \in \{-1, 1\}^{2^s}$ and $x \in S$ we have $|f_{r_i}(x)| = w/30$. Now consider any $x_h, x_j \in S$, where $h \neq j$. It follows that for any $r_i \in \{-1, 1\}^{2^s}$,

$$\Pr_{r_i} [|f_{r_i}(x_h) - f_{r_i}(x_j)| \geq w/15] \geq 1/2$$

(this event occurs when $(r_i)_h = -(r_i)_j$). A basic application of the Chernoff bound implies that

$$\begin{aligned} &\Pr_{r_1, \dots, r_k} [\text{For at least } 1/10 \text{ of the } r_i \text{s,} \\ &|f_{r_i}(x_h) - f_{r_i}(x_j)| \geq w/15] \geq 1 - 2^{-k/30}. \end{aligned}$$

Now, the total number of pairs (x_i, x_j) of databases such that $x_i, x_j \in S$ is at most $2^{2s} \leq 2^{k/200}$. Taking a union bound this implies

$$\begin{aligned} &\Pr_{r_1, \dots, r_k} [\forall h \neq j, \quad \text{For at least } 1/10 \text{ of the } r_i \text{s,} \\ &|f_{r_i}(x_h) - f_{r_i}(x_j)| \geq w/15] \geq 1 - 2^{-k/40} \end{aligned}$$

This implies that we can fix r_1, \dots, r_k such that the following is true.

$$\forall h \neq j, \quad \text{For at least } 1/10 \text{ of the } r_i \text{s, } |f_{r_i}(x_h) - f_{r_i}(x_j)| \geq w/15$$

Thus, for any $x_h \neq x_j \in S$, $\|F(x_h) - F(x_j)\|_\infty \geq w/15$.

Setting $\Delta = 2w$ and $s = \ell/400 > 3\varepsilon w$ (as we did above), and $\eta = w/15$, we satisfy the conditions of Corollary 8.5 and conclude $\Delta \leq (s-1)/\varepsilon$, proving the theorem (via Claim 8.6). \square

The theorem is almost tight: if $k \leq d$ then we can apply the Laplace mechanism to each of the k sensitivity 1 component queries in F with parameter k/ε , and we expect the maximum distortion to be $\Theta(k \ln k / \varepsilon)$. On the other hand, if $d \leq k$ then we can apply the Laplace mechanism to the d -dimensional histogram representing the database, and we expect the maximum distortion to be $\Theta(d \ln d / \varepsilon)$.

The theorem actually shows that, given knowledge of the set S and knowledge that the actual database is an element $x \in S$, the adversary can completely determine x if the L_∞ norm of the distortion is too small. How in real life might the adversary obtain a set S of the type used in the attack? This can occur when a *non-private* database system has been running on a dataset, say, x . For example, x could be a vector in $\{0, 1\}^n$ and the adversary may have learned, through a sequence of linear queries, that $x \in \mathcal{C}$, a linear code of distance, say $n^{2/3}$. Of course, if the database system is not promising privacy there is no problem. The problem arises if the administrator decides to replace the existing system with a differentially private mechanism — after several queries have received noise-free responses. In particular, if the administrator chooses to use (ε, δ) -differential privacy for subsequent k queries then the distortion might fall below the $\Omega(k/\varepsilon)$ lower bound, permitting the attack described in the proof of Theorem 8.8.

The theorem also emphasizes that there is a fundamental difference between auxiliary information about (sets of) members of the database and information about the database *as a whole*. Of course, we already knew this: being told that the number of secret bits sums to exactly 5,000 completely destroys differential privacy, and an adversary that already knew the secret bit of every member of the database except one individual could then conclude the secret bit of the remaining individual.

Additional Consequences. Suppose $k \leq d$, so $\ell = k$ in Theorem 8.8. The linear in k/ε lower bound on noise for k queries sketched in the previous section immediately yields a separation between counting queries and arbitrary 1-sensitivity queries, as the SmallDB construction answers (more than) n queries with noise roughly $n^{2/3}$ while maintaining differential privacy. Indeed, this result also permits us to conclude that there is no small α -net for large sets of arbitrary low sensitivity queries, for $\alpha \in o(n)$ (as otherwise the net mechanism would yield an $(\varepsilon, 0)$ algorithm of desired accuracy).

8.3 Bibliographic notes

The first reconstruction attacks, including Theorem 8.1, are due to Dinur and Nissim [18], who also gave an attack requiring only polynomial time computation and $O(n \log^2 n)$ queries, provided the noise is always $o(\sqrt{n})$. Realizing that attacks requiring n random linear queries, when n is “internet scale,” are infeasible, Dinur, Dwork, and Nissim gave the first positive results, showing that for a sublinear number of subset sum queries, a form of privacy (now known to imply (ε, δ) -differential privacy) can be achieved by adding noise scaled to $o(\sqrt{n})$ [18]. This was exciting because it suggested that, if we think of the database as drawn from an underlying population, then, even for a relatively large number of counting queries, privacy could be achieved with distortion smaller than the sampling error. This eventually lead, via more general queries [31, 6], to differential privacy. The view of these queries as a privacy-preserving programming primitive [6] inspired McSherry’s Privacy Integrated Queries programming platform [59].

The reconstruction attack of Theorem 8.2 appears in [24], where Dwork, McSherry, and Talwar showed that polynomial time reconstruction is possible even if a 0.239 fraction of the responses have wild, arbitrary, noise, provided the others have noise $o(\sqrt{n})$.

The geometric approach, and in particular Lemma 8.4, is due to Hardt and Talwar [45], who also gave a geometry-based algorithm proving these bounds tight for small numbers $k \leq n$ of queries, under a

commonly believed conjecture. Dependence on the conjecture was later removed by Bhaskara et al. [5]. The geometric approach was extended to arbitrary numbers of queries by Nikolov et al. [66], who gave an algorithm with instance-optimal mean squared error. For the few queries case this leads, via a boosting argument, to low expected worst-case error. Theorem 8.8 is due to De [17].

9

Differential Privacy and Computational Complexity

Our discussion of differential privacy has so far ignored issues of computational complexity, permitting both the curator and the adversary to be computationally unbounded. In reality, both curator and adversary may be computationally bounded.

Confining ourselves to a computationally bounded curator restricts what the curator can do, making it harder to achieve differential privacy. And indeed, we will show an example of a class of counting queries that, under standard complexity theoretic assumptions, does not permit *efficient* generation of a synthetic database, even though inefficient algorithms, such as SmallDB and Private Multiplicative Weights, are known. Very roughly, the database rows are digital signatures, signed with keys to which the curator does not have access. The intuition will be that any row in a synthetic database must either be copied from the original — violating privacy — or must be a signature on a *new* message, i.e., a forgery — violating the unforgeability property of a digital signature scheme. Unfortunately, this state of affairs is not limited to (potentially contrived) examples based on digital signatures: it is even difficult to create a synthetic database that maintains relatively

accurate two-way marginals.¹ On the positive side, given a set \mathcal{Q} of queries and an n -row database with rows drawn from a universe \mathcal{X} , a synthetic database can be generated in time polynomial in n , $|\mathcal{X}|$, and $|\mathcal{Q}|$.

If we abandon the goal of a synthetic database and content ourselves with a data structure from which we can obtain a relatively accurate approximation to the answer to each query, the situation is much more interesting. It turns out that the problem is intimately related to the *tracing traitors* problem, in which the goal is to discourage piracy while distributing digital content to paying customers.

If the adversary is restricted to polynomial time, then it becomes easier to achieve differential privacy. In fact, the immensely powerful concept of *secure function evaluation* yields a natural way avoid the trusted curator (while giving better accuracy than randomized response), as well as a natural way to allow multiple trusted curators, who for legal reasons cannot share their data sets, to respond to queries on what is effectively a merged data set. Briefly put, secure function evaluation is a cryptographic primitive that permits a collection of n parties p_1, p_2, \dots, p_n , of which fewer than some fixed fraction are faulty (the fraction varies according to the type of faults; for “honest-but-curious” faults the fraction is 1), to cooperatively compute any function $f(x_1, \dots, x_n)$, where x_i is the input, or *value*, of party p_i , in such a way that no coalition of faulty parties can either disrupt the computation or learn more about the values of the non-faulty parties than can be deduced from the function output and the values of the members of the coalition. These two properties are traditionally called *correctness* and *privacy*. This privacy notion, let us call it *SFE privacy*, is very different from differential privacy. Let V be the set of values held by the faulty parties, and let p_i be a non-faulty party.² SFE privacy permits the faulty parties to learn x_i if x_i can be deduced from $V \cup \{f(x_1, \dots, x_n)\}$; differential privacy would therefore not permit exact release of $f(x_1, \dots, x_n)$. However, secure function evaluation

¹Recall that the two-way marginals are the counts, for every pair of attribute values, of the number of rows in the database having this pair of values.

²In the honest but curious case we can let $V = \{x_j\}$ for any party P_j .

protocols for computing a function f can easily be modified to obtain differentially private protocols for f , simply by defining a new function, g , to be the result of adding Laplace noise $\text{Lap}(\Delta f/\varepsilon)$ to the value of f . In principle, secure function evaluation permits evaluation of g . Since g is differentially private and the SFE privacy property, applied to g , says that nothing can be learned about the inputs that is not learnable from the value of $g(x_1, \dots, x_n)$ together with V , differential privacy is ensured, provided the faulty players are restricted to polynomial time. Thus, secure function evaluation allows a computational notion of differential privacy to be achieved, even without a trusted curator, at no loss in accuracy when compared to what can be achieved with a trusted curator. In particular, counting queries can be answered with constant expected error while ensuring computational differential privacy, with no trusted curator. We will see that, without cryptography, the error must be $\Omega(n^{1/2})$, proving that computational assumptions provably buy accuracy, in the multiparty case.

9.1 Polynomial time curators

In this section we show that, under standard cryptographic assumptions, it is computationally difficult to create a synthetic database that will yield accurate answers to an appropriately chosen class of counting queries, while ensuring even a minimal notion of privacy.

This result has several extensions; for example, to the case in which the set of queries is small (but the data universe remains large), and the case in which the data universe is small (but the set of queries is large). In addition, similar negative results have been obtained for certain natural families of queries, such as those corresponding to conjunctions.

We will use the term *syntheticize* to denote the process of generating a synthetic database in a privacy-preserving fashion³. Thus, the results in this section concern the computational hardness of syntheticizing. Our notion of privacy will be far weaker than differential privacy, so hardness of syntheticizing will imply hardness of generating a synthetic

³In Section 6 a syntheticizer took as input a synopsis; here we are starting with a database, which is a trivial synopsis.

database in a differentially private fashion. Specifically, we will say that syntheticizing is hard if it is hard even to avoid leaking input items in their entirety. That is, some item is always completely exposed.

Note that if, in contrast, leaking a few input items is not considered a privacy breach, then syntheticizing is easily achieved by releasing a randomly chosen subset of the input items. Utility for this “synthetic database” comes from sampling bounds: with high probability this subset will preserve utility even with respect to a large set of counting queries.

When introducing complexity assumptions, we require a *security parameter* in order to express sizes; for example, sizes of sets, lengths of messages, number of bits in a decryption key, and so on, as well as to express computational difficulty. The security parameter, denoted κ , represents “reasonable” sizes and effort. For example, it is assumed that it is feasible to exhaustively search a set whose size is (any fixed) polynomial in the security parameter.

Computational complexity is an asymptotic notion — we are concerned with how the difficulty of a task increases as the sizes of the objects (data universe, database, query family) grow. Thus, for example, we therefore need to think not just of a distribution on databases of a single size (what we have been calling n in the rest of this monograph), but of an ensemble of distributions, indexed by the security parameter. In a related vein, when we introduce complexity we tend to “soften” claims: forging a signature is not impossible — one might be lucky! Rather, we assume that no efficient algorithm succeeds with non-negligible probability, where “efficient” and “non-negligible” are defined in terms of the security parameter. We will ignore these fine points in our intuitive discussion, but will keep them in the formal theorem statements.

Speaking informally, a distribution of databases is *hard to syntheticize* (with respect to some family \mathcal{Q} of queries) if for any efficient (alleged) syntheticizer, with high probability over a database drawn from the distribution, at least one of the database items can be extracted from the alleged syntheticizer’s output. Of course, to avoid triviality, we will also require that when this leaked item is excluded from the input database (and, say, replaced by a random different item),

the probability that it can be extracted from the output is very small. This means that any efficient (alleged) syntheticizer indeed compromises the privacy of input items in a strong sense.

Definition 9.1 below will formalize our utility requirements for a syntheticizer. There are three parameters: α describes the accuracy requirement (being within α is considered accurate); γ describes the fraction of the queries on which a successful synthesis is allowed to be inaccurate, and β will be the probability of failure.

For an algorithm A producing synthetic databases, we say that an output $A(x)$ is (α, γ) -accurate for a query set \mathcal{Q} if $|q(A(x)) - q(x)| \leq \alpha$ for a $1 - \gamma$ fraction of the queries $q \in \mathcal{Q}$.

Definition 9.1 $((\alpha, \beta, \gamma)\text{-Utility})$. Let \mathcal{Q} be a set of queries and \mathcal{X} a data universe. A syntheticizer A has (α, β, γ) -utility for n -item databases with respect to \mathcal{Q} and \mathcal{X} if for any n -item database x :

$$\Pr[A(x) \text{ is } (\alpha, \gamma)\text{-accurate for } \mathcal{Q}] \geq 1 - \beta$$

where the probability is over the coins of A .

Let $\mathcal{Q} = \{\mathcal{Q}_n\}_{n=1,2,\dots}$ be a query family ensemble, $\mathcal{X} = \{\mathcal{X}_n\}_{n=1,2,\dots}$ be a data universe ensemble. An algorithm is said to be *efficient* if its running time is $\text{poly}(n, \log(|\mathcal{Q}_n|), \log(|\mathcal{X}_n|))$.

In the next definition we describe what it means for a family of distributions to be hard to syntheticize. A little more specifically we will say what it means to be hard to generate synthetic databases that provide (α, γ) -accuracy. As usual, we have to make this an asymptotic statement.

Definition 9.2 $((\mu, \alpha, \beta, \gamma, \mathcal{Q})\text{-Hard-to-Syntheticize Database Distribution})$. Let $\mathcal{Q} = \{\mathcal{Q}_n\}_{n=1,2,\dots}$ be a query family ensemble, $\mathcal{X} = \{\mathcal{X}_n\}_{n=1,2,\dots}$ be a data universe ensemble, and let $\mu, \alpha, \beta, \gamma \in [0, 1]$. Let n be a database size and \mathcal{D} an ensemble of distributions, where \mathcal{D}_n is over collections of $n + 1$ items from X_n .

We denote by $(x, i, x'_i) \sim \mathcal{D}_n$ the experiment of choosing an n -element database, an index i chosen uniformly from $[n]$, and an additional element x'_i from \mathcal{X}_n . A sample from \mathcal{D}_n gives us a pair of databases: x and the result of replacing the i th element of x (under

a canonical ordering) with x'_i . Thus, we think of \mathcal{D}_n as specifying a distribution on n -item databases (and their neighbors).

We say that \mathcal{D} is $(\mu, \alpha, \beta, \gamma, \mathcal{Q})$ -hard-to-Syntheticize if there exists an efficient algorithm T such that for any alleged efficient syntheticizer A the following two conditions hold:

1. With probability $1 - \mu$ over the choice of database $x \sim \mathcal{D}$ and the coins of A and T , if $A(x)$ maintains α -utility for a $1 - \gamma$ fraction of queries, then T can recover one of the rows of x from $A(x)$:

$$\Pr_{\substack{(x, i, x'_i) \sim D_n \\ \text{coin flips of } A, T}} [(A(x) \text{ maintains } (\alpha, \beta, \gamma)\text{-utility}) \text{ and } (x \cap T(A(x)) = \emptyset)] \leq \mu$$

2. For every efficient algorithm A , and for every $i \in [n]$, if we draw (x, i, x'_i) from D , and replace x_i with x'_i to form x' , T cannot extract x_i from $A(x')$ except with small probability:

$$\Pr_{\substack{(x, i, x'_i) \sim D_n \\ \text{coin flips of } A, T}} [x_i \in T(A(x'))] \leq \mu.$$

Later, we will be interested in offline mechanisms that produce arbitrary synopses, not necessarily synthetic databases. In this case we will be interested in the related notion of *hard to sanitize* (rather than hard to Syntheticize), for which we simply drop the requirement that A produce a synthetic database.

9.2 Some hard-to-Syntheticize distributions

We now construct three distributions that are hard to syntheticize.

A *signature scheme* is given by a triple of (possibly randomized) algorithms (Gen, Sign, Verify):

- Gen : $1^{\mathbb{N}} \rightarrow \{(\text{SK}, \text{VK})_n\}_{n=1,2,\dots}$ is used to generate a pair consisting of a (secret) *signing* key and a (public) *verification* key. It takes only the security parameter $\kappa \in \mathbb{N}$, written in unary, as input, and produces a pair drawn from $(\text{SK}, \text{VK})_\kappa$, the distribution on (signature, verification) key pairs indexed by κ ; we let

$p_s(\kappa), p_v(\kappa), \ell s(\kappa)$ denote the lengths of the signing key, verification key, and signature, respectively.

- $\text{Sign} : \text{SK}_\kappa \times \{0, 1\}^{\ell(\kappa)} \rightarrow \{0, 1\}^{\ell s(\kappa)}$ takes as input a signing key from a pair drawn from $(\text{SK}, \text{VK})_\kappa$ and a message m of length $\ell(\kappa)$, and produces a signature on m ;
- $\text{Verify} : \text{VK}_\kappa \times \{0, 1\}^* \times \{0, 1\}^{\ell(\kappa)} \rightarrow \{0, 1\}$ takes as input a verification key, a string σ , and a message m of length $\ell(\kappa)$, and checks that σ is indeed a valid signature of m under the given verification key.

Keys, message lengths, and signature lengths are all polynomial in κ .

The notion of security required is that, given any polynomial (in κ) number of valid (message, signature) pairs, it is hard to forge *any* new signature, even a new signature of a previously signed message (recall that the signing algorithm may be randomized, so there may exist multiple valid signatures of the same message under the same signing key). Such a signature scheme can be constructed from any one-way function. Speaking informally, these are functions that are easy to compute — $f(x)$ can be computed in time polynomial in the length (number of bits) of x , but hard to invert: for every probabilistic polynomial time algorithm, running in time polynomial in the security parameter κ , the probability, over a randomly chosen x in the domain of f , of finding *any* valid pre-image of $f(x)$, grows more slowly than the inverse of any polynomial in κ .

Hard to Syntheticize Distribution I: Fix an arbitrary signature scheme. The set \mathcal{Q}_κ of counting queries contains one counting query q_{vk} for each verification key $vk \in \text{VK}_\kappa$. The data universe \mathcal{X}_κ consists of the set of all possible (message, signature) pairs of the form for messages of length $\ell(\kappa)$ signed with keys in VK_κ .

The distribution \mathcal{D}_κ on databases is defined by the following sampling procedure. Run the signature scheme generator $\text{Gen}(1^\kappa)$ to obtain (sk, vk) . Randomly choose $n = \kappa$ messages in $\{0, 1\}^{\ell(\kappa)}$ and run the signing procedure for each one, obtaining a set of n (message, signature) pairs all signed with key sk . This is the database x . Note that all the messages in the database are signed with the *same* signing key.

A data universe item (m, σ) satisfies the predicate q_{vk} if and only if $\text{Verify}(vk, m, \sigma) = 1$, i.e., σ is a valid signature for m according to verification key vk .

Let $x \in_R \mathcal{D}_\kappa$ be a database, and let sk be the signing key used, with corresponding verification key vk . Assuming that the syntheticizer has produced y , it must be the case that almost all rows of y are valid signatures under vk (because the fractional count of x for the query vk is 1). By the unforgeability properties of the signature scheme, all of these must come from the input database x — the polynomial time bounded curator, running in time $\text{poly}(\kappa)$, cannot generate a new valid (message, signature) pair. (Only slightly) more formally, the probability that an efficient algorithm could produce a (message, signature) pair that is verifiable with key vk , but is not in x , is negligible, so with overwhelming probability any y that is produced by an efficient syntheticizer will only contain rows of x .⁴ This contradicts (any reasonable notion of) privacy.

In this construction, both \mathcal{Q}_κ (the set of verification keys) and \mathcal{X}_κ (the set of (message, signature) pairs) are large (superpolynomial in κ). When both sets are small, efficient differentially private generation of synthetic datasets is possible. That is, there is a differentially private syntheticizer whose running time is polynomial in $n = \kappa$, $|\mathcal{Q}_\kappa|$ and $|\mathcal{X}_\kappa|$: compute noisy counts using the Laplace mechanism to obtain a synopsis and then run the syntheticizer from Section 6. Thus, when both of these have size polynomial in κ the running time of the syntheticizer is polynomial in κ .

We now briefly discuss generalizations of the first hardness result to the cases in which one of these sets is small (but the other remains large).

Hard to Syntheticize Distribution II: In the database distribution above, we chose a single (sk, vk) key pair and generated a database of

⁴The quantification order is important, as otherwise the syntheticizer could have the signing key hardwired in. We first fix the syntheticizer, then run the generator and build the database. The probability is over all the randomness in the experiment: choice of key pair, construction of the database, and randomness used by the syntheticizer.

messages, all signed using sk ; hardness was obtained by requiring the syntheticizer to generate a new signature under sk , in order for the syntheticized database to provide an accurate answer to the query q_{vk} . To obtain hardness for syntheticizing when the size of the set of queries is only polynomial in the security parameter, we again use digital signatures, signed with a unique key, but we cannot afford to have a query for each possible verification key vk , as these are too numerous.

To address this, we make two changes:

1. Database rows now have the form (verification key, message, signature). more precisely, the data universe consists of (key,message,signature) triples $\mathcal{X} = \{(vk, m, s) : vk \in \text{VK}_\kappa, m \in \{0, 1\}^{\ell(\kappa)}, s \in \{0, 1\}^{\ell s(\kappa)}\}$.
2. We add to the query class exactly $2p_v(\kappa)$ queries, where $p_v(\kappa)$ is the length of the verification keys produced by running the generation algorithm $\text{Gen}(1^\kappa)$. The queries have the form (i, b) where $1 \leq i \leq p_v(\kappa)$ and $b \in \{0, 1\}$. The meaning of the query “ (i, b) ” is, “What fraction of the database rows are of the form (vk, m, s) where $\text{Verify}(vk, m, s) = 1$ and the i th bit of vk is b ?”. By populating a database with messages signed according to a single key vk , we ensure that the responses to these queries should be close to one for all $1 \leq i \leq p(\kappa)$ when $vk_i = b$, and close to zero when $vk_i = 1 - b$.

With this in mind, the hard to syntheticize distribution on databases is constructed by the following sampling procedure: Generate a signature-verification key pair $(sk, vk) \leftarrow \text{Gen}(1^\kappa)$, and choose $n = \kappa$ messages m_1, \dots, m_n uniformly from $\{0, 1\}^{\ell(\kappa)}$. The database x will have n rows; for $j \in [n]$ the j th row is the verification key, the j th message and its valid signature, i.e., the tuple $(vk, m_j, \text{Sign}(m_j, sk))$. Next, choose i uniformly from $[n]$. To generate the $(n + 1)$ st item x'_i , just generate a new message-signature pair (using the same key sk).

Hard to Syntheticize Distribution III: To prove hardness for the case of a polynomial (in κ) sized message space (but superpolynomial sized query set) we use a *pseudorandom function*. Roughly speaking, these are polynomial time computable functions with small descriptions that

cannot efficiently be distinguished, based only on their input-output behavior, from truly random functions (whose descriptions are long). This result only gives hardness of syntheticizing if we insist on maintaining utility for *all* queries. Indeed, if we are interested only in ensuring on-average utility, then the base generator for counting queries described in Section 6 yields an efficient algorithm for syntheticizing when the universe \mathcal{X} is of polynomial size, even when \mathcal{Q} is exponentially large.

Let $\{f_s\}_{s \in \{0,1\}^\kappa}$ be a family of pseudo-random functions from $[\ell]$ to $[\ell]$, where $\ell \in \text{poly}(\kappa)$. More specifically, we need that the set of all pairs of elements in $[\ell]$ is “small,” but larger than κ ; this way the κ -bit string describing a function in the family is shorter than the $\ell \log_2 \ell$ bits needed to describe a random function mapping $[\ell]$ to $[\ell]$. Such a family of pseudorandom functions can be constructed from any one-way function.

Our data universe will be the set of all pairs of elements in $[\ell]$: $\mathcal{X} = \{(a, b) : a, b \in [\ell]\}$. \mathcal{Q}_κ will contain two types of queries:

1. There will be one query for each function $\{f_s\}_{s \in \{0,1\}^\kappa}$ in the family. A universe element $(a, b) \in \mathcal{X}$ satisfies the query s if and only if $f_s(a) = b$.
2. There will be a relatively small number, say κ , truly random queries. Such a query can be constructed by randomly choosing, for each $(a, b) \in \mathcal{X}$, whether or not (a, b) will satisfy the query.

The hard to syntheticize distribution is generated as follows. First, we select a random string $s \in \{0,1\}^\kappa$, specifying a function in our family. Next, we generate, for $n = \kappa$ distinct values a_1, \dots, a_n chosen at random from $[\ell]$ without replacement, the universe element $(a, f_s(a))$.

The intuition is simple, relies only on the first type of query, and does not make use of the distinctness of the a_i . Given a database x generated according to our distribution, where the pseudo-random function is given by s , the syntheticizer must create a synthetic database (almost) all of whose rows must satisfy the query s . The intuition is that it can’t *reliably* find input-output pairs that do not appear in x . A little more precisely, for an arbitrary element $a \in [\ell]$ such that no

row in x is of the form $(a, f_s(a))$, the pseudo-randomness of f_s says that an efficient syntheticizer should have probability at most negligibly more than $1/\ell$ of finding $f_s(a)$. In this sense the pseudo-randomness gives us properties similar to, although somewhat weaker than, what we obtained from digital signatures.

Of course, for any given $a \in [\ell]$, the syntheticizer can indeed guess with probability $1/\ell$ the value $f_s(a)$, so without the second type of query, nothing obvious would stop it from ignoring x , choosing an arbitrary a , and outputting a database of n copies of (a, b) , where b is chosen uniformly at random from $[\ell]$. The intuition is now that such a synthetic database would give the wrong fraction — either zero or one, when the right answer should be about $1/2$ — on the truly random queries.

Formally, we have:

Theorem 9.1. Let $f : \{0, 1\}^\kappa \rightarrow \{0, 1\}^\kappa$ be a one-way function. For every $a > 0$, and for every integer $n = \text{poly}(\kappa)$, there exists a query family \mathcal{Q} of size $\exp(\text{poly}(\kappa))$, a data universe \mathcal{X} of size $O(n^{2+2a})$, and a distribution on databases of size n that is $(\mu, \alpha, \beta, 0, \mathcal{Q})$ -hard-to-syntheticize (i.e., hard to syntheticize for worst-case queries) for $\alpha \leq 1/3$, $\beta \leq 1/10$ and $\mu = 1/40n^{1+a}$.

The above theorem shows hardness of sanitizing with synthetic data. Note, however, that when the query set is small one can always simply release noisy counts for every query. We conclude that sanitizing for small query classes (with large data universes) is a task that separates efficient syntheticizing from efficient synopsis generation (sanitization with arbitrary outputs).

9.2.1 Hardness results for general synopses

The hardness results of the previous section apply only to syntheticizers — offline mechanisms that create synthetic databases. There is a tight connection between hardness for more general forms of privacy-preserving offline mechanisms, which we have been calling *offline query release mechanisms* or synopsis generators, and the existence of *traitor tracing* schemes, a method of content distribution in which (short) key

strings are distributed to subscribers in such a way that a sender can broadcast encrypted messages that can be decrypted by any subscriber, and any useful “pirate” decoder constructed by a coalition of malicious subscribers can be traced to at least one colluder.

A (private-key, stateless) traitor-tracing scheme consists of algorithms Setup, Encrypt, Decrypt and Trace. The Setup algorithm generates a key bk for the broadcaster and N subscriber keys k_1, \dots, k_N . The Encrypt algorithm encrypts a given bit using the broadcaster’s key bk . The Decrypt algorithm decrypts a given ciphertext using any of the subscriber keys. The Trace algorithm gets the key bk and oracle access to a (pirate, stateless) decryption box, and outputs the index $i \in \{1, \dots, N\}$ of a key k_i that was used to create the pirate box.

An important parameter of a traitor-tracing scheme is its *collusion-resistance*: a scheme is t -resilient if tracing is guaranteed to work as long as no more than t keys are used to create the pirate decoder. When $t = N$, tracing works even if all the subscribers join forces to try and create a pirate decoder. A more complete definition follows.

Definition 9.3. A scheme $(\text{Setup}, \text{Encrypt}, \text{Decrypt}, \text{Trace})$ as above is a *t -resilient traitor-tracing scheme* if (i) the ciphertexts it generates are semantically secure (roughly speaking, polynomial time algorithms cannot distinguish encryptions of 0 from encryptions of 1), and (ii) no polynomial time adversary A can “win” in the following game with non-negligible probability (over the coins of Setup, A , and Trace):

A receives the number of users N and a security parameter κ and (adaptively) requests the keys of up to t users $\{i_1, \dots, i_t\}$. The adversary then outputs a pirate decoder Dec . The Trace algorithm is run with the key bk and black-box access⁵ to Dec ; it outputs the name $i \in [N]$ of a user or the error symbol \perp . We say that an adversary A “wins” if it is both the case that Dec has a non-negligible advantage in decrypting ciphertexts (even a weaker condition than creating a usable pirate decryption device), and the output of Trace is not in $\{i_1, \dots, i_t\}$, meaning that the adversary avoided detection.

⁵Black-box access to an algorithm means that one has no access to the algorithm’s internals; one can only feed inputs to the algorithm and observe its outputs.

The intuition for why traitor-tracing schemes imply hardness results for counting query release is as follows. Fix a traitor tracing scheme. We must describe databases and counting queries for which query release is computationally hard.

For any given $n = \kappa$, the database $x \in \{\{0, 1\}^d\}^n$ will contain user keys from the traitor tracing scheme of a colluding set of n users; here d is the length of the decryption keys obtained when the Setup algorithm is run on input 1^κ . The query family \mathcal{Q}_κ will have a query q_c for each possible ciphertext c asking “For what fraction of the rows $i \in [n]$ does c decrypt to 1 under the key in row i ?”. Note that, since every user can decrypt, if the sender distributes an encryption c of the bit 1, the answer will be 1: all the rows decrypt c to 1, so the fraction of such rows is 1. If instead the sender distributes an encryption c' of the bit 0, the answer will be 0: since no row decrypts c' to 1, the fraction of rows decrypting c' to 1 is 0. Thus, the exact answer to a query q_c , where c is an encryption of a 1-bit message b , is b itself.

Now, suppose there were an efficient offline differentially private query release mechanism for queries in \mathcal{Q} . The colluders could use this algorithm to efficiently produce a synopsis of the database enabling a data analyst to efficiently compute approximate answers to the queries q_c . If these approximations are at all non-trivial, then the analyst can use these to correctly decrypt. That is, the colluders could use this to form a pirate decoder box. But traitor tracing ensures that, for any such box, the Trace algorithm can recover the key of at least one user, i.e., a row of the database. This violates differential privacy, contradicting the assumption that there is an efficient differentially private algorithm for releasing \mathcal{Q} .

This direction has been used to rule out the existence of efficient offline sanitizers for a particular class of $2^{\tilde{O}(\sqrt{n})}$ counting queries; this can be extended to rule out the existence of efficient *on-line* sanitizers answering $\tilde{\Theta}(n^2)$ counting queries drawn adaptively from a second (large) class.

The intuition for why hardness of offline query release for counting queries implies traitor tracing is that failure to protect privacy immediately yields some form of traceability; that is, the *difficulty* of providing an object that yields (approximate) functional equivalence for a set of

rows (decryption keys) while preserving privacy of each individual row (decryption key) — that is, the difficulty of producing an untraceable decoder — is precisely what we are looking for in a traitor tracing scheme.

In a little more detail, given a hard-to-sanitize database distribution and family of counting queries, a randomly drawn n -item database can act like a “master key,” where the secret used to decrypt messages is the *counts* of random queries on this database. For a randomly chosen subset S of $\text{polylog}(n)$ queries, a random set of $\text{polylog}(n)$ rows drawn from the database (very likely) yields good approximation to all queries in S . Thus, individual user keys can be obtained by randomly partitioning the database into $n/\text{polylog}(n)$ sets of $\text{polylog}(n)$ rows and assigning each set to a different user. These sets are large enough that with overwhelming probability their counts on a random collection of say $\text{polylog}(n)$ queries are *all* close to the counts of the original database.

To complete the argument, one designs an encryption scheme in which decryption is equivalent to computing approximate counts on small sets of random queries. Since by definition a pirate decryption box can decrypt, the a pirate box can be used to compute approximate counts. If we view this box as a sanitization of the database we conclude (because sanitizing is hard) that the decryption box can be “traced” to the keys (database items) that were used to create it.

9.3 Polynomial time adversaries

Definition 9.4 (Computational Differential Privacy). A randomized algorithm $C_\kappa : \mathcal{X}^n \rightarrow Y$ is ε -computationally differentially private if and only if for all databases x, y differing in a single row, and for all nonuniform polynomial (in κ) algorithms T ,

$$\Pr[T(C_\kappa(x)) = 1] \leq e^\varepsilon \Pr[T(C_\kappa(y)) = 1] + \nu(\kappa),$$

where $\nu(\cdot)$ is any function that grows more slowly than the inverse of any polynomial and the algorithm C_κ runs in time polynomial in n , $\log |\mathcal{X}|$, and κ .

Intuitively, this says that if the adversary is restricted to polynomial time then computationally differentially private mechanisms provide the same degree of privacy as do $(\varepsilon, \nu(\kappa))$ -differentially private algorithms. In general there is no hope of getting rid of the $\nu(\kappa)$ term; for example, when encryption is involved there is always some (negligibly small) chance of guessing the decryption key.

Once we assume the adversary is restricted to polynomial time, we can use the powerful techniques of *secure multiparty computation* to provide *distributed* online query release algorithms, replacing the trusted server with a distributed protocol that simulates a trusted curator. Thus, for example, a set of hospitals, each holding the data of many patients, can collaboratively carry out statistical analyses of the union of their patients, while ensuring differential privacy for each patient. A more radical implication is that individuals can maintain their own data, opting in or out of each specific statistical query or study, all the while ensuring differential privacy of their own data.

We have already seen one distributed solution, at least for the problem of computing a sum of n bits: randomized response. This solution requires no computational assumptions, and has an expected error of $\Theta(\sqrt{n})$. In contrast, the use of cryptographic assumptions permits much more accurate and extensive analyses, since by simulating the curator it can run a distributed implementation of the Laplace mechanism, which has constant expected error.

This leads to the natural question of whether there is some other approach, not relying on cryptographic assumptions, that yields better accuracy in the distributed setting than does randomized response. Or more generally, is there a separation between what can be accomplished with computational differential privacy and what can be achieved with “traditional” differential privacy? That is, does cryptography provably buy us something?

In the multiparty setting the answer is yes. Still confining our attention to summing n bits, we have:

Theorem 9.2. For $\varepsilon < 1$, every n -party $(\varepsilon, 0)$ -differentially private protocol for computing the sum of n bits (one per party) incurs error $\Omega(n^{1/2})$ with high probability.

A similar theorem holds for (ε, δ) -differential privacy provided $\delta \in o(1/n)$.

Proof. (sketch) Let X_1, \dots, X_n be uniform independent bits. The transcript T of the protocol is a random variable $T = T(P_1(X_1), \dots, P_n(X_n))$, where for $i \in [n]$ the protocol of player i is denoted P_i . Conditioned on $T = t$, the bits X_1, \dots, X_n are still independent bits, each with bias $O(\varepsilon)$. Further, by differential privacy, the uniformity of the X_i , and Bayes' Law we have:

$$\frac{\Pr[X_i = 1|T = t]}{\Pr[X_i = 0|T = t]} = \frac{\Pr[T = t|X_i = 1]}{\Pr[T = t|X_i = 0]} \leq e^\varepsilon < 1 + 2\varepsilon.$$

To finish the proof we note that the sum of n independent bits, each with constant bias, falls outside any interval of size $o(\sqrt{n})$ with high probability. Thus, with high probability, the sum $\sum_i X_i$ is not in the interval $[\text{output}(T) - o(n^{1/2}), \text{output}(T) + o(n^{1/2})]$. \square

A more involved proof shows a separation between computational differential privacy and ordinary differential privacy even for the two-party case. It is a fascinating open question whether computational assumptions buy us anything in the case of the trusted curator. Initial results are negative: for *small* numbers of *real-valued* queries, i.e., for a number of queries that does not grow with the security parameter, there is a natural class of utility measures, including L_p distances and mean-squared errors, for which any computationally private mechanism can be converted to a statistically private mechanism that is roughly as efficient and achieves almost the same utility.

9.4 Bibliographic notes

The negative results for polynomial time bounded curators and the connection to traitor tracing are due to Dwork et al. [28]. The connection to traitor tracing was further investigated by Ullman [82], who showed that, assuming the existence of 1-way functions, it is computationally hard to answer $n^{2+o(1)}$ arbitrary linear queries with differential privacy (even if without privacy the answers are easy to compute). In “Our Data, Ourselves,” Dwork, Kenthapadi, McSherry, Mironov, and

Naor considered a distributed version of the precursor of differential privacy, using techniques from secure function evaluation in place of the trusted curator [21]. A formal study of *computational* differential privacy was initiated in [64], and the separation between the accuracy that can be achieved with $(\varepsilon, 0)$ -differential privacy in the multiparty and single curator cases in Theorem 9.2 is due to McGregor et al. [58]. The initial results regarding whether computational assumptions on the adversary buys anything in the case of a trusted curator are due to Groce et al. [37].

Construction of pseudorandom functions from any one-way function is due to Håstad et al. [40].

10

Differential Privacy and Mechanism Design

One of the most fascinating areas of game theory is mechanism design, which is the science of designing incentives to get people to do what you want them to do. Differential privacy has proven to have interesting connections to mechanism design in a couple of unexpected ways. It provides a tool to quantify and control privacy loss, which is important if the people the mechanism designer is attempting to manipulate care about privacy. However, it also provides a way to limit the sensitivity of the outcome of a mechanism to the choices of any single person, which turns out to be a powerful tool even in the absence of privacy concerns. In this section, we give a brief survey of some of these ideas.

Mechanism Design is the problem of *algorithm design* when the inputs to the algorithm are controlled by individual, self-interested agents, rather than the algorithm designer himself. The algorithm maps its reported inputs to some outcome, over which the agents have preferences. The difficulty is that the agents may mis-report their data if doing so will cause the algorithm to output a different, preferred outcome, and so the mechanism designer must design the algorithm so that the agents are always incentivized to report their true data.

The concerns of mechanism design are very similar to the concerns of private algorithm design. In both cases, the inputs to the algorithm are thought of as belonging to some third party¹ which has preferences over the outcome. In mechanism design, we typically think of individuals as getting some explicit value from the outcomes of the mechanism. In private algorithm design, we typically think of the individual as experiencing some explicit harm from (consequences of) outcomes of the mechanism. Indeed, we can give a utility-theoretic definition of differential privacy which is equivalent to the standard definition, but makes the connection to individual utilities explicit:

Definition 10.1. An algorithm $A : \mathbb{N}^{|\mathcal{X}|} \rightarrow R$ is ϵ -differentially private if for every function $f : R \rightarrow \mathbb{R}_+$, and for every pair of neighboring databases $x, y \in \mathbb{N}^{|\mathcal{X}|}$:

$$\exp(-\epsilon)\mathbb{E}_{z \sim A(y)}[f(z)] \leq \mathbb{E}_{z \sim A(x)}[f(z)] \leq \exp(\epsilon)\mathbb{E}_{z \sim A(y)}[f(z)].$$

We can think of f as being some function mapping outcomes to an arbitrary agent's utility for those outcomes. With this interpretation, a mechanism is ϵ -differentially private, if for every agent it promises that their participation in the mechanism cannot affect their expected future utility by more than a factor of $\exp(\epsilon)$ *independent of what their utility function might be*.

Let us now give a brief definition of a problem in mechanism design. A mechanism design problem is defined by several objects. There are n agents $i \in [n]$, and a set of outcomes \mathcal{O} . Each agent has a type, $t_i \in \mathcal{T}$ which is known only to her, and there is a utility function over outcomes $u : \mathcal{T} \times \mathcal{O} \rightarrow [0, 1]$. The utility that agent i gets from an outcome $o \in \mathcal{O}$ is $u(t_i, o)$, which we will often abbreviate as $u_i(o)$. We will write $t \in \mathcal{T}^n$ to denote vectors of all n agent types, with t_i denoting the type of agent i , and $t_{-i} \equiv (t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n)$ denoting the vector of types of all agents *except* agent i . The type of an agent i completely specifies her utility over outcomes — that is, two agents $i \neq j$ such that $t_i = t_j$ will evaluate each outcome identically: $u_i(o) = u_j(o)$ for all $o \in \mathcal{O}$.

¹In the privacy setting, the database administrator (such as a hospital) might already have access to the data itself, but is nevertheless acting so as to protect the interests of the agents who own the data when it endeavors to protect privacy.

A mechanism M takes as input a set of reported types, one from each player, and selects an outcome. That is, a mechanism is a mapping $M : \mathcal{T}^n \rightarrow \mathcal{O}$. Agents will choose to report their types strategically so as to optimize their utility, possibly taking into account what (they think) the other agents will be doing. In particular, they need not report their true types to the mechanism. If an agent is always incentivized to report some type, no matter what her opponents are reporting, then reporting that type is called a *dominant strategy*. If reporting one's true type is a dominant strategy for every agent, then the mechanism is called *truthful*, or equivalently, *dominant strategy truthful*.

Definition 10.2. Given a mechanism $M : \mathcal{T}^n \rightarrow \mathcal{O}$, truthful reporting is an ϵ -approximate *dominant strategy* for player i if for every pair of types $t_i, t'_i \in T$, and for every vector of types t_{-i} :

$$u(t_i, M(t_i, t_{-i})) \geq u(t_i, M(t'_i, t_{-i})) - \epsilon.$$

If truthful reporting is an ϵ -approximate dominant strategy for every player, we say that M is ϵ -approximately dominant strategy truthful. If $\epsilon = 0$, then M is *exactly truthful*.

That is, a mechanism is truthful if no agent can improve her utility by misrepresenting her type, no matter what the other players report.

Here we can immediately observe a syntactic connection to the definition of differential privacy. We may identify the type space T with the data universe X . The input to the mechanism therefore consists of a database of size n , consisting of the reports of each agent. In fact, when an agent is considering whether she should truthfully report her type t_i or lie, and misreport her type as t'_i , she is deciding which of two databases the mechanism should receive: (t_1, \dots, t_n) , or $(t_1, \dots, t_{i-1}, t'_i, t_{i+1}, \dots, t_n)$. Note that these two databases differ only in the report of agent i ! That is, they are *neighboring databases*. Thus, differential privacy gives a guarantee of approximate truthfulness!

10.1 Differential privacy as a solution concept

One of the starting points for investigating the connection between differential privacy and game theory is observing that differential privacy

is a *stronger* condition than approximate truthfulness. Note that for $\epsilon \leq 1$, $\exp(\epsilon) \leq 1 + 2\epsilon$ and so the following proposition is immediate.

Proposition 10.1. If a mechanism M is ϵ -differentially private, then M is also 2ϵ -approximately dominant strategy truthful.

As a solution concept, this has several robustness properties that strategy proof mechanisms do not. By the composition property of differential privacy, the composition of 2 ϵ -differentially private mechanisms remains 4ϵ -approximately dominant strategy truthful. In contrast, the incentive properties of general strategy proof mechanisms may not be preserved under composition.

Another useful property of differential privacy as a solution concept is that it generalizes to group privacy: suppose that t and $t' \in \mathcal{T}^n$ are not neighbors, but instead differ in k indices. Recall that by group privacy we then have for any player i : $\mathbb{E}_{o \sim M(t)}[u_i(o)] \leq \exp(k\epsilon)\mathbb{E}_{o \sim M(t')}[u_i(o)]$. That is, changes in up to k types changes the expected output by at most $\approx (1+k\epsilon)$, when $k \ll 1/\epsilon$. Therefore, differentially private mechanisms make truthful reporting a $2k\epsilon$ -approximate dominant strategy *even for coalitions of k agents* — i.e., differential privacy automatically provides robustness to collusion. Again, this is in contrast to general dominant-strategy truthful mechanisms, which in general offer no guarantees against collusion.

Notably, differential privacy allows for these properties in very general settings *without the use of money!* In contrast, the set of exactly dominant strategy truthful mechanisms when monetary transfers are not allowed is extremely limited.

We conclude with a drawback of using differential privacy as a solution concept as stated: not only is truthfully reporting one's type an approximate dominant strategy, *any report* is an approximate dominant strategy! That is, differential privacy makes the outcome approximately independent of any single agent's report. In some settings, this shortcoming can be alleviated. For example, suppose that M is a differentially private mechanism, but that agent utility functions are defined to be functions both of the outcome of the mechanism, *and* of the reported type t'_i of the agent: formally, we view the outcome space as $\mathcal{O}' = \mathcal{O} \times T$. When the agent reports type t'_i to the mechanism, and

the mechanism selects outcome $o \in \mathcal{O}$, then the utility experienced by the agent is controlled by the outcome $o' = (o, t'_i)$. Now consider the underlying utility function $u : T \times \mathcal{O}' \rightarrow [0, 1]$. Suppose we have that *fixing* a selection o of the mechanism, truthful reporting is a dominant strategy — that is, for all types t_i, t'_i , and for all outcomes $o \in \mathcal{O}$:

$$u(t_i, (o, t_i)) \geq u(t_i, (o, t'_i)).$$

Then it remains the fact that truthful reporting to an ϵ -differentially private mechanism $M : T^n \rightarrow \mathcal{O}$ remains a 2ϵ approximate dominant strategy, because for any misreport t'_i that player i might consider, we have:

$$\begin{aligned} u(t_i, (M(t), t_i)) &= \mathbb{E}_{o \sim M(t)}[u(t_i, (o, t_i))] \\ &\geq (1 + 2\epsilon)\mathbb{E}_{o \sim M(t'_i, t_{-i})}[u(t_i, (o, t_i))] \\ &\geq \mathbb{E}_{o \sim M(t'_i, t_{-i})}[u(t_i, (o, t'_i))] \\ &= u(t_i, (M(t'_i, t_{-i}), t'_i)). \end{aligned}$$

However, we no longer have that every report is an approximate dominant strategy, because player i 's utility can depend arbitrarily on $o' = (o, t'_i)$, and only o (and not player i 's report t'_i itself) is differentially private. This will be the case in all examples we consider here.

10.2 Differential privacy as a tool in mechanism design

In this section, we show how the machinery of differential privacy can be used as a tool in designing novel mechanisms.

10.2.1 Warmup: digital goods auctions

To warm up, let us consider a simple special case of the first application of differential privacy in mechanism design. Consider a *digital goods auction*, i.e., one where the seller has an unlimited supply of a good with zero marginal cost to produce, for example a piece of software or other digital media. There are n unit demand buyers for this good, each with unknown valuation $v_i \in [0, 1]$. Informally, the valuation v_i of a bidder i represents the maximum amount of money that buyer i

would be willing to pay for a good. There is no prior distribution on the bidder valuations, so a natural revenue benchmark is the revenue of the *best fixed price*. At a price $p \in [0, 1]$, each bidder i with $v_i \geq p$ will buy. Therefore the total revenue of the auctioneer is

$$\text{Rev}(p, v) = p \cdot |\{i : v_i \geq p\}|.$$

The optimal revenue is the revenue of the best fixed price: $\text{OPT} = \max_p \text{Rev}(p, v)$. This setting is well studied: the best known result for exactly dominant strategy truthful mechanisms is a mechanism which achieves revenue at least $\text{OPT} - O(\sqrt{n})$.

We show how a simple application of the exponential mechanism achieves revenue at least $\text{OPT} - O\left(\frac{\log n}{\epsilon}\right)$. That is, the mechanism trades exact for approximate truthfulness, but achieves an exponentially better revenue guarantee. Of course, it also inherits the benefits of differential privacy discussed previously, such as resilience to collusion, and composability.

The idea is to select a price from the exponential mechanism, using as our “quality score” the revenue that this price would obtain. Suppose we choose the range of the exponential mechanism to be $\mathcal{R} = \{\alpha, 2\alpha, \dots, 1\}$. The size of the range is $|\mathcal{R}| = 1/\alpha$. What have we lost in potential revenue if we restrict ourselves to selecting a price from \mathcal{R} ? It is not hard to see that

$$\text{OPT}_{\mathcal{R}} \equiv \max_{p \in \mathcal{R}} \text{Rev}(p, v) \geq \text{OPT} - \alpha n.$$

This is because if p^* is the price that achieves the optimal revenue, and we use a price p such that $p^* - \alpha \leq p \leq p^*$, every buyer who bought at the optimal price continues to buy, and provides us with at most α less revenue per buyer. Since there are at most n buyers, the total lost revenue is at most αn .

So how do we parameterize the exponential mechanism? We have a family of discrete ranges \mathcal{R} , parameterized by α . For a vector of values v and a price $p \in \mathcal{R}$, we define our quality function to be $q(v, p) = \text{Rev}(v, p)$. Observe that because each value $v_i \in [0, 1]$, we can restrict attention to prices $p \leq 1$ and hence, the *sensitivity* of q is $\Delta = 1$: changing one bidder valuation can only change the revenue at a fixed

price by at most $v_i \leq 1$. Therefore, if we require ϵ -differential privacy, by Theorem 3.11, we get that with high probability, the exponential mechanism returns some price p such that

$$\text{Rev}(p, v) \geq (\text{OPT} - \alpha n) - O\left(\frac{1}{\epsilon} \ln\left(\frac{1}{\alpha}\right)\right).$$

Choosing our discretization parameter α to minimize the two sources of error, we find that this mechanism with high probability finds us a price that achieves revenue

$$\text{Rev}(p, v) \geq \text{OPT} - O\left(\frac{\log n}{\epsilon}\right).$$

What is the right level to choose for the privacy parameter ϵ ? Note that here, we do not necessarily view privacy itself as a goal of our computation. Rather, ϵ is a way of trading off the revenue guarantee with an upper bound on agent's incentives to deviate. In the literature on large markets in economics, a common goal when exact truthfulness is out of reach is “asymptotic truthfulness” – that is, the maximum incentive that any agent has to deviate from his truthful report tends to 0 as the size of the market n grows large. To achieve a result like that here, all we need to do is set ϵ to be some diminishing function in the number of agents n . For example, if we take $\epsilon = 1/\log(n)$, then we obtain a mechanism that is asymptotically exactly truthful (i.e., as the market grows large, the approximation to truthfulness becomes exact). We can also ask what our approximation to the optimal revenue is as n grows large. Note that our approximation to the optimal revenue is only additive, and so even with this setting of ϵ , we can still guarantee revenue at least $(1 - o(1))\text{OPT}$, so long as OPT grows more quickly than $\log(n)^2$ with the size of the population n .

Finally, notice that we could make the reported value v_i of each agent i binding. In other words, we could allocate an item to agent i and extract payment of the selected posted price p whenever $v_i \geq p$. If we do this, the mechanism is approximately truthful, because the price is picked using a differentially private mechanism. Additionally, it is not the case that *every* report is an approximate dominant strategy: if an agent over-reports, she may be forced to buy the good at a price higher than her true value.

10.2.2 Approximately truthful equilibrium selection mechanisms

We now consider the problem of approximately truthful equilibrium selection. We recall the definition of a *Nash Equilibrium*: Suppose each player has a set of actions \mathcal{A} , and can choose to play any action $a_i \in \mathcal{A}$. Suppose, moreover, that *outcomes* are merely choices of actions that the agents might choose to play, and so agent utility functions are defined as $u : \mathcal{T} \times \mathcal{A}^n \rightarrow [0, 1]$. Then:

Definition 10.3. A set of actions $a \in \mathcal{A}^n$ is an ϵ -approximate Nash equilibrium if for all players i and for all actions a'_i :

$$u_i(a) \geq u_i(a'_i, a_{-i}) - \epsilon$$

In other words, every agent is simultaneously playing an (approximate) best response to what the other agents are doing, assuming they are playing according to a .

Roughly speaking, the problem is as follows: suppose we are given a game in which each player knows their own payoffs, but not others' payoffs (i.e., the players do not know what the types are of the other agents). The players therefore do not know the equilibrium structure of this game. Even if they did, there might be multiple equilibria, with different agents preferring different equilibria. Can a mechanism offered by an intermediary incentivize agents to truthfully report their utilities and follow the equilibrium it selects?

For example, imagine a city in which (say) Google Navigation is the dominant service. Every morning, each person enters their starting point and destination, receives a set of directions, and chooses his/her route according to those directions. Is it possible to design a navigation service such that: Each agent is incentivized to both (1) report truthfully, and (2) then follow the driving directions provided? Both misreporting start and end points, and truthfully reporting start and end points, but then following a different (shorter) path are to be disincentivized.

Intuitively, our two desiderata are in conflict. In the commuting example above, if we are to guarantee that every player is incentivized to truthfully follow their suggested route, then we must compute an

equilibrium of the game in question given players' reports. On the other hand, to do so, our suggested route to some player i must depend on the reported location/destination pairs of other players. This tension will pose a problem in terms of incentives: if we compute an equilibrium of the game given the reports of the players, an agent can potentially benefit by misreporting, causing us to compute an equilibrium of the wrong game.

This problem would be largely alleviated, however, if the report of agent i only has a tiny effect on the actions of agents $j \neq i$. In this case, agent i could hardly gain an advantage through his effect on other players. Then, assuming that everyone truthfully reported their type, the mechanism would compute an equilibrium of the correct game, and by definition, each agent i could do no better than follow the suggested equilibrium action. In other words, if we could compute an approximate equilibrium of the game under the constraint of *differential privacy*, then truthful reporting, followed by taking the suggested action of the coordination device would be a Nash equilibrium. A moment's reflection reveals that the goal of privately computing an equilibrium is not possible in small games, in which an agent's utility is a highly sensitive function of the actions (and hence utility functions) of the other agents. But what about in large games?

Formally, suppose we have an n player game with action set \mathcal{A} , and each agent with type t_i has a utility function $u_i : \mathcal{A}^n \rightarrow [0, 1]$. We say that this game is Δ -large if for all players $i \neq j$, vectors of actions $a \in \mathcal{A}^n$, and pairs of actions $a_j, a'_j \in \mathcal{A}$:

$$|u_i(a_j, a_{-j}) - u_i(a'_j, a_{-j})| \leq \Delta.$$

In other words, if some agent j unilaterally changes his action, then his affect on the payoff of any other agent $i \neq j$ is at most Δ . Note that if agent j changes his own action, then his payoff can change arbitrarily. Many games are “large” in this sense. In the commuting example above, if Alice changes her route to work she may substantially increase or decrease her commute time, but will only have a minimal impact on the commute time of any other agent Bob. The results in this section are strongest for $\Delta = O(1/n)$, but hold more generally.

First we might ask whether we need privacy at all—could it be the case that in a large game, any algorithm which computes an equilibrium of a game defined by reported types has the stability property that we want? The answer is no. As a simple example, consider n people who must each choose whether to go to the beach (B) or the mountains (M). People privately know their types—each person’s utility depends on his own type, his action, and the fraction of other people p who go to the beach. A Beach type gets a payoff of $10p$ if he visits the beach, and $5(1 - p)$ if he visits the mountain. A mountain type gets a payoff $5p$ from visiting the beach, and $10(1 - p)$ from visiting the mountain. Note that this is a large (i.e., low sensitivity) game — each player’s payoffs are insensitive in the actions of others. Further, note that “everyone visits beach” and “everyone visits mountain” are both equilibria of the game, regardless of the realization of types. Consider the mechanism that attempts to implement the following social choice rule—“if the number of beach types is less than half the population, send everyone to the beach, and vice versa.” It should be clear that if mountain types are just in the majority, then each mountain type has an incentive to misreport as a beach type; and vice versa. As a result, even though the game is “large” and agents’ actions do not affect others’ payoffs significantly, simply computing equilibria from reported type profiles does not in general lead to even approximately truthful mechanisms.

Nevertheless, it turns out to be possible to give a mechanism with the following property: it elicits the type t_i of each agent, and then computes an α -approximate correlated equilibrium of the game defined by the reported types.² (In some cases, it is possible to strengthen this result to compute an approximate *Nash equilibrium* of the underlying game.) It draws an action profile $a \in \mathcal{A}^n$ from the correlated equilibrium, and reports action a_i to each agent i . The algorithm has the guarantee that simultaneously for all players i , the joint distribution a_{-i} on reports to all players *other than* i is differentially private in

²A correlated equilibrium is defined by a joint distribution on profiles of actions, \mathcal{A}^n . For an action profile a drawn from the distribution, if agent i is told only a_i , then playing action a_i is a best response given the induced conditional distribution over a_{-i} . An α -approximate correlated equilibrium is one where deviating improves an agent’s utility by at most α .

the reported type of agent i . When the algorithm computes a correlated equilibrium of the underlying game, this guarantee is sufficient for a restricted form of approximate truthfulness: agents who have the option to opt-in or opt-out of the mechanism (but not to misreport their type if they opt-in) have no disincentive to opt-out, because no agent i can substantially change the distribution on actions induced on *the other players* by opting out. Moreover, given that he opts in, no agent has incentive not to follow his suggested action, as his suggestion is part of a correlated equilibrium. When the mechanism computes a Nash equilibrium of the underlying game, then the mechanism becomes truthful even when agents have the ability to mis-report their type to the mechanism when they opt in.

More specifically, when these mechanisms compute an α -approximate Nash equilibrium while satisfying ϵ -differential privacy, every agent following the honest behavior (i.e., first opting in and reporting their true type, then following their suggested action) forms an $(2\epsilon + \alpha)$ -approximate Nash equilibrium. This is because, by privacy, reporting your true type is a 2ϵ -approximate dominant strategy, and given that everybody reports their true type, the mechanism computes an α -approximate equilibrium of the true game, and hence by definition, following the suggested action is an α -approximate best response. There exist mechanisms for computing an α -approximate equilibrium in large games with $\alpha = O\left(\frac{1}{\sqrt{n}\epsilon}\right)$. Therefore, by setting $\epsilon = O\left(\frac{1}{n^{1/4}}\right)$, this gives an η -approximately truthful equilibrium selection mechanism for

$$\eta = 2\epsilon + \alpha = O\left(\frac{1}{n^{1/4}}\right).$$

In other words, it gives a mechanism for coordinating equilibrium behavior in large games that is asymptotically truthful in the size of the game, all without the need for monetary transfers.

10.2.3 Obtaining exact truthfulness

So far we have discussed mechanisms that are *asymptotically truthful* in large population games. However, what if we want to insist on mechanisms that are *exactly* dominant strategy truthful, while maintaining

some of the nice properties enjoyed by our mechanisms so far: for example, that the mechanisms do not need to be able to extract monetary payments? Can differential privacy help here? It can—in this section, we discuss a framework which uses differentially private mechanisms as a building block toward designing exactly truthful mechanisms without money.

The basic idea is simple and elegant. As we have seen, the exponential mechanism can often give excellent utility guarantees while preserving differential privacy. This doesn't yield an exactly truthful mechanism, but it gives every agent very little incentive to deviate from truthful behavior. What if we could pair this with a second mechanism which need not have good utility guarantees, but gives each agent a strict positive incentive to report truthfully, i.e., a mechanism that essentially only punishes non-truthful behavior? Then, we could randomize between running the two mechanisms. If we put enough weight on the punishing mechanism, then we inherit its strict-truthfulness properties. The remaining weight that is put on the exponential mechanism contributes to the utility properties of the final mechanism. The hope is that since the exponential mechanism is approximately strategy proof to begin with, the randomized mechanism can put small weight on the strictly truthful punishing mechanism, and therefore will have good utility properties.

To design punishing mechanisms, we will have to work in a slightly non-standard environment. Rather than simply picking an outcome, we can model a mechanism as picking an outcome, and then an agent as choosing a *reaction* to that outcome, which together define his utility. Mechanisms will then have the power to *restrict the reactions allowed by the agent based on his reported type*. Formally, we will work in the following framework:

Definition 10.4 (The Environment). An environment is a set N of n players, a set of types $t_i \in \mathcal{T}$, a finite set \mathcal{O} of outcomes, a set of reactions R and a utility function $u : T \times \mathcal{O} \times R \rightarrow [0, 1]$.

We write $r_i(t, s, \hat{R}_i) \in \arg \max_{r \in \hat{R}_i} u_i(t, s, r)$ to denote *is* optimal reaction among choices $\hat{R}_i \subseteq R$ to alternative s if he is of type t .

A direct revelation mechanism \mathcal{M} defines a game which is played as follows:

1. Each player i reports a type $t'_i \in \mathcal{T}$.
2. The mechanism chooses an alternative $s \in \mathcal{O}$ and a subset $\hat{R}_i \subseteq R$ of reactions, for each player i .
3. Each player i chooses a reaction $r_i \in \hat{R}_i$ and experiences utility $u(t_i, s, r_i)$.

Agents play so as to maximize their own utility. Note that since there is no further interaction after the 3rd step, rational agents will pick $r_i = r_i(t_i, s, \hat{R}_i)$, and so we can ignore this as a strategic step. Let $\mathcal{R} = 2^R$. Then a mechanism is a randomized mapping $\mathcal{M} : \mathcal{T} \rightarrow \mathcal{O} \times \mathcal{R}^n$.

Let us consider the utilitarian welfare criterion: $F(t, s, r) = \frac{1}{n} \sum_{i=1}^n u(t_i, s, r_i)$. Note that this has sensitivity $\Delta = 1/n$, since each agent's utility lies in the range $[0, 1]$. Hence, if we simply choose an outcome s and allow each agent to play their best response reaction, the exponential mechanism is an ϵ -differentially private mechanism, which by Theorem 3.11, achieves social welfare at least $\text{OPT} - O\left(\frac{\log |\mathcal{O}|}{\epsilon n}\right)$ with high probability. Let us denote this instantiation of the exponential mechanism, with quality score F , range \mathcal{O} and privacy parameter ϵ , as \mathcal{M}_ϵ .

The idea is to randomize between the exponential mechanism (with good social welfare properties) and a strictly truthful mechanism which punishes false reporting (but with poor social welfare properties). If we mix appropriately, then we will get an exactly truthful mechanism with reasonable social welfare guarantees.

Here is one such punishing mechanism which is simple, but not necessarily the best for a given problem:

Definition 10.5. The commitment mechanism $M^P(t')$ selects $s \in \mathcal{O}$ uniformly at random and sets $\hat{R}_i = \{r_i(t'_i, s, R_i)\}$, i.e., it picks a random outcome and forces everyone to react as if their reported type was their true type.

Define the *gap* of an environment as

$$\gamma = \min_{i, t_i \neq t'_i, t_{-i}} \max_{s \in \mathcal{O}} (u(t_i, s, r_i(t_i, s, R_i)) - u(t_i, s, r_i(t'_i, s, R_i))) ,$$

i.e., γ is a lower bound over players and types of the worst-case cost (over s) of mis-reporting. Note that for each player, this worst-case is realized with probability at least $1/|\mathcal{O}|$. Therefore we have the following simple observation:

Lemma 10.2. For all i, t_i, t'_i, t_{-i} :

$$u(t_i, \mathcal{M}^P(t_i, t_{-i})) \geq u(t_i, \mathcal{M}^P(t'_i, t_{-i})) + \frac{\gamma}{|\mathcal{O}|}.$$

Note that the commitment mechanism is strictly truthful: every individual has at least a $\frac{\gamma}{|\mathcal{O}|}$ incentive not to lie.

This suggests an exactly truthful mechanism with good social welfare guarantees:

Definition 10.6. The punishing exponential mechanism $\mathcal{M}_\epsilon^P(t)$ defined with parameter $0 \leq q \leq 1$ selects the exponential mechanism $\mathcal{M}_\epsilon(t)$ with probability $1 - q$ and the punishing mechanism $\mathcal{M}^P(t)$ with complementary probability q .

Observe that by linearity of expectation, we have for all t_i, t'_i, t_{-i} :

$$\begin{aligned} u(t_i, \mathcal{M}_\epsilon^P(t_i, t_{-i})) &= (1 - q) \cdot u(t_i, \mathcal{M}_\epsilon(t_i, t_{-i})) + q \cdot u(t_i, \mathcal{M}^P(t_i, t_{-i})) \\ &\geq (1 - q) (u(t_i, \mathcal{M}_\epsilon(t'_i, t_{-i})) - 2\epsilon) \\ &\quad + q \left(u(t_i, \mathcal{M}^P(t'_i, t_{-i})) + \frac{\gamma}{|\mathcal{O}|} \right) \\ &= u(t_i, \mathcal{M}_\epsilon^P(t'_i, t_{-i})) - (1 - q)2\epsilon + q \frac{\gamma}{|\mathcal{O}|} \\ &= u(t_i, \mathcal{M}_\epsilon^P(t'_i, t_{-i})) - 2\epsilon + q \left(2\epsilon + \frac{\gamma}{|\mathcal{O}|} \right). \end{aligned}$$

The following two theorems show incentive and social welfare properties of this mechanism.

Theorem 10.3. If $2\epsilon \leq \frac{q\gamma}{|\mathcal{O}|}$ then \mathcal{M}_ϵ^P is strictly truthful.

Note that we also have utility guarantees for this mechanism. Setting the parameter q so that we have a truthful mechanism:

$$\begin{aligned}
& \mathbb{E}_{s, \hat{R} \sim \mathcal{M}_\epsilon^P}[F(t, s, r(t, s, \hat{R}))] \\
& \geq (1 - q) \cdot \mathbb{E}_{s, \hat{R} \sim \mathcal{M}_\epsilon}[F(t, s, r(t, s, \hat{R}))] \\
& = \left(1 - \frac{2\epsilon|\mathcal{O}|}{\gamma}\right) \cdot \mathbb{E}_{s, \hat{R} \sim \mathcal{M}_\epsilon}[F(t, s, r(t, s, \hat{R}))] \\
& \geq \left(1 - \frac{2\epsilon|\mathcal{O}|}{\gamma}\right) \cdot \left(\max_{t, s, r} F(t, s, r) - O\left(\frac{1}{\epsilon n} \log |\mathcal{O}|\right)\right) \\
& \geq \max_{t, s, r} F(t, s, r) - \frac{2\epsilon|\mathcal{O}|}{\gamma} - O\left(\frac{1}{\epsilon n} \log |\mathcal{O}|\right).
\end{aligned}$$

Setting

$$\epsilon \in O\left(\sqrt{\frac{\log |\mathcal{O}| \gamma}{|\mathcal{O}| n}}\right)$$

we find:

$$\mathbb{E}_{s, \hat{R} \sim \mathcal{M}_\epsilon^P}[F(t, s, r(t, s, \hat{R}))] \geq \max_{t, s, r} F(t, s, r) - O\left(\sqrt{\frac{|\mathcal{O}| \log |\mathcal{O}|}{\gamma n}}\right).$$

Note that in this calculation, we assume that $\epsilon \leq \gamma/(2|\mathcal{O}|)$ so that $q = \frac{2\epsilon|\mathcal{O}|}{\gamma} \leq 1$ and the mechanism is well defined. This is true for sufficiently large n . That is, we have shown:

Theorem 10.4. For sufficiently large n , M_ϵ^P achieves social welfare at least

$$\text{OPT} - O\left(\sqrt{\frac{|\mathcal{O}| \log |\mathcal{O}|}{\gamma n}}\right).$$

Note that this mechanism is truthful without the need for payments!

Let us now consider an application of this framework: the facility location game. Suppose that a city wants to build k hospitals to minimize the average distance between each citizen and their closest hospital. To simplify matters, we make the mild assumption that the city is built on a discretization of the unit line.³ Formally, let

³If this is not the case, we can easily raze and then re-build the city.

$L(m) = \{0, \frac{1}{m}, \frac{2}{m}, \dots, 1\}$ denote the discrete unit line with step-size $1/m$. $|L(m)| = m+1$. Let $\mathcal{T} = R_i = L(m)$ for all i and let $|\mathcal{O}| = L(m)^k$. Define the utility of agent i to be:

$$u(t_i, s, r_i) = \begin{cases} -|t_i - r_i|, & \text{If } r_i \in s; \\ -1, & \text{otherwise.} \end{cases}$$

In other words, agents are associated with points on the line, and an outcome is an assignment of a location on the line to each of the k facilities. Agents can react to a set of facilities by deciding which one to go to, and their cost for such a decision is the distance between their own location (i.e., their type) and the facility that they have chosen. Note that $r_i(t_i, s)$ is here the closest facility $r_i \in s$.

We can instantiate Theorem 10.4. In this case, we have: $|\mathcal{O}| = (m+1)^k$ and $\gamma = 1/m$, because any two positions $t_i \neq t'_i$ differ by at least $1/m$. Hence, we have:

Theorem 10.5. M_ϵ^P instantiated for the facility location game is strictly truthful and achieves social welfare at least:

$$\text{OPT} - O\left(\sqrt{\frac{km(m+1)^k \log m}{n}}\right).$$

This is already very good for small numbers of facilities k , since we expect that $\text{OPT} = \Omega(1)$.

10.3 Mechanism design for privacy aware agents

In the previous section, we saw that differential privacy can be useful as a tool to design mechanisms, *for agents who care only about the outcome chosen by the mechanism*. We here primarily viewed privacy as a tool to accomplish goals in traditional mechanism design. As a side effect, these mechanisms also preserved the privacy of the reported player types. Is this itself a worthy goal? *Why* might we want our mechanisms to preserve the privacy of agent types?

A bit of reflection reveals that agents might care about privacy. Indeed, basic introspection suggests that in the real world, agents value the ability to keep certain “sensitive” information private, for example,

health information or sexual preferences. In this section, we consider the question of how to model this value for privacy, and various approaches taken in the literature.

Given that agents might have preferences for privacy, it is worth considering the design of mechanisms that preserve privacy *as an additional goal*, even for tasks such as welfare maximization that we can already solve non-privately. As we will see, it is indeed possible to generalize the VCG mechanism to *privately* approximately optimize social welfare in *any* social choice problem, with a smooth trade-off between the privacy parameter and the approximation parameter, all while guaranteeing exact dominant strategy truthfulness.

However, we might wish to go further. In the presence of agents with preferences for privacy, if we wish to design truthful mechanisms, we must somehow model their preferences for privacy in their utility function, and then design mechanisms which are truthful with respect to these new “privacy aware” utility functions. As we have seen with differential privacy, it is most natural to model privacy as a property of the mechanism itself. Thus, our utility functions are not merely functions of the outcome, but functions of the outcome and of the mechanism itself. In almost all models, agent utilities for outcomes are treated as linearly separable, that is, we will have for each agent i ,

$$u_i(o, \mathcal{M}, t) \equiv \mu_i(o) - c_i(o, \mathcal{M}, t).$$

Here $\mu_i(o)$ represents agent i 's utility for outcome o and $c_i(o, \mathcal{M}, t)$ the (privacy) cost that agent i experiences when outcome o is chosen with mechanism \mathcal{M} .

We will first consider perhaps the simplest (and most naïve) model for the privacy cost function c_i . Recall that for $\epsilon \ll 1$, differential privacy promises that for each agent i , and for every possible utility function f_i , type vector $t \in \mathcal{T}^n$, and deviation $t'_i \in \mathcal{T}$:

$$|\mathbb{E}_{o \sim M(t_i, t_{-i})}[f_i(o)] - \mathbb{E}_{o \sim M(t'_i, t_{-i})}[f_i(o)]| \leq 2\epsilon \mathbb{E}_{o \sim M(t)}[f_i(o)].$$

If we view f_i as representing the “expected future utility” for agent i , it is therefore natural to model agent i 's cost for having his data used in an ϵ -differentially private computation as being linear in ϵ . That is,

we think of agent i as being parameterized by some value $v_i \in \mathbb{R}$, and take:

$$c_i(o, \mathcal{M}, t) = \epsilon v_i,$$

where ϵ is the smallest value such that \mathcal{M} is ϵ -differentially private. Here we imagine v_i to represent a quantity like $\mathbb{E}_{o \sim M(t)}[f_i(o)]$. In this setting, c_i does not depend on the outcome o or the type profile t .

Using this naïve privacy measure, we discuss a basic problem in private data analysis: how to collect the data, when the owners of the data value their privacy and insist on being compensated for it. In this setting, there is no “outcome” that agents value, other than payments, there is only dis-utility for privacy loss. We will then discuss shortcomings of this (and other) measures of the dis-utility for privacy loss, as well as privacy in more general mechanism design settings when agents *do* have utility for the outcome of the mechanism.

10.3.1 A private generalization of the VCG mechanism

Suppose we have a general social choice problem, defined by an outcome space \mathcal{O} , and a set of agents N with arbitrary preferences over the outcomes given by $u_i : \mathcal{O} \rightarrow [0, 1]$. We might want to choose an outcome $o \in \mathcal{O}$ to maximize the *social welfare* $F(o) = \frac{1}{n} \sum_{i=1}^n u_i(o)$. It is well known that in any such setting, the *VCG* mechanism can implement the outcome o^* which exactly maximizes the social welfare, while charging payments that make truth-telling a dominant strategy. What if we want to achieve the same result, while also preserving privacy? How must the privacy parameter ϵ trade off with our approximation to the optimal social welfare?

Recall that we could use the exponential mechanism to choose an outcome $o \in \mathcal{O}$, with quality score F . For privacy parameter ϵ , this would give a distribution \mathcal{M}_ϵ defined to be $\Pr[\mathcal{M}_\epsilon = o] \propto \exp\left(\frac{\epsilon F(o)}{2n}\right)$. Moreover, this mechanism has good social welfare properties: with probability $1 - \beta$, it selects some o such that: $F(o) \geq F(o^*) - \frac{2}{\epsilon n} \left(\ln \frac{|\mathcal{O}|}{\beta} \right)$. But as we saw, differential privacy only gives ϵ -approximate truthfulness.

However, it can be shown that \mathcal{M}_ϵ is the solution to the following exact optimization problem:

$$\mathcal{M}_\epsilon = \arg \max_{\mathcal{D} \in \Delta \mathcal{O}} \left(\mathbb{E}_{o \sim \mathcal{D}}[F(o)] + \frac{2}{\epsilon n} H(\mathcal{D}) \right),$$

where H represents the *Shannon Entropy* of the distribution \mathcal{D} . In other words, the exponential mechanism is the distribution which exactly maximizes the expected social welfare, *plus* the entropy of the distribution weighted by $2/(\epsilon n)$. This is significant for the following reason: it is known that any mechanism that *exactly* maximizes expected player utilities in any finite range (known as maximal in distributional range mechanisms) can be paired with payments to be made exactly dominant strategy truthful. The exponential mechanism is the distribution that *exactly* maximizes expected social welfare, plus entropy. In other words, if we imagine that we have added a single additional player whose utility is exactly the entropy of the distribution, then the exponential mechanism is maximal in distributional range. Hence, it can be paired with payments that make truthful reporting a dominant strategy for all players — in particular, for the n real players. Moreover, it can be shown how to charge payments in such a way as to preserve privacy. The upshot is that for any social choice problem, the social welfare can be approximated in a manner that both preserves differential privacy, and is exactly truthful.

10.3.2 The sensitive surveyor's problem

In this section, we consider the problem of a data analyst who wishes to conduct a study using the private data of a collection of individuals. However, he must *convince* these individuals to hand over their data! Individuals experience costs for privacy loss. The data analyst can mitigate these costs by guaranteeing differential privacy and compensating them for their loss, while trying to get a representative sample of data.

Consider the following stylized problem of the sensitive surveyor Alice. She is tasked with conducting a survey of a set of n individuals N , to determine what proportion of the individuals $i \in N$ satisfy some property $P(i)$. Her ultimate goal is to discover the true value of this statistic, $s = \frac{1}{n} |\{i \in N : P(i)\}|$, but if that is not possible, she will be

satisfied with some estimate \hat{s} such that the error, $|\hat{s} - s|$, is minimized. We will adopt a notion of accuracy based on large deviation bounds, and say that a surveying mechanism is α -accurate if $\Pr[|\hat{s} - s| \geq \alpha] \leq \frac{1}{3}$. The inevitable catch is that individuals value their privacy and will not participate in the survey for free. Individuals experience some *cost* as a function of their loss in privacy when they interact with Alice, and must be compensated for this loss. To make matters worse, these individuals are rational (i.e., selfish) agents, and are apt to misreport their costs to Alice if doing so will result in a financial gain. This places Alice's problem squarely in the domain of mechanism design, and requires Alice to develop a scheme for trading off statistical accuracy with cost, all while managing the incentives of the individuals.

As an aside, this stylized problem is broadly relevant to any organization that makes use of collections of potentially sensitive data. This includes, for example, the use of search logs to provide search query completion and the use of browsing history to improve search engine ranking, the use of social network data to select display ads and to recommend new links, and the myriad other data-driven services now available on the web. In all of these cases, value is being derived from the statistical properties of a collection of sensitive data in exchange for some payment.⁴

Collecting data in exchange for some fixed price could lead to a biased estimate of population statistics, because such a scheme will result in collecting data only from those individuals who value their privacy less than the price being offered. However, without interacting with the agents, we have no way of knowing what price we can offer so that we will have broad enough participation to guarantee that the answer we collect has only small bias. To obtain an accurate estimate of the statistic, it is therefore natural to consider buying private data using an auction — as a means of discovering this price. There are two obvious obstacles which one must confront when conducting an auction for private data, and an additional obstacle which is less obvious but more insidious. The first obstacle is that one must have a quantitative

⁴The payment need not be explicit and/or dollar denominated — for example, it may be the use of a “free” service.

formalization of “privacy” which can be used to measure agents’ costs under various operations on their data. Here, differential privacy provides an obvious tool. For small values of ϵ , because $\exp(\epsilon) \approx (1 + \epsilon)$, and so as discussed earlier, a simple (but possibly naive) first cut at a model is to view each agent as having some *linear* cost for participating in a private study. We here imagine that each agent i has an unknown value for privacy v_i , and experiences a cost $c_i(\epsilon) = \epsilon v_i$ when his private data is used in an ϵ -differentially private manner.⁵ The second obstacle is that our objective is to trade off with *statistical accuracy*, and the latter is not well-studied objective in mechanism design.

The final, more insidious obstacle, is that an individual’s cost for privacy loss may be highly correlated with his private data itself! Suppose we only know Bob has a high value for privacy of his AIDS status, but do not explicitly know his AIDS status itself. This is already dis-
closive because Bob’s AIDS status is likely correlated with his value for privacy, and knowing that he has a high cost for privacy lets us update our belief about what his private data might be. More to the point, suppose that in the first step of a survey of AIDS prevalence, we ask each individual to report their value for privacy, with the intention of then running an auction to choose which individuals to buy data from. If agents report truthfully, we may find that the reported values naturally form two clusters: low value agents, and high value agents. In this case, we may have learned something about the population statistic even before collecting any data or making any payments—and therefore, the agents will have already experienced a cost. As a result, the agents may misreport their value, which could introduce a bias in the survey results. This phenomenon makes direct revelation mechanisms problematic, and distinguishes this problem from classical mechanism design.

Armed with a means of quantifying an agent i ’s loss for allowing his data to be used by an ϵ -differentially-private algorithm ($c_i(\epsilon) = \epsilon v_i$), we are almost ready to describe results for the sensitive surveyor’s problem. Recall that a differentially private algorithm is some mapping $M : \mathcal{T}^n \rightarrow \mathcal{O}$, for a general type space \mathcal{T} . It remains to define what

⁵As we will discuss later, this assumption can be problematic.

exactly the type space \mathcal{T} is. We will consider two models. In both models, we will associate with each individual a bit $b_i \in \{0, 1\}$ which represents whether they satisfy the sensitive predicate $P(i)$, as well as a value for privacy $v_i \in \mathbb{R}^+$.

1. In the *insensitive value model*, we calculate the ϵ parameter of the private mechanism by letting the type space be $\mathcal{T} = \{0, 1\}$: i.e., we measure privacy cost only with respect to how the mechanism treats the sensitive bit b_i , and ignore how it treats the reported values for privacy, v_i .⁶
2. In the *sensitive value model*, we calculate the ϵ parameter of the private mechanism by letting the type space be $\mathcal{T} = (\{0, 1\} \times \mathbb{R}^+)$: i.e., we measure privacy with respect to how it treats the pair (b_i, v_i) for each individual.

Intuitively, the insensitive value model treats individuals as ignoring the potential privacy loss due to correlations between their values for privacy and their private bits, whereas the sensitive value model treats individuals as assuming these correlations are worst-case, i.e., their values v_i are just as disclosive as their private bits b_i . It is known that in the insensitive value model, one can derive approximately optimal direct revelation mechanisms that achieve high accuracy and low cost. By contrast, in the *sensitive value model*, no individually rational direct revelation mechanism can achieve any non-trivial accuracy.

This leaves a somewhat unsatisfying state of affairs. The sensitive value model captures the delicate issues that we really want to deal with, and yet there we have an impossibility result! Getting around this result in a satisfying way (e.g., by changing the model, or the powers of the mechanism) remains an intriguing open question.

10.3.3 Better measures for the cost of privacy

In the previous section, we took the naive modeling assumption that the cost experienced by participation in an ϵ -differentially private mechanism M was $c_i(o, \mathcal{M}, t) = \epsilon v_i$ for some numeric value v_i . This measure

⁶That is, the part of the mapping dealing with reported values need not be differentially private.

is problematic for several reasons. First, although differential privacy promises that any agent's loss in utility is *upper bounded* by a quantity that is (approximately) linear in ϵ , there is no reason to believe that agents' costs are *lower bounded* by such a quantity. That is, while taking $c_i(o, \mathcal{M}, t) \leq \epsilon v_i$ is well motivated, there is little support for making the inequality an equality. Second, (it turns out) *any* privacy measure which is a deterministic function only of ϵ (not just a linear function) leads to problematic behavioral predictions.

So how else might we model c_i ? One natural measure is the *mutual information* between the reported type of agent i , and the outcome of the mechanism. For this to be well defined, we must be in a world where each agent's type t_i is drawn from a known prior, $t_i \sim \mathcal{T}$. Each agent's strategy is a mapping $\sigma_i : \mathcal{T} \rightarrow \mathcal{T}$, determining what type he reports, given his true type. We could then define

$$c_i(o, \mathcal{M}, \sigma) = I(\mathcal{T}; \mathcal{M}(t_{-i}, \sigma(\mathcal{T}))),$$

where I is the mutual information between the random variable \mathcal{T} representing the prior on agent i 's type, and $\mathcal{M}(t_{-i}, \sigma(\mathcal{T}))$, the random variable representing the outcome of the mechanism, given agent i 's strategy.

This measure has significant appeal, because it represents how "related" the output of the mechanism is to the true type of agent i . However, in addition to requiring a prior over agent types, observe an interesting paradox that results from this measure of privacy loss. Consider a world in which there are two kinds of sandwich breads: Rye (R), and Wheat (W). Moreover, in this world, sandwich preferences are highly embarrassing and held private. The prior on types \mathcal{T} is uniform over R and W, and the mechanism \mathcal{M} simply gives agent i a sandwich of the type that he purports to prefer. Now consider two possible strategies, σ_{truthful} and σ_{random} . σ_{truthful} corresponds to truthfully reporting sandwich preferences (and subsequently leads to eating the preferred sandwich type), while σ_{random} randomly reports independent of true type (and results in the preferred sandwich only half the time). The cost of using the random strategy is $I(\mathcal{T}; \mathcal{M}(t_{-i}, \sigma_{\text{random}}(\mathcal{T}))) = 0$, since the output is independent of agent i 's type. On the other hand, the cost of truthfully reporting is $I(\mathcal{T}; \mathcal{M}(t_{-i}, \sigma_{\text{truthful}}(\mathcal{T}))) = 1$, since

the sandwich outcome is now the identity function on agent i 's type. However, from the perspective of any outside observer, the two strategies are indistinguishable! In both cases, agent i receives a uniformly random sandwich. Why then should anyone choose the random strategy? So long as an adversary *believes* they are choosing randomly, they should choose the honest strategy.

Another approach, which does not need a prior on agent types, is as follows. We may model agents as having a cost function c_i that satisfies:

$$|c_i(o, \mathcal{M}, t)| = \ln \left(\max_{t_i, t'_i \in \mathcal{T}} \frac{\Pr[\mathcal{M}(t_i, t_{-i}) = o]}{\Pr[\mathcal{M}(t'_i, t_{-i}) = o]} \right).$$

Note that if \mathcal{M} is ϵ -differentially private, then

$$\max_{t \in \mathcal{T}^n} \max_{o \in \mathcal{O}} \max_{t_i, t'_i \in \mathcal{T}} \ln \left(\frac{\Pr[\mathcal{M}(t_i, t_{-i}) = o]}{\Pr[\mathcal{M}(t'_i, t_{-i}) = o]} \right) \leq \epsilon.$$

That is, we can view differential privacy as bounding the *worst-case* privacy loss over all possible outcomes, whereas the measure proposed here considers only the privacy loss for the outcome o (and type vector t) actually realized. Thus, for any differentially private mechanism \mathcal{M} , $|c_i(o, \mathcal{M}, t)| \leq \epsilon$ for all o, t , but it will be important that the cost can vary by outcome.

We can then consider the following allocation rule for maximizing social welfare $F(o) = \sum_{i=1}^n u_i(o)$.⁷ We discuss the case when $|\mathcal{O}| = 2$ (which does not require payments), but it is possible to analyze the general case (with payments), which privately implements the VCG mechanism for any social choice problem.

1. For each outcome $o \in \mathcal{O}$, choose a random number r_o from the distribution $\Pr[r_o = x] \propto \exp(-\epsilon|x|)$.
2. Output $o^* = \arg \max_{o \in \mathcal{O}} (F(o) + r_o)$.

The above mechanism is ϵ -differentially private, and that it is truthful for privacy aware agents, so long as for each agent i , and for the two outcomes $o, o' \in \mathcal{O}$, $|\mu_i(o) - \mu_i(o')| > 2\epsilon$. Note that this will be true

⁷This allocation rule is extremely similar to, and indeed can be modified to be identical to the exponential mechanism.

for small enough ϵ so long as agent utilities for outcomes are distinct. The analysis proceeds by considering an arbitrary fixed realization of the random variables r_o , and an arbitrary deviation t'_i from truthful reporting for the i th agent. There are two cases: In the first case, the deviation does not change the outcome o of the mechanism. In this case, *neither* the agent's utility for the outcome μ_i , nor his cost for privacy loss c_i change at all, and so the agent does not benefit from deviating. In the second case, if the outcome changes from o to o' when agent i deviates, it must be that $\mu_i(o') < \mu_i(o) - 2\epsilon$. By differential privacy, however, $|c_i(o, \mathcal{M}, t) - c_i(o', \mathcal{M}, t)| \leq 2\epsilon$, and so the change in privacy cost cannot be enough to make it beneficial.

Finally, the most conservative approach to modeling costs for privacy generally considered is as follows. Given an ϵ -differentially private mechanism \mathcal{M} , assume only that

$$c_i(o, \mathcal{M}, t) \leq \epsilon v_i,$$

for some number v_i . This is similar to the linear cost functions that we considered earlier, but crucially, here we assume only an upper bound. This assumption is satisfied by all of the other models for privacy cost that we have considered thus far. It can be shown that many mechanisms that combine a differentially private algorithm with a punishing mechanism that has the ability to restrict user choices, like those that we considered in Section 10.2.3, maintain their truthfulness properties in the presence of agents with preferences for privacy, so long as the values v_i are bounded.

10.4 Bibliographical notes

This section is based off of a survey of Pai and Roth [70] and a survey of Roth [73]. The connections between differential privacy and mechanism design were first suggested by Jason Hartline and investigated by McSherry and Talwar in their seminal work, “Mechanism Design via Differential Privacy” [61], where they considered the application of differential privacy to designing approximately truthful digital goods auctions. The best result for exactly truthful mechanisms in the digital goods setting is due to Balcan et al. [2].

The problem of designing exactly truthful mechanisms using differential privacy as a tool was first explored by Nissim, Smorodinsky, and Tennenholz in [69], who also first posed a criticism as differential privacy (by itself) used as a solution concept. The example in this section of using differential privacy to obtain exactly truthful mechanisms is taken directly from [69]. The sensitive surveyors problem was first considered by Ghosh and Roth [36], and expanded on by [56, 34, 75, 16]. Fleischer and Lyu [34] consider the Bayesian setting discussed in this section, and Ligett and Roth [56] consider the worst-case setting with take-it-or-leave-it offers, both in an attempt to get around the impossibility result of [36]. Ghosh and Ligett consider a related model in which participation decisions (and privacy guarantees) are determined only in equilibrium [35].

The question of conducting mechanism design in the presence of agents who explicitly value privacy as part of their utility function was first raised by the influential work of Xiao [85], who considered (among other measures for privacy cost) the mutual information cost function. Following this, Chen et al. [15] and Nissim et al. [67] showed how in two distinct models, truthful mechanisms can sometimes be designed even for agents who value privacy. Chen Chong, Kash, Moran, and Vadhan considered the outcome-based cost function that we discussed in this section, and Nissim, Orlandi, and Smorodinsky considered the conservative model of only upper bounding each agent's cost by a linear function in ϵ . The “sandwich paradox” of valuing privacy according to mutual information is due to Nissim, Orlandi, and Smorodinsky.

Huang and Kannan proved that the exponential mechanism could be made exactly truthful with the addition of payments [49]. Kearns Pai, Roth, and Ullman showed how differential privacy could be used to derive asymptotically truthful equilibrium selection mechanisms [54] by privately computing correlated equilibria in large games. These results were strengthened by Rogers and Roth [71], who showed how to privately compute approximate *Nash* equilibria in large congestion games, which leads to stronger incentive properties of the mechanism. Both of these papers use the solution concept of “Joint Differential Privacy,”

which requires that for every player i , the joint distribution on messages sent to *other* players $j \neq i$ be differentially private in *is* report. This solution concept has also proven useful in other settings of private mechanism design settings, including an algorithm for computing private matchings by Hsu et al. [47].

11

Differential Privacy and Machine Learning

One of the most useful tasks in data analysis is machine learning: the problem of automatically finding a simple rule to accurately predict certain unknown characteristics of never before seen data. Many machine learning tasks can be performed under the constraint of differential privacy. In fact, the constraint of privacy is not necessarily at odds with the goals of machine learning, both of which aim to extract information from the distribution from which the data was drawn, rather than from individual data points. In this section, we survey a few of the most basic results on private machine learning, without attempting to cover this large field completely.

The goal in machine learning is very often similar to the goal in private data analysis. The *learner* typically wishes to learn some simple rule that explains a data set. However, she wishes this rule to generalize — that is, it should be that the rule she learns not only correctly describes the data that she has on hand, but that it should also be able to correctly describe *new* data that is drawn from the same distribution. Generally, this means that she wants to learn a rule that captures distributional information about the data set on hand, in a way that does not depend too specifically on any single data point. Of

course, this is exactly the goal of private data analysis — to reveal *distributional information* about the private data set, without revealing too much about any single individual in the dataset. It should come as no surprise then that machine learning and private data analysis are closely linked. In fact, as we will see, we are often able to perform private machine learning *nearly as accurately, with nearly the same number of examples* as we can perform non-private machine learning.

Let us first briefly define the problem of machine learning. Here, we will follow Valiant's *PAC* (Or *Probably Approximately Correct*) model of machine learning. Let $\mathcal{X} = \{0, 1\}^d$ be the domain of “unlabeled examples.” Think of each $x \in \mathcal{X}$ as a vector containing d boolean attributes. We will think of vectors $x \in \mathcal{X}$ as being paired with *labels* $y \in \{0, 1\}$.

Definition 11.1. A *labeled example* is a pair $(x, y) \in \mathcal{X} \times \{0, 1\}$: a vector paired with a label.

A learning problem is defined as a distribution \mathcal{D} over labeled examples. The goal will be to find a function $f : \mathcal{X} \rightarrow \{0, 1\}$ that correctly labels almost all of the examples drawn from the distribution.

Definition 11.2. Given a function $f : \mathcal{X} \rightarrow \{0, 1\}$ and a distribution \mathcal{D} over labeled examples, the *error rate* of f on \mathcal{D} is:

$$\text{err}(f, \mathcal{D}) = \Pr_{(x,y) \sim \mathcal{D}}[f(x) \neq y]$$

We can also define the error rate of f over a finite sample D :

$$\text{err}(f, D) = \frac{1}{|D|} |\{(x, y) \in D : f(x) \neq y\}|.$$

A learning *algorithm* gets to observe some number of labeled examples drawn from \mathcal{D} , and has the goal of finding a function f with as small an error rate as possible when measured on \mathcal{D} . Two parameters in measuring the quality of a learning algorithm are its running time, and the number of examples it needs to see in order to find a good hypothesis.

Definition 11.3. An algorithm A is said to PAC-learn a class of functions C over d dimensions if for every $\alpha, \beta > 0$, there exists an

$m = \text{poly}(d, 1/\alpha, \log(1/\beta))$ such that for every distribution \mathcal{D} over labeled examples, A takes as input m labeled examples drawn from \mathcal{D} and outputs a hypothesis $f \in C$ such that with probability $1 - \beta$:

$$\text{err}(f, \mathcal{D}) \leq \min_{f^* \in C} \text{err}(f^*, \mathcal{D}) + \alpha$$

If $\min_{f^* \in C} \text{err}(f^*, \mathcal{D}) = 0$, the learner is said to operate in the *realizable* setting (i.e., there exists some function in the class which perfectly labels the data). Otherwise, the learner is said to operate in the *agnostic* setting. If A also has run time that is polynomial in $d, 1/\alpha$, and $\log(1/\beta)$, then the learner is said to be *efficient*. If there is an algorithm which PAC-learns C , then C is said to be PAC-learnable.

The above definition of learning allows the learner to have direct access to labeled examples. It is sometimes also useful to consider models of learning in which the algorithm only has oracle access to some noisy information about \mathcal{D} .

Definition 11.4. A *statistical query* is some function $\phi : \mathcal{X} \times \{0, 1\} \rightarrow [0, 1]$. A *statistical query oracle* for a distribution over labeled examples \mathcal{D} with tolerance τ is an oracle $\mathcal{O}_{\mathcal{D}}^\tau$ such that for every statistical query ϕ :

$$|\mathcal{O}_{\mathcal{D}}^\tau(\phi) - \mathbb{E}_{(x,y) \sim \mathcal{D}}[\phi(x, y)]| \leq \tau$$

In other words, an SQ oracle takes as input a statistical query ϕ , and outputs some value that is guaranteed to be within $\pm \tau$ of the expected value of ϕ on examples drawn from \mathcal{D} .

The statistical query model of learning was introduced to model the problem of learning in the presence of noise.

Definition 11.5. An algorithm A is said to SQ-learn a class of functions C over d dimensions if for every $\alpha, \beta > 0$ there exists an $m = \text{poly}(d, 1/\alpha, \log(1/\beta))$ such that A makes at most m queries of tolerance $\tau = 1/m$ to $\mathcal{O}_{\mathcal{D}}^\tau$, and with probability $1 - \beta$, outputs a hypothesis $f \in C$ such that:

$$\text{err}(f, \mathcal{D}) \leq \min_{f^* \in C} \text{err}(f^*, \mathcal{D}) + \alpha$$

Note that an SQ learning algorithm does not get any access to \mathcal{D} except through the SQ oracle. As with PAC learning, we can talk about an SQ learning algorithm operating in either the realizable or the agnostic setting, and talk about the computational efficiency of the learning algorithm. We say that a class C is SQ learnable if there exists an SQ learning algorithm for C .

11.1 The sample complexity of differentially private machine learning

Perhaps the first question that one might ask, with respect to the relationship between privacy and learning, is “When is it possible to privately perform machine learning”? In other words, you might ask for a PAC learning algorithm that takes as input a dataset (implicitly assumed to be sampled from some distribution \mathcal{D}), and then privately output a hypothesis f that with high probability has low error over the distribution. A more nuanced question might be, “How many *additional* samples are required to privately learn, as compared with the number of samples *already required* to learn without the constraint of differential privacy?” Similarly, “How much additional run-time is necessary to privately learn, as compared with the run-time required to learn non-privately?” We will here briefly sketch known results for $(\varepsilon, 0)$ -differential privacy. In general, better results for (ε, δ) -differential privacy will follow from using the advanced composition theorem.

A foundational *information theoretic result* in private machine learning is that private PAC learning is possible with a polynomial number of samples if and only if non-private PAC learning is possible with a polynomial number of samples, even in the agnostic setting. In fact, the increase in sample complexity necessary is relatively small — however, this result does not preserve *computational efficiency*. One way to do this is directly via the exponential mechanism. We can instantiate the exponential mechanism with a range $R = C$, equal to the class of queries to be learned. Given a database D , we can use the quality score $q(f, D) = -\frac{1}{|D|} |\{(x, y) \in D : f(x) \neq y\}|$: i.e., we seek to minimize the fraction of misclassified examples in the private dataset. This is clearly

a $1/n$ sensitive function of the private data, and so we have via our utility theorem for the exponential mechanism that with probability $1 - \beta$, this mechanism returns a function $f \in C$ that correctly labels an $\text{OPT} - \frac{2(\log |C| + \log \frac{1}{\beta})}{\varepsilon n}$ fraction of the points in the database correctly. Recall, however, that in the learning setting, we view the database D as consisting of n i.i.d. draws from some distribution over labeled examples \mathcal{D} . Recall the discussion of sampling bounds in Lemma 4.3. A Chernoff bound combined with a union bound tells us that with high probability, if D consists of n i.i.d. samples drawn from \mathcal{D} , then for all $f \in C$: $|\text{err}(f, D) - \text{err}(f, \mathcal{D})| \leq O(\sqrt{\frac{\log |C|}{n}})$. Hence, if we wish to find a hypothesis that has error within α of the optimal error on the distribution \mathcal{D} , it suffices to draw a database D consisting of $n \geq \log |C| / \alpha^2$ samples, and learn the best classifier f^* on D .

Now consider the problem of privately PAC learning, using the exponential mechanism as described above. Recall that, by Theorem 3.11, it is highly unlikely that the exponential mechanism will return a function f with utility score that is inferior to that of the optimal f^* by more than an additive factor of $O((\Delta u / \varepsilon) \log |C|)$, where in this case Δu , the sensitivity of the utility function, is $1/n$. That is, with high probability the exponential mechanism will return a function $f \in C$ such that:

$$\begin{aligned} \text{err}(f, D) &\leq \min_{f^* \in C} \text{err}(f^*, D) + O\left(\frac{(\log |C|)}{\varepsilon n}\right) \\ &\leq \min_{f^* \in C} \text{err}(f^*, \mathcal{D}) + O\left(\sqrt{\frac{\log |C|}{n}}\right) + O\left(\frac{(\log |C|)}{\varepsilon n}\right). \end{aligned}$$

Hence, if we wish to find a hypothesis that has error within α of the optimal error on the distribution \mathcal{D} , it suffices to draw a database D consisting of:

$$n \geq O\left(\max\left(\frac{\log |C|}{\varepsilon \alpha}, \frac{\log |C|}{\alpha^2}\right)\right),$$

which is not asymptotically any more than the database size that is required for non-private learning, whenever $\varepsilon \geq \alpha$.

A corollary of this simple calculation¹ is that (ignoring computational efficiency), a class of functions C is PAC learnable if and only if it is privately PAC learnable.

Can we say something stronger about a concept class C that is SQ learnable? Observe that if C is efficiently SQ learnable, then the learning algorithm for C need only access the data through an SQ oracle, which is very amenable to differential privacy: note that an SQ oracle answers an expectation query defined over a predicate $\phi(x, y) \in [0, 1]$, $\mathbb{E}_{(x,y) \sim \mathcal{D}}[\phi(x, y)]$, which is only $1/n$ sensitive when estimated on a database D which is a sample of size n from \mathcal{D} . Moreover, the learning algorithm does not need to receive the answer exactly, but can be run with any answer a that has the property that: $|\mathbb{E}_{(x,y) \sim \mathcal{D}}[\phi(x, y)] - a| \leq \tau$: that is, the algorithm can be run using *noisy answers* on *low sensitivity queries*. The benefit of this is that we can answer such queries computationally efficiently, using the Laplace mechanism — but at the expense of requiring a potentially large sample size. Recall that the Laplace mechanism can answer m $1/n$ sensitive queries with $(\varepsilon, 0)$ -differential privacy and with expected worst-case error $\alpha = O(\frac{m \log m}{\varepsilon n})$. Therefore, an *SQ* learning algorithm which requires the answers to m queries with accuracy α can be run with a sample size of $n = O(\max(\frac{m \log m}{\varepsilon \alpha}, \frac{\log m}{\alpha^2}))$. Let us compare this to the sample size required for a non-private *SQ* learner. If the *SQ* learner needs to make m queries to tolerance α , then by a Chernoff bound and a union bound, a sample size of $O(\log m / \alpha^2)$ suffices. Note that for $\varepsilon = O(1)$ and error $\alpha = O(1)$, the non-private algorithm potentially requires exponentially fewer samples. However, at the error tolerance $\alpha \leq 1/m$ as allowed in the definition of SQ learning, the sample complexity for private SQ learning is no worse than the sample complexity for non-private SQ learning, for $\varepsilon = \Theta(1)$.

The upshot is that *information theoretically*, privacy poses very little hinderance to machine learning. Moreover, for any algorithm that accesses the data only though an SQ oracle,² then the reduction to

¹Together with corresponding lower bounds that show that for general C , it is not possible to non-privately PAC learn using a sample with $o(\log |C| / \alpha^2)$ points.

²And in fact, almost every class (with the lone exception of *parity functions*) of functions known to be PAC learnable is also learnable using only an SQ oracle.

private learning is immediate via the Laplace mechanism, and preserves computational efficiency!

11.2 Differentially private online learning

In this section, we consider a slightly different learning problem, known as the problem of *learning from expert advice*. This problem will appear somewhat different from the classification problems that we discussed in the previous section, but in fact, the simple algorithm presented here is extremely versatile, and can be used to perform classification among many other tasks which we will not discuss here.

Imagine that you are betting on horse races, but unfortunately know nothing about horses! Nevertheless, you have access to the opinions of some k experts, who every day make a prediction about which horse is going to win. Each day you can choose one of the experts whose advice you will follow, and each day, following your bet, you learn which horse actually won. How should you decide which expert to follow each day, and how should you evaluate your performance? The experts are not perfect (in fact they might not even be any good!), and so it is not reasonable to expect you to make the correct bet all of the time, or even most of the time if none of the experts do so. However, you might have a weaker goal: can you bet on horses in such a way so that you do almost as well as *the best expert, in hindsight*?

Formally, an online learning algorithm A operates in the following environment:

1. Each day $t = 1, \dots, T$:
 - (a) A chooses an expert $a_t \in \{1, \dots, k\}$
 - (b) A observes a loss $\ell_i^t \in [0, 1]$ for each expert $i \in \{1, \dots, k\}$ and experiences loss $\ell_{a_t}^t$.

For a sequence of losses $\ell^{\leq T} \equiv \{\ell^t\}_{t=1}^T$, we write:

$$L_i(\ell^{\leq T}) = \frac{1}{T} \sum_{t=1}^T \ell_i^t$$

to denote the total average loss of expert i over all T rounds, and write

$$L_A(\ell^{\leq T}) = \frac{1}{T} \sum_{t=1}^T \ell_{a_t}^t$$

to denote the total average loss of the algorithm.

The *regret* of the algorithm is defined to be the difference between the loss that it actually incurred, and the loss of the *best* expert in hindsight:

$$\text{Regret}(A, \ell^{\leq T}) = L_A(\ell^{\leq T}) - \min_i L_i(\ell^{\leq T}).$$

The goal in online learning is to design algorithms that have the guarantee that for *all possible loss sequences* $\ell^{\leq T}$, even adversarialy chosen, the regret is guaranteed to tend to zero as $T \rightarrow \infty$. In fact, this is possible using the multiplicative weights algorithm (known also by many names, e.g., the Randomized Weighted Majority Algorithm, Hedge, Exponentiated Gradient Descent, and multiplicative weights being among the most popular).

Remark 11.1. We have already seen this algorithm before in Section 4 — this is just the multiplicative weights update rule in another guise! In fact, it would have been possible to derive all of the results about the private multiplicative weights mechanism directly from the regret bound we state in Theorem 11.1.

Algorithm 15 The Multiplicative Weights (or Randomized Weighted Majority (RWM)) algorithm, version 1. It takes as input a stream of losses ℓ^1, ℓ^2, \dots and outputs a stream of actions a_1, a_2, \dots . It is parameterized by an update parameter η .

RWM(η):

```

For each  $i \in \{1, \dots, k\}$ , let  $w_i \leftarrow 1$ .
for  $t = 1, \dots$  do
    Choose action  $a_t = i$  with probability proportional to  $w_i$ 
    Observe  $\ell^t$  and set  $w_i \leftarrow w_i \cdot \exp(-\eta \ell_i^t)$ , for each  $i \in [k]$ 
end for

```

It turns out that this simple algorithm already has a remarkable regret bound.

Theorem 11.1. For any adversarially chosen sequence of losses of length T , $\ell^{\leq T} = (\ell^1, \dots, \ell^T)$ the Randomized Weighted Majority algorithm with update parameter η has the guarantee that:

$$\mathbb{E}[\text{Regret}(\text{RWM}(\eta), \ell^{\leq T})] \leq \eta + \frac{\ln(k)}{\eta T}, \quad (11.1)$$

where k is the number of experts. Choosing $\eta = \sqrt{\frac{\ln k}{T}}$ gives:

$$\mathbb{E}[\text{Regret}(\text{RWM}(\eta), \ell^{\leq T})] \leq 2\sqrt{\frac{\ln k}{T}}.$$

This remarkable theorem states that even faced with an adversarial sequence of losses, the Randomized Weighted Majority algorithm can do as well, on average, as the best expert among k in hindsight, minus only an additional additive term that goes to zero at a rate of $O(\sqrt{\frac{\ln k}{T}})$. In other words, after at most $T \leq 4\frac{\ln k}{\alpha^2}$ rounds, the regret of the randomized weighted majority algorithm is guaranteed to be at most α ! Moreover, this bound is the best possible.

Can we achieve something similar, but under the constraint of differential privacy? Before we can ask this question, we must decide *what is the input database*, and at what granularity we would like to protect privacy? Since the input is the collection of loss vectors $\ell^{\leq T} = (\ell^1, \dots, \ell^T)$, it is natural to view $\ell^{\leq T}$ as the database, and to view a neighboring database $\hat{\ell}^{\leq T}$ as one that differs in the entire loss vector in any single timestep: i.e., one in which for some fixed timestep t , $\hat{\ell}^i = \ell^i$ for all $i \neq t$, but in which ℓ^t and $\hat{\ell}^t$ can differ arbitrarily. The output of the algorithm is the sequence of actions that it chooses, a_1, \dots, a_T , and it is this that we wish to be output in a differentially private manner.

Our first observation is that the randomized weighted majority algorithm chooses an action at each day t in a familiar manner! We here rephrase the algorithm in an equivalent way:

It chooses an action a_t with probability proportional to: $\exp(-\eta \sum_{j=1}^{t-1} \ell_i^j)$, which is simply the exponential mechanism with quality score $q(i, \ell^{\leq T}) = \sum_{j=1}^{t-1} \ell_i^j$, and privacy parameter $\varepsilon = 2\eta$. Note that because each $\ell_i^t \in [0, 1]$, the quality function has sensitivity 1. Thus,

Algorithm 16 The Multiplicative Weights (or Randomized Weighted Majority (RWM)) algorithm, rephrased. It takes as input a stream of losses ℓ^1, ℓ^2, \dots and outputs a stream of actions a_1, a_2, \dots . It is parameterized by an update parameter η .

RWM(η):

```

for  $t = 1, \dots$  do
    Choose action  $a_t = i$  with probability proportional to
     $\exp(-\eta \sum_{j=1}^{t-1} \ell_i^j)$ 
    Observe  $\ell^t$ 
end for

```

each round t , the randomized weighted majority algorithm chooses an action a_t in a way that preserves 2η differential privacy, so to achieve privacy ε it suffices to set $\eta = \varepsilon/2$.

Moreover, over the course of the run of the algorithm, it will choose an action T times. If we want the *entire* run of the algorithm to be (ε, δ) -differentially private for some ε and δ , we can thus simply apply our composition theorems. Recall that by Theorem 3.20, since there are T steps in total, if each step of the algorithm is $(\varepsilon', 0)$ -differentially private for $\varepsilon' = \varepsilon/\sqrt{8T \ln(1/\delta)}$, then the entire algorithm will be (ε, δ) differentially private. Thus, the following theorem is immediate by setting $\eta = \varepsilon'/2$:

Theorem 11.2. For a sequence of losses of length T , the algorithm RWM(η) with $\eta = \frac{\varepsilon}{\sqrt{32T \ln(1/\delta)}}$ is (ε, δ) -differentially private.

Remarkably, we get this theorem *without modifying the original randomized weighted majority algorithm at all*, but rather just by setting η appropriately. In some sense, we are getting privacy for free! We can therefore use Theorem 11.1, the utility theorem for the RWM algorithm, without modification as well:

Theorem 11.3. For any adversarially chosen sequence of losses of length T , $\ell^{\leq T} = (\ell^1, \dots, \ell^T)$ the Randomized Weighted Majority

algorithm with update parameter $\eta = \frac{\varepsilon}{\sqrt{32T \ln(1/\delta)}}$ has the guarantee that:

$$\begin{aligned}\mathbb{E}[\text{Regret}(\text{RWM}(\eta), \ell^{\leq T})] &\leq \frac{\varepsilon}{\sqrt{32T \ln(1/\delta)}} + \frac{\sqrt{32 \ln(1/\delta)} \ln k}{\varepsilon \sqrt{T}} \\ &\leq \frac{\sqrt{128 \ln(1/\delta)} \ln k}{\varepsilon \sqrt{T}},\end{aligned}$$

where k is the number of experts.

Since the per-round loss at each time step t is an independently chosen random variable (over the choices of a_t) with values bounded in $[-1, 1]$, we can also apply a Chernoff bound to get a high probability guarantee:

Theorem 11.4. For any adversarially chosen sequence of losses of length T , $\ell^{\leq T} = (\ell^1, \dots, \ell^T)$ the Randomized Weighted Majority algorithm with update parameter $\eta = \frac{\varepsilon}{\sqrt{32T \ln(1/\delta)}}$ produces a sequence of actions such that with probability at least $1 - \beta$:

$$\begin{aligned}\text{Regret}(\text{RWM}(\eta), \ell^{\leq T}) &\leq \frac{\sqrt{128 \ln(1/\delta)} \ln k}{\varepsilon \sqrt{T}} + \sqrt{\frac{\ln k / \beta}{T}} \\ &= O\left(\frac{\sqrt{\ln(1/\delta)} \ln(k/\beta)}{\varepsilon \sqrt{T}}\right).\end{aligned}$$

This bound is nearly as good as the best possible bound achievable even without privacy (i.e., the RWM bound) — the regret bound is larger only by a factor of $\Omega(\frac{\sqrt{\ln(k) \ln(1/\delta)}}{\varepsilon})$. (We note that by using a different algorithm with a more careful analysis, we can remove this extra factor of $\sqrt{\ln k}$). Since we are in fact using the same algorithm, efficiency is of course preserved as well. Here we have a powerful example in machine learning where privacy is nearly “free.” Notably, just as with the non-private algorithm, our utility bound only gets better the longer we run the algorithm, while keeping the privacy guarantee the same.³

³Of course, we have to set the update parameter appropriately, just as we have to do with the non-private algorithm. This is easy when the number of rounds T is known ahead of time, but can also be done adaptively when the number of rounds is not known ahead of time.

11.3 Empirical risk minimization

In this section, we apply the randomized weighted majority algorithm discussed in the previous section to a special case of the problem of empirical risk minimization to learn a linear function. Rather than assuming an adversarial model, we will assume that *examples* are drawn from some known distribution, and we wish to learn a classifier from some finite number of samples from this distribution so that our loss will be low on *new* samples drawn from the same distribution.

Suppose that we have a distribution \mathcal{D} over *examples* $x \in [-1, 1]^d$, and for each such vector $x \in [-1, 1]^d$, and for each vector $\theta \in [0, 1]^d$ with $\|\theta\|_1 = 1$, we define the loss of θ on example x to be $\text{Loss}(\theta, x) = \langle \theta, x \rangle$. We wish to find a vector θ^* to minimize the *expected* loss over examples drawn from \mathcal{D} :

$$\theta^* = \arg \min_{\theta \in [0, 1]^d : \|\theta\|_1=1} \mathbb{E}_{x \sim \mathcal{D}} [\langle \theta, x \rangle].$$

This problem can be used to model the task of finding a low error linear classifier. Typically our only access to the distribution \mathcal{D} is through some collection of examples $S \subset [-1, 1]^d$ drawn i.i.d. from \mathcal{D} , which serves as the input to our learning algorithm. We will here think of this sample S as our private database, and will be interested in how well we can privately approximate the error of θ^* as a function of $|S|$ (the *sample complexity* of the learning algorithm).

Our approach will be to reduce the problem to that of learning with expert advice, and apply the private version of the randomized weighted majority algorithm as discussed in the last section:

1. The *experts* will be the d standard basis vectors $\{e_1, \dots, e_d\}$, where $e_i = (0, \dots, 0, \underbrace{1}_i, 0, \dots, 0)$.
2. Given an example $x \in [-1, 1]^d$, we define a loss vector $\ell(x) \in [-1, 1]^d$ by setting $\ell(x)_i = \langle e_i, x \rangle$ for each $i \in \{1, \dots, d\}$. In other words, we simply set $\ell(x)_i = x_i$.
3. At time t , we choose a loss function ℓ^t by sampling $x \sim \mathcal{D}$ and setting $\ell^t = \ell(x)$.

Note that if we have a sample S from \mathcal{D} of size $|S| = T$, then we can run the RWM algorithm on the sequence of losses as described above for a total of T rounds. This will produce a sequence of outputs a_1, \dots, a_T , and we will define our final classifier to be $\theta^T \equiv \frac{1}{T} \sum_{i=1}^T a_i$. (Recall that each a_i is a standard basis vector $a_i \in \{e_1, \dots, e_d\}$, and so we have $\|\theta^T\|_1 = 1$).

We summarize the algorithm below:

Algorithm 17 An algorithm for learning linear functions. It takes as input a private database of examples $S \subset [-1, 1]^d$, $S = (x_1, \dots, x_T)$, and privacy parameters ε and δ .

LinearLearner(S, ε, δ):

```

Let  $\eta \leftarrow \frac{\varepsilon}{\sqrt{32T \ln(1/\delta)}}$ 
for  $t = 1$  to  $T = |S|$  do
    Choose vector  $a_t = e_i$  with probability proportional to
     $\exp(-\eta \sum_{j=1}^{t-1} \ell_i^j)$ 
    Let loss vector  $\ell^t = (\langle e_1, x_t \rangle, \langle e_2, x_t \rangle, \dots, \langle e_d, x_t \rangle)$ .
end for
Output  $\theta^T = \frac{1}{T} \sum_{t=1}^T a_t$ .
```

We have already seen that LinearLearner is private, since it is simply an instantiation of the randomized weighted majority algorithm with the correct update parameter η :

Theorem 11.5. $\text{LinearLearner}(S, \varepsilon, \delta)$ is (ε, δ) -differentially private.

It remains to analyze the classification accuracy of LinearLearner, which amounts to considering the regret bound of the private RWM algorithm.

Theorem 11.6. If S consists of T i.i.d. samples $x \sim \mathcal{D}$, then with probability at least $1 - \beta$, LinearLearner outputs a vector θ^T such that:

$$\mathbb{E}_{x \sim \mathcal{D}}[\langle \theta^T, x \rangle] \leq \min_{\theta^*} \mathbb{E}_{x \sim \mathcal{D}}[\langle \theta^*, x \rangle] + O\left(\frac{\sqrt{\ln(1/\delta)} \ln(d/\beta)}{\varepsilon \sqrt{T}}\right),$$

where d is the number of experts.

Proof. By Theorem 11.4, we have the following guarantee with probability at least $1 - \beta/2$:

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T \langle a_t, x_t \rangle &\leq \min_{i \in \{1, \dots, d\}} \left\langle e_i, \frac{1}{T} \sum_{t=1}^T x_t \right\rangle + O\left(\frac{\sqrt{\ln(1/\delta)} \ln(d/\beta)}{\varepsilon \sqrt{T}}\right) \\ &= \min_{\theta^* \in [0,1]^d: \|\theta^*\|_1=1} \left\langle \theta^*, \frac{1}{T} \sum_{t=1}^T x_t \right\rangle + O\left(\frac{\sqrt{\ln(1/\delta)} \ln(d/\beta)}{\varepsilon \sqrt{T}}\right). \end{aligned}$$

In the first equality, we use the fact that the minimum of a linear function over the simplex is achieved at a vertex of the simplex. Noting that each $x_t \sim \mathcal{D}$ independently and that each $\langle x_t, e_i \rangle$ is bounded in $[-1, 1]$, we can apply Azuma's inequality twice to bound the two quantities with probability at least $1 - \beta/2$:

$$\begin{aligned} &\left| \frac{1}{T} \sum_{t=1}^T \langle a_t, x_t \rangle - \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{x \sim \mathcal{D}} \langle a_t, x \rangle \right| \\ &= \left| \frac{1}{T} \sum_{t=1}^T \langle a_t, x_t \rangle - \mathbb{E}_{x \sim \mathcal{D}} \langle \theta^T, x \rangle \right| \leq O\left(\sqrt{\frac{\ln(1/\beta)}{T}}\right) \end{aligned}$$

and

$$\max_{i \in \{1, \dots, d\}} \left| \left\langle e_i, \frac{1}{T} \sum_{t=1}^T x_t \right\rangle - \mathbb{E}_{x \sim \mathcal{D}} \langle e_i, x \rangle \right| \leq O\left(\sqrt{\frac{\ln(d/\beta)}{T}}\right).$$

Hence we also have:

$$\max_{\theta^* \in [0,1]^d: \|\theta^*\|_1=1} \left| \left\langle \theta^*, \frac{1}{T} \sum_{t=1}^T x_t \right\rangle - \mathbb{E}_{x \sim \mathcal{D}} \langle \theta^*, x \rangle \right| \leq O\left(\sqrt{\frac{\ln d/\beta}{T}}\right).$$

Combining these inequalities gives us our final result about the output of the algorithm θ^T :

$$\mathbb{E}_{x \sim \mathcal{D}} \langle \theta^T, x \rangle \leq \min_{\theta^* \in [0,1]^d: \|\theta^*\|_1=1} \mathbb{E}_{x \sim \mathcal{D}} \langle \theta^*, x \rangle + O\left(\frac{\sqrt{\ln(1/\delta)} \ln(d/\beta)}{\varepsilon \sqrt{T}}\right).$$

□

11.4 Bibliographical notes

The PAC model of machine learning was introduced by Valiant in 1984 [83], and the SQ model was introduced by Kearns [53]. The randomized weighted majority algorithm is originally due to Littlestone and Warmuth [57], and has been studied in many forms. See Blum and Mansour [9] or Arora et al. [1] for a survey. The regret bound that we use for the randomized weighted majority algorithm is given in [1].

Machine learning was one of the first topics studied in differential privacy, beginning with the work of Blum et al. [7], who showed that algorithms that operate in the SQ-learning framework could be converted into privacy preserving algorithms. The sample complexity of differentially private learning was first considered by Kasiviswanathan, Lee, Nissim, Raskhodnikova, and Smith, “What can we Learn Privately?” [52], which characterize the sample complexity of private learning up to polynomial factors. For more refined analysis of the sample complexity of private learning, see [3, 4, 12, 19].

There is also extensive work on efficient machine learning algorithms, including the well known frameworks of SVMs and empirical risk minimizers [13, 55, 76]. Spectral learning techniques, including PCA and low rank matrix approximation have also been studied [7, 14, 33, 42, 43, 51].

Private learning from expert advice was first considered by Dwork et al. [26]. The fact that the randomized weighted majority algorithm is privacy preserving without modification (when the update parameter is set appropriately) is folklore (following from advanced composition [32]) and has been widely used; for example, in [48]. For a more general study of private online learning, see [50], and for a more general study of empirical risk minimization, see [50, 13].

12

Additional Models

So far, we have made some implicit assumptions about the model of private data analysis. For example, we have assumed that there is some trusted curator who has direct access to the private dataset, and we have assumed that the adversary only has access to the output of the algorithm, not to any of its internal state during its execution. But what if this is not the case? What if we trust no one to look at our data, even to perform the privacy preserving data analysis? What if some hacker might gain access to the internal state of the private algorithm while it is running? In this section, we relax some of our previously held assumptions and consider these questions.

In this section we describe some additional computational models that have received attention in the literature.

- The *local model* is a generalization of randomized response (see Section 2), and is motivated by situations in which individuals do not trust the curator with their data. While this lack of trust can be addressed using secure multiparty computation to simulate the role played by the trusted curator, there are also some techniques that do not require cryptography.

The next two models consider streams of *events*, each of which may be associated with an individual. For example, an event may be a search by a particular person on an arbitrary term. In a given event stream, the (potentially many) events associated with a given individual can be arbitrarily interleaved with events associated with other individuals.

- In *pan-privacy* the curator is trusted, but may be subject to compulsory non-private data release, for example, because of a subpoena, or because the entity holding the information is purchased by another, possibly less trustworthy, entity. Thus, in pan-privacy the *internal state* of the algorithm is also differentially private, as is the joint distribution of the internal state and the outputs.
- The *continual observation* model addresses the question of maintaining privacy when the goal is to continually monitor and report statistics about events, such as purchases of over-the-counter medications that might be indicative of an impending epidemic. Some work addresses pan-privacy under continual observation.

12.1 The local model

So far, we have considered a *centralized* model of data privacy, in which there exists a database administrator who has direct access to the private data. What if there is instead no trusted database administrator? Even if there is a suitable trusted party, there are many reasons not to want private data aggregated by some third party. The very existence of an aggregate database of private information raises the possibility that at some *future time*, it will come into the hands of an untrusted party, either maliciously (via data theft), or as a natural result of organizational succession. A superior model — from the perspective of the owners of private data — would be a local model, in which agents could (randomly) answer questions in a differentially private manner about their own data, without ever sharing it with anyone else. In the context of predicate queries, this seems to severely limit the expressivity of a private mechanism's interaction with the data: The mechanism can ask each user whether or not her data satisfies a given predicate, and

the user may flip a coin, answering truthfully only with slightly higher probability than answering falsely. In this model what is possible?

The local privacy model was first introduced in the context of learning. The local privacy model formalizes randomized response: there is no central database of private data. Instead, each individual maintains possession of their own data element (a database of size 1), and answers questions about it only in a differentially private manner. Formally, the database $x \in \mathbb{N}^{|\mathcal{X}|}$ is a collection of n elements from some domain \mathcal{X} , and each $x_i \in x$ is held by an individual.

Definition 12.1 (Local Randomizer). An ε -local randomizer $R : \mathcal{X} \rightarrow W$ is an ε -differentially private algorithm that takes as input a database of size $n = 1$.

In the local privacy model, algorithms may interact with the database only through a local randomizer oracle:

Definition 12.2 (LR Oracle). An LR oracle $LR_D(\cdot, \cdot)$ takes as input an index $i \in [n]$ and an ε -local randomizer R and outputs a random value $w \in W$ chosen according to the distribution $R(x_i)$, where $x_i \in D$ is the element held by the i th individual in the database.

Definition 12.3 ((Local Algorithm)). An algorithm is ε -local if it accesses the database D via the oracle LR_D , with the following restriction: If $LR_D(i, R_1), \dots, LR_D(i, R_k)$ are the algorithm's invocations of LR_D on index i , where each R_j is an ε_j -local randomizer, then $\varepsilon_1 + \dots + \varepsilon_k \leq \varepsilon$.

Because differential privacy is composable, it is easy to see that ε -local algorithms are ε -differentially private.

Observation 12.1. ε -local algorithms are ε -differentially private.

That is to say, an ε -local algorithm interacts with the data using only a sequence of ε -differentially private algorithms, each of which computes only on a database of size 1. Because nobody other than its owner ever touches any piece of private data, the local setting is far more secure: it does not require a trusted party, and there is no central party who might be subject to hacking. Because even the algorithm

never sees private data, the internal state of the algorithm is always differentially private as well (i.e., local privacy implies pan privacy, described in the next section). A natural question is how restrictive the local privacy model is. In this section, we merely informally discuss results. The interested reader can follow the bibliographic references at the end of this section for more information. We note that an alternative name for the local privacy model is the *fully distributed* model.

We recall the definition of the statistical query (SQ) model, introduced in Section 11. Roughly speaking, given a database x of size n , the statistical query model allows an algorithm to access this database by making a polynomial (in n) number of noisy linear queries to the database, where the error in the query answers is some inverse polynomial in n . Formally:

Definition 12.4. A *statistical query* is some function $\phi : \mathcal{X} \times \{0, 1\} \rightarrow [0, 1]$. A *statistical query oracle* for a distribution over labeled examples \mathcal{D} with tolerance τ is an oracle $\mathcal{O}_{\mathcal{D}}^{\tau}$ such that for every statistical query ϕ :

$$\left| \mathcal{O}_{\mathcal{D}}^{\tau}(\phi) - \mathbb{E}_{(x,y) \sim \mathcal{D}}[\phi(x, y)] \right| \leq \tau$$

In other words, an SQ oracle takes as input a statistical query ϕ , and outputs some value that is guaranteed to be within $\pm \tau$ of the expected value of ϕ on examples drawn from \mathcal{D} .

Definition 12.5. An algorithm A is said to SQ-learn a class of functions C if for every $\alpha, \beta > 0$ there exists an $m = \text{poly}(d, 1/\alpha, \log(1/\beta))$ such that A makes at most m queries of tolerance $\tau = 1/m$ to $\mathcal{O}_{\mathcal{D}}^{\tau}$, and with probability $1 - \beta$, outputs a hypothesis $f \in C$ such that:

$$\text{err}(f, \mathcal{D}) \leq \min_{f^* \in C} \text{err}(f^*, \mathcal{D}) + \alpha$$

More generally, we can talk about an algorithm (for performing any computation) as operating in the SQ model if it accesses the data only through an SQ oracle:

Definition 12.6. An algorithm A is said to operate in the SQ model if there exists an m such that A makes at most m queries of tolerance $\tau = 1/m$ to $\mathcal{O}_{\mathcal{D}}^{\tau}$, and does not have any other access to the database. A is efficient if m is polynomial in the size of the database, D .

It turns out that up to polynomial factors in the size of the database and in the number of queries, any algorithm that can be implemented in the SQ model can be implemented and analyzed for privacy in the local privacy model, and vice versa. We note that there is a distinction between an algorithm being implemented in the SQ model, and its privacy analysis being carried out in the local model: almost all of the algorithms that we have presented in the end access the data using noisy linear queries, and so can be thought of as acting in the SQ model. However, their privacy guarantees are analyzed in the centralized model of data privacy (i.e., because of some “global” part of the analysis, as in the sparse vector algorithm).

In the following summary, we will also recall the definition of PAC learning, also introduced in Section 11:

Definition 12.7. An algorithm A is said to PAC-learn a class of functions C if for every $\alpha, \beta > 0$, there exists an $m = \text{poly}(d, 1/\alpha, \log(1/\beta))$ such that for every distribution \mathcal{D} over labeled examples, A takes as input m labeled examples drawn from \mathcal{D} and outputs a hypothesis $f \in C$ such that with probability $1 - \beta$:

$$\text{err}(f, \mathcal{D}) \leq \min_{f^* \in C} \text{err}(f^*, \mathcal{D}) + \alpha$$

If $\min_{f^* \in C} \text{err}(f^*, \mathcal{D}) = 0$, the learner is said to operate in the *realizable* setting (i.e., there exists some function in the class which perfectly labels the data). Otherwise, the learner is said to operate in the *agnostic* setting. If A also has run time that is polynomial in $d, 1/\alpha$, and $\log(1/\beta)$, then the learner is said to be *efficient*. If there is an algorithm which PAC-learns C , then C is said to be PAC-learnable. Note that the main distinction between an SQ learning algorithm and a PAC learning algorithm, is that the PAC learning algorithm gets direct access to the database of examples, whereas the SQ learning algorithm only has access to the data through a noisy SQ oracle.

What follows is some of our understanding of the limitations of the SQ model and problems which separate it from the centralized model of data privacy.

1. A single sensitivity-1 query can be answered to error $O(1)$ in the centralized model of data privacy using the Laplace mechanism, but requires error $\Theta(\sqrt{n})$ in the local data privacy model.
2. The set of function classes that we can (properly) learn in the local privacy model is exactly the set of function classes that we can properly learn in the SQ model (up to polynomial factors in the database size and query complexity of the algorithm). In contrast, the set of things we can (properly or agnostically) learn in the centralized model corresponds to the set of things we can learn in the PAC model. SQ learning is strictly weaker, but this is not a huge handicap, since parity functions are essentially the only interesting class that is PAC learnable but not SQ learnable. We remark that we refer explicitly to proper learning here (meaning the setting in which there is some function in the class which perfectly labels the data). In the PAC model there is no information theoretic difference between proper and agnostic learning, but in the SQ model the difference is large: see the next point.
3. The set of queries that we can release in the local privacy model are exactly those queries that we can agnostically learn in the SQ model. In contrast, the set of things we can release in the centralized model corresponds to the set of things we can agnostically learn in the PAC model. This is a much bigger handicap — even conjunctions (i.e., marginals) are not agnostically learnable in the SQ model. This follows from the information theoretic reduction from agnostic learning (i.e., *distinguishing*) to query release that we saw in Section 5 using the iterative construction mechanism.

We note that if we are only concerned about computationally bounded adversaries, then in principle distributed agents can use *secure multiparty computation* to simulate private algorithms in the centralized setting. While this does not actually give a differential privacy guarantee, the result of such simulations will be indistinguishable from the result of differentially private computations, from the point of view of a computationally bounded adversary. However, general secure multiparty computation protocols typically require huge amounts of message passing (and hence sometimes have unreasonably large run times),

whereas algorithms in the local privacy model tend to be extremely simple.

12.2 Pan-private streaming model

The goal of a pan-private algorithm is to remain differentially private even against an adversary that can, on rare occasions, observe the algorithm's internal state. Intrusions can occur for many reasons, including hacking, subpoena, or *mission creep*, when data collected for one purpose are used for a different purpose ("Think of the children!"). Pan-private streaming algorithms provide protection against all of these. Note that ordinary streaming algorithms do *not* necessarily provide privacy against intrusions, as even a low-memory streaming algorithm can hold a small number of data items in memory, which would be completely exposed in an intrusion. On the technical side, intrusions can be *known* to the curator (subpoena) or unknown (hacking). These can have very different effects, as a curator aware of an intrusion can take protective measures, such as re-randomizing certain variables.

12.2.1 Definitions

We assume a data stream of unbounded length composed of elements in a universe \mathcal{X} . It may be helpful to keep in mind as motivation data analysis on a query stream, in which queries are accompanied by the IP address of the issuer. For now, we ignore the query text itself; the universe \mathcal{X} is the universe of potential IP addresses. Thus, intuitively, *user-level* privacy protects the presence or absence of an IP address in the stream, independent of the number of times it arises, should it actually be present at all. In contrast, *event-level* privacy merely protects the privacy of individual accesses. For now, we focus on user-level privacy.

As usual in differentially private algorithms, the adversary can have arbitrary control of the input stream, and may have arbitrary auxiliary knowledge obtained from other sources. It can also have arbitrary computational power.

We assume the algorithm runs until it receives a special signal, at which point it produces (observable) outputs. The algorithm may optionally continue to run and produce additional outputs later, again in response to a special signal. Since outputs are observable we do not provide privacy for the special signals.

A streaming algorithm experiences a sequence of internal states, and produces a (possibly unbounded) sequence of outputs. Let I denote the set of possible internal states of the algorithm, and σ the set of possible output sequences. We assume that the adversary can only observe internal states and the output sequence; it cannot see the data in the stream (although it may have auxiliary knowledge about some of these data) and it has no access to the *length* of the input sequence.

Definition 12.8 (\mathcal{X} -Adjacent Data Streams). We think of data streams as being of unbounded length; *prefixes* have finite length. Data streams S and S' are \mathcal{X} -adjacent if they differ only in the presence or absence of *all* occurrences of a single element $u \in \mathcal{X}$. We define \mathcal{X} -adjacency for stream prefixes analogously.

User-Level Pan-Privacy. An algorithm **Alg** mapping data stream prefixes to the range $I \times \sigma$, is *pan-private against a single intrusion* if for all sets $I' \subseteq I$ of internal states and $\sigma' \subseteq \sigma$ of output sequences, and for all pairs of adjacent data stream prefixes S, S'

$$\Pr[\mathbf{Alg}(S) \in (I', \sigma')] \leq e^\varepsilon \Pr[\mathbf{Alg}(S') \in (I', \sigma')],$$

where the probability spaces are over the coin flips of the algorithm **Alg**.

This definition speaks only of a single intrusion. For multiple intrusions we must consider interleavings of observations of internal states and outputs.

The relaxation to *event-level privacy* is obtained by modifying the notion of adjacency so that, roughly speaking, two streams are event-adjacent if they differ in a single instance of a single element in \mathcal{X} ; that is, one instance of one element is deleted/added. Clearly, event-level privacy is a much weaker guarantee than user-level privacy.

Remark 12.1. If we assume the existence of a very small amount of secret storage, not visible to the adversary, then many problems for which we have been unable to obtain pan-private solutions have (non-pan-) private streaming solutions. However, the *amount* of secret storage is not so important as its *existence*, since secret storage is vulnerable to the social pressures against which pan-privacy seeks to protect the data (and the curator).

Pan-Private Density Estimation. Quite surprisingly, pan-privacy can be achieved even for *user-level* privacy of many common streaming computations. As an example, consider the problem of *density estimation*: given a universe \mathcal{X} of data elements and a stream σ , the goal is to estimate the fraction of \mathcal{X} that actually appears in the stream. For example, the universe consists of all teenagers in a given community (represented by IP addresses), and the goal is to understand what fraction visit the Planned Parenthood website.

Standard low-memory streaming solutions for density estimation involve recording the results of deterministic computations of at least some input items, an approach that is inherently not pan-private. Here is a simple, albeit high-memory, solution inspired by randomized response. The algorithm maintains a bit b_a for each IP address a (which may appear any number of times in the stream), initialized uniformly at random. The stream is processed one element at a time. On input a the algorithm flips a bit biased to 1; that is, the biased bit will take value 0 with probability $1/2 - \varepsilon$, and value 1 with probability $1/2 + \varepsilon$. The algorithm follows this procedure independent of the number of times IP address a appears in the data stream. This algorithm is $(\varepsilon, 0)$ -differentially private. As with randomized response, we can estimate the fraction of “real” 1’s by $z = 2(y - |\mathcal{X}|/2)/|\mathcal{X}|$, where y is the actual number of 1’s in the table after the stream is processed. To ensure pan-privacy, the algorithm publishes a noisy version of z . As with randomized response, the error will be on the order of $1/\sqrt{|\mathcal{X}|}$, yielding meaningful results when the density is high.

Other problems enjoying user-level pan-private algorithms include:

- Estimating, for any t , the fraction of elements appearing exactly t times;

- Estimating the *t-cropped mean*: roughly, the average, over all elements, of the minimum of t and the number of occurrences of the element in the data stream;
- Estimating the fraction of k -heavy hitters (elements of \mathcal{X} that appear at least k times in the data stream).

Variants of these problems can also be defined for *fully dynamic* data, in which counts can be decremented as well as incremented. For example, density estimation (what fraction appeared in the stream?) becomes “How many (or what fraction) of elements have a (net) count equal to zero?” These, too, can be solved with user-level pan-privacy, using differentially private variations of *sketching* techniques from the streaming literature.

12.3 Continual observation

Many applications of data analysis involve repeated computations, either because the entire goal is one of monitoring of, for example, traffic conditions, search trends, or incidence of influenza. In such applications the system is required to continually produce outputs. We therefore need techniques for achieving *differential privacy under continual observation*.

As usual, differential privacy will require having essentially the same distribution on outputs for each pair of adjacent databases, but how should we define adjacency in this setting? Let us consider two example scenarios.

Suppose the goal is to monitor public health by analyzing statistics from an H1N1 self-assessment Web site.¹ Individuals can interact with the site to learn whether symptoms they are experiencing may be indicative of the H1N1 flu. The user fills in some demographic data (age, zipcode, sex), and responds to queries about his symptoms (fever over 100.4°F?, sore throat?, duration of symptoms?). We would expect a given individual to interact very few times with the H1N1 self-assessment site (say, if we restrict our attention to a six-month

¹<https://h1n1.cloudapp.net> provided such a service during the winter of 2010; user-supplied data were stored for analysis with the user’s consent.

period). For simplicity, let us say this is just once. In such a setting, it is sufficient to ensure *event-level* privacy, in which the privacy goal is to hide the presence or absence of a single event (interaction of one user with the self-assessment site).

Suppose again that the goal is to monitor public health, this time by analyzing search terms submitted to a medical search engine. Here it may no longer be safe to assume an individual has few interactions with the Web site, even if we restrict attention to a relatively short period of time. In this case we would want *user-level* privacy, ensuring that the entire set of a user’s search terms is protected simultaneously.

We think of continual observation algorithms as taking steps at discrete time intervals; at each step the algorithm receives an input, computes, and produces output. We model the data as arriving in a stream, at most one data element in each time interval. To capture the fact that, in real life, there are periods of time in which nothing happens, null events are modeled by a special symbol in the data stream. Thus, the intuitive notion of “ t time periods” corresponds to processing a sequence of t elements in the stream.

For example, the motivation behind the counter primitive below is to count the number of times that something has occurred since the algorithm was started (the counter is very general; we don’t specify *a priori* what it is counting). This is modeled by an input stream over $\{0, 1\}$. Here, “0” means “nothing happened,” “1” means the event of interest occurred, and for $t = 1, 2, \dots, T$ the algorithm outputs an approximation to the number of 1s seen in the length t prefix of the stream.

There are three natural options:

1. Use randomized response for each time period and add this randomized value to the counter;
2. Add noise distributed according to $\text{Lap}(1/\varepsilon)$ to the true value for each time step and add this perturbed value to the counter;
3. Compute the true count at each time step, add noise distributed according to $\text{Lap}(T/\varepsilon)$ to the count, and release this noisy count.

All of these options result in noise on the order of at least $\Omega(\sqrt{T}/\varepsilon)$. The hope is to do much better by exploiting structure of the query set.

Let \mathcal{X} be the universe of possible input symbols. Let S and S' be stream prefixes (i.e., finite streams) of symbols drawn from \mathcal{X} . Then $\text{Adj}(S, S')$ (“ S is adjacent to S' ”) if and only if there exist $a, b \in \mathcal{X}$ so that if we change some of the instances of a in S to instances of b , then we get S' . More formally, $\text{Adj}(S, S')$ iff $\exists a, b \in \mathcal{X}$ and $\exists R \subseteq [|S|]$, such that $S|_{R:a \rightarrow b} = S'$. Here, R is a set of indices in the stream prefix S , and $S|_{R:a \rightarrow b}$ is the result of replacing all the occurrences of a at these indices with b . Note that adjacent prefixes are always of the same length.

To capture event-level privacy, we restrict the definition of adjacency to the case $|R| \leq 1$. To capture user-level privacy we do not constrain the size of R in the definition of adjacency.

As noted above, one option is to publish a noisy count at each time step; the count published at time t reflects the approximate number of 1s in the length t prefix of the stream. The privacy challenge is that early items in the stream are subject to nearly T statistics, so for $(\varepsilon, 0)$ -differential privacy we would be adding noise scaled to T/ε , which is unacceptable. In addition, since the 1s are the “interesting” elements of the stream, we would like that the distortion be scaled to the number of 1s seen in the stream, rather than to the length of the stream. This rules out applying randomized response to each item in the stream independently.

The algorithm below follows a classical approach for converting static algorithms to dynamic algorithms.

Assume T is a power of 2. The intervals are the natural ones corresponding to the labels on a complete binary tree with T leaves, where the leaves are labeled, from left to right, with the intervals $[0, 0], [1, 1], \dots, [T - 1, T - 1]$ and each parent is labeled with the interval that is the union of the intervals labeling its children. The idea is to compute and release a noisy count for each label $[s, t]$; that is, the released value corresponding to the label $[s, t]$ is a noisy count of the number of 1s in positions $s, s + 1, \dots, t$ of the input stream. To learn the approximate cumulative count at time $t \in [0, T - 1]$ the analyst uses the binary representation of t to determine a set of at most $\log_2 T$

Counter (T, ε)

Initialization. Initialize $\xi = \log_2 T / \varepsilon$, and sample Counter $\sim \text{Lap}(\xi)$.

Intervals. For $i \in \{1, \dots, \log T\}$, associate with each string $s \in \{0, 1\}^i$ the time interval S of $2^{\log T - i}$ time periods $\{s \circ 0^{\log T - i}, \dots, s \circ 1^{\log T - i}\}$. The interval *begins in time* $s \circ 0^{\log T - i}$ and *ends in time* $s \circ 1^{\log T - i}$.

Processing. In time period $t \in \{0, 1, \dots, T - 1\}$, let $x_t \in \{0, 1\}$ be the t -th input bit:

1. For every interval I beginning at time t , initialize c_I to an independent random draw: $c_I \leftarrow \text{Lap}((\log_2 T) / \varepsilon)$;
2. For every interval I containing t , add x_t to c_I : $c_I \leftarrow c_I + x_t$;
3. For every interval I that ends in time t , output c_I .

Figure 12.1: Event-level private counter algorithm (not pan-private).

disjoint intervals whose union is $[0, t]$, and computes the sum of the corresponding released noisy counts.² See Figure 12.1.

Each stream position $t \in [0, T - 1]$ appears in at most $1 + \log_2 T$ intervals (because the height of the tree is $\log_2 T$), and so each element in the stream affects at most $1 + \log_2 T$ released noisy counts. Thus, adding noise to each interval count distributed according to $\text{Lap}((1 + \log_2 T) / \varepsilon)$ ensures $(\varepsilon, 0)$ -differential privacy. As for accuracy, since the binary representation of any index $t \in [0, T - 1]$ yields a disjoint set of at most $\log_2 T$ intervals whose union is $[0, t]$ we can apply Lemma 12.2 below to conclude that the expected error is tightly concentrated around $(\log_2 T)^{3/2}$. The maximum expected error, over all times t , is on the order of $(\log_2 T)^{5/3}$.

Lemma 12.2. Let Y_1, \dots, Y_k be independent variables with distribution $\text{Lap}(b_i)$. Let $Y = \sum_i Y_i$ and $b_{\max} = \max_i b_i$. Let $\nu \geq \sqrt{\sum_i (b_i)^2}$, and $0 < \lambda < \frac{2\sqrt{2}\nu^2}{b_{\max}}$. Then

$$\Pr[Y > \lambda] \leq \exp\left(-\frac{\lambda^2}{8\nu^2}\right).$$

²This algorithm can be optimized slightly (for example, we never use the count corresponding to the root, eliminating one level from the tree), and it can be modified to handle the case in which T is not a power of 2 and, more interestingly, when T is not known *a priori*.

Proof. The moment generating function of Y_i is $\mathbb{E}[\exp(hY_i)] = 1/(1 - h^2 b_i^2)$, where $|h| < 1/b_i$. Using the inequality $(1 - x)^{-1} \leq 1 + 2x \leq \exp(2x)$ for $0 \leq x < 1/2$, we have $\mathbb{E}[\exp(hY_i)] \leq \exp(2h^2 b_i^2)$, if $|h| < 1/2b_i$. We now calculate, for $0 < h < 1/\sqrt{2}b_{\max}$:

$$\begin{aligned}\Pr[Y > \lambda] &= \Pr[\exp(hY) > \exp(h\lambda)] \\ &\leq \exp(-h\lambda)\mathbb{E}[\exp(hY)] \\ &= \exp(-h\lambda)\prod_i \mathbb{E}[\exp(hY_i)] \\ &\leq \exp(-h\lambda + 2h^2\nu^2).\end{aligned}$$

By assumption, $0 < \lambda < \frac{2\sqrt{2}\nu^2}{b_{\max}}$. We complete the proof by setting $h = \lambda/4\nu^2 < 1/\sqrt{2}b_{\max}$. \square

Corollary 12.3. Let $Y, \nu, \{b_i\}_i, b_{\max}$ be as in Lemma 12.2. For $\delta \in (0, 1)$ and $\nu > \max\{\sqrt{\sum_i b_i^2}, b_{\max}\sqrt{\ln(2/\delta)}\}$, we have that $\Pr[|Y| > \nu\sqrt{8\ln(2/\delta)}] \leq \delta$.

In our case, all the b_i 's are the same (e.g., $b = (\log_2 T)/\varepsilon$). Taking $\nu = \sqrt{k}b$ we have the following corollary:

Corollary 12.4. For all $\lambda < \alpha(\sqrt{k}b) < 2\sqrt{2}kb = 2\sqrt{2k}\nu$,

$$\Pr[Y > \lambda] \leq e^{-\alpha^2/8}.$$

Note that we have taken the unusual step of adding noise to the count *before* counting, rather than after. In terms of the outputs it makes no difference (addition is commutative). However, it has an interesting effect on the algorithm's internal states: they are differentially private! That is, suppose the intrusion occurs at time t , and consider any $i \in [0, t]$. Since there are at most $\log_2 T$ intervals containing step i (in the algorithm we abolished the interval corresponding to the root), x_i affects at most $\log_2 T$ of the noisy counts, and so x_i is protected against the intrusion for exactly the same reason that it is protected in the algorithm's outputs. Nevertheless, the algorithm in Figure 12.1 is *not* pan-private even against a single intrusion. This is because, while its internal state and its outputs are each independently differentially private, the joint distribution does not ensure ε -differential privacy. To

see why this is so, consider an intruder that sees the internal state at time t and knows the entire data stream except x_{t+1} , and let $I = [a, b]$ be an interval containing both t and $t + 1$. Since the adversary knows $x_{[0,t]}$, it can subtract from c_I the contribution from the stream occurring up through time t (that is, it subtracts off from the observed c_I at time t the values x_a, x_{a+1}, \dots, x_t , all of which it knows). From this the intruder learns the value of the Laplace draw to which c_I was initialized. When c_I is published at the end of step b , the adversary subtracts from the published value this initial draw, together with the contributions of all elements in $x_{[a,b]}$ except x_{t+1} , which it does not know. What remains is the unknown x_{t+1} .

12.3.1 Pan-private counting

Although the algorithm in Figure 12.1 is easily modified to ensure *event-level pan-privacy against a single intrusion*, we give a different algorithm here in order to introduce a powerful *bijection* technique which has proved useful in other applications. This algorithm maintains in its internal state a single noisy counter, or accumulator, as well as noise values for each interval. The output at any given time period t is the sum of the accumulator and the noise values for the intervals containing t . When an interval I ends, its associated noise value, η_I , is erased from memory.

Theorem 12.5. The counter algorithm of Figure 12.2, when run with parameters T, ε , and suffering at most one intrusion, yields an $(\varepsilon, 0)$ -pan-private counter that, with probability at least $1 - \beta$ has maximum error, over its T outputs, of $O(\log(1/\beta) \cdot \log^{2.5} T/\varepsilon)$. We note also that in every round *individually* (rather than in all rounds simultaneously), with all but β probability, the error has magnitude at most $O(\log(1/\beta) \cdot \log^{1.5} T/\varepsilon)$.

Proof. The proof of accuracy is the same as that for the algorithm in Figure 12.1, relying on Corollary 12.4. We focus here on the proof of pan-privacy.

During an intrusion between atomic steps t^* and $t^* + 1$, that is, immediately following the processing of element t^* in the input stream

Pan-Private Counter (T, ε)

Initialization. Initialize $\xi = (1 + \log T)/\varepsilon$, and sample Counter $\sim \text{Lap}(\xi)$.

Intervals. For $i \in \{1, \dots, \log T\}$, associate with each string $s \in \{0, 1\}^i$ the time interval S of $2^{\log T - i}$ time periods $\{s \circ 0^{\log T - i}, \dots, s \circ 1^{\log T - i}\}$. The interval *begins in time* $s \circ 0^{\log T - i}$ and *ends in time* $s \circ 1^{\log T - i}$.

Processing. In time period $t \in \{0, 1, \dots, T - 1\}$, let $x_t \in \{0, 1\}$ be the t -th input bit:

1. Counter \leftarrow Counter + x_t ;
2. For every interval I which begins in time t , sample noise $\eta_I \sim \text{Lap}(\xi)$;
3. Let $I_1, \dots, I_{\log T}$ be the $\log T$ intervals that contain t . Output Counter + $\sum_{i=1}^{\log T} \eta_{I_i}$.
4. For every interval I that ends in time t , erase η_I .

Figure 12.2: Event-level pan-private counter algorithm.

(recall that we begin numbering the elements with 0), the view of the adversary consists of (1) the noisy cumulative count (in the variable “count”), (2) the interval noise values η_S in memory when the intrusion occurs, and (3) the complete sequence of all of the algorithm’s outputs in rounds $0, 1, \dots, t$. Consider adjacent databases x and x' , which differ in time t , say, without loss of generality, $x_t = 1$ and $x'_t = 0$, and an intrusion immediately following time period $t^* \geq t$ (we will discuss the case $t^* < t$ below). We will describe a bijection between the vector of noise values used in executions on x and executions on x' , such that corresponding noise values induce identical adversary views on x and x' , and the probabilities of adjacent noise values differ only by an e^ε multiplicative factor. This implies ε -differential pan-privacy.

By assumption, the true count just after the time period $t^* \geq t$ is larger when the input is x than it is when the input is x' . Fix an arbitrary execution E_x when the input stream is x . This amounts to fixing the randomness of the algorithm, which in turn fixes the noise values generated. We will describe the corresponding execution $E_{x'}$ by describing how its noise values differ from those in E_x .

The program variable Counter was initialized with Laplace noise. By increasing this noise by 1 in $E_{x'}$ the value of Counter just after step t^* is identical in $E_{x'}$ and E_x . The noise variables in memory immediately following period t^* are independent of the input; these will be

unchanged in $E_{x'}$. We will make the sequence of outputs in $E_{x'}$ *identical* to those in E_x by changing a collection of $\log T$ interval noise values η_S that are *not* in memory when the adversary intrudes, so that the sum of *all* noise values in all rounds up through $t - 1$ is unchanged, but the sum from round t on is larger by 1 for database x' than for x . Since we *increased* the initialization noise for Counter, we now need to *decrease* the sum of interval noise values for periods $0, \dots, t - 1$ by 1, and leave unchanged the sum of interval noise values from period t .

To do this, we find a collection of disjoint intervals whose union is $\{0, \dots, t - 1\}$. There is always such a collection, and it is always of size at most $\log T$. We can construct it iteratively by, for i decreasing from $\lfloor \log(t - 1) \rfloor$ to 0, choosing the interval of size 2^i that is contained in $\{0, \dots, t - 1\}$ and is not contained in a previously chosen interval (if such an interval exists). Given this set of disjoint intervals, we notice also that they all end by time $t - 1 < t \leq t^*$, and so their noises are not in memory when the adversary intrudes (just following period t^*). In total (taking into account also changing the initial noise value for Counter), the complete view seen by the adversary is identical and the probabilities of the (collection of) noise values used for x and x' differ by at most an e^ε multiplicative factor.

Note that we assumed $t^* \geq t$. If $t^* < t$ then the initial noise added to Counter in $E_{x'}$ will be the same as in E_x , and we need to add 1 to the sum of interval noises in every time period from t through T (the sum of interval noises before time t remains unchanged). This is done as above, by finding a disjoint collection of at most $\log T$ intervals that exactly covers $\{t, \dots, T - 1\}$. The noise values for these intervals are not yet in memory when the intrusion occurs in time $t^* < t$, and the proof follows similarly. \square

12.3.2 A logarithmic (in T) lower bound

Given the upper bound of Theorem 12.5, where the error depends only poly-logarithmically on T , it is natural to ask whether *any* dependence is inherent. In this section we show that a logarithmic dependence on T is indeed inherent.

Theorem 12.6. Any differentially private event-level algorithm for counting over T rounds must have error $\Omega(\log T)$ (even with $\varepsilon = 1$).

Proof. Let $\varepsilon = 1$. Suppose for the sake of contradiction that there exists a differentially private event-level counter for streams of length T that guarantees that with probability at least $2/3$, its count at all time periods is accurate up to a maximum error of $(\log_2 T)/4$. Let $k = (\log_2 T)/4$. We construct a set S of T/k inputs as follows. Divide the T time periods into T/k consecutive phases, each of length k (except, possibly, the last one). For $i = 1, \dots, T/k$, the i -th input $x^i \in S$ has 0 input bits everywhere except during the i th phase. That is, $x^i = 0^{k \cdot i} \circ 1^k \circ 0^{k \cdot ((T/k) - (i+1))}$.

For $1 \leq i \leq T/k$, we say an output *matches* i if just before the i th phase the output is less than $k/2$ and at the end of the i th phase the output is at least $k/2$. By accuracy, on input x^i the output should match i with probability at least $2/3$. By ε differential privacy, this means that for every $i, j \in [T/k]$ such that $i \neq j$, the output on input x^i should match j with probability at least

$$\begin{aligned} e^{-2\varepsilon \cdot k} &= e^{-\varepsilon \log(T^{1/2})} \\ &= e^{-\log(T^{1/2})} = 1/\sqrt{T}. \end{aligned}$$

This is a contradiction, because the events that the output matches j are disjoint for different j , and yet the sum of their probabilities on input x^i exceeds 1. \square

12.4 Average case error for query release

In Sections 4 and 5, we considered various mechanisms for solving the private query release problem, where we were interested in *worst case error*. That is, given a class of queries \mathcal{Q} , of size $|\mathcal{Q}| = k$, we wished to recover a vector of answers $\hat{a} \in \mathbb{R}^k$ such that for *each* query $f_i \in \mathcal{Q}$, $|f_i(x) - \hat{a}_i| \leq \alpha$ for some worst-case error rate α . In other words, if we let $a \in \mathbb{R}^k$ denote the vector of *true* answers, with $a_i \equiv f_i(x)$, then we require a bound of the form: $\|a - \hat{a}\|_\infty \leq \alpha$. In this section, we consider a weakened utility guarantee, on the ℓ_2 (rather than ℓ_∞) error: a bound of the form $\|a - \hat{a}\|_2 \leq \alpha$. A bound of this form does not guarantee

that we have low error for *every* query, but it does guarantee that on average, we have small error.

Although this sort of bound is weaker than worst-case error, the mechanism is particularly simple, and it makes use of an elegant geometric view of the query release problem that we have not seen until now.

Recall that we can view the database x as a vector $x \in \mathbb{N}^{|\mathcal{X}|}$ with $\|x\|_1 = n$. We can similarly also view the queries $f_i \in \mathcal{Q}$ as vectors $f_i \in \mathbb{N}^{|\mathcal{X}|}$, such that $f_i(x) = \langle f_i, x \rangle$. It will therefore be helpful to view our class of queries \mathcal{Q} as a matrix $A \in \mathbb{R}^{k \times |\mathcal{X}|}$, with the i th row of A being the vector f_i . We can then see that our answer vector $a \in \mathbb{R}^k$ is, in matrix notation:

$$A \cdot x = a.$$

Let's consider the domain and range of A when viewed as a linear map. Write $B_1 = \{x \in \mathbb{R}^{|\mathcal{X}|} : \|x\|_1 = 1\}$ denote the unit ℓ_1 ball in $|\mathcal{X}|$ dimensional space. Observe that $x \in nB_1$, since $\|x\|_1 = n$. We will refer to nB_1 as “Database Space.” Write $K = AB_1$. Note similarly that for all $x \in nB_1$, $a = A \cdot x \in nK$. We will refer to nK as “answer space.” We make a couple of observations about K : Note that because B_1 is centrally symmetric, so is K — that is, $K = -K$. Note also that $K \subset \mathbb{R}^k$ is a convex polytope with vertices $\pm A^1, \dots, \pm A^{|\mathcal{X}|}$ equal to the columns of A , together with their negations.

The following algorithm is extremely simple: it simply answers every query independently with the Laplace mechanism, and then *projects back into answer space*. In other words, it adds independent Laplace noise to every query, which as we have seen, by itself leads to distortion that is linear in k (or at least \sqrt{k} , if we relax to (ε, δ) -differential privacy). However, the resulting vector \tilde{a} of answers is likely not consistent with *any* database $y \in nB_1$ in database space. Therefore, rather than returning \tilde{a} , it instead returns some consistent answer vector $\hat{a} \in nK$ that is as close to \tilde{a} as possible. As we will see, this projection step improves the accuracy of the mechanism, while having no effect on privacy (since it is just post-processing!).

We first observe that Project is differentially private.

Theorem 12.7. For any $A \in [0, 1]^{k \times |\mathcal{X}|}$, $\text{Project}(x, A, \varepsilon)$ preserves (ε, δ) -differential privacy.

Algorithm 18 The K -Projected Laplace Mechanism. It takes as input a matrix $A \in [0, 1]^{k \times |\mathcal{X}|}$, a database $x \in nB_1$, and a privacy parameters ε and δ .

Project($x, A, \varepsilon, \delta$):

Let $a = A \cdot x$

For each $i \in [k]$, sample $\nu_i \sim \text{Lap}(\sqrt{8k \ln(1/\delta)}/\varepsilon)$, and let $\tilde{a} = a + \nu$.

Output $\hat{a} = \arg \min_{\hat{a} \in nK} \|\hat{a} - \tilde{a}\|_2^2$.

Proof. We simply note that \tilde{a} is the output of the Laplace mechanism on k sensitivity 1 queries, which is (ε, δ) -differentially private by Theorems 3.6 and 3.20. Finally, since \hat{a} is derived from \tilde{a} without any further access to the private data, the release of \hat{a} is differentially private by the post-processing guarantee of differential privacy, Proposition 2.1. \square

Theorem 12.8. For any class of linear queries A and database x , let $a = A \cdot x$ denote the true answer vector. Let \hat{a} denote the output of the mechanism **Project**: $\hat{a} = \text{Project}(x, A, \varepsilon)$. With probability at least $1 - \beta$:

$$\|a - \hat{a}\|_2^2 \leq \frac{kn\sqrt{192 \ln(1/\delta) \ln(2|\mathcal{X}|/\beta)}}{\varepsilon}.$$

To prove this theorem, we will introduce a couple of simple concepts from convex geometry. For a convex body $K \subset \mathbb{R}^k$, its *polar body* is K° defined to be $K^\circ = \{y \in \mathbb{R}^k : \langle y, x \rangle \leq 1 \text{ for all } x \in K\}$. The *Minkowski Norm* defined by a convex body K is

$$\|x\|_K \equiv \min\{r \in \mathbb{R} \text{ such that } x \in rK\}.$$

The *dual norm* of $\|x\|_K$ is the Minkowski norm induced by the polar body of K , i.e., $\|x\|_{K^\circ}$. This norm also has the following form:

$$\|x\|_{K^\circ} = \max_{y \in K} \langle x, y \rangle.$$

The key fact we will use is *Hölder's Inequality*, which is satisfied by all centrally symmetric convex bodies K :

$$|\langle x, y \rangle| \leq \|x\|_K \|y\|_{K^\circ}.$$

Proof of Theorem 12.8. The proof will proceed in two steps. First we will show that: $\|a - \hat{a}\|_2^2 \leq 2\langle \hat{a} - a, \tilde{a} - a \rangle$, and then we will use Holder's inequality to bound this second quantity.

Lemma 12.9.

$$\|a - \hat{a}\|_2^2 \leq 2\langle \hat{a} - a, \tilde{a} - a \rangle$$

Proof. We calculate:

$$\begin{aligned} \|\hat{a} - a\|_2^2 &= \langle \hat{a} - a, \hat{a} - a \rangle \\ &= \langle \hat{a} - a, \tilde{a} - a \rangle + \langle \hat{a} - a, \hat{a} - \tilde{a} \rangle \\ &\leq 2\langle \hat{a} - a, \tilde{a} - a \rangle. \end{aligned}$$

The inequality follows from calculating:

$$\begin{aligned} \langle \hat{a} - a, \tilde{a} - a \rangle &= \|\tilde{a} - a\|_2^2 + \langle \hat{a} - \tilde{a}, \tilde{a} - a \rangle \\ &\geq \|\hat{a} - \tilde{a}\|_2^2 + \langle \hat{a} - \tilde{a}, \tilde{a} - a \rangle \\ &= \langle \hat{a} - \tilde{a}, \hat{a} - a \rangle, \end{aligned}$$

Where the final inequality follows because by choice of \hat{a} , for all $a' \in nK$: $\|\tilde{a} - \hat{a}\|_2^2 \leq \|\tilde{a} - a'\|_2^2$. \square

We can now complete the proof. Recall that by definition, $\tilde{a} - a = \nu$, the vector of i.i.d. Laplace noise added by the Laplace mechanism. By Lemma 12.9 and Holder's inequality, we have:

$$\begin{aligned} \|a - \hat{a}\|_2^2 &\leq 2\langle \hat{a} - a, \nu \rangle \\ &\leq 2\|\hat{a} - a\|_K \|\nu\|_{K^\circ}. \end{aligned}$$

We bound these two terms separately. Since by definition $\hat{a}, a \in nK$, we have $\max(\|\hat{a}\|_K, \|a\|_K) \leq n$, and so by the triangle inequality, $\|\hat{a} - a\|_K \leq 2n$.

Next, observe that since $\|\nu\|_{K^\circ} = \max_{y \in K} \langle y, \nu \rangle$, and since the maximum of a linear function taken over a polytope is attained at a vertex, we have: $\|\nu\|_{K^\circ} = \max_{i \in [\|\mathcal{X}\|]} |\langle A^i, \nu \rangle|$.

Because each $A^i \in \mathbb{R}^k$ is such that $\|A^i\|_\infty \leq 1$, and recalling that for any scalar q , if $Z \sim \text{Lap}(b)$, then $qZ \sim \text{Lap}(qb)$, we can apply Lemma by

Lemma 12.2 to bound the weighted sums of Laplace random variables $\langle A^i, \nu \rangle$. Doing so, we have that with probability at least $1 - \beta$:

$$\max_{i \in [\lvert \mathcal{X} \rvert]} |\langle A^i, \nu \rangle| \leq \frac{8k\sqrt{\ln(1/\delta) \ln(\lvert \mathcal{X} \rvert / \beta)}}{\epsilon}.$$

Combining all of the above bounds, we get that with probability $1 - \beta$:

$$\|a - \hat{a}\|_2^2 \leq \frac{16nk\sqrt{\ln(1/\delta) \ln(\lvert \mathcal{X} \rvert / \beta)}}{\epsilon}. \quad \square$$

Let's interpret this bound. Observe that $\|a - \hat{a}\|_2^2 = \sum_{i=1}^k (a_i - \hat{a}_i)^2$, and so this is a bound on the sum of squared errors over all queries. Hence, the *average* per-query squared error of this mechanism is only:

$$\frac{1}{k} \sum_{i=1}^k (a_i - \hat{a}_i)^2 \leq \frac{16n\sqrt{\ln(1/\delta) \ln(\lvert \mathcal{X} \rvert / \beta)}}{\epsilon}.$$

In contrast, the private multiplicative weights mechanism guarantees that $\max_{i \in [k]} |a_i - \hat{a}_i| \leq \tilde{O}(\sqrt{n} \log |\mathcal{X}|^{1/4} / \epsilon^{1/2})$, and so matches the average squared error guarantee of the projected Laplace mechanism, with a bound of: $\tilde{O}(n\sqrt{\log |\mathcal{X}|} / \epsilon)$. However, the multiplicative weights mechanism (and especially its privacy analysis) is much more complex than the Projected Laplace mechanism! In particular, the *private* part of the K -Projected Laplace mechanism is simply the Laplace mechanism itself, and requires no coordination between queries. Interestingly — and, it turns out, necessarily — coordination occurs in the projection phase. Since projection is in post-precessing, it incurs no further privacy loss; indeed, it can be carried out (online, if necessary) by the data analyst himself.

12.5 Bibliographical notes

The local model of data privacy has its roots in randomized response, as first proposed by Warner in 1965 [84]. The local model was formalized by Kasiviswanathan et al. [52] in the context of learning, who proved that private learning in the local model is equivalent to non-private

learning in the statistical query (SQ) model. The set of queries which can be *released* in the local model was shown to be exactly equal to the set of queries that can be *agnostically learned* in the SQ model by Gupta et al. [38].

Pan-Privacy was introduced by Dwork et al. [27], and further explored by Mir et al. [62]. The pan-private density estimation, as well as a low-memory variant using hashing, appear in [27].

Privacy under continual observation was introduced by Dwork et al. [26]; our algorithm for counting under continual observation is from that paper, as is the lower bound on error. Similar algorithms were given by Chan et al. [11]. The proof of concentration of measure inequality for the sums of Laplace random variables given in Lemma 12.2 is from [11].

The Projected Laplace mechanism for achieving low average error was given by Nikolov et al. [66], who also give *instance optimal* algorithms for the (average error) query release problem for any class of queries. This work extends a line of work on the connections between differential privacy and geometry started by Hardt and Talwar [45], and extended by Bhaskara et al. [5] and Dwork et al. [30].

Dwork, Naor, and Vadhan proved an exponential gap between the number of queries that can be answered (with non-trivial error) by stateless and stateful differentially private mechanisms [29]. The lesson learned — that coordination is essential for accurately and privately answering very large numbers of queries — seems to rule out the independent noise addition in the Projected Laplace mechanism. The statefulness of that algorithm appears in the projection step, resolving the paradox.

13

Reflections

13.1 Toward practicing privacy

Differential Privacy was designed with internet-scale data sets in mind. Reconstruction attacks along the lines of those in Section 8 can be carried out by a *polynomial time* bounded adversary asking only $O(n)$ queries on databases of size n . When n is on the order of hundreds of millions, and each query requires a linear amount of computation, such an attack is unrealistic, even though the queries can be parallelized. This observation led to the earliest steps toward differential privacy: If the adversary is restricted to a *sublinear* number of counting queries, then $o(\sqrt{n})$ noise per query — less than the sampling error! — is sufficient for preserving privacy (Corollary 3.21).

To what extent can differential privacy be brought to bear on smaller data sets, or even targeted attacks that isolate a small subset of a much larger database, without destroying statistical utility? First, an analysis may require a number of queries that begins to look something like the size of this smaller set. Second, letting n now denote the size of the smaller set or small database, and letting k be the number of queries, fractional errors on the order of \sqrt{k}/n are harder to ignore when n is small. Third, the $\sqrt{\ln(1/\delta)}/\varepsilon$ factor in the advanced

composition theorem becomes significant. Keeping in mind the reconstruction attacks when noise is $o(\sqrt{n})$, there appears to be little room to maneuver for arbitrary sets of $k \approx n$ low-sensitivity queries.

There are several promising lines of research for addressing these concerns.

The Query Errors Don't Tell the Whole Story. As an example of this phenomenon, consider the problem of linear regression. The input is a collection of labeled data points of the form (x, y) , where $x \in \mathbb{R}^d$ and $y \in \mathbb{R}$, for arbitrary dimension d . The goal is to find $\theta \in \mathbb{R}^d$ that “predicts” y “as well as possible,” given x , under the assumption that the relationship is linear. If the goal is simply to “explain” the given data set, differential privacy may well introduce unacceptable error. Certainly the specific algorithm that simply computes

$$\operatorname{argmin}_{\theta} \left| \sum_{i=1}^n \theta \cdot x_i - y_i \right|^2$$

and adds appropriately scaled Laplace noise independently to each coordinate of θ may produce a $\tilde{\theta}$ that differs substantially from θ . But if the goal is to learn a predictor that will do well for *future, unseen* inputs (x, y) then a slightly different computation is used to avoid overfitting, and the (possibly large) difference between the private and non-private coefficient vectors does *not* translate into a gap in classification error! A similar phenomenon has been observed in model fitting.

Less Can Be More. Many analyses ask for more than they actually use. Exploitation of this principle is at the heart of Report Noisy Max, where for the accuracy “price” of one measurement we learn one of the largest of many measurements. By asking for “less” (that is, not requiring that all noisy measurements be released, but rather only asking for the largest one), we obtain “more” (better accuracy). A familiar principle in privacy is to *minimize* collection and reporting. Here we see this play out in the realm of what must be *revealed*, rather than what must be used in the computation.

Quit When You are NOT Ahead. This is the philosophy behind Propose-Test-Release, in which we test in a privacy-preserving way

that small noise is sufficient for a particular intended computation on the given data set.

Algorithms with Data-Dependent Accuracy Bounds. This can be viewed as a generalization of Quit When You are Not Ahead. Algorithms with data-dependent accuracy bounds can deliver excellent results on “good” data sets, as in Propose-Test-Release, and the accuracy can degrade gradually as the “goodness” decreases, an improvement over Propose-Test-Release.

Exploit “Nice” Query Sets. When (potentially large) sets of linear queries are presented as a batch it is possible, by analyzing the geometry of the query *matrix* to obtain higher quality answers than would be obtained were the queries answered independently¹.

Further Relaxation of Differential Privacy We have seen that (ϵ, δ) -differential privacy is a meaningful relaxation of differential privacy that can provide substantially improved accuracy bounds. Moreover, such a relaxation can be essential to these improvements. For example, Propose-Test-Release algorithms can only offer (ϵ, δ) -differential privacy for $\delta > 0$. What about other, but still meaningful, relaxations of differential privacy? *Concentrated Differential Privacy* is such a relaxation that is incomparable to (ϵ, δ) -differential privacy and that permits better accuracy. Roughly speaking, it ensures that large privacy loss happens with very small probability; for example, for all k the probability of privacy loss $k\epsilon$ falls exponentially in k^2 . In contrast, (ϵ, δ) -differential privacy is consistent with having *infinite* privacy loss with probability δ ; on the other hand, privacy lost 2ϵ can happen in concentrated differential privacy with constant probability, while in (ϵ, δ) -differential privacy it will only occur with probability bounded by δ , which we typically take to be cryptographically small.

Why might we feel comfortable with this relaxation? The answer lies in behavior under composition. As an individual’s data participate

¹More accurately, the analysis is of the object $K = AB_1^k$, where A is the query matrix and B_1^k is the k -dimensional L_1 ball; note that K is the feasible region in answer space when the database has one element.

in many databases and many different computations, perhaps the real worry is the combined threat of multiple exposures. This is captured by privacy under composition. Concentrated differential privacy permits better accuracy while yielding the same behavior under composition as (ε, δ) (and $(\varepsilon, 0)$) differential privacy.

Differential privacy also faces a number of cultural challenges. One of the most significant is non-algorithmic thinking. Differential privacy is a property of an algorithm. However, many people who work with data describe their interactions with the data in fundamentally non-algorithmic terms, such as, “First, I *look at* the data.” Similarly, data cleaning is often described in non-algorithmic terms. If data are reasonably plentiful, and the analysts are energetic, then the “Raw Data” application of the Subsample and Aggregate methodology described in Example 7.3 suggests a path toward enabling non-algorithmic, interactions by trusted analysts who will follow directions. In general, it seems plausible that on high-dimensional and on internet-scale data sets non-algorithmic interactions will be the exception.

What about ε ? In Example 3.7 we applied Theorem 3.20 to conclude that to bound the cumulative lifetime privacy loss at $\varepsilon = 1$ with probability $1 - e^{-32}$, over participation in 10,000 databases, it is sufficient that each database be $(1/801, 0)$ -differentially private. While $k = 10,000$ may be an overestimate, the dependence on k is fairly weak (\sqrt{k}), and in the worst case these bounds are tight, ruling out a more relaxed bound than $\varepsilon_0 = 1/801$ for each database *over the lifetime of the database*. This is simply too strict a requirement in practice.

Perhaps we can ask a different question: Fix ε , say, $\varepsilon = 1$ or $\varepsilon = 1/10$; now ask: How can multiple ε ’s be apportioned? Permitting ε privacy loss *per query* is too weak, and ε loss over the lifetime of the database is too strong. Something in between, say, ε per study or ε per researcher, may make sense, although this raises the questions of who is a “researcher” and what constitutes a “study.” This affords substantially more protection against accidental and intentional privacy compromise than do current practices, from enclaves to confidentiality contracts.

A different proposal is less prescriptive. This proposal draws from second-generation regulatory approaches to reducing environmental

degradation, in particular pollution release registries such as the Toxic Release Inventory that have been found to encourage better practices through transparency. Perhaps a similar effect could arise with private data analysis: an Epsilon Registry describing data uses, granularity of privacy protection, a “burn rate” of privacy loss per unit time, and a cap on total privacy loss permitted before data are retired, when accompanied with a financial penalty for infinite (or very large) loss, can lead to innovation and competition, deploying the talents and resources of a larger set of researchers and privacy professionals in the search for differentially private algorithms.

13.2 The differential privacy lens

An online etymological dictionary describes the original 18th century meaning of the term of the word “statistics” as “science dealing with data about the condition of a state or community.” This resonates with differential privacy in the breach: if the presence or absence of the data of a small number of individuals changes the outcome of an analysis then in some sense the outcome is “about” these few individuals, and is not describing the condition of the community as a whole. Put differently, stability to small perturbations in the data is both the hallmark of differential privacy and the essence of a common conception of the term “statistical.” Differential privacy is enabled by stability (Section 7) and ensures stability (by definition). In some sense it forces all queries to be statistical in nature. As stability is also increasingly understood to be a key necessary and sufficient condition for learnability, we observe a tantalizing moral equivalence between learnability, differential privacy, and stability.

With this in mind, it is not surprising that differential privacy is also a means to ends other than privacy, and indeed we saw this with game theory in Section 10. The power of differential privacy comes from its amenability to composition. Just as composition allows us to build complex differentially private algorithms from smaller differentially private building blocks, it provides a programming language for constructing stable algorithms for complex analytical tasks. Consider, for example, the problem of eliciting a set of bidder values, and using them to price

a collection of goods that are for sale. Informally, *Walrasian equilibrium prices* are prices such that every individual can simultaneously purchase their *favorite* bundle of goods *given the prices*, while ensuring that demand exactly equals the supply of each good. It would seem at first blush, then, that simply computing these prices, and assigning each person their favorite bundle of goods given the prices would yield a mechanism in which agents were incentivized to tell the truth about their valuation function — since how could any agent do better than receiving their favorite bundle of goods? However, this argument fails — because in a Walrasian equilibrium, agents receive their favorite bundle of goods *given the prices*, but the prices are computed as a function of the reported valuations, so an industrious but dishonest agent could potentially gain by manipulating the computed prices. However, this problem is solved (and an approximately truthful mechanism results) if the equilibrium prices are computed using a differentially private algorithm — precisely because individual agents have almost no effect on the distribution of prices computed. Note that this application is made possible by the use of the tools of differential privacy, but is completely orthogonal to privacy concerns. More generally, this connection is more fundamental: computing *equilibria* of various sorts using algorithms that have the stability property guaranteed by differential privacy leads to approximately truthful mechanisms implementing these equilibrium outcomes.

Differential privacy also helps in ensuring generalizability in adaptive data analysis. Adaptivity means that the questions asked and hypotheses tested depend on outcomes of earlier questions. Generalizability means that the outcome of a computation or a test on the data set is close to the ground truth of the distribution from which the data are sampled. It is known that the naive paradigm of answering queries with the exact empirical values on a fixed data set fails to generalize even under a limited amount of adaptive questioning. Remarkably, answering with differential privacy not only ensures privacy, but with high probability it ensures generalizability even for exponentially many adaptively chosen queries. Thus, the deliberate introduction of noise using the techniques of differential privacy has profound and promising implications for the validity of traditional scientific inquiry.

Appendices

A

The Gaussian Mechanism

Let $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^d$ be an arbitrary d -dimensional function, and define its ℓ_2 sensitivity to be $\Delta_2 f = \max_{\text{adjacent } x, y} \|f(x) - f(y)\|_2$. The *Gaussian Mechanism with parameter σ* adds noise scaled to $\mathcal{N}(0, \sigma^2)$ to each of the d components of the output.

Theorem A.1. Let $\varepsilon \in (0, 1)$ be arbitrary. For $c^2 > 2 \ln(1.25/\delta)$, the Gaussian Mechanism with parameter $\sigma \geq c\Delta_2 f / \varepsilon$ is (ε, δ) -differentially private.

Proof. There is a database D and a query f , and the mechanism will return $f(D) + \eta$, where the noise is normally distributed. We are adding noise $\mathcal{N}(0, \sigma^2)$. For now, assume we are talking about real-valued functions, so

$$\Delta f = \Delta_1 f = \Delta_2 f.$$

We are looking at

$$\left| \ln \frac{e^{(-1/2\sigma^2)x^2}}{e^{(-1/2\sigma^2)(x+\Delta f)^2}} \right|. \quad (\text{A.1})$$

We are investigating the probability, given that the database is D , of observing an output that occurs with a very different probability

under D than under an adjacent database D' , where the probability space is the noise generation algorithm. The numerator in the ratio above describes the probability of seeing $f(D) + x$ when the database is D , the denominator corresponds the probability of seeing *this same value* when the database is D' . This is a ratio of probabilities, so it is always positive, but the logarithm of the ratio may be negative. Our random variable of interest — the privacy loss — is

$$\ln \frac{e^{(-1/2\sigma^2)x^2}}{e^{(-1/2\sigma^2)(x+\Delta f)^2}}$$

and we are looking at its absolute value.

$$\begin{aligned} \left| \ln \frac{e^{(-1/2\sigma^2)x^2}}{e^{(-1/2\sigma^2)(x+\Delta f)^2}} \right| &= \left| \ln e^{(-1/2\sigma^2)[x^2 - (x+\Delta f)^2]} \right| \\ &= \left| -\frac{1}{2\sigma^2}[x^2 - (x^2 + 2x\Delta f + \Delta f^2)] \right| \\ &= \left| \frac{1}{2\sigma^2}(2x\Delta f + (\Delta f)^2) \right|. \end{aligned} \quad (\text{A.2})$$

This quantity is bounded by ε whenever $x < \sigma^2\varepsilon/\Delta f - \Delta f/2$. To ensure privacy loss bounded by ε with probability at least $1 - \delta$, we require

$$\Pr[|x| \geq \sigma^2\varepsilon/\Delta f - \Delta f/2] < \delta,$$

and because we are concerned with $|x|$ we will find σ such that

$$\Pr[x \geq \sigma^2\varepsilon/\Delta f - \Delta f/2] < \delta/2.$$

We will assume throughout that $\varepsilon \leq 1 \leq \Delta f$.

We will use the tail bound

$$\Pr[x > t] \leq \frac{\sigma}{\sqrt{2\pi}} e^{-t^2/2\sigma^2}.$$

We require:

$$\begin{aligned} \frac{\sigma}{\sqrt{2\pi}} \frac{1}{t} e^{-t^2/2\sigma^2} &< \delta/2 \\ \Leftrightarrow \sigma \frac{1}{t} e^{-t^2/2\sigma^2} &< \sqrt{2\pi}\delta/2 \\ \Leftrightarrow \frac{t}{\sigma} e^{t^2/2\sigma^2} &> 2/\sqrt{2\pi}\delta \\ \Leftrightarrow \ln(t/\sigma) + t^2/2\sigma^2 &> \ln(2/\sqrt{2\pi}\delta). \end{aligned}$$

Taking $t = \sigma^2 \varepsilon / \Delta f - \Delta f / 2$, we get

$$\begin{aligned} & \ln((\sigma^2 \varepsilon / \Delta f - \Delta f / 2) / \sigma) + (\sigma^2 \varepsilon / \Delta f - \Delta f / 2)^2 / 2\sigma^2 > \ln(2/\sqrt{2\pi}\delta) \\ &= \ln\left(\sqrt{\frac{2}{\pi}} \frac{1}{\delta}\right). \end{aligned}$$

Let us write $\sigma = c\Delta f / \varepsilon$; we wish to bound c . We begin by finding the conditions under which the first term is non-negative.

$$\begin{aligned} \frac{1}{\sigma} \left(\sigma^2 \frac{\varepsilon}{\Delta f} - \frac{\Delta f}{2} \right) &= \frac{1}{\sigma} \left[\left(c^2 \frac{(\Delta f)^2}{\varepsilon^2} \right) \frac{\varepsilon}{\Delta f} - \frac{\Delta f}{2} \right] \\ &= \frac{1}{\sigma} \left[c^2 \left(\frac{\Delta f}{\varepsilon} \right) - \frac{\Delta f}{2} \right] \\ &= \frac{\varepsilon}{c\Delta f} \left[c^2 \left(\frac{\Delta f}{\varepsilon} \right) - \frac{\Delta f}{2} \right] \\ &= c - \frac{\varepsilon}{2c}. \end{aligned}$$

Since $\varepsilon \leq 1$ and $c \geq 1$, we have $c - \varepsilon/(2c) \geq c - 1/2$. So $\ln(\frac{1}{\sigma}(\sigma^2 \frac{\varepsilon}{\Delta f} - \frac{\Delta f}{2})) > 0$ provided $c \geq 3/2$. We can therefore focus on the t^2/σ^2 term.

$$\begin{aligned} \left(\frac{1}{2\sigma^2} \frac{\sigma^2 \varepsilon}{\Delta f} - \frac{\Delta f}{2} \right)^2 &= \frac{1}{2\sigma^2} \left[\Delta f \left(\frac{c^2}{\varepsilon} - \frac{1}{2} \right) \right]^2 \\ &= \left[(\Delta f)^2 \left(\frac{c^2}{\varepsilon} - \frac{1}{2} \right) \right]^2 \left[\frac{\varepsilon^2}{c^2(\Delta f)^2} \right] \frac{1}{2} \\ &= \frac{1}{2} \left(\frac{c^2}{\varepsilon} - \frac{1}{2} \right)^2 \frac{\varepsilon^2}{c^2} \\ &= \frac{1}{2}(c^2 - \varepsilon + \varepsilon^2/4c^2). \end{aligned}$$

Since $\varepsilon \leq 1$ the derivative of $(c^2 - \varepsilon + \varepsilon^2/4c^2)$ with respect to c is positive in the range we are considering ($c \geq 3/2$), so $c^2 - \varepsilon + \varepsilon^2/4c^2 \geq c^2 - 8/9$ and it suffices to ensure

$$c^2 - 8/9 > 2 \ln\left(\sqrt{\frac{2}{\pi}} \frac{1}{\delta}\right).$$

In other words, we need that

$$c^2 > 2 \ln(\sqrt{2/\pi}) + 2 \ln(1/\delta) + \ln(e^{8/9}) = \ln(2/\pi) + \ln(e^{8/9}) + 2 \ln(1/\delta),$$

which, since $(2/\pi)e^{8/9} < 1.55$, is satisfied whenever $c^2 > 2 \ln(1.25/\delta)$.

Let us partition \mathbb{R} as $\mathbb{R} = R_1 \cup R_2$, where $R_1 = \{x \in \mathbb{R} : |x| \leq c\Delta f/\varepsilon\}$ and $R_2 = \{x \in \mathbb{R} : |x| > c\Delta f/\varepsilon\}$. Fix any subset $S \subseteq \mathbb{R}$, and define

$$\begin{aligned} S_1 &= \{f(x) + x \mid x \in R_1\} \\ S_2 &= \{f(x) + x \mid x \in R_2\}. \end{aligned}$$

We have

$$\begin{aligned} \Pr_{x \sim \mathcal{N}(0, \sigma^2)}[f(x) + x \in S] &= \Pr_{x \sim \mathcal{N}(0, \sigma^2)}[f(x) + x \in S_1] \\ &\quad + \Pr_{x \sim \mathcal{N}(0, \sigma^2)}[f(x) + x \in S_2] \\ &\leq \Pr_{x \sim \mathcal{N}(0, \sigma^2)}[f(x) + x \in S_1] + \delta \\ &\leq e^\varepsilon \left(\Pr_{x \sim \mathcal{N}(0, \sigma^2)}[f(y) + x \in S_1] \right) + \delta, \end{aligned}$$

yielding (ε, δ) -differential privacy for the Gaussian mechanism in one dimension.

High Dimension. To extend this to functions in R^m , define $\Delta f = \Delta_2 f$. We can now repeat the argument, using Euclidean norms. Let v be any vector satisfying $\|v\| \leq \Delta f$. For a fixed pair of databases x, y we are interested in $v = f(x) - f(y)$, since this is what our noise must obscure. As in the one dimensional, case we seek conditions on σ under which the privacy loss

$$\left| \ln \frac{e^{(-1/2\sigma^2)\|x-\mu\|^2}}{e^{(-1/2\sigma^2)\|x+v-\mu\|^2}} \right|$$

is bounded by ε ; here x is chosen from $\mathcal{N}(0, \Sigma)$, where (Σ) is a diagonal matrix with entries σ^2 , whence $\mu = (0, \dots, 0)$.

$$\begin{aligned} \left| \ln \frac{e^{(-1/2\sigma^2)\|x-\mu\|^2}}{e^{(-1/2\sigma^2)\|x+v-\mu\|^2}} \right| &= \left| \ln e^{(-1/2\sigma^2)[\|x-\mu\|^2 - \|x+v-\mu\|^2]} \right| \\ &= \left| \frac{1}{2\sigma^2} (\|x\|^2 - \|x+v\|^2) \right|. \end{aligned}$$

We will use the fact that the distribution of a spherically symmetric normal is independent of the orthogonal basis from which its constituent normals are drawn, so we may work in a basis that is aligned with v . Fix such a basis b_1, \dots, b_m , and draw x by first drawing signed lengths $\lambda_i \sim \mathcal{N}(0, \sigma^2)$, for $i \in [m]$, then defining $x^{[i]} = \lambda_i b_i$, and finally letting $x = \sum_{i=1}^m x^{[i]}$. Assume without loss of generality that b_1 is parallel to v . We are interested in $|\|x\|^2 - \|x+v\|^2|$.

Consider the right triangle with base $v + x^{[1]}$ and edge $\sum_{i=2}^m x^{[i]}$ orthogonal to v . The hypotenuse of this triangle is $x + v$.

$$\begin{aligned} \|x + v\|^2 &= \|v + x^{[1]}\|^2 + \sum_{i=2}^m \|x^{[i]}\|^2 \\ \|x\|^2 &= \sum_{i=1}^m \|x^{[i]}\|^2. \end{aligned}$$

Since v is parallel to $x^{[1]}$ we have $\|v + x^{[1]}\|^2 = (\|v\| + \lambda_1)^2$. Thus, $\|x + v\|^2 - \|x\|^2 = \|v\|^2 + 2\lambda_1 \cdot \|v\|$. Recall that $\|v\| \leq \Delta f$, and $\lambda \sim \mathcal{N}(0, \sigma)$, so we are now exactly back in the one-dimensional case, writing λ_1 instead of x in Equation (A.2):

$$\left| \frac{1}{2\sigma^2} (\|x\|^2 - \|x+v\|^2) \right| \leq \left| \frac{1}{2\sigma^2} (2\lambda_1 \Delta f - (\Delta f)^2) \right|$$

and the rest of the argument proceeds as above. \square

The argument for the high dimensional case highlights a weakness of (ε, δ) -differential privacy that does not exist for $(\varepsilon, 0)$ -differential privacy. Fix a database x . In the $(\varepsilon, 0)$ -case, the guarantee of indistinguishability holds for all adjacent databases *simultaneously*. In the

(ε, δ) case indistinguishability only holds “prospectively,” i.e., for any fixed y adjacent to x , the probability that the mechanism will allow the adversary to distinguish x from y is small. In the proof above, this is manifested by the fact that we fixed $v = f(x) - f(y)$; we did not have to argue about all possible directions of v simultaneously, and indeed we cannot, as once we have fixed our noise vector $x \sim \mathcal{N}(0, \Sigma)$, so that the output on x is $o = f(x) + x$, there may exist an adjacent y such that output $o = f(x) + x$ is much more likely when the database is y than it is on x .

A.1 Bibliographic notes

Theorem A.1 is folklore initially observed by the authors of [23]. A generalization to non-spherical gaussian noise appears in [66].

B

Composition Theorems for (ε, δ) -DP

B.1 Extension of Theorem 3.16

Theorem B.1. Let $T_1(D) : D \mapsto T_1(D) \in \mathcal{C}_1$ be an (ε, δ) -d.p. function, and for any $s_1 \in \mathcal{C}_1$, $T_2(D, s_1) : (D, s_1) \mapsto T_2(D, s_1) \in \mathcal{C}_2$ be an (ε, δ) -d.p. function given the second input s_1 . Then we show that for any neighboring D, D' , for any $S \subseteq \mathcal{C}_2 \times \mathcal{C}_1$, we have, using the notation in our paper

$$P((T_2, T_1) \in S) \leq e^{2\varepsilon} P'((T_2, T_1) \in S) + 2\delta. \quad (\text{B.1})$$

Proof. For any $C_1 \subseteq \mathcal{C}_1$, define

$$\mu(C_1) = (P(T_1 \in C_1) - e^\varepsilon P'(T_1 \in C_1))_+,$$

then μ is a measure on \mathcal{C}_1 and $\mu(\mathcal{C}_1) \leq \delta$ since T_1 is (ε, δ) -d.p. As a result, we have for all $s_1 \in \mathcal{C}_1$,

$$P(T_1 \in ds_1) \leq e^\varepsilon P'(T_1 \in ds_1) + \mu(ds_1). \quad (\text{B.2})$$

Also note that by the definition of (ε, δ) -d.p., for any $s_1 \in \mathcal{C}_1$,

$$\begin{aligned} P((T_2, s_1) \in S) &\leq (e^\varepsilon P'((T_2, s_1) \in S) + \delta) \wedge 1 \\ &\leq (e^\varepsilon P'((T_2, s_1) \in S)) \wedge 1 + \delta. \end{aligned} \quad (\text{B.3})$$

Then (B.2) and (B.3) give (B.1):

$$\begin{aligned}
P((T_2, T_1) \in S) &\leq \int_{S_1} P((T_2, s_1) \in S) P(T_1 \in ds_1) \\
&\leq \int_{S_1} ((e^\epsilon P'((T_2, s_1) \in S)) \wedge 1 + \delta) P(T_1 \in ds_1) \\
&\leq \int_{S_1} ((e^\epsilon P'((T_2, s_1) \in S)) \wedge 1) P(T_1 \in ds_1) + \delta \\
&\leq \int_{S_1} ((e^\epsilon P'((T_2, s_1) \in S)) \wedge 1) \\
&\quad \times (e^\epsilon P'(T_1 \in ds_1) + \mu(ds_1)) + \delta \\
&\leq e^{2\epsilon} \int_{S_1} P'((T_2, s_1) \in S) P'(T_1 \in ds_1) + \mu(S_1) + \delta \\
&\leq e^{2\epsilon} P'((T_2, T_1) \in S) + 2\delta. \tag{B.4}
\end{aligned}$$

In the equations above, S_1 denotes the projection of S onto \mathcal{C}_1 . The event $\{(T_2, s_1) \in S\}$ refers to $\{(T_2(D, s_1), s_1) \in S\}$ (or $\{(T_2(D'), s_1) \in S\}$). \square

Using induction, we have:

Corollary B.2 (general composition theorem for (ϵ, δ) -d.p. algorithms). Let $T_1 : D \mapsto T_1(D)$ be (ϵ, δ) -d.p., and for $k \geq 2$, $T_k : (D, s_1, \dots, s_{k-1}) \mapsto T_k(D, s_1, \dots, s_{k-1}) \in \mathcal{C}_k$ be (ϵ, δ) -d.p., for all given $(s_{k-1}, \dots, s_1) \in \bigotimes_{j=1}^{k-1} \mathcal{C}_j$. Then for all neighboring D, D' and all $S \subseteq \bigotimes_{j=1}^k \mathcal{C}_j$

$$P((T_1, \dots, T_k) \in S) \leq e^{k\epsilon} P'((T_1, \dots, T_k) \in S) + k\delta.$$

Acknowledgments

We would like to thank many people for providing careful comments and corrections on early drafts of this book, including Vitaly Feldman, Justin Hsu, Simson Garfinkel, Katrina Ligett, Dong Lin, David Parkes, Ryan Rogers, Guy Rothblum, Ian Schmutte, Jon Ullman, Salil Vadhan, Zhiwei Steven Wu, and the anonymous referees. This book was used in a course taught by Salil Vadhan and Jon Ullman, whose students also provided careful feedback. This book has also benefited from conversations with many other colleagues, including Moritz Hardt, Ilya Mironov, Sasho Nikolov, Kobbi Nissim, Mallesh Pai, Benjamin Pierce, Adam Smith, Abhradeep Thakurta, Abhishek Bhowmick, Kunal Talwar, and Li Zhang. We are grateful to Madhu Sudan for proposing this monograph.

References

- [1] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: A meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.
- [2] M.-F. Balcan, A. Blum, J. D. Hartline, and Y. Mansour. Mechanism design via machine learning. In *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, pages 605–614. IEEE, 2005.
- [3] A. Beimel, S. P. Kasiviswanathan, and K. Nissim. Bounds on the sample complexity for private learning and private data release. In *Theory of Cryptography*, pages 437–454. Springer, 2010.
- [4] A. Beimel, K. Nissim, and U. Stemmer. Characterizing the sample complexity of private learners. In *Proceedings of the Conference on Innovations in Theoretical Computer Science*, pages 97–110. Association for Computing Machinery, 2013.
- [5] A. Bhaskara, D. Dadush, R. Krishnaswamy, and K. Talwar. Unconditional differentially private mechanisms for linear queries. In H. J. Karloff and T. Pitassi, editors, *Proceedings of the Symposium on Theory of Computing Conference, Symposium on Theory of Computing, New York, NY, USA, May 19–22, 2012*, pages 1269–1284. 2012.
- [6] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the SuLQ framework. In Chen Li, editor, *Principles of Database Systems*, pages 128–138. ACM, 2005.
- [7] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the sulq framework. In *Principles of Database Systems*. 2005.

- [8] A. Blum, K. Ligett, and A. Roth. A learning theory approach to non-interactive database privacy. In Cynthia Dwork, editor, *Symposium on Theory of Computing*, pages 609–618. Association for Computing Machinery, 2008.
- [9] A. Blum and Y. Mansour. Learning, regret minimization, and equilibria, 2007.
- [10] J. L. Casti. *Five Golden Rules: Great Theories of 20th-Century Mathematics and Why They Matter*. Wiley, 1996.
- [11] T. H. Hubert Chan, E. Shi, and D. Song. Private and continual release of statistics. In *Automata, Languages and Programming*, pages 405–417. Springer, 2010.
- [12] K. Chaudhuri and D. Hsu. Sample complexity bounds for differentially private learning. In *Proceedings of the Annual Conference on Learning Theory (COLT 2011)*. 2011.
- [13] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *Journal of machine learning research: JMLR*, 12:1069, 2011.
- [14] K. Chaudhuri, A. Sarwate, and K. Sinha. Near-optimal differentially private principal components. In *Advances in Neural Information Processing Systems 25*, pages 998–1006. 2012.
- [15] Y. Chen, S. Chong, I. A. Kash, T. Moran, and S. P. Vadhan. Truthful mechanisms for agents that value privacy. *Association for Computing Machinery Conference on Electronic Commerce*, 2013.
- [16] P. Dandekar, N. Fawaz, and S. Ioannidis. Privacy auctions for recommender systems. In *Internet and Network Economics*, pages 309–322. Springer, 2012.
- [17] A. De. Lower bounds in differential privacy. In *Theory of Cryptography Conference*, pages 321–338. 2012.
- [18] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *Proceedings of the Association for Computing Machinery SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 202–210. 2003.
- [19] J. C. Duchi, M. I. Jordan, and M. J. Wainwright. Local privacy and statistical minimax rates. *arXiv preprint arXiv:1302.3203*, 2013.
- [20] C. Dwork. Differential privacy. In *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP)(2)*, pages 1–12. 2006.

- [21] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pages 486–503. 2006.
- [22] C. Dwork and J. Lei. Differential privacy and robust statistics. In *Proceedings of the 2009 International Association for Computing Machinery Symposium on Theory of Computing (STOC)*. 2009.
- [23] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference '06*, pages 265–284. 2006.
- [24] C. Dwork, F. McSherry, and K. Talwar. The price of privacy and the limits of l_p decoding. In *Proceedings of the Association for Computing Machinery Symposium on Theory of Computing*, pages 85–94. 2007.
- [25] C. Dwork and M. Naor. On the difficulties of disclosure prevention in statistical databases or the case for differential privacy. *Journal of Privacy and Confidentiality*, 2010.
- [26] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. Differential privacy under continual observation. In *Proceedings of the Association for Computing Machinery Symposium on Theory of Computing*, pages 715–724. Association for Computing Machinery, 2010.
- [27] C. Dwork, M. Naor, T. Pitassi, G. N. Rothblum, and Sergey Yekhanin. Pan-private streaming algorithms. In *Proceedings of International Conference on Super Computing*. 2010.
- [28] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum, and S. P. Vadhan. On the complexity of differentially private data release: Efficient algorithms and hardness results. In *Symposium on Theory of Computing '09*, pages 381–390. 2009.
- [29] C. Dwork, M. Naor, and S. Vadhan. The privacy of the analyst and the power of the state. In *Foundations of Computer Science*. 2012.
- [30] C. Dwork, A. Nikolov, and K. Talwar. Efficient algorithms for privately releasing marginals via convex relaxations. In *Proceedings of the Annual Symposium on Computational Geometry (SoCG)*. 2014.
- [31] C. Dwork and K. Nissim. Privacy-preserving datamining on vertically partitioned databases. In *Proceedings of Cryptology 2004*, vol. 3152, pages 528–544. 2004.
- [32] C. Dwork, G. N. Rothblum, and S. P. Vadhan. Boosting and differential privacy. In *Foundations of Computer Science*, pages 51–60. 2010.

- [33] C. Dwork, K. Talwar, A. Thakurta, and L. Zhang. Analyze gauss: Optimal bounds for privacy-preserving pca. In *Symposium on Theory of Computing*. 2014.
- [34] L. Fleischer and Y.-H. Lyu. Approximately optimal auctions for selling privacy when costs are correlated with data. In *Association for Computing Machinery Conference on Electronic Commerce*, pages 568–585. 2012.
- [35] A. Ghosh and K. Ligett. Privacy and coordination: Computing on databases with endogenous participation. In *Proceedings of the fourteenth ACM conference on Electronic commerce (EC)*, pages 543–560, 2013.
- [36] A. Ghosh and A. Roth. Selling privacy at auction. In *Association for Computing Machinery Conference on Electronic Commerce*, pages 199–208. 2011.
- [37] A. Groce, J. Katz, and A. Yerukhimovich. Limits of computational differential privacy in the client/server setting. In *Proceedings of the Theory of Cryptography Conference*. 2011.
- [38] A. Gupta, M. Hardt, A. Roth, and J. Ullman. Privately releasing conjunctions and the statistical query barrier. In *Symposium on Theory of Computing '11*, pages 803–812. 2011.
- [39] A. Gupta, A. Roth, and J. Ullman. Iterative constructions and private data release. In *Theory of Cryptography Conference*, pages 339–356. 2012.
- [40] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal of Computing*, 28, 1999.
- [41] M. Hardt, K. Ligett, and F. McSherry. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems 25*, pages 2348–2356. 2012.
- [42] M. Hardt and A. Roth. Beating randomized response on incoherent matrices. In *Proceedings of the Symposium on Theory of Computing*, pages 1255–1268. Association for Computing Machinery, 2012.
- [43] M. Hardt and A. Roth. Beyond worst-case analysis in private singular vector computation. In *Proceedings of the Symposium on Theory of Computing*. 2013.
- [44] M. Hardt and G. N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *Foundations of Computer Science*, pages 61–70. IEEE Computer Society, 2010.

- [45] M. Hardt and K. Talwar. On the geometry of differential privacy. In *Proceedings of the Association for Computing Machinery Symposium on Theory of Computing*, pages 705–714. Association for Computing Machinery, 2010.
- [46] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. Pearson, D. Stephan, S. Nelson, and D. Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS Genet*, 4, 2008.
- [47] J. Hsu, Z. Huang, A. Roth, T. Roughgarden, and Z. S. Wu. Private matchings and allocations. arXiv preprint arXiv:1311.2828, 2013.
- [48] J. Hsu, A. Roth, and J. Ullman. Differential privacy for the analyst via private equilibrium computation. In *Proceedings of the Association for Computing Machinery Symposium on Theory of Computing (STOC)*, pages 341–350, 2013.
- [49] Z. Huang and S. Kannan. The exponential mechanism for social welfare: Private, truthful, and nearly optimal. In *IEEE Annual Symposium on the Foundations of Computer Science (FOCS)*, pages 140–149. 2012.
- [50] P. Jain, P. Kothari, and A. Thakurta. Differentially private online learning. *Journal of Machine Learning Research — Proceedings Track*, 23:24.1–24.34, 2012.
- [51] M. Kapralov and K. Talwar. On differentially private low rank approximation. In Sanjeev Khanna, editor, *Symposium on Discrete Algorithms*, pages 1395–1414. SIAM, 2013.
- [52] S. P. Kasiviswanathan, H. K. Lee, Kobbi Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.
- [53] M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the Association for Computing Machinery (JAssociation for Computing Machinery)*, 45(6):983–1006, 1998.
- [54] M. Kearns, M. Pai, A. Roth, and J. Ullman. Mechanism design in large games: Incentives and privacy. In *Proceedings of the 5th conference on Innovations in theoretical computer science (ITCS)*, 2014.
- [55] D. Kifer, A. Smith, and A. Thakurta. Private convex empirical risk minimization and high-dimensional regression. *Journal of Machine Learning Research*, 1:41, 2012.
- [56] K. Ligett and A. Roth. Take it or leave it: Running a survey when privacy comes at a cost. In *Internet and Network Economics*, pages 378–391. Springer, 2012.

- [57] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. In *Annual Symposium on Foundations of Computer Science, 1989*, pages 256–261. IEEE, 1989.
- [58] A. McGregor, I. Mironov, T. Pitassi, O. Reingold, K. Talwar, and S. P. Vadhan. The limits of two-party differential privacy. In *Foundations of Computer Science*, pages 81–90. IEEE Computer Society, 2010.
- [59] F. McSherry. Privacy integrated queries (codebase). Available on Microsoft Research downloads website. See also the Proceedings of SIGMOD 2009.
- [60] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *Foundations of Computer Science*, pages 94–103. 2007.
- [61] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *Foundations of Computer Science*, pages 94–103. 2007.
- [62] D. Mir, S. Muthukrishnan, A. Nikolov, and R. N. Wright. Pan-private algorithms via statistics on sketches. In *Proceedings of the Association for Computing Machinery SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 37–48. Association for Computing Machinery, 2011.
- [63] I. Mironov. On significance of the least significant bits for differential privacy. In T. Yu, G. Danezis, and V. D. Gligor, editors, *Association for Computing Machinery Conference on Computer and Communications Security*, pages 650–661. Association for Computing Machinery, 2012.
- [64] I. Mironov, O. Pandey, O. Reingold, and S. P. Vadhan. Computational differential privacy. In *Proceedings of CRYPTOLOGY*, pages 126–142. 2009.
- [65] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets (how to break anonymity of the netflix prize dataset). In *Proceedings of IEEE Symposium on Security and Privacy*. 2008.
- [66] A. Nikolov, K. Talwar, and L. Zhang. The geometry of differential privacy: the sparse and approximate cases. *Symposium on Theory of Computing*, 2013.
- [67] K. Nissim, C. Orlandi, and R. Smorodinsky. Privacy-aware mechanism design. In *Association for Computing Machinery Conference on Electronic Commerce*, pages 774–789. 2012.
- [68] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the Association for Computing Machinery Symposium on Theory of Computing*, pages 75–84. 2007.

- [69] K. Nissim, R. Smorodinsky, and M. Tennenholtz. Approximately optimal mechanism design via differential privacy. In *Innovations in Theoretical Computer Science*, pages 203–213. 2012.
- [70] M. Pai and A. Roth. Privacy and mechanism design. *SIGecom Exchanges*, 2013.
- [71] R. Rogers and A. Roth. Asymptotically truthful equilibrium selection in large congestion games. arXiv preprint arXiv:1311.2625, 2013.
- [72] A. Roth. Differential privacy and the fat-shattering dimension of linear queries. In *Approximation, Randomization, and Combinatorial Optimization, Algorithms and Techniques*, pages 683–695. Springer, 2010.
- [73] A. Roth. Buying private data at auction: the sensitive surveyor’s problem. *Association for Computing Machinery SIGecom Exchanges*, 11(1):1–8, 2012.
- [74] A. Roth and T. Roughgarden. Interactive privacy via the median mechanism. In *Symposium on Theory of Computing ’10*, pages 765–774. 2010.
- [75] A. Roth and G. Schoenebeck. Conducting truthful surveys, cheaply. In *Proceedings of the ACM Conference on Electronic Commerce*, pages 826–843. 2012.
- [76] B. I. P. Rubinstein, P. L. Bartlett, L. Huang, and N. Taft. Learning in a large function space: Privacy-preserving mechanisms for svm learning. arXiv preprint arXiv:0911.5708, 2009.
- [77] R. Schapire. The boosting approach to machine learning: An overview. In D. D. Denison, M. H. Hansen, C. Holmes, B. Mallick, and B. Yu, editors, *Nonlinear Estimation and Classification*. Springer, 2003.
- [78] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 39:297–336, 1999.
- [79] R. E. Schapire and Y. Freund. *Boosting: Foundations and Algorithms*. MIT Press, 2012.
- [80] A. Smith and A. G. Thakurta. Differentially private feature selection via stability arguments, and the robustness of the lasso. In *Proceedings of Conference on Learning Theory*. 2013.
- [81] L. Sweeney. Weaving technology and policy together to maintain confidentiality. *Journal of Law, Medicines Ethics*, 25:98–110, 1997.
- [82] J. Ullman. Answering $n^{\{2+o(1)\}}$ counting queries with differential privacy is hard. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, *Symposium on Theory of Computing*, pages 361–370. Association for Computing Machinery, 2013.

- [83] L. G. Valiant. A theory of the learnable. *Communications of the Association for Computing Machinery*, 27(11):1134–1142, 1984.
- [84] S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.
- [85] D. Xiao. Is privacy compatible with truthfulness? In *Proceedings of the Conference on Innovations in Theoretical Computer Science*, pages 67–86. 2013.

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Privacy, Big Data, and the Public Good: Frameworks for Engagement

Edited by

Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum

Terms and conditions

for the provision of publications in PDF form for promotional and/or review and/or panel-symposium purposes

PDF versions of Cambridge publications are occasionally made available to authors or others for promotional and/or review and/or panel-symposium purposes.

PDFs are provided on the terms and conditions below.

- The PDF of the book is made available for the recipient's own use and reference, to inform the recipient of the content of the book.
- The PDF may not be copied by the recipient to other parties except to those in the recipient's employment, or who have a contractual relationship to the recipient, for direct promotional and/or review purposes.
- The PDF may not be altered in any way.
- The PDF may not be sold, nor may any charge be made for use of the PDF or of any output from it.
- The PDF must be destroyed by the recipient once it has been used for the purpose for which it was intended.

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Contents

Editors' Introduction

Part I. Conceptual Framework

1. Monitoring, Datafication, and Consent: Legal Approaches to Privacy in the Big Data Context

Katherine J. Strandburg

2. Big Data's End Run around Anonymity and Consent

Solon Barocas and Helen Nissenbaum

3. The Economics and Behavioral Economics of Privacy

Alessandro Acquisti

4. Changing the Rules: General Principles for Data Use and Analysis

Paul Ohm

5. Enabling Reproducibility in Big Data Research: Balancing Confidentiality and Scientific Transparency

Victoria Stodden

Part II. Practical Framework

6. The Value of Big Data for Urban Science

Steven E. Koonin and Michael J. Holland

7. Data for the Public Good: Challenges and Barriers in the Context of Cities

Robert M. Goerge

8. A European Perspective on Research and Big Data Analysis

Peter Elias

9. The New Deal on Data: A Framework for Institutional Controls

Daniel Greenwood, Arkadiusz Stopczynski, Brian Sweatt, Thomas Hardjono, and Alex Pentland

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

10. Engineered Controls for Dealing with Big Data

Carl Landwehr

11. Portable Approaches to Informed Consent and Open Data

John Wilbanks

Part III. Statistical Framework

12. Extracting Information from Big Data: A Privacy and Confidentiality Perspective

Frauke Kreuter and Roger Peng

13. Using Statistics to Protect Privacy

Alan F. Karr and Jerome P. Reiter

14. Differential Privacy: A Cryptographic Approach to Private Data Analysis

Cynthia Dwork

This is a preliminary version of the book Privacy, Big Data, and the Public Good: Frameworks for Engagement, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Contributors

Alessandro Acquisti

Heinz College

Carnegie Mellon University

5000 Forbes Avenue, HBH 2105C

Pittsburgh, PA 15213

Solon Barocas

Department of Media, Culture & Communication

New York University

East Building 7th Floor, 239 Green Street

New York, NY 10003

Cynthia Dwork

Microsoft Research Silicon Valley

1065 La Avenida

Mountain View, CA 94043

Peter Elias

University of Warwick

Warwick Institute for Employment Research

Room C0.03

Coventry CV4 7AL

England

Robert M. George

University of Chicago

Chapin Hall

1313 East 60th Street

Chicago, IL 60637

Daniel Greenwood

Human Dynamics Group

MIT Media Lab

77 Massachusetts Avenue, E15-384b

Cambridge, MA 20139

Thomas Hardjono

MIT Kerberos & Internet Trust Consortium

77 Massachusetts Avenue, W92-152

Cambridge, MA 01890

Michael J. Holland

Center for Urban Science + Progress, NYU

1 MetroTech Center, 19th Floor

Brooklyn, NY 11201

Alan F. Karr

National Institute of Statistical Sciences

19 T. W. Alexander Drive

P.O. Box 140006

Research Triangle Park, NC 27709-4006

Steven E. Koonin

Center for Urban Science + Progress, NYU

1 MetroTech Center, 19th Floor

Brooklyn, NY 11201

Frauke Kreuter

University of Maryland

1218 Lefrak Hall

College Park, MD 20742, USA

Carl Landwehr

George Washington University

1923 Kenbar Court

McLean, Virginia 22101

Helen Nissenbaum

Department of Media, Culture & Communication

New York University

East Building 7th Floor, 239 Green Street

New York, NY 10003

Paul Ohm

University of Colorado Law School

433 Wolf Law Building

401 UCB

Roger Peng

Johns Hopkins Bloomberg School of Public Health

615 N. Wolfe St.

Baltimore, MD 21205

Alex Pentland

Human Dynamics Group

MIT Media Lab

77 Massachusetts Avenue, E15-387

Cambridge, MA 20139

Jerome P. Reiter

Duke University

112A Old Chemistry Building

Durham, NC 27708-0251

Victoria Stodden

Columbia University

1255 Amsterdam Ave, 10th floor

New York, NY 10027

Arkadiusz Stopczynski

Human Dynamics Group

MIT Media Lab

77 Massachusetts Avenue, E15-386

Cambridge, MA 02139-4307

This is a preliminary version of the book **Privacy, Big Data, and the Public Good: Frameworks for Engagement**, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Katherine J. Strandburg
New York University School of Law
40 Washington Square South, 430K
New York, NY 10012

Brian Sweat
Human Dynamics Group
MIT Media Lab
77 Massachusetts Avenue, E15-384b
Cambridge, MA 20139

John Wilbanks
Sage Bionetworks and Kauffman Foundation
425 L Street NW #906
Washington DC 20001 USA

Editors' Introduction

Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum

Massive amounts of new data on human beings can now be accessed and analyzed. And the new “big data”¹ are much more likely to be harvested from a wide variety of different sources. Much has been made of the many uses of such data for pragmatic purposes, including selling goods and services, winning political campaigns, and identifying possible terrorists. Yet “big data” can also be harnessed to serve the public good in other ways: scientists can use new forms of data to do research that improves the lives of human beings; federal, state, and local governments can use data to improve services and reduce taxpayer costs and public organizations can use information to advocate for public causes, for example.

Much has also been made of the privacy and confidentiality issues associated with access. Statisticians are not alone in thinking that consumers should worry about privacy issues, and that an ethical framework should be in place to guide data scientists² - the European Commission and the US government have begun to address the problem (Elias and Greenwood, this volume). Yet there are many unanswered questions. What are the ethical and legal requirements for scientists and government officials seeking to use big data to serve the public good without harming individual citizens? What are the rules of engagement with these new data sources? What are the best ways to provide access while protecting confidentiality? Are there reasonable mechanisms to compensate citizens for privacy loss?

The goal of this book is to answer some of these questions. The book’s authors paint an intellectual landscape that includes the legal, economic, and statistical context necessary to frame the many privacy issues, including the value to the public of data access, clarifying personal data ownership questions, and raising issues of agency in personal data. The authors also identify core practical approaches that use new technologies to simultaneously maximize the utility of data access while minimizing information risk. As is appropriate for such a new and evolving field, each chapter also identifies important questions that require future research.

The work in this book is also intended to be accessible to an audience broader than those in the research and policy spheres. In addition to informing the public, we hope that the book will be useful to people trying to provide data access within confidentiality constraints in the roles as data custodians for federal, state, and local agencies, or decision makers on institutional review boards.

Historical and Future Use of Data for the Public Good

Good data are critically important for good public decisions. For example, national and international government policies depend on GDP estimates – indeed, international crises have been exacerbated when statistical agencies have cooked the data books³. Good data are also important for good science – as Daniel Kahneman famously pointed out, the first big breakthrough in our understanding of the mechanism of association was an improvement in a method of measurement (Kahneman, 2011).

Historically the leading producer of high quality data has been statistical government agencies engaged in collecting data through large scale statistically representative surveys. There are several reasons for this. One was the sheer scale of the necessary activity: generating representative samples required an expensive, constantly updated, population frame, and extensive investments in survey methodology, data storage, cleaning and dissemination. The second was that the public trusted the government to protect confidentiality, and statistical agencies invested heavily in the appropriate statistical disclosure limitation methodologies. The third was that the statistical agencies were seen to be objective, and not trying to sell a product. The Census Bureau's mission statement reflects all three of these reasons

“The Census Bureau's mission is to serve as the *leading source of quality* data about the nation's people and economy. We honor *privacy, protect confidentiality*, share our expertise globally, and conduct our work openly. We are guided on this mission by *scientific objectivity*, our strong and capable workforce, our devotion to research-based innovation, and our abiding commitment to our customers.”⁴ (emphases added).

The public good has clearly been served by the creation and careful dissemination of data by both the government and the research community. Of course, the nature of the data, as well as the dissemination modality, has evolved over time. The development and release of large scale public use datasets like the Current Population Survey, and later, the National Longitudinal Surveys of Youth and the Panel Study of Income Dynamics and the German Socio-Economic Panel transformed our understanding of labor markets, while protecting respondent confidentiality. The development of large scale administrative datasets and their access through secure data enclaves, have lowered costs, increased sample size and reduced respondent burden (Groen, 2012), as well as created completely new classes of information (Abowd & Vilhuber, 2011).

Big data, which we use here as shorthand for the data necessary to support new types of data intensive research (Hey, Tansley, & Tolle, 2009), have the potential to have an even more profound effect on the public good. Information derived from big data is likely to be one of the foundational elements for running future society, by for example, generating real time information about economic and social activity, or by generating new insights into human behavior (Einav & Levin, 2013). Yet the pathway to developing this foundation is not clear, since experience provides little guidance. The data that are

currently used to inform decisions - survey and administrative data – have benefitted from decades of statistical research, as well as clear rules defining ownership and responsibility. Statistical agencies, the major custodians, have developed clear pathways to both protect and access the data. By contrast, the value of big data to inform evidence based policy is still being established, and the ownership of big data, typically distributed across multiple entities, is not as well defined. Yet big data have many elements of a common natural resource, and sensible rules should be developed in order to avoid a tragedy of the commons, and create an commonly pooled resource for improving scientific understanding for the public good (Ostrom, 1990).

Privacy, Big Data, and the Public Good: The Contributions of This Book

The vast changes in the data world have brought with them changes in the role of traditional data collectors and producers. Data on human beings are now much less likely to be purposively collected by researchers and government agencies, and there is thus less knowledge on how to protect privacy. There are serious consequences: the lack of dissemination experience of non-governmental collectors can lead to massive privacy breaches (the 2006 AOL data release is but one of the most famous examples⁵). Even worse, if no dissemination is allowed, the quality of privately held data is largely unknown⁶ absent detailed researcher inspection and validation. Similarly, Institutional Review Boards with few reference guidelines are likely to slow down or prevent research on human subjects with complex data.⁷

Because of the importance of the topic, there is a rich and vibrant literature; authors in this book have provided, for the first time in one place, an accessible summary of existing research in many of the important aspects. They have also identified practical suggestions – to help guide practitioners and Institutional Review Boards – as well as identified important areas of future research.

In opening the conceptual framework section, *Katherine Strandburg* argues that the acquisition, transfer and aggregation of data on a massive scale for data mining and predictive analysis raises questions that simply are not answered by the paradigms that have dominated privacy law to date. She develops a taxonomy of current United States privacy law and uses that taxonomy to elucidate the mismatch between current law and big data privacy concerns. *Barocas and Nissenbaum* argue that big data involves practices that have radically disrupted entrenched information flows. From modes of acquiring to aggregation, analysis, and application, these disruptions affect actors, information types, and transmission principles. . Privacy and big data are simply incompatible and the time has come to reconfigure choices that we made decades ago to

enforce certain constraints. They argue that it is time for the background of rights, obligations, and legitimate expectations to be explored and enriched so that notice and consent can do the work for which it is best suited. *Acquisti* discusses how the economics and behavioral economics of privacy can be applied to the investigation of the implications of consumer data mining and business analytics. An important insight is that personal information, when shared, can become a public good whose analysis can reduce inefficiencies and increase economic welfare; when abused, it can lead to transfer of economic wealth from data subjects to data holders. The interesting economic question then becomes, who will bear the costs if privacy enhancing technologies become more popular in the age of big data: data subjects (whose benefits from business analytics and big data would shrink with the amount of information they share), data holders (who may face increasing costs associated with collecting and handling consumer data), or both? There are some practical implications. *Ohm* provides an overview of how information privacy laws regulate the use of Big Data techniques, if at all. He discusses whether these laws strike an appropriate balance between allowing the benefits of Big Data and protecting individual privacy, and if not, how might the laws be better extended and amended to better strike this balance. He notes that most information privacy law focuses on collection or disclosure and not use. Once data has been legitimately obtained, few laws dictate what may be done with the information. The chapter proposes five general approaches for change. *Stodden* motivates the scientific rationale for access to data and computational methods to enable the verification and validation of published research findings. She describes the legal landscape in the context of big data research and suggests two guiding principles to facilitate reproducibility and re-use of research data and code within and beyond the scientific context.

Koonin and Holland open the section on practical frameworks by addressing the motivations for new urban science, and the value for cities – particularly with respect to analysis of the infrastructure, the environment and the people. They discuss the key technical issues necessary to build a data infrastructure for curation, analytics, visualization, machine learning, data mining, as well as modeling and simulation to keep up with the volume and speed of data. *Goerge* uses an example of the creation of a data warehouse which links data on multiple services provided by the public sector to individuals and families as a way to highlight both the barriers to and opportunities for cities to use data. He identifies the key issues that need to be addressed - what data to develop and access from counties, states, the federal government and private sources; how to develop the capacity to use data; how to present data and be transparent; and how best to keep data secure so that individuals and organizations are protected – as well as the key barriers. *Elias* provides a broader perspective than simply the U.S by noting that many of the legal and ethical issues associated with big data have wider relevance. Much can be learned from examining the progress that has been made across Europe to develop

a harmonised approach to legislation designed to provide individuals and organisations with what has become known as the ‘right to privacy’. The legislative developments have had and are continuing to have substantial impact on cross-border access to micro-data for research purposes; that impact is also examined.

Greenwood, Stopczynski, Sweatt, Hardjono and Pentland explore the emergence of the Big Data society, arguing that the “personal data sector” of the economy needs productive collaboration between the Government, the private sector, and the citizen to create new markets – just like the automobile and oil industries did in prior centuries. They sketch a model of data access which is governed by Living Informed Consent, where the user is entitled to know what data is being collected about her by which entities, empowered to understand the implications of data sharing, and finally put in charge of the sharing authorizations. They envision the establishment of a New Deal on Data, grounded in principles, such as the opt-in nature of data provision, the boundaries of the data usage, and parties accessing the data. *Landwehr* takes a very pragmatic approach. He notes that regardless of what data policies have been agreed to, access must be allowed through controls engineered into the data infrastructure. Without sound technical enforcement, incidents of abuse, misuse, theft of data and even invalid scientific conclusions based on undetectably altered data can be expected. He discusses what features those access controls might have – delineating the characteristics of subjects, objects, and access modes and notes that areas of research that could change the picture in the future include advances in practical cryptographic solutions to computing on encrypted data, which could reduce the need to trust hardware and system software. Advances in methods for building systems in which information flow, rather than access control, is the basis for policy enforcement could also open the door for better enforcement of comprehensible policies. *Wilbanks* is similarly practical. He provides an overview of frameworks that are available to permit data reuse and discusses how legal and technical systems can be structured to allow people to donate their data to science.

He argues that traditional frameworks to permit data reuse have been left behind by the mix of advanced techniques for re-identification and cheap technologies for the creation of data about individuals. He provides an overview of the approaches developed in technological and organizational systems to “create” privacy where it has been eroded while allowing data reuse, but also discusses a new approach of “radical honesty” towards data contribution and the development of “portable” approaches to informed that could potentially support a broad range of research without the unintended fragmentation of data created by traditional consent systems

Kreuter and Peng open the statistical section with a discussion of the new statistical challenges associated with inference in the context of big data. It pays particular attention to the importance of providing access to researchers in order to both develop new statistical approaches to address the issues of coverage and non response, the need for

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

transparency in the data collection and data analysis as well as to expand the research in linkage and matching to evaluate the quality of the linked sources, including research on the missing elements. *Karr and Reiter* explore the interactions between data dissemination, big data and statistical inference. They identify a set of lessons that stewards of big data can learn from statistical agencies' experiences about the measurement of disclosure risk and data utility. Their conclusion is that the sheer scale and potential use of big data will require that analysis be taken to the data rather than the data to the analyst or the analyst to the data. They argue that a viable way forward for big data access is an integrated system including (i) unrestricted access to highly redacted data, most likely some version of synthetic data, followed with (ii) means for approved researchers to access the confidential data via remote access solutions, glued together by (iii) verification servers that allow users to assess the quality of their inferences with the redacted data so as to be more efficient with their use (if necessary) of the remote data access. *Dwork* concludes with a vision for the future. She shows how differential privacy provides a mathematically rigorous theory of privacy, a theory amenable to measuring (and minimizing) cumulative privacy loss, as data are analyzed and re-analyzed, shared and linked. There are tradeoffs - differential privacy requires a new way of interacting with data, in which the analyst only accesses data through a privacy mechanism, and in which accuracy and privacy are improved by minimizing the viewing of intermediate results. But the approach provides a measure that captures cumulative privacy loss over multiple releases; it offers the possibility that data usage and release could be accompanied by publication of privacy loss

Thanks As with any book, we have benefitted enormously from the support and help of many people. Our editor, Diana Gillooly has worked tirelessly and efficiently at all phases – going well beyond the call of duty. Our referees took time out of their busy schedules to give thoughtful, constructive guidance to the authors. They include Micah Altman, Mike Batty, Jason Bobe, Aleksandra Bujnowska, Fred Conrad, Josep Domingo-Ferrer, Stephanie Eckman, Mark Elliot, Martin Feldkircher, Simson Garfinkel, Bob Goerge, Eric Grosse, Patricia Hammar, David J. Hand, Dan Harnesk, Kumar Jayasuriya, Gary King, Frauke Kreuter, Tom Kvan, Bethany Letalier, William Lowrance, Lars Lyberg, Tim Mulcahy, Kobbi Nissim, Onora O'Neill, Kathleen Perez Lopez, Carlo Reggiani, Jerome H. Reichman, Guy Rothblum, Subu R. Sangameswar, Fred Schneider, Aleksandra Slavkovic, Tom Snijders, Omer Tene, Vincenc Torra, Paul Uhlir, Richard Valliant and Felix Wu.

The book was financially supported by New York University, through the Center for Urban Science and Progress. Konstantin Baetz, Veronika Zakrocki, Felicitas Mittereder, Reinhard Sauckel and Dominik Braun provided superb administrative support, Shaylee Nielson and Christian Rafidi at CUSP made sure editors and authors were paid, Mark

This is a preliminary version of the book Privacy, Big Data, and the Public Good: Frameworks for Engagement, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Righter in NYU General Counsel's office provided legal assistance and Ardis Kadiu, Spark451 gave assistance with our website. The book was also supported by the Privacy and Confidentiality SubCommittee of the American Statistical Association, and the Institute for Employment Research of the German Federal Employment agency.

We are also very grateful to the Steve Pierson, American Statistical Association; Kim Alfred, CUSP, and Kelly Berschauer, Microsoft Research for their help with outreach to key stakeholders.

Notes

¹ There are many definitions; one fairly representative version says the term “big data” “.. refers to large, diverse, complex, longitudinal, and/or distributed data sets generated from instruments, sensors, Internet transactions, email, video, click streams, and/or all other digital sources available today and in the future”¹; another uses the velocity, volume and variety rubric “data is now available faster, has greater coverage and scope, and includes new types of observations and measurements that previously were not available”(Einav & Levin, 2013)

² <http://blog.revolutionanalytics.com/2013/09/statistician-survey-results.html>

³ <http://www.ft.com/cms/s/0/82b15932-18fe-11e1-92d8-00144feabdc0.html#axzz2g7W3pWOJ>

⁴ <http://www.census.gov/aboutus/#>

⁵ http://en.wikipedia.org/wiki/AOL_search_data_leak

⁶ See, for example, <http://www.theguardian.com/commentisfree/2013/may/04/adp-forecasting-monthly-bls-jobs-reports>

⁷ http://sites.nationalacademies.org/DBASSE/BBCSS/CurrentProjects/DBASSE_080452

References

Abowd, J. M., & Vilhuber, L. (2011). National estimates of gross employment and job flows from the Quarterly Workforce Indicators with demographic and industry detail. *Journal of Econometrics*, 161(1), 82–99. Retrieved from <http://ideas.repec.org/a/eee/econom/v161y2011i1p82-99.html>

Einav, L., & Levin, J. D. (2013). The Data Revolution and Economic Analysis. *National Bureau of Economic Research Working Paper Series*, No. 19035. Retrieved from <http://www.nber.org/papers/w19035>

This is a preliminary version of the book Privacy, Big Data, and the Public Good: Frameworks for Engagement, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Groen, J. (2012). Sources of Error in Survey and Administrative Data: The Importance of Reporting Procedures. *Journal of Official Statistics*, 173–198.

Hey, T., Tansley, S., & Tolle, K. (2009). *The Fourth Paradigm: Data Intensive Scientific Discovery*. Microsoft Research.

Kahneman, D. (2011). *Thinking Fast and Slow*. Farrar, Straus and Giroux.

Ostrom, E. (1990). GOVERNING THE COMMONS The Evolution of Institutions for Collective Action. Cambridge University Press, Cambridge UK.

Part I

Conceptual Framework

This section begins by considering the existing legal constraints on the collection and use of big data in the privacy and confidentiality context. It then identifies gaps in the current legal landscape and issues in designing a coherent set of policies that both protect privacy and yet permit the potential benefits that come with big data. Three themes emerge in this section: that the concepts used in the larger discussion of privacy and big data require updating; how we understand and assess harms from privacy violations needs updating; and that we must rethink established approaches to managing privacy in the big data context.

The notion of “big data” is interpreted as a change in paradigm, rather than solely a change in technology. This illustrates the first central theme of this section. Baracas and Nissenbaum define big data as a “paradigm, rather than a particular technology,” while Strandburg differentiates between collections of data, and collections of data that have been “datafied,” that is, “aggregated in a computationally manipulable format.” She claims that such datafication is a key step in heightening privacy concerns and creating a greater need for a coherent regulatory structure for data acquisition. Traditional regulatory tools for managing privacy, notice and consent, have failed to provide a viable market mechanism allowing a form of self regulation governing industry data collection. Strandburg elucidated the current legal restrictions and guidance on data collection in the industrial setting, including the Fair Information Practice Principles (FIPPs) dating from 1973 and underlying the Fair Credit Reporting Act (FCRA) from 1970 and the Privacy Act from 1974. Strandburg advocates for a more nuanced assessment of tradeoffs in the big data context, moving away from individualized assessments of the costs of privacy violations. The collection of private data for monitoring purposes should have its privacy law strengthened, in particular a substantive distinction should be made based on datafication and the repurposing of data that was collected as a byproduct of providing services. Additionally, she suggests taking a substantive approach to the ideas of notice and consent in particular to clarify their meaning for large entities.

The inadequacy of assessment of harm from privacy is another major theme of this section, extending it from the level of the individual to that of groups or classes, and even society as a whole. Acquisti notes that this requires a greater understanding of the breadth of type of privacy breaches, of the nature of harm as diffused over time, and an improved valuation of privacy in the big data context. Consumers may value their own privacy in variously flawed ways. They may have incomplete information for example, or an overabundance of information rendering processing impossible, or use heuristics that systematize deviations for rational decisions making. Acquisti notes that privacy

protection can both potentially increase and decrease economic efficiency in the marketplace, and that deriving benefits from big data may not conflict with benefits from assuring privacy protection.

In order to address these issues, several authors ask us to rethink traditional approaches to privacy. This is the third overarching theme of this section. Baracas and Nissenbaum argues that the concepts of anonymity and informed consent do not create solutions to the privacy issue. As datasets become increasingly linked, anonymity is largely impossible to guarantee in the future. This also implied that it is impossible to truly give informed consent, since we cannot, by definition, know what the risks are from revealing personal data either for individuals or for society as a whole.

The use of privately collected data is largely unregulated. Ohm describes the few regulations that do apply (such as the Health Information Portability and Accountability Act (HIPAA), the Privacy Act, and the Fair Credit Reporting Act (FCRA)) and explain that the United States employs a “sectoral” approach to privacy regulation, in that different economic areas have separate privacy laws. Ohm also calls into question the traditional notion of notice in the case of big data. To whom are you to give notice, and for what? The results of big data analysis can be unpredictable and sometimes unexplainable, adding another reason it is difficult to assess privacy risks accurately in the big data context.

Ohm advocates for a new conceptualization of legal policy regarding privacy in the big data context that uses five guiding principles for reform: 1) that rules take into account the varying levels of inherent risk to individuals across different datasets, 2) traditional definitions of personally identifiable information need to be rethought, 3) regulation has a role in creating and policing walls between datasets, 4) those analyzing big data must be reminded, with the frequency in proportion to the sensitivity of the data, that they are dealing with people, and 5) the ethics of big data research must be an open topic for continual reassessment.

In the final chapter, Stodden focuses on this theme of research integrity in the big data context. She notes a conflict between research requirements regarding the replication of computational results which can require data access, and traditional methods of privacy protection via sequestration. She advocates establishing “middle ground” solutions whenever possible that maximize verification of computational findings, while taking into account any legal and ethical barriers. Permitting authorized researchers access to confidential data within a “walled garden” can increase the ability of others to independently replication big data findings, for example. Two principles are presented to help guide thinking regarding reproducibility and verification in big data research: the Principle of Scientific Licensing; and the Principle of Scientific Data and Code Sharing. That is, in the scientific context, legal encumbrances to data sharing for purposes of independent verification should be minimized wherever possible, and access to the data

This is a preliminary version of the book Privacy, Big Data, and the Public Good: Frameworks for Engagement, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

and methods associated with published findings should be maximized subject to legal and ethical restrictions.

Chapter 1

Monitoring, Datafication, and Consent: Legal Approaches to Privacy in the Big Data Context

Katherine J. Strandburg

Knowledge is power. ‘Big data’ has great potential to benefit society. At the same time, its availability creates significant potential for mistaken, misguided, or malevolent uses of personal information. The conundrum for law is to provide space for big data to fulfill its potential for societal benefit, while protecting citizens adequately from related individual and social harms. Current privacy law evolved to address different concerns and must be adapted to confront big data’s challenges. This chapter addresses only one aspect of privacy law: the regulation of private sector acquisition, aggregation, and transfer of personal information.¹ It provides an overview and taxonomy of current law, highlighting the mismatch between current law and the big data context, with the goal of informing the debate about how to bring big data practice and privacy regulation into optimal harmony.

Part I briefly describes how privacy regulation in the United States has evolved in response to a changing technological and social milieu. Part II introduces a taxonomy of privacy laws relating to data acquisition, based on the following features: (1) whether the law provides a rule- or a fact-based standard; (2) whether the law is substantive or procedural, in a sense defined below; and (3) which mode(s) of data acquisition are covered by the law. It also argues that the recording, aggregation, and organization of information into a form that can be used for data mining, here dubbed ‘datafication’, has distinct privacy implications that often go unrecognized by current law. Part III provides a selective overview of relevant privacy laws in light of that taxonomy. Section A discusses the most standards-like legal regimes, such as the privacy torts, for which determining liability generally involves a fact-specific analysis of the behavior of both data subjects and those who acquire or transfer the data (‘data handlers’). Section B discusses the Federal Trade Commission’s (FTC’s) ‘unfair and deceptive trade practices’ standard,² which depends on a fact-specific inquiry into the behavior of data handlers, but makes general assumptions about data subjects. Section C discusses rule-like regimes, such as the Privacy Rule³ of the Health Insurance Portability and Accountability Act⁴ (HIPAA Rule). Part IV points out some particular features of the mismatch between current law’s conceptualization of the issues and the big data context, using the taxonomy developed in Part II as an aid to the analysis. It then makes several suggestions about how to devise a better fit.

I. The Evolution of U.S. Privacy Law

Outside of the law enforcement context, privacy law was erected on the foundation of Warren and Brandeis's famous 1890 article, *The Right to Privacy*.⁵ The privacy torts built on that foundation were concerned primarily with individualized harms of emotional distress, embarrassment, and humiliation arising out of 'intrusion upon seclusion' or 'public disclosure of private facts'. Privacy law also aimed to protect confidentiality in certain kinds of relationships, often involving professional expertise, in which information asymmetry and power imbalances create a potential for exploitation. These torts provide compensation for individualized injuries caused by egregious deviations from social norms. In principle, and often in fact, the tort paradigm employs a highly contextualized analysis of the actions of plaintiff and defendant and the relationship between them.

In the 1970s, the development of digital computers and the increasing complexity of the administrative state led to an expansion in 'computer-based record-keeping operations' by governments and certain other large institutions, such as banks. This expansion raised fears of misuse, unfairness, lack of transparency in decision making, and chilling of autonomous behavior distinct from the concerns about emotional distress and reputation at the heart of the privacy torts. Fair Information Practice Principles (FIPPs), which have become the mainstay of data privacy law, were developed during this period as an approach to those issues. The Fair Credit Reporting Act (FCRA),⁶ adopted in 1970, and the Privacy Act of 1974,⁷ regulating data use by government agencies, were based on FIPPs.

A set of five FIPPs were proposed in 1973 in a report commissioned by the Department of Health, Education, and Welfare (HEW Report):

1. *There must be no personal data record-keeping systems whose very existence is secret.*
2. *There must be a way for a person to find out what information about the person is in a record and how it is used.*
3. *There must be a way for a person to prevent information about the person that was obtained for one purpose from being used or made available for other purposes without the person's consent.*
4. *There must be a way for a person to correct or amend a record of identifiable information about the person.*
5. *Any organization creating, maintaining, using, or disseminating records of identifiable personal data must assure the reliability of the data for their intended use and must take precautions to prevent misuses of the data.*⁸

These principles, along with three sets of related ‘safeguard requirements’,⁹ attempted to cope with the scale of data collection by substituting transparency and consent for the individualized fact-specific approach of the privacy torts. The HEW Report recognized the difficulty of legislating substantive rules in light of the “enormous number and variety of institutions dealing with personal data,” arguing that institutions should be “deterred from inappropriate practices, rather than being forced by regulation to adopt specific practices.”¹⁰

Another important version of FIPPs was formulated by the OECD in 1980 (OECD FIPPs), articulating eight principles, which expanded on the HEW FIPPs and include a “Collection Limitation Principle” that there “should be limits to the collection of personal data,” a “Data Quality Principle” that data collected should be “relevant to the purposes for which they are to be used” and “accurate, complete and kept up-to-date,” and a “Purpose Specification Principle” that purposes should be specified in advance and “subsequent use limited to the fulfillment of those purposes or such others as are not incompatible with those purposes.”¹¹

At the turn of the 21st century, the FTC, which has taken the primary role in commercial privacy regulation, recommended a rather slimmed down set of FIPPs for online privacy:

- (1) *Notice – Web sites would be required to provide consumers clear and conspicuous notice of their information practices....*
- (2) *Choice – Web sites would be required to offer consumers choices as to how their personal identifying information is used beyond the use for which the information was provided (e.g., to consummate a transaction)....*
- (3) *Access – Web sites would be required to offer consumers reasonable access to the information a Web site has collected about them, including a reasonable opportunity to review information and to correct inaccuracies or delete information.*
- (4) *Security – Web sites would be required to take reasonable steps to protect the security of the information they collect from consumers.*¹²

In practice, outside of a few sectors, such as health care, the FIPPs approach in the United States has been whittled down to a focus on procedures for ensuring notice and consent. What explains this progression? Cheap and ubiquitous information technology makes large-scale data aggregation and use possible for a wide variety of private sector entities in addition to the government agencies and large institutions, such as banks, that were the subject of privacy concerns in the 1970s and 1980s. The resulting expansion of private sector data collection and use has competing implications for privacy regulation. On the one hand, the acquisition and use of an increasing quantity and scope of personal

information by an increasingly large and various set of private entities heightens privacy concerns relating to data security and breach, accountability and transparency, and unpredictable data uses. On the other hand, substantive regulation of a large and diverse array of private sector entities is politically controversial, regulations that effectively span the field of data handlers are hard to devise, and monitoring the data practices of such a large number of players is difficult and expensive.

As a result, the trend was to assume (or at least hope) that notice and consent would provide a market mechanism for encouraging industry self-regulation of data privacy. In light of recent acceleration in data collection and the development of big data approaches to mining aggregated data, it now is widely recognized that the notice and consent paradigm is inadequate to confront the privacy issues posed by the big data explosion.¹³ The notice and consent paradigm assumes that citizens are able to assess the potential benefits and costs of data acquisition sufficiently accurately to make informed choices. This assumption was something of a legal fiction when applied to data collected by government agencies and regulated industries in the 1970s. It is most certainly a legal fantasy today, for a variety of reasons including the increasing use of complex and opaque predictive data-mining techniques, the interrelatedness of personal data, and the unpredictability of potential harms from its nearly ubiquitous collection.¹⁴

II. A Taxonomy of Privacy Laws Relevant to Data Acquisition

As mentioned in the introduction, it is useful to organize this selective overview of U.S. privacy law relating to data acquisition and transfer according to a taxonomy focusing on three characteristics: (A) whether the law takes a rule-like or fact-specific standards-like approach; (B) whether the law regulates substance or procedure, in a sense defined below; and (C) what modes of data acquisition are covered by the law. While these distinctions are not bright lines, an approximate categorization is useful in analyzing the uncomfortable fit between current privacy law and big data projects.

A. Rules or Standards

Privacy law regimes vary according to the extent to which they impose flexible fact-specific standards or generally applicable rules, but can be divided roughly into three groups. Laws in the first group, such as the torts of intrusion upon seclusion and public disclosure of private facts, assess liability using standards that depend on detailed fact-intensive inquiry into the activities of both subjects and acquirers of personal information. Laws in the second group, such as Section 5 of the FTC Act, employ standards-based assessment of the activities of data holders, while relying on presumptions about data subjects. Laws in the third group, such as the HIPAA Rule, mandate compliance with rules.

Trade-offs between rules and standards are endemic to law.¹⁵ Ex ante, rules provide clearer direction for behavior, while standards provide leeway so behavior can be tailored more optimally to specific contexts. Ex post, rules are cheaper to enforce and leave less room for bias, while standards leave more discretion for crafting context-sensitive and fair outcomes. These tensions are dynamic. Because of these trade-offs, courts and legislatures often attempt to draw at least some bright lines (such as a line between private and public) to guide the application of standards, while rule-like regimes often become complex (or even byzantine) as lawmakers try to anticipate all relevant contingencies.

B. Substance or Procedure

Privacy law also grapples with trade-offs between substantive and procedural regulation. Compliance with substantive regulation, as I use the term here, is determined by asking: Was it legally acceptable for Data Handler A to acquire or transfer this information about Data Subject B in this situation and in this way? Compliance with procedural regulation, on the other hand, is determined by asking: Did Data Handler A follow the appropriate procedures in acquiring or transferring information about Data Subject B?

Though substantive regulation is preferable when goals are well defined and outcomes are observable by enforcement agencies, procedural regulation is advantageous in some situations. Regulated entities may be better situated than lawmakers, by virtue of expertise or superior information, to make substantive determinations, especially when circumstances are expected to evolve, perhaps by technological advance, but data holders may have incentives to use their discretion in socially undesirable ways. Procedural regulation may be used to limit their ability to do so. Substantive outcomes may be difficult to specify or to observe. Procedural regulations may structure behavior so as to make desirable outcomes more likely and may make compliance easier to audit. Procedural regulation also may help to prevent negligence and mistake. Procedural and substantive approaches often are combined in privacy regulation. For example, some laws require different procedures based on substantive distinctions between data-handling entities, types of data, or purposes of data acquisition.

C. Modes of Data Acquisition

There are three basic avenues for acquiring big data: monitoring, acquisition as a byproduct of another activity, and transfer of pre-existing information. Monitoring, as I use the term here, applies broadly to the recording of information in plain view and information acquisition by means such as wiretapping and spyware. Acquisition as a byproduct of another activity is common for service providers such as telecommunications providers, utilities, online websites and apps, search engines, and governments.

However it is acquired, if information is to be used in a big data project it must be recorded, quantified, formatted, and stored digitally to make it usable for computational knowledge discovery. Note that what I will call ‘datafication’ is distinct from digitization. Cellphone photos are digital, but they are not datafied unless they are aggregated in a computationally manipulable format. Datafication has independent privacy implications because recording and organizing information changes the uses to which it can be put, both for good and for ill.¹⁶ Importantly, because computation methods are continually developed and refined, datafication is likely to open the door to uses that were not feasible (and hence obviously not anticipated) at the time the data was acquired.

To illustrate the role of datafication in monitoring, consider video surveillance. Without cameras, the only record of what happens on city streets is in the minds of the human beings who pass by. Those memories are scattered, degrade quickly, may be inaccurate, and are very costly to aggregate and to search in retrospect, making it difficult for police to capture burglars, muggers, and murderers. Installing surveillance cameras watched by security guards provides more focused attention on what is occurring in view of the camera, making it more likely that a given event will be noticed and recalled. Security cameras also may deter crime (at least in their vicinity). Adding an analog video recorder makes a more accurate reconstruction of events taking place in view of the camera possible and decreases the cost of surveillance. Still, the difficulty of manually reviewing videotape limits the circumstances under which tapes are likely to be reviewed and makes aggregation of information from different cameras difficult. The storage costs of archiving videotape also constrain how far back one can look. Substituting a digital recorder increases the efficiency of manual review, while storage and transmission become easier and cheaper, meaning that recordings are likely to be archived for longer periods.

As each of these changes is made, monitoring becomes more efficient and effective for law enforcement purposes. The potential for privacy invasion also increases. A security guard might happen to see someone she knows heading into a psychiatrist’s office. A building employee might review videotapes to see who recently visited the psychiatrist or search through digital archives to see whether a particular person visited the psychiatrist five years ago. But digital recording also facilitates datafication – long-term storage in a format that is searchable, computationally manipulable, and may be aggregated with information from other security cameras. Datafication opens up the potential for uses that may have been unanticipated or even technologically infeasible at the time of collection and are qualitatively different from the original purposes of the surveillance. For example, the development of facial recognition algorithms might make it possible to aggregate the data from surveillance cameras to track particular individuals’ movements, to investigate correlations between the places people go, or to combine the video surveillance data with other data to create profiles of the types of people who visit

psychiatrists. Some of these uses may be socially beneficial and some socially harmful on balance. The data can be used to track a political activist or ex-lover as well as a robber fleeing the scene of a mugging. Once datafied, the information can be used by law enforcement officials, researchers, or hackers if they have access. The point thus is that datafication heightens privacy concerns and changes the trade-offs involved in monitoring and its regulation.

When service providers acquire information as a byproduct of providing service, there are several conceptually distinct possibilities. Data may be (1) acquired for service provision but not datafied; (2) acquired, datafied, and used only for service provision; (3) acquired for service provision but datafied for other purposes; or (4) acquired and datafied for service provision but repurposed for other uses, perhaps by transferring it to third parties. Service providers also can leverage their access to their users' computers, mobile phones, or other property to monitor information that would not be acquired as a byproduct of providing service.

For example, consider a provider of a subscription streaming video service. The provider will obtain subscription and payment information as a byproduct of providing its services and might datafy that information to streamline its services. It might also repurpose that information by using it to advertise its own other services or transferring it to third parties for advertising purposes. The provider also will necessarily receive and process information about which videos are requested by which customers, when they are streamed, and so forth. It may have no need to datafy that information to provide the service, but might choose to do so in order to use or transfer the data for some other purpose, such as creating advertising profiles. If the service provider uses cookies to record what webpages its customers visits and what ads they click on, it is no longer obtaining information as a byproduct of service provision. It is engaging in monitoring.

As a normative matter we might want the law to distinguish carefully between these different modes of data acquisition. As we shall see in Part III, current law has been slow to recognize these distinctions, especially in the service provider context.

III. Selective Overview of Privacy Laws

This section considers a sampling of important U.S. privacy laws in light of the taxonomy developed in Part II. Section A deals with legal regimes that rely primarily on standards based on fact-specific analysis of the behavior of both data subjects and data handlers. Section B deals with standards based primarily on data handler behavior, relying on general assumptions about data subjects. Section C deals with legal regimes that employ a mostly rule-like approach. Of course, these categorizations are rough approximations, since the mixture of rules and standards can vary in many different ways.

A. Two-Sided Standards

Privacy laws that apply two-sided fact-specific standards include the intrusion upon seclusion tort and California's state constitutional right to privacy. Two-sided fact-specific standards are most effectively applied in situations involving specific and traceable individualized harms. Like torts more generally, privacy laws employing two-sided standards have difficulty handling small widespread harms, probabilistic harms, and harms that are difficult to trace to a particular relationship, transaction, or incident.

1. The Intrusion upon Seclusion Tort

The intrusion upon seclusion tort is a substantive legal regime that has been employed primarily to regulate monitoring. Privacy torts are usually creatures of state common law (though sometimes they are codified by state legislatures). They were strongly influenced by William Prosser, author of a famous torts treatise. During the first half of the 20th century,¹⁷ Prosser studied and attempted to systematize the common law of invasion of privacy that had developed in the wake of Warren and Brandeis's 1890 article. Prosser's definitions of the privacy torts were incorporated into the 1967 draft of the Second Restatement of Torts, for which he was the chief reporter.¹⁸ Restatements attempt to guide and structure common law discretion, thus importing rule-like aspects into the analysis. Though states' adoption and interpretation of the privacy torts vary, the Second Restatement's definitions have been adopted in most states that recognize the torts.

This discussion focuses on the tort of intrusion upon seclusion, which is most relevant to data acquisition (though the tort of public disclosure of private facts might also apply to data acquisition in some circumstances). The Second Restatement defines intrusion upon seclusion as follows:

One who intentionally intrudes, physically or otherwise, upon the solitude or seclusion of another or his private affairs or concerns, is subject to liability to the other for invasion of his privacy, if the intrusion would be highly offensive to a reasonable person.¹⁹

An intrusion claim thus requires (1) an actionable intrusion that is (2) highly offensive.²⁰

a. Actionable Intrusion As might be expected from its definition, actionable intrusion traditionally involves monitoring. The Restatement's definition of actionable intrusion seeks to draw a rule-like line between protectable "solitude or seclusion" and "private affairs and concerns" and unprotectable public activity. The Restatement's examples of actionable intrusion mostly involve traditionally "private" zones, such as the home, mail or telephone communications, and financial matters:

The invasion may be by physical intrusion into a place in which the plaintiff has secluded himself, as when the defendant forces his way into the plaintiff's room in a hotel or insists over the plaintiff's objection in entering his home. It may also be by the use of the defendant's senses, with or without mechanical aids, to oversee or overhear the plaintiff's private affairs, as by looking into his upstairs windows with binoculars or tapping his telephone wires. It may be by some other form of investigation or examination into his private concerns, as by opening his private and personal mail, searching his safe or his wallet, examining his private bank account, or compelling him by a forged court order to permit an inspection of his personal documents.²¹

The Restatement also describes what is *not* actionable because it can be observed in public:

[T]here is no liability for the examination of a public record concerning the plaintiff.... Nor is there liability for observing him or even taking his photograph while he is walking on the public highway.... Even in a public place, however, there may be some matters about the plaintiff, such as his underwear or lack of it, that are not exhibited to the public gaze.²²

Despite this attempt to draw a line, the Restatement's commentary recognizes the importance of specific facts about the social context, relationships between the parties, and means of intrusion in determining whether an actionable intrusion has occurred. Thus, the Restatement distinguishes "calling [an individual] to the telephone on one occasion or even two or three, to demand payment of a debt"²³ from repeated phone calls "at meal times, late at night and at other inconvenient times."²⁴ If "the telephone calls are repeated with such persistence and frequency as to amount to a course of hounding the plaintiff," the caller may be liable.²⁵ By referring to the use of binoculars, telephone taps, and forged court orders, the Restatement also recognizes that the *means* of intrusion must be taken account in determining whether intrusion has occurred.

The case law further illustrates the doubly fact-specific nature of the intrusion definition. There is no bright line rule, for example, as to whether plaintiffs can complain of intrusion upon seclusion in the workplace. The assessment of actionable intrusion may depend on factors such as the degree to which the location was open to others, the use of deception, hidden cameras, or microphones, and the purpose of the intrusion. For example, spying on an employee from a space above the restroom²⁶ and secretly videotaping conversations between telephone psychics for television broadcast in a shared workplace have been deemed actionable.²⁷ On the other hand, intrusive investigations are sometimes justified if they are tailored to an appropriate purpose. Thus

in *Hernandez v. Hillsides, Inc.*, the employer invaded privacy expectations by “secretly install[ing] a hidden video camera that was both operable and operating (electricity-wise), and that could be made to monitor and record activities inside plaintiffs’ office, at will, by anyone who plugged in the receptors, and who had access to the remote location in which both the receptors and recording equipment were located” in order to investigate use of its computers for viewing pornography. Nonetheless, the invasion was not actionable because the employer had aimed the camera at a particular computer, controlled access to the surveillance equipment, and operated the equipment only during limited times.²⁸ Similarly, while drug testing of employees generally has not been deemed an intrusion, courts have found particular methods of collecting urine specimens intrusive.²⁹

Many intrusion upon seclusion cases involve journalists or researchers. In such cases, First Amendment values often outweigh claims to intrusion, depending upon the nature of the intrusion and the egregiousness of its means. For example, while mere deception in the course of journalistic investigation is not actionable, intrusion into a home,³⁰ the use of hidden cameras and recording devices,³¹ or taking advantage of intimate or confidential relationships³² sometimes leads to liability. In *Shulman v. Group W Productions*, for example, a reporter on a ‘ride-along’ with an emergency medical helicopter team videotaped an accident scene and recorded conversations between a victim and one of the team’s nurses, who was wearing a microphone. While the reporter’s general presence at and video recording of the accident scene was not an intrusion, the court refused to dismiss claims based on the recording of the victim’s utterances picked up by the nurse’s microphone and on the reporter’s presence in the helicopter while the accident victim was transported to the hospital. As the court explained: “[The reporter], perhaps, did not intrude into that zone of privacy merely by being present at a place where he could hear such conversations with unaided ears. But by placing a microphone on Carnahan’s person, amplifying and recording what she said and heard, defendants may have listened in on conversations the parties could reasonably have expected to be private.”³³ In allowing this claim to proceed, the court emphasized the traditional expectation of confidentiality in conversations with providers of medical care.³⁴

Courts also rely on the factual context to determine whether defendants intrude upon seclusion by obtaining records from others. In *Hall v. Harleysville Insurance Co.*,³⁵ for example, the court sustained an intrusion claim alleging that workers’ compensation investigators had improperly obtained a claimant’s credit reports. In doing so, the court distinguished *Chicarella v. Passant*,³⁶ which had dismissed a similar claim alleging that an automobile insurance company had obtained the plaintiff’s hospital records by pretext while investigating his general credibility. *Hall* opined that credit reports are “a thorough and complete analysis of [an individual’s] financial position,”³⁷ while the hospital records in *Chicarella* contained only “brief descriptions of claimant’s medical treatment.”³⁸

Courts sometimes have allowed intrusion claims based on excessive and prolonged surveillance in public, despite the tort's purported inapplicability to public behavior. A 1913 case found liability for "rough or open shadowing," opining that "[a]ctual pursuit and public surveillance of person and home are suggestive of criminality fatal to public esteem and productive of public contempt or ridicule."³⁹ In *Galella v. Onassis*, the court held that prolonged and intrusive monitoring of the public activities of Jacqueline Onassis and her family by a paparazzo amounted to a "torrent of almost unrelieved abuse into everyday activity" and constituted actionable intrusion.⁴⁰

*Nader v. General Motors*⁴¹ considered allegations that General Motors had intruded upon Nader's seclusion via a pattern of tactics including following the well-known consumer advocate in public, interviewing his friends and associates in ways intended to harm his reputation, seeking to entrap him into illicit sexual relationships, harassing him with phone calls, and tapping his phones. The court focused on an allegation that a GM agent had "followed [Nader] into a bank, getting sufficiently close to him to see the denomination of the bills he was withdrawing from his account." As the court explained:

A person does not automatically make public everything he does merely by being in a public place, and the mere fact that Nader was in a bank did not give anyone the right to try to discover the amount of money he was withdrawing. On the other hand, if the plaintiff acted in such a way as to reveal that fact to any casual observer, then, it may not be said that the appellant intruded into his private sphere.

While noting that "it is manifest that the mere observation of the plaintiff in a public place does not amount to an invasion of his privacy," the court opined that public surveillance nonetheless could be "so overzealous as to render it actionable." A concurring judge would have gone farther, arguing that

it does not strain credulity or imagination to conceive of the systematic "public" surveillance of another as being the implementation of a plan to intrude on the privacy of another. Although acts performed in "public," especially if taken singly or in small numbers, may not be confidential, at least arguably a right to privacy may nevertheless be invaded through extensive or exhaustive monitoring and cataloguing of acts normally disconnected and anonymous.⁴²

Recently, in the Fourth Amendment context, courts have begun to recognize that prolonged monitoring of public activities, especially when accompanied by datafication, can intrude upon a "reasonable expectation of privacy," as required for a Fourth Amendment violation.⁴³ In *United States v. Jones*,⁴⁴ five concurring justices agreed that

there could be a reasonable expectation of privacy under the Fourth Amendment in aggregated location data.⁴⁵ While the “reasonable expectation of privacy” test is not identical to the test for actionable intrusion, a similar approach could be imported into intrusion analysis. Justice Sotomayor’s concurrence noted that GPS tracking “generates a precise, comprehensive record of a person’s public movements that reflects a wealth of detail about her familial, political, professional, religious, and sexual associations.” Her concurrence also implicitly recognized the privacy implications of datafication. Thus, the GPS location record invaded reasonable expectation of privacy because it was “precise” and “comprehensive” and because the “Government can store such records and efficiently mine them for information years into the future.”⁴⁶ Justice Sotomayor also emphasized the fact that the GPS monitoring at issue is “cheap in comparison to conventional surveillance techniques.”⁴⁷ It is too soon to tell what influence, if any, these developments will have on tort law.

The intrusion upon seclusion tort rarely has been extended to data acquisition as a byproduct of providing service. *Dwyer v. American Express* rejected such an attempt, emphasizing that “[b]y using the American Express card, a cardholder is voluntarily, and necessarily, giving information to defendants that, if analyzed, will reveal a cardholder’s spending habits and shopping preferences,” and concluding that “[w]e cannot hold that a defendant has committed an unauthorized intrusion by compiling the information voluntarily given to it and then renting its compilation.”⁴⁸

In *Pulla v. Amoco*,⁴⁹ however, the court upheld a jury finding of intrusion upon seclusion when a credit card company employee improperly accessed and reviewed the company’s records to investigate another employee’s use of sick leave. The court emphasized the repurposing of the data: “whatever Amoco’s interest may have been, its methods were unduly intrusive when balanced against that interest. Amoco used confidential credit card records for a purpose for which they were never intended, and did so surreptitiously, without notice to, or consent or knowledge of, the credit card holder.” The fact that the records had been obtained and compiled by Amoco as a byproduct of providing credit card services was not decisive.

Because of increasing concerns about the privacy implications of ubiquitous data collection and aggregation, it is possible that more courts will be inclined to take a nuanced approach to intrusion upon seclusion claims involving the datafication, repurposing, or transfer of data acquired as a byproduct of service provision.⁵⁰

b. The Requirement that the Intrusion Be “Highly Offensive” In addition to requiring an actionable intrusion, the Restatement’s version of the intrusion tort requires that the intrusion be “highly offensive,” suggesting that the tort is aimed at harms involving mental or emotional distress. The extent to which tort law should recognize emotional and mental injuries was a subject of controversy in the late 19th and early 20th

centuries, in part because of concerns that courts would be flooded with trivial claims. To address those concerns, the tort of intentional infliction of emotional distress was limited to extreme and outrageous behavior resulting in severe emotional distress. Prosser, who also wrote extensively about intentional infliction of emotional distress,⁵¹ viewed the intrusion upon seclusion tort as a gap-filler aimed at addressing similar harms:

It appears obvious that the interest protected by this branch of the tort is primarily a mental one. It has been useful chiefly to fill in the gaps left by trespass, nuisance, the intentional infliction of mental distress, and whatever remedies there may be for the invasion of constitutional rights.⁵²

The requirement tends to stand in the way of intrusion claims in modern data contexts, where different privacy concerns are salient. For example, the court in *Hall*, while allowing an intrusion claim based on acquisition of credit reports to go forward, left it to the jury to determine “whether disclosure of a person’s credit report would cause shame or embarrassment to a person of ordinary sensibilities.”⁵³ The court in *Chicarella* determined that “the information disclosed by the hospital records is not of the sort which would cause mental suffering, shame or humiliation to a person of ordinary sensibilities.”⁵⁴ In *Busse v. Motorola*, the court dismissed an intrusion claim alleging that mobile phone carriers had transferred customer information to outside researcher consultants for a study of the effects of cellphone use on mortality. The court determined that “none of the ‘personal’ information furnished by the customers, standing alone – names, telephone numbers, addresses or social security numbers – have been held to be private facts,” because those pieces of information were not “facially revealing, compromising or embarrassing.”⁵⁵

The privacy concerns raised by big data stem from the risks associated with data collection and aggregation and do not revolve solely around fear of shame or embarrassment. If the intrusion upon seclusion tort is to play any significant role in regulating data acquisition in the big data context, the conception of a highly offensive intrusion will have to evolve.

2. California’s Constitutional Right to Information Privacy

Several state constitutions incorporate explicit rights to privacy, but most, like the federal Fourth Amendment, apply only to government action. California, however, enforces its constitutional right to privacy against both public and private actors. The elements of a claim are (1) a legally protected privacy interest, (2) a reasonable expectation of privacy under the circumstances, and (3) conduct amounting to a serious invasion of protected privacy interests.⁵⁶ Though there is a requirement of seriousness, offensiveness and mental distress are not elements of the constitutional claim, perhaps reflecting the fact

that informational privacy is “the core value furthered by” California’s constitutional right to privacy.⁵⁷ Courts have just begun to flesh out the scope of the right’s application to modern data acquisition contexts.

In 2012, *Goodman v. HTC America, Inc.* allowed a claim based on allegations that a mobile phone weather app collected location data that was far more accurate than necessary to provide weather information.⁵⁸ While phone users “may have expected their phones to transmit fine GPS data occasionally for certain purposes, they did not expect their phones to continually track them for reasons not related to consumer needs.” The opinion compared the app’s collection of fine location data to GPS monitoring, quoting Justice Sotomayor’s observation that “GPS monitoring generates a precise, comprehensive record of a person’s public movements that reflects a wealth of detail about her familial, political, professional, religious, and sexual associations.”⁵⁹

Goodman may be contrasted with *Fogelstrom v. Lamps Plus*⁶⁰ and *In re iPhone Application Litigation*.⁶¹ In *Fogelstrom*, the court dismissed a California constitutional privacy claim based on a retailer’s asking customers for their ZIP codes and using them to find customers’ home addresses for advertising purposes. The court opined that the retailer’s behavior was “not an egregious breach of social norms, but routine commercial behavior.” In *iPhone*, the court relied on *Fogelstrom* in ruling that the alleged disclosure to third parties of “the unique device identifier number, personal data, and geolocation information from Plaintiffs’ iDevices,” even if surreptitious, was insufficiently egregious to state a claim.

The law in this area is in flux and it remains to be seen how it will evolve.

3. Statutory Prohibitions of Eavesdropping and Recording Conversations

Statutes on both the federal and state levels prohibit eavesdropping and unauthorized interception of communications. Many of these statutes make no inquiry into the data subject’s behavior and hence are rule like. Some, however, apply only under specified conditions of privacy, employing fact-specific standards to determine whether monitoring is permitted.

The federal Electronic Communications Privacy Act (ECPA) imposes a rule against non-consensual monitoring of wire and electronic communications.⁶² ECPA’s ban on monitoring “oral communications,” however, applies only when there is “an expectation that such communication is not subject to interception under circumstances justifying such expectation.”⁶³ Many decisions essentially turn that standard into a rule that monitoring is permitted if a conversation “could be overheard, unaided by any mechanical device, from an uncontrived position.”⁶⁴ Some decisions are more nuanced, however. For example, one court determined that an employee may have had an expectation that his conversations near his workstation were free from electronic

interception by his supervisors even if other employees standing nearby might have overheard.⁶⁵

A California statute prohibits the recording of “confidential communications.”⁶⁶ The California Supreme Court has held that a conversation is confidential if a party has “an objectively reasonable expectation that the conversation is not being overheard or recorded.”⁶⁷ A later court applied this test in a nuanced analysis of a news show’s secret recording of actors’ conversations during casting workshops. The court considered factors such as whether the conversing actors had their backs turned to the undercover reporter, whether they were “chatting amongst themselves in a corner,” and whether the nature of the gathering was such that attendees would expect their conversations to be recorded or disseminated.⁶⁸ Courts also have considered whether the speakers took steps to ensure that their conversations would be confidential.⁶⁹

In Wisconsin, a prohibition on recording a conversation depends on “(1) the volume of the statements; (2) the proximity of other individuals to the speaker, or the potential for others to overhear the speaker; (3) the potential for the communications to be reported; (4) the actions taken by the speaker to ensure his or her privacy; (5) the need to employ technological enhancements for one to hear the speaker’s statements; and (6) the place or location where the statements are made.”⁷⁰ In New York, where the eavesdropping statute prohibits “mechanical overhearing of a conversation,” courts have debated whether the prohibition applies when the speaker has no expectation of privacy.⁷¹

4. Contracts

Though contract law is not privacy law per se, contractual agreements often provide substantive terms for the acquisition or transfer of information. For example, a data holder might transfer data to a researcher, subject to an agreement that the researcher keep the data confidential or disclose it only in de-identified form. Or an individual might agree to participate in a survey based on an agreement to keep specific personal information confidential.

With rare exceptions for terms deemed unconscionable or contrary to public policy, contract law has little to say about the appropriateness of substantive contractual terms. Instead, contract law primarily regulates contract formation, sometimes inquiring into the circumstances surrounding a contract’s formation to assess its validity.⁷² Though standardized ‘contracts of adhesion’ are nothing new,⁷³ contract law is challenged by modern data acquisition contexts, where double-sided fact-specific inquiry into contract formation is not practical.⁷⁴ In such contexts, contract validity tends to be determined by adherence to procedural formalities, such as requiring users to scroll through pages of text before clicking ‘I agree’.

B. One-Sided Standards-based Laws

Many privacy laws, especially in the service provider context, focus on the activities of the data handler, while making general assumptions about data subjects' behavior. This section discusses Section 5 of the FTC Act, which prohibits "unfair or deceptive acts or practices in or affecting commerce."⁷⁵ The FTC enforces Section 5. It also holds hearings and issues reports. The FTC's reports, along with decisions from previous enforcement actions, provide guidance and structure to the interpretation of Section 5's very general standard. As discussed in Part I, the FIPPs of notice and consent have been the linchpins of the FTC's approach to enforcing Section 5.

In its 1998 Report, the FTC explained the notice principle in seemingly substantive terms, suggesting that compliance would depend on whether a data subject had an actual understanding of a data handler's information practices before consenting to collection:

Without notice, a consumer cannot make an informed decision as to whether and to what extent to disclose personal information. Moreover, three of the other principles discussed below – choice/consent, access/participation, and enforcement/redress – are only meaningful when a consumer has notice of an entity's policies, and his or her rights with respect thereto.⁷⁶

The Report immediately shifted its focus away from a substantive effectiveness, however, providing a list of types of information that should be included in a notice and asserting (rather glibly in retrospect):

In the Internet context, notice can be accomplished easily by the posting of an information practice disclosure describing an entity's information practices on a company's site on the Web. To be effective, such a disclosure should be clear and conspicuous, posted in a prominent location, and readily accessible from both the site's home page and any Web page where information is collected from the consumer. It should also be unavoidable and understandable so that it gives consumers meaningful and effective notice of what will happen to the personal information they are asked to divulge.⁷⁷

The Report's discussion of choice/consent also quickly turned procedural, asserting (again glibly in retrospect) that "[i]n the online environment, choice easily can be exercised by simply clicking a box on the computer screen that indicates a user's decision with respect to the use and/or dissemination of the information being collected."⁷⁸

In practice, FTC enforcement has been mostly procedural, focusing on ensuring that online companies have privacy policies, that they are not hidden in obscure places on company websites, and so forth. Though the FTC does concern itself with whether privacy policies are accurate, it does not make substantive inquiries into whether

companies are meeting the goals of meaningful consumer awareness of and consent to their information practices. The defects of this procedural notice and consent approach have been noted by scholars, consumer advocates, and the FTC itself. As it observed in a 2010 report, “Protecting Consumer Privacy in an Era of Rapid Change” (2010 Framework):⁷⁹

[T]he notice-and-choice model, as implemented, has led to long, incomprehensible privacy policies that consumers typically do not read, let alone understand.... In addition, [the model has] struggled to keep pace with the rapid growth of technologies and business models that enable companies to collect and use consumers' information in ways that often are invisible to consumers.⁸⁰

The FTC’s attempts to move beyond notice and consent have, for the most part, stayed within a procedural rubric. Thus, the 2010 Framework endorses a ‘privacy by design’ approach, under which companies should incorporate privacy considerations into all stages of developing and implementing their products, services, and interactions with consumers.⁸¹ The FTC has mandated ‘privacy by design’ practices in the consent decrees settling some of its enforcement actions. Most notably, the consent decree in an action against Google mandated that Google implement a “comprehensive privacy program” that would, among other things “address privacy risks related to the development and management of new and existing products and services for consumers” by conducting “privacy risk assessments” and implement “reasonable privacy controls and procedures” to address the identified risks.⁸²

In a few recent cases involving spyware or smartphone apps that gain access to personal information stored on the phone, the FTC has taken steps toward putting substantive teeth into its interpretation of “unfair and deceptive trade practices” in the privacy context. While most such actions have been based primarily on allegations of specific misrepresentations in companies’ privacy policies, a few cases have relied on a policy’s sin of omission in failing to explain clearly that the company was engaging in monitoring that would not be expected in the context of a particular service.⁸³

C. Rule-based Laws

In several sectors, privacy laws are far more rule-like than the laws described in the previous two sections. Rule-like approaches tend to have been adopted most often in sectors dealing with traditionally sensitive information, where the stakes seem especially high. Three of the most important of these regimes at the federal level cover electronic communications, financial information, and health information. A rule-like approach is desirable only if interactions between data subjects and data handlers are sufficiently standardized to mitigate the costs of lost flexibility, explaining why rule-like laws tend to

be sector specific. Attempts to retain some contextual sensitivity often lead to highly complex sets of rules accounting for various contingencies. Perhaps for this reason, rule-like regimes often apply only to specific types of data handlers, leaving closely related entities unregulated.

1. Electronic Communications Privacy Act

Except with regard to oral communications, ECPA generally imposes a substantive rule against private monitoring.⁸⁴ ECPA also generally prohibits entities that provide telecommunication services to the public from disclosing the contents of those communications to private parties.⁸⁵ The most important exception to these prohibitions is consent. Unfortunately, however, ECPA suffers from a problem that commonly affects rule-like regimes: as times have changed, the terms that define the rule – such as ‘intercept’, ‘electronic storage’, and ‘remote computing service’ – have become increasingly out of step with reality and hence more and more difficult to interpret.⁸⁶ The result is a legal regime that lacks the benefits of either clear rules or flexible standards. As a result, both privacy advocates and telecommunications companies have advocated ECPA reforms.⁸⁷

2. Computer Fraud and Abuse Act

The CFAA prohibits “intentionally access[ing] a computer without authorization or exceed[ing] authorized access” to obtain “information.”⁸⁸ Since enacting it in 1984 to criminalize hacking into government computers, Congress has expanded the CFAA’s scope of application to virtually all computers. Like ECPA, the CFAA is rule like, but suffers from controversy over the interpretation of key terms, such as what it means to “exceed authorized access.”

Comparing the CFAA to the tort of intrusion upon seclusion illustrates the tension between rules and standards. Tort liability depends on contextual factors, such as the type of information stored on the computer, the type of unauthorized access, and the relationship between the person accessing the computer and the computer’s owner. While more likely to get the nuances right, such a standards-based approach provide less clear ex ante guidance and thus might be an unsuitable basis for criminal prosecution. The CFAA attempts to impose a bright line prohibition instead. A bright line rule is likely to be either overly broad or overly narrow, leading to controversy over the terms defining the boundary. A relatively narrow interpretation, under which unauthorized access means technical hacking – the use of technical measures to gain access to computer files – misses some blameworthy intrusions. A broad interpretation, under which one can exceed authorized access by using a fake name in violation of a social network service’s terms of service, sweeps in a huge swath of ordinary social behavior, thereby leaving room for abuse by prosecutors or litigious private parties.

3. Laws Governing Acquisition and Transfer of Financial Information

Financial institutions, such as banks, obtain sensitive financial information as a byproduct of providing services to individuals. The Gramm-Leach-Bliley Act (GLBA)⁸⁹ and related regulations⁹⁰ regulate covered financial institutions' transfers of "personally identifiable financial information" to outside entities. It permits virtually unrestricted transfer to certain "affiliated" companies and essentially bans transfer to non-affiliated companies without consent. The act mandates very specific notice and consent procedures, including annual privacy notices and specific mechanisms for offering consumers to "opt out" of transfer to non-affiliated companies. Essentially, the GLBA approach is a rule-like version of the FTC's notice and consent approach that is tailored to the financial sector.

The Fair Credit Reporting Act (FCRA)⁹¹ places substantive restrictions on the circumstances under which "consumer reporting agencies" are permitted to provide "credit reports" to third parties. Unlike banks, credit reporting agencies do not obtain consumer data as a byproduct of providing services to them. Instead, they essentially monitor financial information for the benefit of third parties. FCRA is a rule-like substantive regulation of credit reporting practices. It specifies the types of information that can be included in credit reports under various circumstances and the purposes, which include the extension of credit, employment, insurance, government benefits, licensing, and other "legitimate business needs," for which credit reports may be obtained.

Like many complex rule-like regimes, FCRA and the GLBA apply only to specifically defined entities and circumstances. Newer entities, such as data brokers and companies like Google, which now collect large amounts of financial data about consumers, are largely unregulated unless their activities come within FCRA's specific definitions.

4. Laws Regulating the Acquisition and Transfer of Health Information

Health care institutions traditionally acquire health information in two basic ways: as a byproduct of treatment or through research. Under the taxonomy presented here, acquiring information for research purposes is a type of monitoring. The regulations governing the acquisition and transfer of personal information in the traditional health care context are extremely complex and comprehensive and are generally rule like, though they include some standards-based elements. The health privacy regime combines detailed procedural elements with important substantive distinctions between different types of transfer and use.

In acquiring and transferring personal information obtained as a result of either treatment or research, medical professionals are bound by professional ethical regulations and obligations, by rules governing medical research, and by HIPAA's Privacy Rule,

which was updated in 2012. Insurers are “covered entities” under HIPAA, while certain “business associates” providing services such as claims processing and data administration also are regulated.⁹² HIPAA does not apply to the increasing number of online entities, such as medical information websites, online medical forums and chat rooms, online genetic testing and medical ‘apps’, that handle large quantities of health information. Many are governed only by consumer privacy law, primarily through the FTC’s Section 5 enforcement discussed in Section B.

The HIPAA Rule applies only to “protected health information,” which generally means “individually identifiable health information” (unless such information is regulated by certain other statutory regimes). Information is considered de-identified, and hence unprotected, if it meets one of two criteria.⁹³ The first, which is rule like, considers information to be de-identified if 18 particular types of data are removed. The second, which is standards like, requires that an expert deploy generally accepted and documented procedures to ensure that there is a very small risk of re-identification. These two alternatives illustrate the trade-off between the flexibility and adaptability of standards and the clarity and lower administrative costs of rules. The rule-based definition is easy to deploy and provides regulated entities with certainty that they have met legal standards, yet has been subject to criticism in light of studies demonstrating the ease with which many records now can be re-identified, especially when aggregated with other datasets. The standards-based criterion is flexible and can, in principle, be adapted to particular circumstances and to changes in re-identification technology, but is expensive and risky for health care providers to use.

Health information privacy law depends substantively on the purpose for which information is to be acquired, used, or transferred.

a. Acquisition, Use, and Transfer for Treatment Medical professionals are afforded substantial discretion in determining appropriate treatments and sharing information with other professionals for treatment purposes. They are, however, bound by professional ethics and norms in these determinations. Both legally and ethically, medical professionals are required to obtain ‘informed consent’ to treatment.⁹⁴ In principle, the informed consent requirement is a substantive and doubly fact-specific standard: “Informed consent is more than simply getting a patient to sign a written consent form. It is a process of communication between a patient and physician that results in the patient's authorization or agreement to undergo a specific medical intervention.”⁹⁵ State law and institutional requirements govern procedures for obtaining informed consent and generally require an in-person explanation of treatment benefits and risks, with an opportunity for patients to ask questions, and sometimes require signed consent forms.⁹⁶ Once a patient has consented to treatment, the HIPAA Privacy Rule imposes few

constraints on the acquisition and transfer of protected health information for purposes of treatment and obtaining payment.⁹⁷

b. Acquisition, Use, and Transfer for Research Medical research projects usually are subject to approval by an Institutional Review Board (IRB). For example, the HHS regulations governing federally funded research (which are applied by many IRBs generally) apply whenever an investigator obtains “identifiable private information,”⁹⁸ which includes “information about behavior that occurs in a context in which an individual can reasonably expect that no observation or recording is taking place, and information which has been provided for specific purposes by an individual and which the individual can reasonably expect will not be made public (for example, a medical record).” IRBs evaluate proposed research according to criteria⁹⁹ that include minimization of risk, reasonable relationship of risk to benefits, and equitable selection of subjects.

IRBs must consider whether “there are adequate provisions to protect the privacy of subjects and to maintain the confidentiality of data.” Informed consent to the research is required and the explanation provided to research subjects must include a “statement describing the extent, if any, to which confidentiality of records identifying the subject will be maintained.” These requirements usually apply both to information obtained during a research project and to existing data that was acquired as a byproduct of treatment. Research based on “existing data” is exempt only if the data “sources are publicly available or if the information is recorded by the investigator in such a manner that subjects cannot be identified, directly or through identifiers linked to the subjects.”¹⁰⁰

The HIPAA Rule generally requires written authorization from a data subject for use or transfer of protected health information for purposes other than treatment and payment, including research. The authorization must provide specifics, including the identity of the recipient and the purpose of the transfer or use. The authorization requirement may be waived only for research activities that an IRB or a ‘privacy board’¹⁰¹ determines, among other things, poses “no more than a minimal risk to the privacy of individuals.” The rule specifies procedural mechanisms for that evaluation.

c. Use and Transfer for Marketing and Other Purposes HIPAA also regulates the use and transfer of protected health information for purposes other than treatment, payment, and research.¹⁰² Use and transfer of genetic information for underwriting purposes is prohibited entirely.¹⁰³ Otherwise, protected health information generally may be used or transferred only with the data subject’s signed authorization,¹⁰⁴ which must include specific elements, including a “specific and meaningful” description of the information to be disclosed, the “specific identification” of the persons to whom the information will be disclosed, and a description of each purpose of the use or

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

disclosure¹⁰⁵ and an explanation that the authorization may be revoked in most circumstances.¹⁰⁶ If the covered entity will receive financial remuneration from a third party as result of a sale of the information or a use or transfer for marketing purposes, the authorization must state that fact.¹⁰⁷

IV. Analysis and Discussion

This part begins with a few observations based on the taxonomy and overview in Parts II and III. It then presents thoughts about how to move toward resolving the challenges big data poses to privacy law.

A. Observations about Current Privacy Law

Applying the taxonomy developed here to the privacy laws briefly described in Part III suggests the following observations:

1. Substantive Regulation and Monitoring

For the most part, privacy law tends to engage in substantive regulation of monitoring and procedural regulation (if any) of data acquisition as a byproduct of service provision. The intrusion upon seclusion tort is a standards-based example of this tendency, since intrusion claims based on use of information obtained as a byproduct of service provision have not been successful for the most part. ECPA, the CFAA, and other surveillance statutes provide rule-like substantive regulation of monitoring. FCRA, which regulates entities that provide credit reports to parties that wish to monitor financial behavior, is significantly substantive, while the GLBA, which concerns the transfer of information about individuals obtained as a byproduct of providing financial services to them, is predominantly procedural. In the health care arena, there is significant substantive regulation of research, which is a form of monitoring.

Though privacy law has not always disentangled monitoring by service providers from data acquisition as a byproduct of service provision, there appears to be a trend toward making this distinction. Thus, *Goodman*, which recognized a claim against a service provider under California's substantive constitutional right to privacy, emphasized that the fine location data at issue was acquired for reasons unrelated to service provision. The FTC also has begun to bring substantive Section 5 claims when service providers sought out information that was unrelated to service provision. This issue is well recognized in the medical context, where regulations not only distinguish research from treatment, but specify in detail how to deal with the fact that research and treatment often overlap.

2. Substantive Regulation and Datafication and Repurposing

U.S. law for the most part has not regulated the datafication or repurposing of information that is acquired as a byproduct of providing service. Cases including *Dwyer*, *Busse*, *Fogelstrom* and *In re iPhone* have denied intrusion upon seclusion and California right to privacy claims based on datafication and repurposing of customer records; indeed *Fogelstrom* and *In re iPhone* characterized the behavior at issue as “routine commercial behavior.” The GLBA distinguishes data transfer to affiliated companies from transfer to non-affiliated companies. That distinction bears some relation to a distinction between use in providing service and repurposing, though it permits affiliated companies to repurpose data for advertising purposes.

Though the idea of purpose limitation as a FIPP goes back to the 1967 HEW Report and “purpose specification” is included in the OECD FIPPs, the FTC’s version does not include a purpose limitation principle. The FTC’s 2010 Framework recognizes the privacy concerns raised by unanticipated uses of previously collected data and emphasizes that “[u]nder well-settled FTC case law and policy, companies must provide prominent disclosures and obtain opt-in consent before using consumer data in a materially different manner than claimed when the data was collected, posted, or otherwise obtained.”¹⁰⁸ The difficulty with this notice-and-consent-based approach to repurposing is that companies easily can get around it by phrasing their privacy policies vaguely as to the uses they make of the data. Indeed, privacy policies rarely, if ever, distinguish between data acquired as a byproduct of service provision and data acquired or datafied for other purposes. For example, Google, which has one of the most detailed privacy policies, distinguishes “information you give us” (for example, by creating a Google account) and “information we get from your use of our services.” Nowhere, however, does the policy inform consumers about which of this is collected to provide the service and which is collected only for other purposes, such as profiling. Similarly, though Google generally describes “how we use information we collect” to offer services, provide tailored advertising, and so forth, its description makes no distinction between data collected for the purpose of providing services and data collected for other purposes.¹⁰⁹ Users are not given the option of opting out of data collection that is unnecessary for service provision.

There are a few exceptions. Most notably, the HIPAA Rule imposes substantive restrictions as to whether information acquired in the course of treatment may be used for other purposes. Indeed, the research use of existing information is treated essentially equivalently to the acquisition of data specifically for research. The court in *Pulla* ruled that intrusion upon seclusion applied when an employee’s credit card records were pulled from the company’s database and used to investigate his use of sick leave.

3. Consent in the Big Data Context

Nearly all privacy law provides a consent exception. When those acquiring personal information deal with data subjects on an individual basis, the question of consent can be assessed in a double-sided factual inquiry. In the big data context, however, such highly contextual inquiry usually is unmanageable due to the sheer number of individuals involved.

Indeed, when big data is acquired by monitoring, perhaps by video surveillance of city streets, it often would be impossible to obtain the consent of data subjects, who have no direct interaction with the data handlers. This is a very different situation from that assumed in medical research regulation, for example, where informed consent plays a central role. In such circumstances, consent exceptions become virtually irrelevant and everything depends on the substantive law.

When data is acquired as a byproduct of service provision, consent remains problematic in the big data context. Notice of what information is being acquired for what purposes must be provided in a standardized format. The more detailed and granular the information provided, the more time-consuming it becomes for data subjects to wade through it, to understand it, and to figure out how it relates to their concerns. The big data context generally offers little opportunity for data subjects to ask individualized questions to clear up any ambiguity or confusion. The validity of the consent process must then be assessed in a one-sided fashion, with relevant facts about data subjects reduced to trivialities, such as whether or not they clicked ‘I agree’. The question generally becomes “Did the data handler follow the right procedures to obtain consent?” rather than “Did the data subject consent?”

4. Bad Fit with Big Data Harms

Current privacy laws are not well adapted to the privacy concerns that arise in the big data context, in which risks tend to be probabilistic and the connection between particular harms and data handler behavior tends to be attenuated because of the importance of aggregation and hard to trace. Harms often are societal and spread over large numbers of individuals. The privacy torts’ emphasis on outrageous behavior is clearly inapropos to the privacy concerns associated with large aggregations of data. Moreover, the reliance on inference fundamental to big data practices means that privacy law’s categorical treatment of certain pieces of information as private or public is unlikely to correlate with big data privacy harms. Recent judicial recognition of this problem in the context of location tracking is only the tip of the iceberg of the data aggregation issue. The ubiquity of data collection also undermines the effectiveness of regulatory regimes that rely on the expertise of entities in particular sectors.

B. Where Do We Go from Here?

This section suggests five approaches that might help to improve the fit between big data and privacy law, most of which build on trends that are already in motion. First, we should recognize that the market cannot make the necessary normative assessments of the trade-offs between big data's benefits and costs and look seriously for alternatives. Second, we should attempt to disentangle the issue of government surveillance from questions of big data use by the private sector by reforming surveillance law so as to raise and individualize the showing required for government access to private sector data. Third, we should strengthen substantive privacy laws with respect to monitoring in the big data context. Fourth, we should adopt a purpose-based framework for regulation in the service provider context. Fifth, we should consider a substantive approach to notice and consent.

1. Assessing Trade-offs in the Big Data Context

The most important mismatch between current privacy law and big data is current law's virtually exclusive focus on individualized assessments of privacy costs. This focus, which is common to both consent-based and fact-specific regimes, is out of sync with the privacy concerns raised by big data for several reasons. First, as I have discussed in more detail elsewhere,¹¹⁰ the interrelatedness of data makes assessment of the potential costs of particular decisions to consent to data collection impossible for individuals to assess in practice, and perhaps even in principle. (To put this in more familiar economic language, the acquisition and use of large amounts of data is fraught with externalities.) This fact, combined with various other factors, including the probabilistic nature and unpredictability of harms, the opacity and complexity of big data practices, and various standard biases familiar from behavioral economics, means that there is no functioning market for assessing citizen's preferences in this regard. (See Chapter 3 in this volume, by Acquisti.)

As a result, we need to devise other ways to assess the trade-offs that must be made in the big data context. To do this well, we need more information about both the privacy impacts of data acquisition and use and the potential benefits of big data analyses. Substantive assessment of the weight to accord privacy concerns in a particular data acquisition context may demand a combination of technical expertise, metrics and statistics for assessing the likelihood and expected extent of privacy impact, and transparency about data collection practices. Research on topics such as the frequency and severity of individualized harms such as identity theft, reputational injury, stalking and fraud, the psychological and sociological effects of ubiquitous data collection and monitoring, and the ways in which personal data are collected, datafied, used, and exchanged in practice is undoubtedly needed.

While better information about the big data ecosystem is important, it may be slow in coming. Moreover, the costs and benefits of big data practices are complex,

unpredictable, and difficult to compare. In the end, normative assessments of the trade-offs will be required.¹¹¹ Because both the benefits and the privacy implications of big data's inferences and predictions are interdependent, those assessments should be made in some collective fashion.

In thinking about how to handle the big data issue, we should look to other arenas in which law has grappled with similar problems of unpredictability, externalities, probabilistic harms, and valuation difficulties. As others have suggested, environmental regulation faces somewhat similar issues of balancing ephemeral broad-based values against the shorter term benefits of private economic activity.¹¹² In the mass tort context, the law has grappled with the difficulty of assigning responsibility for increases in statistical risk and harms that are difficult or impossible to trace back to particular tortfeasors. The law also has many mechanisms beyond direct regulation and tort law for encouraging the internalization of externalities, including targeted taxes, subsidies, audit requirements, and safe harbors (see Chapter 9 in this volume, by Greenwood et al.). Better metrics for assessing privacy impact or substantive standards might be used in conjunction with procedural regulation.¹¹³

We should also consider institutional mechanisms for case-by-case collective normative assessment for some types of big data projects. One such mechanism would be a review by a privacy board involving deliberation and input from experts and members of the public who are likely be affected, both positively and negatively, by the projects. Such a mechanism might be modeled on IRBs. Another (possibly complementary) mechanism for generating a community assessment of potential privacy impact would be a privacy impact notice, prepared by experts but shared with the public.¹¹⁴

2. Disentangle Government Surveillance from Private Sector Data Acquisition and Use

Current U.S. surveillance law sets low or nonexistent barriers to government acquisition of data from private entities.¹¹⁵ Thus concern about government access to data held by service providers and other private entities colors many discussions of big data and privacy. If we believe that it is particularly important to ensure that government officials do not intrude into the private lives of citizens without good reason, we need to reform surveillance law. Instituting higher thresholds and requiring more individualized reasons for government demands for data from private entities would permit the trade-offs involved in private sector data acquisition to be considered on their own merits.

3. Strengthen Substantive Privacy Law with Respect to Monitoring in the Big Data Context

There is relatively strong consensus in current law that monitoring of private behavior without meaningful consent should be substantively regulated via tort law, surveillance

statutes, and regulations governing human subjects research. Part III discussed cases in which this body of law is being extended to cover prolonged surveillance in public and monitoring outside of the scope of service provision. This trend is likely to continue (and in my view should). Recognizing the intrusiveness of aggregating otherwise ‘non-private’ information and distinguishing between data acquisition by monitoring and data acquisition as a byproduct of providing service is normatively attractive and would clarify current debates about big data.

This need not mean that all monitoring should require consent, but it does mean that monitoring is not necessarily acceptable simply because activity takes place nominally in public. The substantive law might provide mechanisms for authorizing monitoring for particular purposes on a case-by-case basis. For example, there might be an exception from privacy liability for monitoring as a part of research that has been approved by an IRB or privacy board and meets specific criteria for waiver of the consent requirement.¹¹⁶

In its 2010 Report, the FTC proposed that data acquisition by companies be limited to “information needed to fulfill a specific, legitimate business need.”¹¹⁷ That proposal is somewhat similar to my suggestion here, but the distinction between monitoring and service provision is more intuitive, less amorphous, and better in line both with current privacy law and with the goals of privacy law. To instantiate the distinction between monitoring and service provision, data handlers should be required to specify in their privacy policies what services they provide, whether they acquire information by monitoring (i.e. not as a byproduct of providing service), what information they acquire as a result of monitoring, whether they datafy that information, and what they do with it. Service providers should bear a heavy burden in demonstrating consent to monitoring, which should be both opt in and explicit. In essence, service providers should not be able to use their access to users’ computers, phones, and so forth as means to avoid substantive restrictions on monitoring any more than home repairpersons would be permitted to install listening devices or photocopy documents.

4. Make Substantive Distinctions Based on Datafication and Repurposing for Data Acquired as a Byproduct of Providing Service

The FTC’s 2010 proposed Framework suggests eliminating consent requirements for the collection and use of data for “commonly accepted practices,”¹¹⁸ including “product and service fulfillment,” “internal operations,” “fraud prevention,” “legal compliance and public purpose,” and “first-party marketing.”¹¹⁹ The reasoning behind this proposal is that some of these uses are obvious to users, some are necessary for policy reasons, and others are “sufficiently accepted” such that requiring consent would be more burdensome than beneficial. Like “legitimate business need,” however, “commonly accepted practices” is a vague and elastic term. Moreover, it has an unfortunate circularity, especially in view of the market failures endemic to notice and choice. Practices may become “commonly

accepted” simply because companies adopt them and data subjects are unaware of or do not understand them.

The Framework also suggests increasing transparency about data practices by providing “clearer, shorter, and more standardized” privacy notices. It is not at all clear how this is to be accomplished, however. For data subjects to understand what data handlers are doing with their data, they need more detail than privacy policies currently provide, not less. Given the dearth of substantive privacy law that might provide a common foundation on which standardized notices could be based, it is challenging to provide the necessary detail in a way that is “clearer, shorter, and more standardized” than current policies.

Distinguishing between service provision and other uses and between mere acquisition of information and datafication would be a better way to make a distinction that would simplify and standardize privacy notices. The concept of service provision is intuitive and does not have the vagueness or circularity of “commonly accepted practices.” Companies could be required to specify the services they provide, which might include recommendation services that are integral to the company’s services, such as Amazon’s product recommendations, but would not generally include advertising, whether first or third party. Service provision could presumptively include ‘internal operations’, ‘fraud prevention’, and ‘legal compliance and public purpose’, which would not have to be spelled out in the description of services provided.

Modern data privacy concerns are triggered primarily when information is datafied. While the concept of ‘datafication’ would have to be fleshed out to be employed in a legal regime, the difference between information that is ephemeral or is kept in a file cabinet and data that is formatted and aggregated for computational manipulation is also intuitive and significant to data subjects. Because datafication increases the risk of certain privacy harms even if the information is used only to provide services, users should be notified of datafication even if the information is used only to provide services.

These distinctions should be embodied in a substantive provision that would (1) permit information acquisition for service provision without notice or consent, (2) permit datafication for purposes of service provision with notice, and (3) prohibit use, datafication, or transfer for other purposes (including first-party advertising) unless there is notice and explicit and specific opt-in consent.¹²⁰ Where notice is required, it should specify what information is involved and what it is used for, including the identities or specific categories of entities to which any information is transferred.

Such a legal regime would not solve all of the problems of notice and consent intrinsic to data aggregation, and might not even result in shorter privacy policies. Nonetheless, it would improve over the current regime by relying on intuitive distinctions that correspond to the privacy concerns that users have regarding datafication and repurposing. Because the privacy notice will be short and no consent will be required if a

company is using personal information only for providing services, the form of the notice will immediately give users a rough idea of whether there is something for them to worry about. Moreover, since opt-in consent and specificity both would be required, companies would have incentives to explain what they are doing both accurately and clearly enough to overcome user privacy concerns.

5. Adopt a Substantive Approach to Notice and Consent, at Least for Large Entities

Another way to improve the functioning of notice and consent in the big data context would be to impose substantive requirements of clear and understandable notice, at least on larger entities. The FTC might be able to take this step immediately on the basis of its Section 5 authority by deeming a notice and consent process “unfair and deceptive” if it cannot be demonstrated that it is understandable by a reasonable consumer. Techniques such as consumer surveys and experiments, similar to those employed in trademark law, could be used to demonstrate understandability. While a substantive evaluation of this sort might be unreasonably expensive for smaller companies, the necessary testing would be well within the means of the larger companies that collect the bulk of personal information.

Acknowledgement The generous support of the Filomen D’Agostino and Max E. Greenberg Research Fund is gratefully acknowledged.

Notes

¹ See Chapter 4 in this volume, by Ohm, for a discussion of laws regulating data use.

² 15 U.S.C. § 45 (commonly known as Section 5 of the FTC Act).

³ 45 CFR § §160, 164(A) and 164(E), available at <http://www.hhs.gov/ocr/privacy/hipaa/administrative/combined/hipaa-simplification-201303.pdf>.

⁴ 110 Stat. 1936 (1996). The latest version of the Rule accounts for various later statutory additions and revisions. See 78 Fed. Reg. 5566 (January 25, 2013) for an accounting of these authorities.

⁵ Samuel D. Warren and Louis D. Brandeis, *The Right to Privacy*, 4 HARV. L. REV. 193 (1890).

⁶ 15 U.S.C. § 1681 et seq.

⁷ 5 U.S.C. § 552a

⁸ Records, Computers and the Rights of Citizens: Report of the Secretary's Advisory Committee on Automated Personal Data Systems (HEW Report) at 4, available at <http://www.rand.org/content/dam/rand/pubs/papers/2008/P5077.pdf>.

⁹ HEW Report at 6–8.

¹⁰ HEW Report at 5–6.

¹¹ OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data (September 23, 1980), available at <http://www.oecd.org/internet/ieconomy/oecdguidelinesontheprotecti...> The OECD adopted a revised set of guidelines in 2013, available at <http://www.oecd.org/sti/ieconomy/privacy.htm#newguidelines>, which are not discussed here.

¹² Privacy Online: Fair Information Practices in the Electronic Marketplace: A Report to Congress (May 2000; 2000 FTC Report), available at <http://www.ftc.gov/sites/default/files/documents/reports/privacy-online-fair-information-practices-electronic-marketplace-federal-trade-commission-report/privacy2000.pdf>. See also Privacy Online: A Report to Congress (June 1998; 1998 FTC Report), available at http://www.ftc.gov/sites/default/files/documents/public_events/exploring-privacy-roundtable-series/priv-23a.pdf.

¹³ Chapter 2 in this volume, by Baracas and Nissenbaum, discusses this issue. See also e.g. Daniel J. Solove, *Privacy Self-Management and the Consent Dilemma*, 126 HARV. L. REV. 1880 (2013); James P. Nehf, OPEN BOOK: THE FAILED PROMISE OF INFORMATION PRIVACY IN AMERICA 191 (2012) (discussing reasons for failure of the self-policing model for privacy); Richard Warner, *Undermined Norms: The Corrosive Effect of Information Processing Technology on Informational Privacy*, 55 ST LOUIS L.J. 1047, 1084–86 (2011) (raising questions about the viability of using consent to limit mass surveillance). See also Fred H. Cate, *Protecting Privacy in Health Research: The Limits of Individual Choice*, 98 CALIF. L. REV. 1765 (2010) (arguing that the requirement of individual choice stands in the way of health research and arguing for alternative approaches to protecting privacy).

¹⁴ See Katherine J. Strandburg, *The Online Market's Consumer Preference Disconnect*, 2013 CHI. LEGAL FORUM 95 for a more detailed discussion of these concerns.

¹⁵ See e.g. Paul M. Schwartz and Daniel J. Solove, *The PII Problem: Privacy and a New Concept of Personally Identifiable Information*, 86 N.Y.U. L. REV. 1814 (2011) (applying a rules versus standards analysis); Derek E. Bambauer, *Rules, Standards, and*

Geeks, 5 BROOK. J. CORP. FIN. & COM. L. 49 (2010) (discussing the “age-old debate between rules and standards” in the context of information security). See also, generally, Louis Kaplow, *Rules versus Standards: An Economic Analysis*, 42 DUKE L.J. 557 (1992); Cass R. Sunstein, *Problems with Rules*, 83 CAL. L. REV. 953 (1995).

¹⁶ For a few of the many discussions of these issues by legal scholars, see Roger A. Clarke, *Information Technology and Dataveillance*, 31 COMM. ACM 498 (1988); Ryan M. Calo, *The Boundaries of Privacy Harm*, 86 IND. L.J. 1131 (2011); David Gray and Danielle Citron, *The Right to Quantitative Privacy*, 98 MINN. L. REV. 62 (2013); Lior Strahilevitz, *Toward a Positive Theory of Privacy Law*, 126 HARV. L. REV. 2010 (2013); Daniel J. Solove, *THE DIGITAL PERSON* (2004); Adam Thierer, *A Framework for Benefit-Cost Analysis in Digital Privacy Debates*, 20 GEO. MASON L. REV. 1055 (2013); Omer Tene and Jules Polonetsky, *Privacy in the Age of Big Data: A Time for Big Decisions*, 64 STAN. L. REV. ONLINE 63 (2012); Jane Yakowitz, *The Tragedy of the Data Commons*, 25 HARV. J. LAW & TECH. 1 (2011); Omer Tene and Jules Polonetsky, *Big Data for All: Privacy and User Control in the Age of Analytics*, 11 NW. J. TECH. & INTELL. PROP. 239 (2013); Julie M. Cohen, *What Privacy Is For*, 126 HARV. L. REV. 1904 (2013); Tal Z. Zarsky, *Transparent Predictions*, 2013 U. ILL. L. REV. 1503; Paul Ohm, *The Underwhelming Benefits of Big Data*, 161 U. PA. L. REV. ONLINE 339 (2013); Ira S. Rubinstein, *Big Data: The End of Privacy or a New Beginning?* 3 INT'L DATA PRIVACY L. 74 (2013); Katherine J. Strandburg, *Free Fall: The Online Market's Consumer Preference Disconnect*, 2013 CHI. LEGAL FORUM 95.

¹⁷ See e.g. William Prosser, *Privacy*, 48 CAL. L. REV. 383 (1960).

¹⁸ Restatement of the Law, Second, Torts § 652 (finalized in 1977). See also Neil M. Richards and Daniel J. Solove, *Prosser's Privacy Law: A Mixed Legacy*, 98 CAL. L. REV. 1887, 1891–99 (2010) (discussing the development of Prosser's thinking about privacy torts and tracing his definition of the intrusion tort back to his 1941 treatise).

¹⁹ Restatement of the Law, Second, Torts, § 652B.

²⁰ Note that courts parse these factors variously, with some including factors relating to the seriousness and justification of the intrusion under the second factor. Here, I focus the discussion of offensiveness on the requirement of a mental or emotional injury, which often stands in the way of intrusion claims based on modern data acquisition, and include all other factors in the discussion of actionable intrusion.

²¹ Id., comment b.

²² Id., comment c.

²³ Id., comment d.

²⁴ Id., Illustration 5.

²⁵ Id.

²⁶ Speer v. Dept. of Rehabilitation and Correction, 646 N.E.2d 273 (Ct. Cl. Ohio 1994).

²⁷ Sanders v. ABC, 978 P.2d 67 (Ca. 1999). But see Marrs v. Marriott Corp., 830 F. Supp. 274 (D. Md. 1992) (no intrusion by videotaping common area open to other employees).

²⁸ Hernandez v. Hillsides, Inc., 211 P.3d 1063 (Ca. 2009) (extensive discussion of the scope of intrusion upon seclusion in the workplace). But see Johnson v. K Mart Corp., 723 N.E.2d 1192 (1st Dist. Ill. 2000) (possible intrusion where employers' agents investigating theft, vandalism and drug use reported about employees' family problems, health problems, sex lives, future work plans, and attitudes about the employer).

²⁹ See e.g. Borse v. Piece Goods Shop, Inc. 611, 622–28 (3rd Cir. 19) (reviewing cases in various jurisdictions and discussing situations under which drug testing might constitute intrusion); Mares v. Conagra Poultry Co., 971 F.2d 492 (10th Cir. 1992) (no intrusion where employer requested information about prescription drug use in order to improve accuracy of drug test results).

³⁰ Dietmann v. Time, Inc., 449 F.2d 245 (9th Cir. 1971).

³¹ See e.g. Webb v. CBS Broad. Inc., 37 Media L. Rep. 1792 (2009).

³² See e.g. Taus v. Loftus, 151 P.3d 1185, 1213–24 (Ca. 2007) (analyzing and discussing cases).

³³ 955 P.2d 469 (Cal. 1998).

³⁴ Id. at 491–92.

³⁵ 896 F. Supp. 478 (E.D. Pa. 1995).

³⁶ 494 A.2d 1109 (Pa. 1985).

³⁷ 896 F. Supp. at 484.

³⁸ Id., citing 494 A.2d at 1114.

³⁹ Schultz v. Frankfurt, 151 Wis. 537, 545 (1913).

⁴⁰ Galella v. Onassis, 353 F. Supp. 196 (SDNY 1972). The Second Circuit affirmed the substantive ruling, though it limited the remedy on First Amendment grounds, 487 F.2d 986 (2d Cir. 1973).

⁴¹ Nader v. General Motors, 25 N.Y.2d 560, 570 (1970).

⁴² Id. at 572 (Breitel, J., concurring).

⁴³ See e.g. *United States v. Maynard*, 615 F.3d 544 (D.C. Cir. 2010). But see *In re Application of the United States for Historical Cell Site Data*, 724 F.3d 600 (5th Cir. 2013) (no reasonable expectation privacy in cellphone service provider location records).

⁴⁴ 132 S. Ct. 945 (2012).

⁴⁵ Id. at 954–57 (Sotomayor, J., concurring); Id. at 963–64 (Alito, J., concurring). The majority opinion rested on other grounds. Id. at 954.

⁴⁶ Id. at 955–56 (Sotomayor, J., concurring).

⁴⁷ Id.

⁴⁸ *Dwyer v. Am. Ex.*, 273 Ill. App. 3d 742 (1st Dist. Ill. 1995).

⁴⁹ 882 F. Supp. 836 (S.D. Iowa 1994).

⁵⁰ See Jane Yakowitz Bambauer, *The New Intrusion*, 88 NOTRE DAME L. REV. 205 (2012). But see Diane L. Zimmerman, *The New Privacy and the Old: Is Applying the Tort Law of Privacy Like Putting High-Button Shoes on the Internet?* 17 COMM. L. & POL’Y 107 (2012) for a skeptical view of the extensibility of the privacy torts to modern data privacy issues.

⁵¹ See e.g. William L. Prosser, *Intentional Infliction of Mental Suffering: A New Tort*, 37 MICH. L. REV. 874, 884 (1939). See also Richards and Solove, 98 CAL. L. REV. at 1895–1900 (discussing how Prosser’s views about the tort of intentional infliction of emotional distress affected his views of the privacy torts).

⁵² Prosser, 48 CAL. L. REV. at 392.

⁵³ 869 F. Supp. at 484.

⁵⁴ 494 A.2d at 1114.

⁵⁵ *Busse v. Motorola*, 351 Ill. App. 3d 67 (1st Dist. Ill. 2004). See also *Fogelstrom v. Lamps Plus*, 882 F. Supp. 836 (S.D. Iowa 1994) (dismissing claim that a retailer intruded upon plaintiff’s seclusion by asking for his ZIP code in the context of a credit card transaction so that it could obtain his home address for marketing purposes where the intrusion was not “highly offensive” because there was no allegation that the information was used for an offensive or improper purpose).

⁵⁶ *Hill v. NCAA*, 865 P.2d 633, 654–57 (1994).

⁵⁷ Id. at 654.

⁵⁸ 2012 U.S. Dist. LEXIS 88496 (W.D. Wash. June 26, 2012).

⁵⁹ 132 S. Ct. at 955 (2012).

⁶⁰ 195 Cal. App. 4th 986 (2d Dist. 2011).

⁶¹ 844 F. Supp. 2d 1040 (N.D. Cal. 2013). See also *In re Google Android Consumer Privacy Litig.*, 2013 U.S. Dist. LEXIS 42724 (N.D. Cal. Mar. 26, 2013).

⁶² See Section III.C.1.

⁶³ 18 U.S.C. § 2510.

⁶⁴ *United States v. Carroll*, 337 F. Supp. 1260. (D.D.C. 1971).

⁶⁵ *Walker v. Darby*, 911 F.2d 1573 (11th Cir. 1990).

⁶⁶ Cal. Penal Code § 632.

⁶⁷ *Flanagan v. Flanagan*, 27 Cal. 4th 766, 768 (2002).

⁶⁸ *Turnbull v. ABC*, 32 Media L. Rep. 2442 (C.D. Cal. 2004).

⁶⁹ *Vera v. O'Keefe*, 40 Media L. Rep. 2564 (S.D. Cal. 2012).

⁷⁰ W.S.A. § 968.27(12); *State v. Duchow*, 749 N.W.2d 913 (Wis. 2008).

⁷¹ N.Y. Penal Law § 250.00; *People v. Fata*, 159 A.D.2d 180 (1990)

⁷² Restatement of the Law, Second, Contracts, §5, comments a and b.

⁷³ See e.g. Restatement of the Law, Second, Contracts, § 211.

⁷⁴ See e.g. Florencia Marotta-Wurgler, *Online Markets vs. Traditional Markets: Some Realities of Online Contracting*, 19 S. CT. ECON. REV. 11 (2011).

⁷⁵ 15 U.S.C. § 45.

⁷⁶ 1998 FTC Report at 7.

⁷⁷ Id. at 7–8.

⁷⁸ Id. at 9.

⁷⁹ Protecting Consumer Privacy in an Era of Rapid Change (2010 FTC Report), available at <http://www.ftc.gov/os/2010/12/101201privacyreport.pdf>.

⁸⁰ Id. at iii.

⁸¹ Id. at 41. See also Ira S. Rubinstein, *Regulating Privacy by Design*, 26 BERKELEY TECH. L.J. 1409 (2011).

⁸² Agreement Containing Consent Order, In the Matter of Google, Inc., FTC File No. 102 3136 (March 30, 2011), available at

<http://www.ftc.gov/sites/default/files/documents/cases/2011/03/110330googlebuzzagreorder.pdf>.

⁸³ See e.g. Matter of Sears Holdings Mgmt. Corp., FTC Docket No. C-4264 (2009), <http://www.ftc.gov/enforcement/cases-and-proceedings/cases/2009/09/sears-holdings-management-corporation-corporation> (alleging “My SHC Community” application violated implied restriction to collecting data about ‘online browsing’ to collect data from sources including non-Internet-related activity); Matter of Chitika, Inc., FTC Docket No. C-4324 (2011), <http://www.ftc.gov/enforcement/cases-and-proceedings/cases/2011/06/chitika-inc-matter> (alleging 10-day expiration of ‘opt-out’ choice violated implied meaning of consumer choice to opt out); Matter of Scanscout, Inc., FTC Docket No. C-4344 (2011), <http://www.ftc.gov/enforcement/cases-and-proceedings/cases/2011/12/scanscout-inc-matter> (alleging use of Flash objects to circumvent users’ cookie deletion violated implications of privacy policy).

⁸⁴ Private parties generally may not “intercept any wire … or electronic communication,” 18 U.S.C. § 2511, “intentionally access without authorization a facility through which an electronic communication service is provided,” 18 U.S.C. § 2701, “intentionally exceed an authorization to access that facility and thereby obtain … a wire or electronic communication while it is in electronic storage in such system,” 18 U.S.C. § 2701, or use electronic means to intercept routing information, such as telephone numbers dialed, 18 U.S.C. § 3121.

⁸⁵ 18 U.S.C. § 2702.

⁸⁶ For recent discussions of these issues see e.g. Charles H. Kennedy, *An ECPA for the 21st Century: The Present Reform Efforts and Beyond*, 20 COMM LAW CONSPECTUS 129 (2011–12), and articles in the symposium *Big Brother in the 21st Century? Reforming the Electronic Communications Privacy Act*, 45 U.S.F. L. Rev. (2012).

⁸⁷ For information about reform efforts, see e.g. digitaldueprocess.org.

⁸⁸ 18 U.S.C. § 1030.

⁸⁹ 15 U.S.C. § 6801.

⁹⁰ See e.g. FTC Privacy of Consumer Financial Information Rule, <http://www.business.ftc.gov/documents/bus67-how-comply-privacy-consumer-financial-information-rule-gramm-leach-bliley-act>.

⁹¹ 15 U.S.C. § 1681 et seq.

⁹² 45 CFR § 160.102.

⁹³ 45 CFR § 164.514.

⁹⁴ See e.g. AMA Ethics Opinion 8.08, available at <http://www.ama-assn.org//ama/pub/physician-resources/medical-ethics/code-medical-ethics/opinion808.page>.

⁹⁵ See <http://www.ama-assn.org//ama/pub/physician-resources/legal-topics/patient-physician-relationship-topics/informed-consent.page>.

⁹⁶ See <http://www.nlm.nih.gov/medlineplus/ency/patientinstructions/000445.htm>.

⁹⁷ 45 CFR § 164.502. Certain disclosures ancillary to patient care, such as to caregiving family members, require an opportunity for the patient to agree or object orally. 45 CFR § 164.510.

⁹⁸ 45 CFR § 46.102.

⁹⁹ 45 CFR § 46.111.

¹⁰⁰ 45 CFR § 46.101(b).

¹⁰¹ 45 CFR § 164.512(i).

¹⁰² Professional ethics and institutional conflict of interest policies may provide additional constraints on marketing, but are not discussed here.

¹⁰³ 45 CFR § 164.502(5).

¹⁰⁴ 45 CFR § 164.508(a). There are limited exceptions to the authorization requirement for certain purposes related to public health or law enforcement.

¹⁰⁵ 45 CFR § 164.508(c).

¹⁰⁶ 45 CFR § 164.508(b)(5).

¹⁰⁷ 45 CFR § 164.508(3) & (4).

¹⁰⁸ 2010 FTC Report at 76–77.

¹⁰⁹ See <http://www.google.com/policies/privacy/>.

¹¹⁰ Katherine J. Strandburg, *Free Fall: The Online Market's Consumer Preference Disconnect*, 2013 CHI. LEGAL FORUM 95.

¹¹¹ See Omer Tene and Jules Polonetsky, *To Track or “Do Not Track”: Advancing Transparency and Individual Control in Online Behavioral Advertising*, 13 MINN. J.L. SCI. & TECH. 281 (2012) (“Policymakers must engage with the underlying normative question” and “cannot continue to sidestep these questions in the hope that ‘users will decide’ for themselves”). But see Thierer, 36 HARV. J. L. & PUBLIC POL’Y at 437 (arguing instead for a “flexible framework” comprising “education, empowerment, and

targeted enforcement of existing legal standards” to “help individuals cope with a world of rapidly evolving technological change and constantly shifting social and market norms as they pertain to information sharing”).

¹¹² See e.g. Dennis D. Hirsch, *Is Privacy Regulation the Environmental Law of the Information Age?* in Katherine J. Strandburg and Daniela S. Raicu, eds., *PRIVACY AND TECHNOLOGIES OF IDENTITY: AN INTERDISCIPLINARY CONVERSATION* (2005); Dennis D. Hirsch, *Achieving Global Privacy Rules through Sector-based Codes of Conduct*, 74 OHIO ST. L.J. 1029 (2013).

¹¹³ See e.g. Ira S. Rubinstein, *Regulating Privacy by Design*, 26 BERKELEY TECH. L.J. 1409 (2011) (suggesting regulatory mechanisms for encouraging and enforcing privacy by design); Kenneth A. Bamberger and Deirdre K. Mulligan, *Privacy on the Books and on the Ground*, 63 STAN. L. REV. 24 (2011) (discussing approaches to managing privacy risks, including privacy audits, currently undertaken by private companies).

¹¹⁴ Michael Froomkin has suggested this type of approach. A. Michael Froomkin, *Privacy Impact Notices*, Privacy Law Scholars Conference 2013, abstract available at <http://privacylaw.berkeleylawblogs.org/2013/05/24/a-michael-froomkin-privacy-impact-notices/>. The 2010 FTC Report suggests that companies develop “comprehensive data management programs” and that “[w]here appropriate, the programs also should direct companies to assess the privacy impact of specific practices, products, and services to evaluate risks and ensure that the company follows appropriate procedures to mitigate those risks.”

¹¹⁵ See e.g. Peter P. Swire, *Financial Privacy and the Theory of High-Tech Government Surveillance*, 77 WASH. U. L.Q. 461 (1999); Fred A. Cate, *Government Data Mining: The Need for a Legal Framework*, 43 HARV. C.R.-C.L. L. Rev. 435 (2008); Seth F. Kreimer, *Watching the Watchers: Surveillance, Transparency, and Political Freedom in the War on Terror*, 7 U. PA. J. CONST. L. 13 (2004). A discussion of the complex set of laws regulating surveillance and data acquisition by governments is beyond the scope of this chapter.

¹¹⁶ For a related suggestion, see Cate, 98 CAL. L. REV. at 1800 (suggesting that “personal information [] be provided to ‘licensed’ or ‘registered’ research facilities without any individual consent, but subject to strict privacy protections” and comparing this suggestion to the regime under FCRA, which “imposes strict requirements on ‘Consumer Reporting Agencies’ which are then allowed to collect consumer financial data without individual consent, but can only provide them to end users for ‘permissible purposes’ and subject to important restrictions on their disclosure and use”).

¹¹⁷ 2010 FTC Report at 45.

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

¹¹⁸ Id. at 53–54.

¹¹⁹ Id. at 53–57.

¹²⁰ The opportunity to opt in to first-party advertising can be provided concisely and with little burden on consumers. Many companies already provide such a process for users to consent to first-party advertising, such as discounts, newsletters, and ‘special offers’.

Chapter 2

Big Data's End Run around Anonymity and Consent

Solon Barocas and Helen Nissenbaum

Introduction

Big data promises to deliver analytic insights that will add to the stock of scientific and social scientific knowledge, significantly improve decision making in both the public and private sector, and greatly enhance individual self-knowledge and understanding. They have already led to entirely new classes of goods and services, many of which have been embraced enthusiastically by institutions and individuals alike. And yet, where these data commit to record details about human behavior, they have been perceived as a threat to fundamental values, including everything from autonomy, to fairness, justice, due process, property, solidarity, and, perhaps most of all, privacy.¹ Given this apparent conflict, some have taken to calling for outright prohibitions on various big data practices, while others have found good reason to finally throw caution (and privacy) to the wind in the belief that big data will more than compensate for its potential costs. Still others, of course, are searching for a principled stance on privacy that offers the flexibility necessary for these promises to be realized while respecting the important values that privacy promotes.

This is a familiar situation because it rehearses many of the long-standing tensions that have characterized each successive wave of technological innovation over the past half-century and their inevitable disruption of constraints on information flows through which privacy had been assured. It should come as no surprise that attempts to deal with new threats draw from the toolbox assembled to address earlier upheavals. Ready-to-hand, anonymity and informed consent remain the most popular tools for relieving these tensions – tensions that we accept, from the outset, as genuine and, in many cases, acute. Taking as a given that big data implicates important ethical and political values,² we direct our focus instead on attempts to avoid or mitigate the conflicts that may arise. We do so because the familiar pair of anonymity and informed consent continues to strike many as the best and perhaps only way to escape the need to actually resolve these conflicts one way or the other.

Anonymity and informed consent emerged as panaceas because they presented ways to ‘have it all’; they would open the data floodgates while ensuring that no one was unexpectedly swept up or away by the deluge. Now, as then, conscientious industry practitioners, policymakers, advocates, and researchers across the disciplines look to anonymity and informed consent as counters to the worrisome aspects of emerging

applications of big data. We can see why anonymity and consent are attractive: anonymization seems to take data outside the scope of privacy, as it no longer maps onto identifiable subjects, while allowing information subjects to give or withhold consent maps onto the dominant conception of privacy as control over information about oneself. In practice, however, anonymity and consent have proven elusive, as time and again critics have revealed fundamental problems in implementing both.³

The argument that we develop in this chapter goes further. Those committed to anonymity and consent do not deny the practical challenges; their solution is to try harder, to be more creative, to utilize more sophisticated mathematical and statistical techniques, and to become astute to the cognitive and motivational contours of users. Although we accept that improvements can result and have resulted from these efforts (e.g. more digestible privacy policies, more robust guarantees of anonymity, more usable choice architectures, and more supple policy), the transition to big data has turned definitional and practical fault lines that have worried policymakers, pundits, practitioners, and scholars into impassable chasms. After tracing progressive difficulties for anonymity and informed consent, respectively, we reveal virtually intractable challenges to both. In the case of anonymity, where important work has already shown it to be rather elusive, we argue that, even where strong guarantees of anonymity can be achieved, common applications of big data undermine the values that anonymity traditionally had protected. Even when individuals are not ‘identifiable’, they may still be ‘reachable’, may still be comprehensibly represented in records that detail their attributes and activities, and may be subject to consequential inferences and predictions taken on that basis. In the case of consent, too, commonly perceived operational challenges have distracted from the ultimate inefficacy of consent as a matter of individual choice and the absurdity of believing that notice and consent can fully specify the terms of interaction between data collector and data subject. Both, we argue, lead to the inescapable conclusion that procedural approaches cannot replace policies based on substantive moral and political principles that serve specific contextual goals and values.

Definitions and Background Theory

Many of the terms in this chapter have ambiguous and often contested meanings. To avoid disagreements originating in terminological differences, we specify the interpretations of two key terms – big data and privacy – assumed throughout the rest of this chapter. We have reason to believe that these interpretations contribute positively to the substantive clarity, but, for the most part, we set these out as starting assumptions.

Big Data

Taking into consideration wide-ranging uses of ‘big data’ in public discussions, specialized applications,⁴ government initiatives,⁵ research agendas,⁶ and diverse scientific,⁷ critical,⁸ and popular publications, we find that the term better reflects a paradigm than a particular technology, method, or practice. There are, of course, characteristic techniques and tools associated with it,⁹ but, more than the sum of these parts, big data, the paradigm, is a way of thinking about knowledge through data and a framework for supporting decision making, rationalizing action, and guiding practice.¹⁰ For better or worse, it is challenging entrenched epistemic and decision-making traditions across various domains, from climate science to medicine, from finance to marketing, from resource management to urban planning, and from security to governance.¹¹ Statistics, computer science, and information technology are crucial enablers and supporters of this paradigm,¹² but the ascent of big data involves, fundamentally, a belief in the power of finely observed patterns, structures, and models drawn inductively from massive datasets.¹³

Privacy as Contextual Integrity

There is some disagreement over how important privacy is among the various ethical and political issues raised by big data.¹⁴ Downplaying privacy, the argument is that *real* problems include how we use the data, whether it is fair to treat people as part of a group, whether data is representative, whether we diminish the range of choices we make about their own lives and fates, whether data about us and the data that we generate belong to us, invoking thereby justice, fairness, autonomy, and property rights. Revealing these wide-ranging ethical dimensions of big data is important, but an impoverished working conception of privacy can result in the failure to appreciate the crucial ways that these other values and privacy interact.

The conception we adopt here gives privacy a wider berth. To begin, we take privacy to be the requirement that information about people (‘personal information’) flows appropriately, where appropriateness means in accordance with informational norms. According to the theory of contextual integrity, from which this conception is drawn, informational norms prescribe information flows according to key actors, types of information, and constraints under which flow occurs (‘transmission principles’). Key actors include recipients, information subjects, and senders, where the last two are often one and the same. Social contexts form the backdrop for this approach to privacy, accounting for the range over which the parameters of actors, information types, and transmission principles vary. Put more concretely, informational norms for a health care context would govern flow between and about people in their context-specific capacities, such as physicians, patients, nurses, insurance companies, pharmacists, and so forth. Types of information would range over relevant fields, including, say, symptoms, diagnoses, prescriptions, as well as biographical information. And notable among

transmission principles, confidentiality is likely to be a prominent constraint on the terms under which information types flow from, say, patients to physicians. In drawing comparisons between contextual integrity and other theories of privacy, one key difference is that control over information about oneself is merely one in an indefinitely large class of transmission principles, not presumed unless the other parameters – (context specific) actors and information types – warrant it.¹⁵

Contextual informational norms, like other social norms, generally, are not fixed and static, but may shift, fade, evolve, and even reverse at varying rates, slowly or suddenly, sometimes due to deliberate cultural, legal, and societal alterations and other times in response to contingencies beyond human or societal control. Science and technology is a significant agent of change; in particular, computing and information technologies have been radically disruptive, enabling information practices that frequently diverge from entrenched informational norms. To explain why such disruptions are morally problematic – or rather to distinguish between those that are and are not – a norm-based account of privacy, such as contextual integrity, must offer a basis for drawing such distinctions. This enables a systematic critical perspective on informational norms in flux. For the theory of contextual integrity, the touchstones of moral legitimacy include interests and general moral and political values (and associated rights), commonly cited in accounts of privacy. Beyond these, however, a further distinctive set of considerations are context-specific ends, purposes, and values. Although this is not the place to elaborate in detail, consider as a quick illustration the rules limiting access to results of an HIV test. Generally, we might consider embarrassment, job security, danger to sexual partners, autonomy, various freedoms, and so on. Beyond these, however, contextual integrity further considers how the shape of access rules may affect whether people choose to undergo testing at all. As such, access rules could influence how effectively the purposes and values of the health care context are achieved. Ideal norms, therefore, are those that promote relevant ends, purposes, and values. And since the world is a messy place, rife with conflict and uncertainty, it is usually on the basis of partial knowledge only that we seek to optimize on these factors. In concrete circumstances where science and technology enable disruptions of entrenched norms, a heuristic supported by contextual integrity sets entrenched norms as default but allows that if novel practices are more effective in promoting interests, general moral and political values, and context-specific ends, purposes, and values, they should be favored over the status quo.

Now we are ready to weave together the disparate threads thus far spun. Big data involves practices that have radically disrupted entrenched information flows. From modes of acquiring to aggregation, analysis, and application, these disruptions affect actors, information types, and transmission principles. Accordingly, privacy, understood as contextual integrity, is fundamentally part of the big data story for it immediately alerts us to the ways any practice conflicts with the expectations we may have based on

entrenched information-flow norms. But that is merely the beginning. Evaluating disruptive practices means judging whether they move us closer or farther from ideal informational flows, that is, whether they are more or less effective in promoting interests, general moral and political values, and context-specific ends, purposes, and values. In other words, we proceed from observing disruptive flows to assessing their comparative impacts on ethical and political values, such as fairness, justice, freedom, autonomy, welfare, and others more specific to the context in question. Take, for example, an applicant who is denied admission to college based on predictive analytics performed on a dataset aggregated from diverse sources, including many that have not traditionally featured into admissions decisions. Imagine further that these additional sources allowed the college to discriminate – perhaps unwittingly – against applicants on the basis of criteria that happen to correlate with socioeconomic status and thus with the likely need for financial aid.¹⁶ While the outcome of such decisions may be judged unfair for many reasons worth discussing, it is the role of privacy – the role of disruptive informational flow – that we wish to note in this case.

Why, one may ask, insist on the centrality of privacy? First, doing so deepens our understanding of privacy and its instrumental value and at the same time highlights the distinctive ways that other ethical values are impinged and sustained, specifically, by the ways information does and does not flow. Privacy is important, in part, because it implicates these other values. Second, doing so also allows us to better formulate interventions, regulations, or remediation for the sake of these values. By keeping in view connections with specific information flows, certain options become salient that might otherwise not have been. Parsing cases in which big data gives rise to discrimination in terms of contextual integrity forces us to be much more specific about the source of that unfairness because it compels us to account for the disruption that made such discrimination possible.¹⁷ And it likewise allows us to ask if anonymity and informed consent limit or mitigate the potential consequences of such disruptions – that is, whether they actually protect the values at stake when novel applications of big data (threaten to) violate contextual integrity.

Anonymity

Anonymity obliterates the link between data and a specific person not so much to protect privacy but, in a sense, to bypass it entirely.¹⁸ Anonymity is an attractive solution to challenges big data poses to privacy when identities associated with information in a dataset are not necessary for the analysis to proceed. For those in search of group-level regularities, anonymity may allow for relatively unfettered access to databases. The greatest consensus around the utility of anonymization seems to have emerged in the sciences, including medicine, public and population health, urban planning, and

education, to name a few, with exciting prospects for advancing knowledge, diminishing risk, and improving decision making.¹⁹ But incumbents in many other sectors have begun to stake out this moral high ground by claiming that their analytics apply only to anonymized datasets, particularly those in marketing and other commercial sectors.²⁰

As we well know, however, anonymity is not unassailable. One of the earliest public demonstrations of its limits came with AOL's release of a large set of anonymized search queries with the stated purpose of facilitating academic research. This well-intended act backfired when a pair of enterprising news reporters identified a number of individuals based on the content of searches.²¹ Following these revelations, efforts to anonymize search query data, which were not particularly persuasive,²² have more or less fizzled out. The promise of anonymization was further chipped away by rigorous demonstrations by Sweeney, joint work by Narayanan and Shmatikov, and ongoing efforts by Dwork,²³ with implications further drawn by Ohm and others in areas of law and policy, where debates rage on.²⁴

It is impossible, within the scope of this article, to render anything close to a thorough account of the contemporary debate around anonymity; we merely mention key positions on threats to anonymity and attempts to defend it that are relevant to the general argument that we wish to develop. According to the literature, the promise of anonymity is impossible to fulfill if individual records happen to contain information – information that falls outside the scope of the commonly defined set of personally identifiable information – that nevertheless uniquely distinguishes a person enough to associate those records to a specific individual. So-called ‘vanity searches’ are an obvious example of this problem,²⁵ as AOL discovered,²⁶ but so, too, are records that contain extremely rich (e.g. location) data that necessarily map onto specific individuals.²⁷ The literature has also demonstrated many less obvious ways in which anonymity cannot be guaranteed due to the threat of so-called re-identification attacks.²⁸ These attacks depend on a variety of methods: overlaying an anonymized dataset with a separate dataset that includes identifying information, looking for areas of overlap (commonly described as a linkage attack)²⁹ or performing a sequence of queries on an anonymized dataset that allow the attacker to deduce that a specific person must be in the dataset because only one person has *all* of the queried attributes (differencing attack).³⁰ Responding to these challenges, computer scientists have developed a number of approaches to limit, if not eliminate, the chances of deducing identity, such as k-anonymity³¹ and differential privacy,³² which work in certain settings by abstracting or perturbing data to a level or degree set by data controllers. At the time of writing, this area of research is burgeoning, even though few real-world applications have been successfully implemented.

Let us review the main threads of this argument: anonymity is an attractive solution to challenges big data poses to privacy when identities associated with information in a dataset are not necessary for the analysis to proceed. Scientific and policy debates have

swirled around whether robust anonymization is possible and whether the impact of intractable challenges is a fringe phenomenon of little practical importance (and thus merely of academic interest) or fatal to the entire enterprise. *The concerns we have are neither about whether anonymization is possible nor about how serious a problem it poses for practical purposes; they are whether, in the first place, anonymization addresses privacy and related ethical issues of big data.* In so saying, we wish to shift the locus of attention away from the usual debates – conceding, at the same time, that they are extremely important and significant – to a different set of questions, where, for the sake of argument, we assume that the problem of anonymization, classically speaking, has been solved.

In order to see why anonymity does not solve ethical problems relating to privacy in a big data age, we should ask why we believe it does. And to do that, we need to ask not only whether in this age we are able to preserve the present-day equivalent of a traditional understanding of anonymity as namelessness, but whether this equivalent preserves what is at stake in protecting anonymity. In short, we need to ask whether it is worthwhile to protect whatever is being protected when, today, we turn to anonymity to avoid the ethical concerns raised by the big data paradigm.

Scholarship, judicial opinions, and legislative arguments have articulated the importance of anonymity in preserving and promoting liberal democratic values. We summarized these in earlier work, where we wrote that anonymity

offers a safe way for people to act, transact, and participate without accountability, without others ‘getting at’ them, tracking them down, or even punishing them. [As such, it] may encourage freedom of thought and expression by promising a possibility to express opinions, and develop arguments, about positions that for fear of reprisal or ridicule they would not or dare not do otherwise. Anonymity may enable people to reach out for help, especially for socially stigmatized problems like domestic violence, fear of HIV or other sexually transmitted infection, emotional problems, suicidal thoughts. It offers the possibility of a protective cloak for children, enabling them to engage in internet communication without fear of social predation or – perhaps less ominous but nevertheless unwanted – overtures from commercial marketers. Anonymity may also provide respite to adults from commercial and other solicitations. It supports socially valuable institutions like peer review, whistle-blowing and voting.³³

In this work, we argued that the value of anonymity inheres not in namelessness, and not even in the extension of the previous value of namelessness to all uniquely identifying information, but instead to something we called ‘reachability’, the possibility of knocking on your door, hauling you out of bed, calling your phone number, threatening you with

sanction, holding you accountable – with or without access to identifying information.³⁴ These are problematic because they may curtail basic ethical and political rights and liberties. But also at stake are contextual ends and values such as intellectual exploration, wholehearted engagement in social and economic life, social trust, and the like. The big data paradigm raises the stakes even further (to a point anonymity simply cannot extend and the concept of reachability did not locate) for a number of related reasons.

'Anonymous Identifiers'

First and perhaps foremost, many of anonymity's proponents have different meanings in mind, few of which describe practices that achieve unreachability. For example, when commercial actors claim that they only maintain anonymous records, they do not mean that they have no way to distinguish a specific person – or his browser, computer, network equipment, or phone – from others. Nor do they mean that they have no way to recognize him as the same person with whom they have interacted previously. They simply mean that they rely on unique persistent identifiers that differ from those in common and everyday use (i.e. a name and other so-called personally identifiable information (PII)). Hence the seemingly oxymoronic notion of an 'anonymous identifier', the description offered by, among others, Google for its forthcoming AdID,³⁵ an alternative to the cookie-based tracking essential for targeted advertising.³⁶ If its very purpose is to enable Google to identify (i.e. recognize) the same person on an ongoing basis, to associate observed behaviors with the record assigned to that person, and to tailor its content and services accordingly, AdID is anonymous only insofar as it does not depend on traditional categories of identity (i.e. names and other PII). As such, the identifier on offer does nothing to alleviate worries individuals might have in the universe of applications that rely on it. This understanding of anonymity instead assumes that the real – and only – issue at stake is how easily the records legitimately amassed by one institution can be associated with those held by *other* institutions, namely an association that would reveal the person's legal or real-world identity.³⁷

The reasons for adopting this peculiar perspective on anonymity becomes clear when we explore why names, in particular, tend to generate such anxiety. As a persistent and common identifier, names have long seemed uniquely worrisome because they hold the potential to act as an obvious basis for seeking out *additional* information that refers to the same person by allowing institutions to match records keyed to the same name. Indeed, this is the very business of commercial data brokers: "Acxiom and other database marketing companies sell services that let retailers simply type in a customer's name and zip code and append all the additional profile information that retailers might want".³⁸ But this is highly misleading because, as scholars have long argued, a given name and address is just one of many possible ways to recognize and associate data with a specific

person.³⁹ Indeed, *any* unique identifier or sufficiently unique pattern can serve as the basis for recognizing the same person in and across multiple databases.⁴⁰

The history of the Social Security Number is highly instructive here: as a unique number assigned to all citizens, the number served as a convenient identifier that *other* institutions could adopt for their own administrative purposes. Indeed, large institutions were often attracted to the Social Security Number because it was necessarily more unique than given names, the more common of which (e.g. John Smith) could easily recur multiple times in the same database. The fact that people had existing reasons to commit this number to memory also explains why other institutions would seize upon it. In so doing, however, these institutions turned the Social Security Number, issued by the government for administering its own welfare programs, into a *common* unique identifier that applied across multiple silos of information. A Social Security Number is now perceived as sensitive, not because of any quality inherent to the number itself, but rather because it serves as one of the few common unique identifiers that enable the straightforward matching of the disparate and detailed records held by many important institutions.

The history of the Social Security Number makes clear that any random string that acts as a unique persistent identifier should be understood as a pseudonym rather than an ‘anonymous identifier’,⁴¹ that pseudonyms place no inherent restrictions on the matching of records, and that the protective value of pseudonyms decreases as they are adopted by or shared with additional institutions.⁴² This is evident in the more recent and rather elaborate process that Facebook has adopted to facilitate the matching of its records with those maintained by outside advertisers while ensuring the putative anonymity of the people to whom those records refer:

*A website uses a formula to turn its users’ email addresses into jumbled strings of numbers and letters. An advertiser does the same with its customer email lists. Both then send their jumbled lists to a third company that looks for matches. When two match, the website can show an ad targeted to a specific person, but no real email addresses changed hands.*⁴³

While there might be some merit to the argument, advanced by a representative of the Interactive Advertising Bureau, that such methods demonstrate that online marketers are not in the business of trying “to get people’s names and hound them”,⁴⁴ they certainly fall short of any common understanding of the value of anonymity. They place no inherent limits on an institution’s ability to recognize the same person in subsequent encounters, to associate, amass, and aggregate facts on that basis, and to draw on these facts in choosing if and how to act on that person. The question is whether, in the big data era, this still constitutes a meaningful form of unreachability.

Comprehensiveness

A further worry is that the comprehensiveness of the records maintained by especially large institutions – records that contain no identifying information – may become so rich that they subvert the very meaning of anonymity.⁴⁵ Turow, for instance, has asked, “[i]f a company knows 100 data points about me in the digital environment, and that affects how that company treats me in the digital world, what’s the difference if they know my name or not?”⁴⁶ The answer from industry is that it seems to matter very little indeed: “The beauty of what we do is we don’t know who you are [...] We don’t want to know anybody’s name. We don’t want to know anything recognizable about them. All we want to do is [...] have these attributes associated with them.”⁴⁷ This better accounts for the common refrain that companies have no particular interest in who someone is because their ability to tailor their offerings and services to individuals is in no way limited by the absence of such information. And it helps to explain the otherwise bizarre statement by Facebook’s Chief Privacy Officer that they “serve ads to you based on your identity [...] but that doesn’t mean you’re identifiable.”⁴⁸ On this account, your legal or real-world identity is of no significance. What matters are the properties and behaviors that your identity comprises – the kinds of details that can be associated with a pseudonym assigned to you without revealing your actual identity. Where these details are sufficiently extensive, as is the case with platforms that deal in big data, and where all of these details can be brought to bear in deciding how to treat people, the protections offered by ‘anonymity’ or ‘pseudonymity’ may amount to very little.⁴⁹ They may enable holders of large datasets to act on individuals, under the cover of anonymity, in precisely the ways anonymity has long promised to defend against. And to the extent that results in differential treatment that limits available choices and interferes with identity construction, it threatens individual autonomy and social justice. For these reasons, Serge Gutwirth and Paul Hert have warned that if it is “possible to control and steer individuals without the need to identify them, the time has probably come to explore the possibility of a shift from personal data protection to data protection tout court.”⁵⁰ In other words, we can no longer turn to anonymity (or, more accurately, pseudonymity) to pull datasets outside the remit of privacy regulations and debate.

Inference

But even this fails to appreciate the novel ways in which big data may subvert the promise of such protections: inference. However troubling the various demonstrations by computer scientists about the challenge of ensuring anonymity, there is perhaps more to fear in the expanding range of facts that institutions can infer and upon which they have become increasingly willing to act. As Brian Dalessandro has explained, “a lot can be predicted about a person’s actions without knowing anything personal about them.”⁵¹ This is a subtle but crucially important point: insights drawn from big data can furnish

additional facts about an individual (in excess of those that reside in the database) without any knowledge of their specific identity or any identifying information. Data mining breaks the basic intuition that identity is the greatest source of potential harm because it substitutes inference for using identifying information as a bridge to get at additional facts. Rather than matching records keyed to the same name (or other PII) in different datasets, data mining derives insights that simply allow firms to guess at these qualities instead. In fact, data mining opens people up to entirely new kinds of assessments because it can extend the range of inferable qualities far beyond whatever information happens to reside in records elsewhere. And as Dalessandro again explains, firms that adopt these tactics may submit to few, if any, constraints, because “PII isn’t really that useful for a lot of predictive modeling tasks.”⁵² This explains a recent anecdote relayed by Hardy: “Some years ago an engineer at Google told me why Google wasn’t collecting information linked to people’s names. ‘We don’t want the name. The name is noise.’ There was enough information in Google’s large database of search queries, location, and online behavior, he said, that you could tell a lot about somebody through indirect means.”⁵³ These indirect means may allow data collectors to draw inferences about precisely those qualities that have long seemed unknowable in the absence of identifying information. Rather than attempt to de-anonymize medical records, for instance, an attacker (or commercial actor) might instead infer a rule that relates a string of more easily observable or accessible indicators to a specific medical condition,⁵⁴ rendering large populations vulnerable to such inferences even in the absence of PII. Ironically, this is often the very thing about big data that generates the most excitement: the capacity to detect subtle correlations and draw actionable inferences. But it is this very same feature that renders the traditional protections afforded by anonymity (again, more accurately, pseudonymity) much less effective.

Research Underwritten by Anonymity

The very robustness of the new guarantees of anonymity promised by emerging scholarship may have perverse effects if findings from the research that they underwrite provide institutions with new paths by which to infer precisely those attributes that were previously impossible to associate with specific individuals in the absence of identifying information. Ironically, this is the very purpose of differential privacy, which attempts to permit useful analysis of datasets while providing research subjects with certain guarantees of anonymity.⁵⁵ *However much these protect volunteers, such techniques may license research studies that result in findings that non-volunteers perceive as menacing because they make certain facts newly inferable that anonymity once promised to keep beyond reach.*

A recent study demonstrating that students suffering from depression could be identified by their Internet traffic patterns alone was met with such a reaction.⁵⁶ Much of

this seemed to stem from one of the applications that the researchers envisioned for their results: “[p]roactively discovering depressive symptoms from passive and unobtrusive Internet usage monitoring.”⁵⁷ The study is noteworthy for our purposes for having taken a number of steps to ensure the anonymity and privacy of its research subjects while simultaneously – if unintentionally – demonstrating the limits of those very same protections for anyone who might be subject to the resulting model. The point is not to pick on these or other academic researchers; rather, it is to show that anonymity is not an escape from the ethical debates that researchers should be having about their obligations not only to their data subjects, but also to others who might be affected by their studies for precisely the reasons they have chosen to anonymize their data subjects.

Informed Consent

Informed consent is believed to be an effective means of respecting individuals as autonomous decision makers with rights of self-determination, including rights to make choices, take or avoid risks, express preferences, and, perhaps most importantly, resist exploitation. Of course, the act of consenting, by itself, does not protect and support autonomy; individuals must first understand how their assent plays out in terms of specific commitments, beliefs, needs, goals, and desires. Thus, where anonymity is unachievable or simply does not make sense, informed consent often is the mechanism sought out by conscientious collectors and users of personal information.

Understood as a crucial mechanism for ensuring privacy, informed consent is a natural corollary of the idea that privacy means control over information about oneself. For some, these are the roots of privacy that must be respected in all environments and against all threats. Its central place in the regulation of privacy, however, was solidified with the articulation and spread of the Fair Information Practice Principles (FIPPs) in the domains of privacy law and countless data protection and privacy regulation schemes around the world. These principles, in broad brushstrokes, demand that data subjects be given notice, that is to say, informed who is collecting, what is being collected, how information is being used and shared, and whether information collection is voluntary or required.⁵⁸

The Internet challenged the ‘level playing field’ embodied in FIPPS.⁵⁹ It opened unprecedented modalities for collecting, disseminating, and using personal information, serving and inspiring a diverse array of interests. Mobile devices, location-based services, the Internet of things, and ubiquitous sensors have expanded the scope even more. For many, the need to protect privacy meant and continues to mean finding a way to support notice and choice without bringing this vibrant ecology to a grinding halt. This need has long been answered by online privacy policies offered to individuals as unilateral terms-of-service contracts (often dubbed ‘transparency and choice’ or ‘notice and consent’). In

so doing, privacy questions have been turned into practical matters of implementation. As in the arena of human subjects research, the practical challenge has been how to design protocols for embedding informed consent into interactions of data subjects and research subjects with online actors and researchers, respectively. In both cases, the challenge is to come up with protocols that appropriately model both notice and consent. What has emerged online are privacy policies similar to those already practiced in hard copy by actors in the financial sector, following the Gramm-Leach-Bliley privacy rules.⁶⁰

Over the course of roughly a decade and a half, privacy policies have remained the linchpin of privacy protection online, despite overwhelming evidence that most of us neither read nor understand them.⁶¹ Sensitive to this reality, regulatory agencies, such as the Federal Trade Commission, have demanded improvements focusing attention on (1) ways privacy policies are expressed and communicated so that they furnish more effective notice and (2) mechanisms that more meaningfully model consent, reviving the never-ending stalemate over opt-in versus opt-out.⁶² While the idea that informed consent *itself* may no longer be a match for challenges posed by big data has been floated by scholars, practitioners, advocates, and even some regulators,⁶³ such thinking has not entered the mainstream. As before, the challenge continues to be perceived as purely operational, as a more urgent need for new and inventive approaches to informing and consenting that truly map onto the states of understanding and assenting that give moral legitimacy to the practices in question.

In this chapter, we take a different path. We accept that informed consent is a useful privacy measure in certain circumstances and against certain threats and that existing mechanisms can and should be improved, but, against the challenges of big data, consent, by itself, has little traction. After briefly reviewing some of the better-known challenges to existing models of informed consent, we explore those we consider insurmountable.

The Transparency Paradox

There is little value in a protocol for informed consent that does not meaningfully model choice and, in turn, autonomy. The ideal offers data or human subjects true freedom of choice based on a sound and sufficient understanding of what the choice entails.

Community best practices provide standards that best approximate the ideal, which, because only an approximation, remains a subject of philosophical and practical debate.⁶⁴ Online tracking has been one such highly contentious debate⁶⁵ – one in which corporate actors have glommed onto the idea of plain language, simple-to-understand privacy policies, and plain-to-see boxes where people can indicate their assent or consent. A number of scholars continue to hold out hopes for this approach,⁶⁶ as do regulators, such as the FTC, who continues to issue guiding principles that reflect such commitments.⁶⁷ But situations involving complex data flows and diverse institutional structures representing disparate interests are likely to confront a challenge we have called ‘the

transparency paradox',⁶⁸ meaning that simplicity and clarity unavoidably results in losses of fidelity. Typical of the big data age is the business of targeted advertising, with its complex ecology of back-end ad networks and their many and diverse adjuncts. For individuals to make considered decisions about privacy in this environment, they need to be informed about the types of information being collected, with whom it is shared, under what constraints, and for what purposes. Anything less than this requires a leap of faith. Simplified, plain-language notices cannot provide information that people need to make such decisions. The detail that would allow for this would overwhelm even savvy users because the practices themselves are volatile and indeterminate as new parties come on board and new practices, squeezing out more value from other sources of information (e.g. social graphs), are constantly augmenting existing flows. Empirical evidence is incontrovertible: the very few people who read privacy policies do not understand them.⁶⁹ But the paradox identified above suggests that even when people understand the text of plain-language notices, they still will not – indeed cannot – be informed in ways relevant to their decisions whether to consent.

Indeterminate, Unending, Unpredictable

What we have said, thus far, emerges from a discussion of notice and choice applied to online behavioral advertising, but with clear parallels for the big data paradigm generally. Consider typical points of contact for data gathering: signing up for a smart utility meter, joining an online social network, joining a frequent flier program, buying goods and services, enrolling in a MOOC, enrolling in a health self-tracking program, traveling, participating in a medical trial, signing up for a supermarket loyalty card, clicking on an online ad, commenting on a book, a movie, or a product, applying for insurance, a job, a rental apartment, or a credit card. Because these mundane activities may yield raw material for subsequent analysis, they offer a potential juncture for obtaining consent, raising the natural question of how to describe information practices in ways that are relevant to privacy so that individuals meaningfully grant or withhold consent. The machinations of big data make this difficult because data moves from place to place and recipient to recipient in unpredictable ways. Further, because its value is not always recognized at collection time, it is difficult to predict how much it will travel, how much it will be in demand, and whether and how much it may be worth. In the language of contextual integrity, unless recipients and transmission principles are specified, the requirements of big data are for a blank check.

While questions of information type and use might, at first, seem straightforward, they are extremely difficult when considered in detail: it may be reasonably easy for a utility company to explain to customers that, with smart meters, it can monitor usage at a fine grain, can derive aggregate patterns within and across customers, and can use these as a basis for important decisions about allocation of resources and for targeted

advisement about individual customers' energy usage. It may clearly explain who will be receiving what information and to what end. With notice such as this, consent is meaningful. However, big data analytics typically do not stop here; an enterprising company may attempt to figure out how many people are associated with a given account, what appliances they own, their routines (work, bedtime, and vacations). It may fold other information associated with the account into the analysis and other information beyond the account – personal or environmental, such as weather. The company may extract further value from the information by collaborating with third parties to introduce further data fields. Not anomalous, practices such as these are the life blood of the big data enterprise for massive corporate data brokers and federal, state, and local government actors. How can they be represented to data subjects as the basis for meaningful consent?

Let us consider the challenges. The chain of senders and recipients is mazelike and potentially indefinite, incorporating institutions whose roles and responsibilities are not circumscribed or well understood. The constraints under which handoffs take place are equally obscure, including payments, reciprocity, obligation, and more. What can it mean to an ordinary person that the information will be shared with Axiom or Choicepoint, let alone the NSA? Characterizing the type of information is even tougher. Is it sufficient for the utility company to inform customers that it is collecting smart meter readings? The case is strong for arguing that notice should cover not only this information but, further, information that can be directly derived from it and even information that more sophisticated analysis might yield, including that which follows from aggregations of smart meter readings with information about other matters, personal or contextual. Intuitions on this matter are challenging, almost by definition, because the value of big data lies in the unexpectedness of the insights that it can reveal.

Even if we knew what it meant to provide adequate notice to ensure meaningful consent, we would still not have confronted the deepest challenges. One is the possibility of detecting surprising regularities across an entire dataset that reveal actionable correlations defying intuition and even understanding. With the best of intentions, holders of large datasets willing to submit them to analyses unguided by explicit hypotheses may discover correlations that they had not sought in advance or anticipated. A lot hangs on what informed consent means in such cases.⁷⁰ Does the data controller's obligation end with informing subjects about data that is explicitly recorded, or must the data controller adopt a more encompassing approach, explaining what further information the institution may be able to glean?⁷¹ If the more encompassing approach is taken, how does the data controller explain that it is impossible to know in advance what further information might be discoverable? These factors diminish the value of informed consent because they seem to require notice that does not delimit future uses of data and the possible consequences

of such uses. As many have now argued, consent under those conditions is not meaningful.⁷²

The Tyranny of the Minority

But big data troubles the long-standing focus on individual choice in a slightly more roundabout way because, as discussed earlier, the willingness of a few individuals to disclose certain information implicates everyone else who happens to share the more easily observable traits that correlate with the revealed trait. This is the tyranny of the minority: the volunteered information of the few can unlock the same information about the many. This differs markedly from the suggestion that individuals are ill equipped to make choices that serve their actual interests; rather, even if we accept that individuals can make informed, rational decisions concerning their own privacy, these decisions nonetheless affect what institutions (to whom these individuals have disclosed information) can now know (i.e. infer) about others.⁷³

Such inferences can be drawn in a number of ways. In registering some kind of connection to another person through the formal process of ‘friending’ on a social networking site, we signal that this is a person with whom we share certain interests, affinities, and history. In associating with this person, we open ourselves up to inferences that peg us as people who share certain qualities with this other person. This is the familiar trope about ‘the company I keep’: what my friends say and do – or rather, what they are willing to say and do on social networking sites – will affect what others think of me. Hence danah boyd’s point that “[i]t’s no longer about what you do that will go down on your permanent record. Everything that everyone else does that concerns you, implicates you, or might influence you will go down on your permanent record.”⁷⁴

Computer scientists have turned this into a formal problem, asking whether techniques drawing from social network analysis and data mining can be used to infer undisclosed attributes of a user based on the disclosed attributes of the user’s friends on social networking sites. And indeed a recent study has demonstrated that, where a certain portion of their friends disclose such facts, social networking sites may be able to infer users’ undisclosed major, graduation year, and dorm.⁷⁵ Other – more widely reported – research has also shown that homosexuality can be inferred with some reliability from the fact that a user holds a number of relationships and interacts with an otherwise disproportionate number of ‘out’ users.⁷⁶ Yet another study, building on this earlier work, has even shown that it is possible to make inferences about people who are not even a part of an online social network (i.e. to learn things about obviously absent *nonmembers*).⁷⁷

These demonstrations have tended to focus on cases of explicit association and the drawing of inferences based on confirmed relations, but, when we move away from discussions of online social networking, we find that no such explicit associations are

necessary to engage in this same kind of guesswork. More significantly, similar inferences can be made about an entire population even if only a small fraction of people who share no ties are willing to disclose. This describes the dynamics of the Target pregnancy prediction score.⁷⁸ In this case, Target did not infer the likelihood of a woman giving birth by looking at her group of friends; rather, the company looked over the records from its baby shower registry to find women who had actively disclosed the fact that they had given birth and then went about trying to figure out if these women's shopping habits, leading up to the baby shower, seemed to differ from other customers' habits such that Target could then recognize the telltale signs in the future shopping habits of *other* women.⁷⁹ Which is to say that Target was able to infer a rule about the relationship between purchases and pregnancy from what must have been a tiny proportion of all its customers who actually decided to tell the company that they recently had a baby. Not only is this the tyranny of the minority, it is a choice forced upon the majority by a minority with whom they have no meaningful or recognized relations.⁸⁰

Computer science researchers are tackling this question head-on: what proportion of people need to disclose that they possess a certain attribute for an adversary to then be able to identify all the other members in the population who also have this attribute? The findings from Mislove et al.'s study are rather startling: "multiple attributes can be inferred globally when as few as 20% of the users reveal their attribute information."⁸¹ Of course, reaching this minimum threshold is really just a matter of arriving at a sufficiently representative sample whose analysis generates findings that are generalizable to an entire population. As such, the value of any particular individual's withheld consent diminishes incrementally the closer the dataset of those who granted consent approaches representativeness – a point beyond which companies may have no further reason to pass. So long as a data collector can overcome sampling bias with a relatively small proportion of the consenting population,⁸² this minority will determine the range of what can be inferred for the majority and it will discourage firms from investing their resources in procedures that help garner the willing consent of more than the bare minimum number of people. In other words, once a critical threshold has been reached, data collectors can rely on more easily observable information to situate all individuals according to these patterns, rendering irrelevant whether or not those individuals have consented to allowing access to the critical information in question. Withholding consent will make no difference to how they are treated!

Conclusion

Those swept up in the great excitement that has placed big data at the forefront of research investment and the national scientific policy agenda may take courage. For them, these findings, particularly those concerning consent, prove once and for all that

privacy is an unsustainable constraint if we are to benefit, truly, from big data. Privacy and big data are simply incompatible and the time has come to reconfigure choices that we made decades ago to enforce certain constraints. The arguments presented here give further reason to dislodge privacy from its pedestal and allow the glorious potential of big data to be fulfilled.⁸³ We think these people are wrong in part because they adhere to a mistaken conception of privacy, often as control or as secrecy. Because they see privacy at odds with any distribution and use of data instead of focusing only on the inappropriate, they set up a false conflict from the start. They also may wrongly be conflating the *operationalization* of informed consent with informed consent *itself*.

Others say that we should remain concerned about ethical issues raised by big data, that, while privacy may be a lost cause, the real problems arise with use.⁸⁴ Those deserving urgent attention include unfair discrimination, being limited in one's life choices, being trapped inside stereotypes, being unable to delineate personal boundaries, being wrongly judged, embarrassed, or harassed.⁸⁵ Pursuing privacy as a way to address these issues is not only retrograde but a fool's errand, a conclusion reinforced by the arguments in our paper. Better, they would say, to route around privacy and pursue directly its ends. We agree that individual interests and ethical and, we would add, context-specific values are vitally important, but we think that it is reckless to sever, prematurely, the conceptual and practical ties between privacy and these moral and political ends. To fathom the ways that big data may threaten interests and values, we must distinguish among the origins and nature of threats to individual and social integrity, between, say, unfair discrimination originating in inappropriate information flows and unfair discrimination originating from other causes. For one thing, different sources may indicate different solutions.

We are not yet ready to give up on privacy, nor completely on anonymity and consent. The paradigm shift of big data calls for a paradigm shift in our responses and, though it may seem that the arguments of this chapter leave no place for anonymity and consent and, for some, therefore, no place for privacy, we reach different conclusions.

Let us begin with informed consent and imagine it foregrounded against a social landscape. In academic and regulatory circles, attention has focused on the foreground, suggesting ways to shape, tweak, and augment informed consent so that it covers everything important about the relationship between a data controller and a data subject. FIPPS and its innumerable descendants are a case in point. These efforts ensure that, in principle, nothing should go unremarked, unrevealed, unnoticed; in practice, informed consent has groaned under the weight of this burden with results – such as the transparency paradox – that have been noted here and elsewhere.

Informed consent also has a great legacy in the domain of human subjects research, where it remains the subject of ongoing deliberation, and has generated a mature philosophical literature. In *Rethinking Informed Consent in Bioethics*, philosophers Neil

Manson and Onora O'Neill address a concern, analogous to the one confronted by privacy researchers and regulators, over how to communicate with human subjects to ensure that consent is meaningful. They observe that the transaction of informed consent in medical treatment and biomedical research can only be understood against a rich social backdrop, which integrates medical practice and research into the background fabric of social and political life. When individuals – human subjects – enter into a study or treatment regime, they engage not as tabula rasa in a vacuum expecting that the protocol of informed consent will specify fully what will happen and respective rights, obligations, and responsibilities. It does not and cannot constitute the complete relationship between the medical researcher or practitioner and the subject. Instead, the protocol is set against a rich background of social and professional roles, ethical standards, and legal and other obligations, which shape a subject's reasonable expectations. Notice generally only covers notable departures from these expectations and consent is a limited and selective waiver of rights that subjects normally would expect to be respected. In other words, individuals understand that

obligations and expectations are presupposed by informed consent practices. When they are waived by giving consent, they are not discarded or marginalized: they are merely waived in limited ways, for a limited time, for a limited purpose. In consenting to an appendectomy I do not consent to other irrelevant incisions, or to incisions by persons other than the relevant surgeon. In consenting to take part in a clinical trial I do not consent to swallow other novel medicines, let alone medicines that are irrelevant to my condition. Informed consent matters because it offers a standard and controllable way of setting aside obligations and prohibitions for limited and specific purposes.⁸⁶

According to O'Neill and Manson, consent is not required for acceptable, expected behaviors, but only for those that depart from it. The burden on notice, therefore, is to describe clearly the violations of norms, standards, and expectations for which a waiver is being asked and not to describe everything that will be done and not done in the course of treatment or research, which both the researcher and the subjects can safely presume. Manson and O'Neill decline to produce a general or universal list of legal and ethical claims that applies to all treatment and research scenarios because, while all would surely include a common set of obvious prohibitions on, say, killing, stealing, injury, torture, fraud, deception, manipulations, and so forth, each would further include prohibitions and prescriptions relevant to the particular treatment or study in which subjects are engaged. For example, subjects may reasonably expect physicians, researchers, and others to perform in accordance with the training and professional commitments required in their respective fields, for example, to prescribe only the treatment and medication they believe to be the best and necessary for a patient's condition.

It is not sufficient for researchers to provide assurances that subjects are given a choice to waive or not to waive; they must be able to justify “actions that otherwise violate important norms, standards or expectations.”⁸⁷ According to O’Neill and Manson, “[a]ny justification of informed consent has therefore to start from a recognition of the underlying legal and ethical claims and legitimate expectations that are selectively waived by consent transactions, and the reasons individuals may have for waiving them in particular cases.”⁸⁸ In other words, selective waivers may not be requested for just anything but are acceptable under two conditions, either concerning actions for which individuals are presumed to have reasons to waive rights and obligations, or concerning actions that promise significant benefits to others and to society at large. In other words, consent cannot exist as an excuse for anything, a limitation further emphasized by the second and third key principles of scientific integrity in the treatment of human subjects, namely, justice and beneficence (or non-maleficence.) Scientists requesting a limited waiver must ensure that subjects are well informed of departures from expected behaviors and they should ensure that the waiver they are requesting is consistent with the reasons their subjects have for waiving these rights. But informed consent is constrained in one further, crucial way – namely, by the requirements of beneficence, non-maleficence, and justice. These constrain what a subject can be asked to consent to.

When we understand informed consent as a limited waiver of rights and obligations, certain aspects of existing practices applied to privacy come to light. To begin, since FIPPs have served as a guide to law and policy, the focus has been on specifying the characteristics of notice and consent and very little on rights and obligations. Drawing on Manson and O’Neill, it is quite clear why this has not worked; it is impossible, even absurd to believe that notice and consent can fully specify the terms of interaction between data collector and data subject. The arguments in our paper attest to this. For too long, we have focused on the foreground, working at it from every angle. In good faith, we have crammed into the notice and consent protocol all our moral and political anxieties, believing that this is the way to achieve the level playing field,⁸⁹ to promote the autonomy of data subjects, to energize a competitive marketplace for good data practices, and more. In our view, this became a futile effort at some point along the way for reasons we and others have repeatedly offered. It is time to contextualize consent by bringing the landscape into focus. It is time for the background of rights, obligations, and legitimate expectations to be explored and enriched so that notice and consent can do the work for which it is best suited.⁹⁰

Until now, the greatest obligation of data gatherers was either to anonymize data and pull it outside various privacy requirements or to inform and obtain consent. After charting the increasing difficulty of fulfilling these obligations in the face of big data, we presented the ultimate challenge: not of practical difficulty but of irrelevance. Where, for example, anonymizing data, adopting pseudonyms, or granting or withholding consent

makes no difference to outcomes for an individual, we had better be sure that the outcomes in question can be defended as morally and politically legitimate. When anonymity and consent do make a difference, we learn from the domain of scientific integrity that simply because someone is anonymous or pseudonymous or has consented does not by itself legitimate the action in question. A burden is upon the collector and user of data to explain why a subject has good reason to consent, even if consenting to data practices that lie outside the norm. That, or there should be excellent reasons why social and contextual ends are served by these practices.

We have argued that background and context-driven rights and obligations have been neglected in favor of anonymity and consent to the detriment of individuals and social integrity. Although our chapter will be deeply vexing to those who have placed anonymization and consent at the foundation of privacy protection, we welcome the shift in focus to the purposes to which data practices are being put and how these comport with individual interests as well as ethical, political, and context-driven values.

Acknowledgements The authors gratefully acknowledge research support from Intel Science and Technology Center for Social Computing, DHHS Strategic Healthcare Information Technology Advanced Research Projects on Security (SHARPS), NSF Cyber-Trust Collaborative Research (CNS-0831124), and Lady Davis Trust, The Hebrew University of Jerusalem.

Notes

¹ For a wide-ranging set of opinions on these matters, see David Bollier, *The Promise and Peril of Big Data* (Washington, DC: The Aspen Institute, 2010) and Janna Anderson and Lee Rainie, *The Future of Big Data* (Washington, DC: Pew Research Center, July 20, 2012).

² For a broad overview, see Solon Barocas, “Data Mining: An Annotated Bibliography,” *Cyber-Surveillance in Everyday Life: An International Workshop* (Toronto, Canada: University of Toronto, 12–15 May 2011), http://www.digitallymediatedsurveillance.ca/wp-content/uploads/2011/04/Barocas_Data_Mining_Annotated_Bibliography.pdf.

³ See e.g. Latanya Sweeney, “K-Anonymity: A Model for Protecting Privacy,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, no. 5 (October 2002): 557–570, doi:10.1142/S0218488502001648; Arvind Narayanan and Vitaly Shmatikov, “Robust De-Anonymization of Large Sparse Datasets” (presented at

the 2008 IEEE Symposium on Security and Privacy, IEEE, 2008), 111–125, doi:10.1109/SP.2008.33; Paul Ohm, “Broken Promises of Privacy: Responding to the Surprising Failure of Anonymization,” *UCLA Law Review* 57, no. 6 (August 2010): 1701–1777; Solon Barocas and Helen Nissenbaum, “On Notice: The Trouble with Notice and Consent” (presented at the Engaging Data: First International Forum on the Application and Management of Personal Electronic Information, Cambridge, MA, 2009); Lorrie Faith Cranor, “Necessary but Not Sufficient: Standardized Mechanisms for Privacy Notice and Choice,” *Journal on Telecommunications and High Technology Law* 10, no. 2 (Summer 2012): 273–445; Alessandro Acquisti and Jens Grossklags, “Privacy and Rationality in Individual Decision Making,” *IEEE Security and Privacy Magazine* 3, no. 1 (January 2005): 26–33, doi:10.1109/MSP.2005.22; Daniel J. Solove, “Privacy Self-Management and the Consent Dilemma,” *Harvard Law Review* 126, no. 7 (May 2013): 1880–1880.

⁴ James Manyika et al., *Big Data: The Next Frontier for Innovation, Competition, and Productivity* (McKinsey Global Institute, 2011).

⁵ *Demystifying Big Data: A Practical Guide to Transforming the Business of Government* (Washington, DC: TechAmerica Foundation, 2012).

⁶ “NSF Advances National Efforts Enabling Data-Driven Discovery” (Washington, DC: National Science Foundation, November 12, 2013).

⁷ E.g. *Big Data* (<http://www.liebertpub.com/big>).

⁸ E.g. *Big Data and Society* (<http://bigdatasoc.blogspot.com/p/big-data-and-society.html>).

⁹ See Tony Hey, Stewart Tansley, and Kristin Tolle, eds., *The Fourth Paradigm: Data-Intensive Scientific Discovery* (Redmond, WA: Microsoft Research, 2009); *Frontiers in Massive Data Analysis* (Washington, DC: The National Academies Press, 2013); Pete Warden, *Big Data Glossary* (Sebastopol, CA: O'Reilly Media, 2011).

¹⁰ Mireille Hildebrandt, “Defining Profiling: A New Type of Knowledge?” in *Profiling the European Citizen: Cross-Disciplinary Perspectives*, ed. Mireille Hildebrandt and Serge Gutwirth (Dordrecht, Netherlands: Springer, 2008), 17–45, doi:10.1007/978-1-4020-6914-7_2; danah boyd and Kate Crawford, “Critical Questions for Big Data,” *Information, Communication & Society* 15, no. 5 (June 2012): 662–679, doi:10.1080/1369118X.2012.678878; Christopher Steiner, *Automate This: How Algorithms Came to Rule Our World* (New York: Portfolio/Penguin, 2012); Viktor Mayer-Schönberger and Kenneth Cukier, *Big Data: A Revolution That Will Transform How We Live, Work, and Think* (New York: Houghton Mifflin Harcourt, 2013).

¹¹ Manyika et al., *Big Data: The Next Frontier for Innovation, Competition, and Productivity*; Steiner, *Automate This: How Algorithms Came to Rule Our World*; Mayer-Schönberger and Cukier, *Big Data: A Revolution That Will Transform How We Live, Work, and Think*.

¹² *Frontiers in Massive Data Analysis*.

¹³ See Usama Fayyad, “The Digital Physics of Data Mining,” *Communications of the ACM* 44, no. 3 (March 1, 2001): 62–65, doi:10.1145/365181.365198; David Weinberger, “The Machine That Would Predict the Future,” *Scientific American* 305, no. 6 (November 15, 2011): 52–57, doi:10.1038/scientificamerican1211-52; Foster Provost and Tom Fawcett, “Data Science and Its Relationship to Big Data and Data-Driven Decision Making,” *Big Data* 1, no. 1 (March 2013): 51–59, doi:10.1089/big.2013.1508; Vasant Dhar, “Data Science and Prediction,” *Communications of the ACM* 56, no. 12 (December 1, 2013): 64–73, doi:10.1145/2500499.

¹⁴ See e.g. Oscar H. Gandy Jr., “Consumer Protection in Cyberspace,” *triplecC: Communication, Capitalism & Critique* 9, no. 2 (2011): 175–189; Cynthia Dwork and Deirdre K. Mulligan, “It’s Not Privacy, and It’s Not Fair,” *Stanford Law Review Online* 66 (September 3, 2013): 35–40; Omer Tene and Jules Polonetsky, “Judged by the Tin Man: Individual Rights in the Age of Big Data,” *Journal on Telecommunications and High Technology Law*, August 15, 2013; Jonas Lerman, “Big Data and Its Exclusions,” *Stanford Law Review Online* 66 (September 3, 2013): 55–63; Kate Crawford and Jason Schultz, “Big Data and Due Process: Toward a Framework to Redress Predictive Privacy Harms,” *Boston College Law Review* 55, no. 1 (2014).

¹⁵ For a more detailed account, see Helen Nissenbaum, *Privacy in Context: Technology, Policy, and the Integrity of Social Life* (Stanford, CA: Stanford University Press, 2010).

¹⁶ Applicants’ ability to pay is already a controversial factor in the admissions decisions of many colleges; in locating less obvious correlates for the ability to pay, analytics may grant colleges the capacity to pursue similar ends without direct access to such information while also shielding such contentious practices from view.

¹⁷ Solon Barocas, “How Data Mining Discriminates,” in *Data Mining: Episteme, Ethos, and Ethics*, PhD dissertation, New York University (Ann Arbor, MI: ProQuest Dissertations and Theses, 2014).

¹⁸ Such was the thinking in the so-called HEW report, where anonymized datasets were treated differently and separately under the heading of ‘statistical databases’. Secretary’s

Advisory Committee on Automated Personal Data Systems, *Records, Computers and the Rights of Citizens* (U.S. Department of Health, Education, and Welfare, July 1973).

¹⁹ White House Office of Science and Technology Policy and the Networking and Information Technology R&D, *Data to Knowledge to Action*, Washington, DC, November 12, 2013, http://www.nitrd.gov/nitrdgroups/index.php?title=Data_to_Knowledge_to_Action.

²⁰ Emily Steel and Julia Angwin, “On the Web’s Cutting Edge, Anonymity in Name Only,” *The Wall Street Journal*, August 4, 2010.

²¹ Michael Barbaro and Tom Zeller, “A Face Is Exposed for AOL Searcher No. 4417749,” *The New York Times*, August 9, 2006.

²² Vincent Toubiana and Helen Nissenbaum, “An Analysis of Google Logs Retention Policies,” *Journal of Privacy and Confidentiality* 3, no. 1 (2011): 2.

²³ Sweeney, “K-Anonymity: A Model for Protecting Privacy;” Narayanan and Shmatikov, “Robust De-Anonymization of Large Sparse Datasets”; Cynthia Dwork, “Differential Privacy” (presented at the ICALP’06 Proceedings of the 33rd International Conference on Automata, Languages and Programming, Berlin: Springer, 2006), 1–12, doi:10.1007/11787006_1; Cynthia Dwork, “A Firm Foundation for Private Data Analysis,” *Communications of the ACM* 54, no. 1 (January 1, 2011): 86, doi:10.1145/1866739.1866758.

²⁴ Ohm, “Broken Promises of Privacy: Responding to the Surprising Failure of Anonymization”; Jane Yakowitz, “Tragedy of the Data Commons,” *Harvard Journal of Law & Technology* 25, no. 1 (Autumn 2012): 1–67; Felix T Wu, “Defining Privacy and Utility in Data Sets,” *University of Colorado Law Review* 84, no. 4 (2013): 1117–1177; Jane Bambauer, Krishnamurti Muralidhar, and Rathindra Sarathy, “Fool’s Gold: An Illustrated Critique of Differential Privacy,” *Vanderbilt Journal of Entertainment & Technology Law* 16 (2014).

²⁵ Christopher Soghoian, “The Problem of Anonymous Vanity Searches,” *I/S: A Journal of Law and Policy for the Information Society* 3, no. 2 (2007).

²⁶ Barbaro and Zeller, “A Face Is Exposed for AOL Searcher No. 4417749.”

²⁷ Yves-Alexandre de Montjoye et al., “Unique in the Crowd: The Privacy Bounds of Human Mobility,” *Scientific Reports* 3 (2013): 1376, doi:10.1038/srep01376.

²⁸ Khaled El Emam et al., “A Systematic Review of Re-Identification Attacks on Health Data,” ed. Roberta W Scherer, *PLoS ONE* 6, no. 12 (December 2, 2011): e28071, doi:10.1371/journal.pone.0028071.s001.

²⁹ Narayanan and Shmatikov, “Robust De-Anonymization of Large Sparse Datasets.”

³⁰ Dwork, “A Firm Foundation for Private Data Analysis.”

³¹ Sweeney, “K-Anonymity: a Model for Protecting Privacy.”

³² Dwork, “Differential Privacy.”

³³ Helen Nissenbaum, “The Meaning of Anonymity in an Information Age,” *The Information Society* 15, no. 2 (May 1999): 142, doi:10.1080/019722499128592.

³⁴ Of course, this is why anonymity, in certain contexts and under certain conditions, can be problematic.

³⁵ Alistair Barr, “Google May Ditch ‘Cookies’ as Online Ad Tracker,” *USA Today*, September 17, 2013.

³⁶ Ashkan Soltani, “Questions on the Google AdID,” *Ashkan Soltani*, September 19, 2013, <http://ashkansoltani.org/2013/09/19/questions-on-the-google-adid/>.

³⁷ In practice, this has tended to refer to what we commonly conceive as ‘contact information’.

³⁸ Natasha Singer, “Acxiom, the Quiet Giant of Consumer Database Marketing,” *The New York Times*, June 16, 2012.

³⁹ Gary T. Marx, “What’s in a Name? Some Reflections on the Sociology of Anonymity,” *The Information Society* 15, no. 2 (May 1999): 99–112, doi:10.1080/019722499128565.

⁴⁰ Arvind Narayanan and Vitaly Shmatikov, “Myths and Fallacies of ‘Personally Identifiable Information’,” *Communications of the ACM* 53, no. 6 (June 1, 2010): 24–26, doi:10.1145/1743558.

⁴¹ This explains the ongoing attempt, as part of European Data Protection reform, to broaden the definition of ‘personal data’ to cover any such data that allows for the “singling out” of individuals, whether or not they can be identified as traditionally understood. The Article 29 Working Party, for instance, has advised “that a natural person can be considered identifiable when, within a group of persons, (s)he can be distinguished from others and consequently be treated differently. This means that the notion of identifiability includes singling out.” *Statement of the Working Party on Current Discussions Regarding the Data Protection Reform Package* (European

Commission, February 27, 2013). For a more detailed discussion of the contours of this debate in the context of online behavioral advertising, see Frederik Zuiderveen Borgesius, “Behavioral Targeting: A European Legal Perspective,” *IEEE Security and Privacy Magazine* 11, no. 1 (January 2013): 82–85, doi:10.1109/MSP.2013.5.

⁴² This is not to suggest that pseudonyms are valueless. Most immediately, pseudonyms limit the potential to infer gender, race, national origin, religion, or class position from names that possess any such obvious associations. *One-off* pseudonyms (i.e., unique identifiers that are *not* common to multiple databases) also do not lend themselves to the kind of straightforward matching of records facilitated by traditional categories of identity. In principle, only the institution that assigns a one-off pseudonym to a specific person can recognize that person *according* to that pseudonym. And where this pseudonym has been abandoned or replaced (e.g. by expiring or deleting a cookie), even the institution that assigned it to a specific individual will no longer be able to recognize or associate prior observations with that person.

⁴³ Jennifer Valentino-Devries and Jeremy Singer-Vine, “They Know What You’re Shopping for,” *The Wall Street Journal*, December 7, 2012.

⁴⁴ “‘Drinking from a Fire Hose’: Has Consumer Data Mining Gone Too Far?,” *Knowledge@Wharton*, November 22, 2011, <http://knowledge.wharton.upenn.edu/article.cfm?articleid=2886>.

⁴⁵ Steel and Angwin, “On the Web’s Cutting Edge, Anonymity in Name Only.”

⁴⁶ “‘Drinking From a Fire Hose’: Has Consumer Data Mining Gone Too Far?”

⁴⁷ Cindy Waxer, “Big Data Blues: The Dangers of Data Mining,” *Computerworld*, November 4, 2013, http://www.computerworld.com/s/article/print/9243719/Big_data_blues_The_dangers_of_data_mining.

⁴⁸ Valentino-Devries and Singer-Vine, “They Know What You’re Shopping For.”

⁴⁹ This, too, explains the dispute over an article in the current draft of the proposed revision of the European Data Protection laws that stipulates that profiling “based solely on the processing of pseudonymous data should be presumed not to significantly affect the interests, rights or freedoms of the data subject.” For more details about this point of debate, see Monika Ermert, “EU Data Protection: Bumpy Piece of Road Ahead,” *Internet Policy Review*, October 24, 2013, <http://policyreview.info/articles/news/eu-data-protection-bumpy-piece-road-ahead/209>.

⁵⁰ Serge Gutwirth and Paul Hert, “Regulating Profiling in a Democratic Constitutional State,” in *Profiling the European Citizen: Cross-Disciplinary Perspectives*, ed. Mireille Hildebrandt and Serge Gutwirth (Dordrecht, Netherlands: Springer, 2008), 289, doi:10.1007/978-1-4020-6914-7_14.

⁵¹ Brian Dalessandro, “The Science of Privacy,” *Ad:Tech*, July 30, 2013, <http://blog.ad-tech.com/the-science-of-privacy/>.

⁵² Dalessandro, “The Science of Privacy.”

⁵³ Quentin Hardy, “Rethinking Privacy in an Era of Big Data,” *The New York Times*, June 4, 2012, <http://bits.blogs.nytimes.com/2012/06/04/rethinking-privacy-in-an-era-of-big-data/>.

⁵⁴ Solon Barocas, “Extending the Frontier of the Inferable: Proxies, Proximity, and Privacy,” in *Data Mining: Episteme, Ethos, and Ethics*.

⁵⁵ Dwork, “A Firm Foundation for Private Data Analysis.”

⁵⁶ The authors of the study, somewhat surprised by the fierce reaction to news of their findings, assembled and responded to various criticisms: Frances H. Montgomery et al., “Monitoring Student Internet Patterns: Big Brother or Promoting Mental Health?” *Journal of Technology in Human Services* 31, no. 1 (January 2013): 61–70, doi:10.1080/15228835.2012.756600.

⁵⁷ Raghavendra Katikalapudi et al., “Associating Internet Usage with Depressive Behavior among College Students,” *IEEE Technology and Society Magazine* 31, no. 4 (Winter 2012): 73–80, doi:10.1109/MTS.2012.2225462.

⁵⁸ Among other things, FIPPs also require that adequate steps be taken to secure the information.

⁵⁹ OECD, *The Evolving Privacy Landscape: 30 Years after the OECD Privacy Guidelines*, vol. 176, April 6, 2011, doi:10.1787/5kgf09z90c31-en.

⁶⁰ Gramm-Leach-Bliley Act, 15 USC, § 6801–6809. See also Chapters 1 and 7 in this volume.

⁶¹ Yannis Bakos, Florencia Marotta-Wurgler, and David R. Trossen, “Does Anyone Read the Fine Print? Testing a Law and Economics Approach to Standard Form Contracts,” *SSRN Electronic Journal* (2009), doi:10.2139/ssrn.1443256; Aleecia M. McDonald and Lorrie Faith Cranor, “The Cost of Reading Privacy Policies,” *I/S: a Journal of Law and Policy for the Information Society* 4, no. 3 (2008): 540–565; Aleecia M. McDonald et al., “A Comparative Study of Online Privacy Policies and Formats” (presented at the PETS

2009, Springer, 2009), 37–55. Also discussed in Helen Nissenbaum, “A Contextual Approach to Privacy Online,” *Daedalus*, 140, no. 4 (Fall 2011): 32–48.

⁶² Julie Brill, “Reclaim Your Name: Privacy in the Age of Big Data” (presented at the Sloan Cyber Security Lecture, Brooklyn, NY, 2013).

⁶³ In 2012, Microsoft hosted a series of ‘dialogs’ in which scholars, executives, advocates, and regulators discussed the future of notice and consent in the wake of big data, many of whom expressed this sentiment. For a summary of the full range of opinions at these events, see Fred H. Cate and Viktor Mayer-Schönberger, “Notice and Consent in a World of Big Data,” *International Data Privacy Law* 3, no. 2 (May 20, 2013): 67–73, doi:10.1093/idpl/ipt005. Similar arguments have been advanced in Mireille Hildebrandt, “Who Is Profiling Who? Invisible Visibility,” in *Reinventing Data Protection?* ed. Serge Gutwirth et al. (Dordrecht, Netherlands: Springer, 2009), 239–252, doi:10.1007/978-1-4020-9498-9_14; Christopher Kuner et al., “The Challenge of ‘Big Data’ for Data Protection,” *International Data Privacy Law* 2, no. 2 (April 23, 2012): 47–49, doi:10.1093/idpl/ips003; *Big Data and Analytics: Seeking Foundations for Effective Privacy Guidance* (Washington, DC: The Centre for Information Policy Leadership, February 28, 2013); Omer Tene and Jules Polonetsky, “Big Data for All: Privacy and User Control in the Age of Analytics,” *Northwestern Journal of Technology and Intellectual Property* 11, no. 5 (April 2013): 239–272; Ira Rubinstein, “Big Data: The End of Privacy or a New Beginning?” *International Data Privacy Law* 3, no. 2 (May 20, 2013): 74–87, doi:10.1093/idpl/ips036.

⁶⁴ Nir Eyal, “Informed Consent,” in *The Stanford Encyclopedia of Philosophy*, ed. Edward N Zalta, 2012.

⁶⁵ Lorrie Faith Cranor, “Can Users Control Online Behavioral Advertising Effectively?” *IEEE Security and Privacy Magazine* 10, no. 2 (n.d.): 93–96, doi:10.1109/MSP.2012.32; Pedro Giovanni Leon et al., “What Do Online Behavioral Advertising Privacy Disclosures Communicate to Users?” (presented at the WPES ’12 Proceedings of the 2012 ACM workshop on Privacy in the electronic society, New York, NY: ACM Press, 2012), 19–30, doi:10.1145/2381966.2381970; Omer Tene and Jules Polonetsky, “To Track or ‘Do Not Track’: Advancing Transparency and Individual Control in Online Behavioral Advertising,” *Minnesota Journal of Law, Science & Technology* 13, no. 1 (Winter 2012): 281–357; Frederik J. Zuiderveen Borgesius, “Consent to Behavioural Targeting in European Law - What Are the Policy Implications of Insights from Behavioural Economics?” *SSRN Electronic Journal* (2013), doi:10.2139/ssrn.2300969; Joseph Turow, “Self-Regulation and the Construction of Media Harms: Notes on the

Battle over Digital ‘Privacy,’ ” in *Routledge Handbook of Media Law*, ed. Monroe E Price, Stefaan Verhulst, and Libby Morgan (New York, NY: Routledge, 2013).

⁶⁶ “Carnegie Mellon Leads NSF Project to Help People Understand Web Privacy Policies,” *Carnegie Mellon News* (Pittsburgh, PA: Carnegie Mellon University, August 20, 2013). See Usable Privacy Policy Project: <http://www.usableprivacy.org/>.

⁶⁷ *Protecting Consumer Privacy in an Era of Rapid Change* (Washington, DC: Federal Trade Commission, March 2012).

⁶⁸ Helen Nissenbaum, “A Contextual Approach to Privacy Online,” *Daedalus* 140, no. 4 (October 2011): 32–48, doi:10.1162/DAED_a_00113.

⁶⁹ For a summary of the relevant research, see Solove, “Privacy Self-Management and the Consent Dilemma.”

⁷⁰ Mireille Hildebrandt, “Profiling and the Rule of Law,” *Identity in the Information Society* 1, no. 1 (December 19, 2008): 55–70, doi:10.1007/s12394-008-0003-1.

⁷¹ Data miners were concerned with the principle of consent for this very reason from early on in the field’s history; see e.g. Daniel E. O’Leary, “Some Privacy Issues in Knowledge Discovery: The OECD Personal Privacy Guidelines,” *IEEE Expert: Intelligent Systems and Their Applications* 10, no. 2 (1995): 48–59.

⁷² Herman T. Tavani, “KDD, Data Mining, and the Challenge for Normative Privacy,” *Ethics and Information Technology* 1, no. 4 (1999): 265–273, doi:10.1023/A:1010051717305; Mireille Hildebrandt, “Who Is Profiling Who?”, Mireille Hildebrandt, “Profiling and Aml,” in *The Future of Identity in the Information Society*, ed. Kai Rannenberg, Denis Royer, and André Deuker (Berlin: Springer, 2009), 273–310, doi:10.1007/978-3-642-01820-6_7; Serge Gutwirth and Mireille Hildebrandt, “Some Caveats on Profiling,” in *Data Protection in a Profiled World*, ed. Serge Gutwirth, Yves Poulet, and Paul De Hert (Dordrecht, Netherlands: Springer, 2010), 31–41, doi:10.1007/978-90-481-8865-9_2; Cate and Mayer-Schönberger, “Notice and Consent in a World of Big Data”; *Big Data and Analytics: Seeking Foundations for Effective Privacy Guidance*; Tene and Polonetsky, “Big Data for All: Privacy and User Control in the Age of Analytics”; Rubinstein, “Big Data: The End of Privacy or a New Beginning?”

⁷³ We should stress that this is not the same argument, advanced by a number of scholars, that the capacity to assess any particular individual depends on the willingness (or unwillingness, as the case may be) of other individuals to reveal data about themselves. The common example in these arguments is that *I am a good customer because you are less profitable*. Taking the example of car insurance, Tverdek explains that what

constitutes a ‘good’ driver is a statistical artifact that can only be made in contrast to a statistically ‘reckless’ driver. But the phenomena that we are describing here, to stick with the same example, is the capacity for insurers to predict that I am bad driver because I share certain qualities with the limited number of other bad drivers who chose to report their accidents. Edward Tverdek, “Data Mining and the Privatization of Accountability,” *Public Affairs Quarterly* 20, no. 1 (2006): 67–94. See also Scott R. Peppet, “Unraveling Privacy: The Personal Prospectus and the Threat of a Full Disclosure Future,” *Northwestern University Law Review* 105, no. 3 (2011): 1153–1203, and, more recently, Evgeny Morozov, “The Real Privacy Problem,” *MIT Technology Review*, October 22, 2013.

⁷⁴ danah boyd, “Networked Privacy” (presented at the Personal Democracy Forum 2011, New York, NY, 2011).

⁷⁵ Alan Mislove et al., “You Are Who You Know: Inferring User Profiles in Online Social Networks” (presented at the WSDM ’10 Proceedings of the Third ACM International Conference on Web Search and Data Mining, New York, NY: ACM Press, 2010), 251–260, doi:10.1145/1718487.1718519.

⁷⁶ Carter Jernigan and Behram F. T. Mistree, “Gaydar: Facebook Friendships Expose Sexual Orientation,” *First Monday* 14, no. 10 (September 25, 2009), doi:10.5210/fm.v14i10.2611.

⁷⁷ Emöke-Ágnes Horvát et al., “One Plus One Makes Three (for Social Networks),” ed. Sergio Gómez, *PLoS ONE* 7, no. 4 (April 6, 2012): e34740, doi:10.1371/journal.pone.0034740.s011.

⁷⁸ Charles Duhigg, “How Companies Learn Your Secrets,” *The New York Times Magazine*, February 16, 2012.

⁷⁹ Rachel Nolan, “Behind the Cover Story: How Much Does Target Know?” *The New York Times*, February 21, 2012, <http://6thfloor.blogs.nytimes.com/2012/02/21/behind-the-cover-story-how-much-does-target-know/>.

⁸⁰ Scholars have described this as the problem of ‘categorical privacy’,⁸⁰ whereby an individual’s apparent membership in a group reveals more about them than can be observed directly (i.e., inferring that they likely possess the same traits as other group members). But the focus of this line of thinking has been on the impossibility of individuals foreseeing these potential inferences, the problem of inaccurate stereotyping, and the absence of associational ties, rather than the limited amount of examples that would be necessary to induce the rule and then apply it to others. See, in particular,

Anton Vedder, “KDD: the Challenge to Individualism,” *Ethics and Information Technology* 1, no. 4 (1999): 275–281, doi:10.1023/A:1010016102284.

⁸¹ Mislove et al., “You Are Who You Know: Inferring User Profiles in Online Social Networks,” 255. We should put this result into context to ensure that we do not overstate its significance: Mislove et al. were looking at the relatively innocuous details posted by college students on Facebook, specifically their major, year of expected graduation, and college (at this particular university, students are assigned to a residential college where they tend to remain for the duration of their college career). The stakes are not especially high in this case. That said, these inferences were only based on the very limited set of variables (in fact, they only looked at the same attributes that they hoped to infer), rather than the far richer data that social networking sites accrue about their users. The authors speculate that inferences about far more sensitive attributes would be possible if the analysis were to consider a larger set of criteria that might prove statistically relevant.

⁸² Scholars have already proposed such a method: Christina Aperjis and Bernardo A. Huberman, “A Market for Unbiased Private Data: Paying Individuals According to Their Privacy Attitudes,” *First Monday* 17, no. 5 (May 4, 2012), doi:10.5210/fm.v17i5.4013.

⁸³ Yakowitz, “Tragedy of the Data Commons.”

⁸⁴ Cate and Mayer-Schönberger, “Notice and Consent in a World of Big Data”; *Big Data and Analytics: Seeking Foundations for Effective Privacy Guidance*; Tene and Polonetsky, “Big Data for All: Privacy and User Control in the Age of Analytics”; Rubinstein, “Big Data: the End of Privacy or a New Beginning?”

⁸⁵ Oscar H. Gandy, *Coming to Terms with Chance: Engaging Rational Discrimination and Cumulative Disadvantage* (Burlington, VT: Ashgate, 2009); Dwork and Mulligan, “It’s Not Privacy, and It’s Not Fair”; Tene and Polonetsky, “Judged by the Tin Man”; Omer Tene and Jules Polonetsky, “A Theory of Creepy: Technology, Privacy and Shifting Social Norms,” *Journal on Telecommunications and High Technology Law*, September 16, 2013; Gutwirth and Hildebrandt, “Some Caveats on Profiling”; Tal Z. Zarsky, “Mine Your Own Business!: Making the Case for the Implications of the Data Mining of Personal Information in the Forum of Public Opinion,” *Yale Journal of Law & Technology* 5 (2004): 1–57.

⁸⁶ Neil C. Manson and Onora O’Neill, *Rethinking Informed Consent in Bioethics* (New York: Cambridge University Press, 2012), 73.

⁸⁷ Ibid., 75.

⁸⁸ Ibid., 73.

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

⁸⁹ Secretary's Advisory Committee on Automated Personal Data Systems, *Records, Computers and the Rights of Citizens*.

⁹⁰ This idea has been picked up by the Federal Trade Commission, which, as one of the key recommendations of its 2012 report, suggested that “companies do not need to provide choice before collecting and using consumers’ data for practices that are consistent with the context of the transaction, consistent with the company’s relationship with the consumer,” *Protecting Consumer Privacy in an Era of Rapid Change*, vii.

Chapter 3

The Economics and Behavioral Economics of Privacy

Alessandro Acquisti

Introduction

Imagine a world in which consumers' preferences can be so precisely estimated by observing their online behavior, that firms are able to anticipate consumers' needs, offering the right product at exactly the right time. Imagine that same world, but now consider that extensive knowledge of consumers' preferences also allows precise inferences about their reservation prices (the maximum price each consumer will pay for a good), so that firms can charge different prices for the same product to each of their buyers, and absorb the entire surplus arising from an economic transaction.

Imagine a world in which the collection and analysis of individual health data allow researchers to discover the causes of rare deceases and the cures for common ones. Now, consider the same world, but imagine that employers are able to predict job candidates' future health conditions from few data points extracted from the latter's social network profiles – and then, imagine those employers making hiring decision based on those predictions, without the candidate's consent or even awareness.

The economics of privacy attempts to study the costs and benefits associated with personal information – for the data subject, the data holder, and for society as a whole. As a field of research, it has been active for some decades. Progresses in data mining, business analytics, and so-called big data, have the potential for magnifying the size and augmenting the scope of economic benefits and dangers alike. This chapter overviews the growing body of theoretical and empirical research on the economics and behavioral economics of privacy, and discusses how these streams of research can be applied to the investigation of the implications of consumer data mining and business analytics. Among the many possible interpretations of privacy, this capture focuses on its informational aspects: the trade-offs arising from the protection or disclosure of personal data.

Since the second half of the last century, progresses in information technology and the transformation of advanced economies into service economies have made it possible for organizations to monitor, collect, store, and analyze increasing amounts of individual data. Those progresses have also raised significant, and in some cases novel, privacy concerns. Attempting to analyze privacy in the age of big data from an economic perspective does not imply the assumption that all modern privacy issues have explicit monetary dimensions. Rather, this type of analysis stems from the realization that, with or without individuals' awareness, decisions that data subjects and data holders make about

personal data often carry complex trade-off. The mining of personal data can help increase welfare, lower search costs, and reduce economic inefficiencies; at the same time, it can be source of losses, economic inequalities, and power imbalances between those who hold the data and those whose data is controlled. For instance, a firm may reduce its inventory costs by mining and analyzing the behavior of many individual consumers; however, the infrastructure needed to carry out analysis may require substantial investments, and if the analysis is conducted in manners that raise consumers' privacy concerns, those investments may backfire. Likewise, a consumer may benefit from contributing her data to a vast database of individuals' preferences (for instance, by sharing music interests with an online vendor, and receiving in turn targeted recommendations for new music to listen to); that same consumer, having lost control over that data, may end up suffering from identity theft, price discrimination, or stigma associated with the information unintended parties can acquire about her.

This chapter offers an overview of the lessons that economics (and behavioral economics) can tell us regarding privacy in the age of big data. The chapter suggests that the microeconomic theory of privacy has brought forward arguments both supporting the view that privacy protection may increase economic efficiency in a marketplace, and decreases it: personal information, when shared, can become a public good whose analysis can reduce inefficiencies and increase economic welfare; when abused, it can lead to transfer of economic wealth from data subjects to data holders. Similarly, empirical evidence has been offered of both the benefits and costs, for data subjects and data holders alike, of privacy protection. It is unlikely that economics can answer questions such as what is the "optimal" amount of privacy and disclosure for an individual and for society – but it can help us think about the trade-offs associated with personal information.

The rest of this chapter first provides a brief summary of some relevant results from the micro economic theory of privacy (Section [Error! Reference source not found.](#)). It then describes the potential trade-offs associated with privacy and disclosure in the age of big data (Sections [Error! Reference source not found.](#) and [Error! Reference source not found.](#)), consumers' privacy valuations (Section [Error! Reference source not found.](#)), and behaviors (Section [Error! Reference source not found.](#)). It concludes by discussing the role of privacy enhancing technologies and market forces (Section [Error! Reference source not found.](#)) in balancing the value of data and the value of privacy.

Privacy and Economic Theory

Among the many heterogeneous dimensions of privacy (Solove 2006), formal economic analysis has predominantly (albeit not solely) focused on privacy as concealment of personal information – a form of information asymmetry (Akerlof 1970). For instance, before a consumer interacts with a seller, the seller may not have knowledge of the

consumer's "type" (such as her preferences, or her reservation price for a product). After the consumer has interacted with the seller (for instance, she has completed a purchase of a certain product at a certain price), it is the consumer who may not know how the seller is going to use the information it acquired through the transaction. The former is a case of hidden information; the latter, of hidden action.

Some of the earliest explicit economic discussions of privacy appeared in the literature near the end of the 1970s and the beginning of the 1980s – thanks in particular to the work of scholars belonging to the so-called Chicago School. Among them, Stigler (1980) argued that the protection of privacy may lower the quality of information about economic agents available in the marketplace. Hence, excessive protection of privacy rights may end up being economically inefficient and redistributive, as it may deny to the market the signals needed to allocate, compensate, and efficiently price productive factors. Similarly, Posner (1981) argued that concealing personal information may transfer costs from one party to another: for instance, the employer who cannot fully scrutinize the job candidate may end up paying the price of hiring an unsuitable employee. According to this view, legislative initiatives that favor privacy protection by restricting the activities of companies are likely to create inefficiencies, raise firm costs, and ultimately decrease economic welfare.

Hirshleifer (1980), however, took positions that may be considered alternative to Stigler and Posner. He noted that economic studies based on the assumption of neo-classically rational economic agents may not adequately capture the nuances of transactions that occur outside the logic of the market, such as those involving privacy. Earlier, Hirshleifer (1971) had also noted that investment in private information gathering may be inefficient: using private information may have redistributive effects, which leads to overinvestment in information gathering. A similar conclusion is found by Taylor (2004b), who finds that market forces alone may not guarantee efficient economic outcomes (under competition, firms have a private incentive to invest more than socially optimal into collecting larger amount of consumer data).

The contraposition of the results found by Stigler (1980) or Posner (1981) and those by Hirshleifer (1971) or Taylor (2004b) highlights a common theme in the economic literature on privacy: privacy costs and privacy benefits are inextricably related. Works by Varian (1996), Noam (1996), Taylor (2004a) and Acquisti and Varian (2005) offer further examples.

Varian (1996) noted that a general ban on the dissemination of personal data would not be in the interest of the consumer herself, as well as the firms she interacts with. A consumer may naturally be interested in disclosing certain personal traits to other firms (for example, her preferences and tastes, so as to receive services). However, the same consumer may have an interest in keeping other types of information private (for example, her reservation price for a particular good). Noam (1996), applying Ronald

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Coase's 'theorem' to the study of privacy, argued that in absence of transaction costs, the interaction between consumers interested in the protection of their data and firms interested in accessing will, under free market exchanges, lead to an equilibrium in which the agent with the greatest interest in either accessing the data or protecting it from access will be the one to actually achieve its goal – independently of the initial assignments of privacy rights or rights over access to consumer data. However, transaction costs and uncertainties regarding the initial assignment of rights over personal information are likely to be substantial in the interaction between consumers and firms, in which case it would no longer be guaranteed that market forces alone would produce the most efficient privacy outcomes. Similarly, both Taylor (2004a) and Acquisti and Varian (2005) studied the economic impact of tracking technologies that make customer identification possible. In these types of models (which typically analyze intertemporal interactions between consumers and merchants, and focus on consumers' reservation prices as private information the merchant is interested in inferring), when consumers are rational decision makers, a regulatory regime for privacy protection turns out not to be necessary. For instance, in Acquisti and Varian (2005), consumers who expect to be tracked can engage in strategic behaviors that render tracking counterproductive; to avoid this, firms must use consumer information to offer personalized services that consumers will value.

This series of microeconomic results suggests that not only does privacy protection (or lack thereof) carry both potential costs and potential benefits for data subjects and data holders alike; but also that economic theory should not be expected to be answer the question "what is the economic impact of privacy (or lack thereof) on consumer and aggregate welfare?" in an unambiguous, unequivocal manner. Economic analysis certainly can help us carefully investigate local trade-offs associated with privacy, but the economic consequences of privacy are nuanced, and the evolution of technologies for data mining and business intelligence are more likely to emphasize, rather than resolve, those nuances, as we highlight in the following section.

Markets for Privacy

Due to the concurrent evolution of Internet technologies, online business models, and data mining and business analytics tools, economic transactions of privacy relevance occur nowadays in different types of market. We distinguish three markets for privacy in this section.

The first type of transactions that have privacy relevance actually occur in the market for ordinary, non-privacy goods: in the process of exchanging or acquiring other products or services, individuals often reveal personal information, which may be collected, analyzed, and then used by the counterpart in the transaction in a variety of ways. In this case, the exchange of personal data, and the privacy implications of such exchange, are a secondary aspect of a primary transaction involving a good which is not, *per se*, privacy

related. An example of a transaction happening in this market may be the purchase of a book completed on an online merchant's site.

The second type of privacy-related transactions occur in what may be called the market for personal data. This market itself includes a variety of exchanges. One form of exchange involves 'infomediaries' that trade consumer data among themselves or with other data-holding firms. For instance, firms like Acxiom or credit reporting agencies like Transunion both acquire from, and sell to, consumer data by interacting with other consumer-facing firms. The data subjects are not generally active agents in these transactions. A second form of exchange involves so-called free products or services provided to consumers in exchange for their data. This market includes search engines and online social networks. In these exchanges, consumers are directly involved in the transaction, although the exchange of their personal information is not always a *visible*, explicit component of the transaction: while the price for services in this type of exchanges may be nominally zero, the customer is effectively purchasing the service by selling her data.

A third form of privacy-related transactions occur in what may be called the market for privacy. In this market, consumers explicitly seek products and services to manage and protect their personal information. For instance, they may acquire a privacy enhancing technology to protect their communications or hide their browsing behavior. The business models associated with providing consumers with more protection over their data have evolved rapidly, also due to the attention paid to the potential benefits associated with the sharing and mining of consumer data. Indeed, some business models end up being a bridge between the market for privacy and the market for personal data, in that they aim at giving consumers more 'ownership' over (exchanges involving) their personal information, including – sometimes – the potential ability to monetize it.

Privacy Trade-offs

The evolution and success of data mining and business analytics tools is and will keep affecting the markets described in the previous sections and their emerging trade-offs, especially in the form of both positive and negative externalities that arise when a consumers' data are aggregated and analyzed together with the data of many other consumers.¹ As anticipated in Section **Error! Reference source not found.**, the resulting costs and benefits for data subjects, data holders, and society at large are complex and nuanced. On the one hand, expected benefits can emerge from disclosed data for both data holders and data subjects (as well as opportunity costs when information is not shared or collected), together with the expected costs of the investments necessary to collect and process that data. On the other hand, expected benefits can arise from *protecting* data and expected costs can arise from privacy intrusions; however, costs are also associated with the protection of personal data. While a complete analysis of such dual benefits and costs

associated with either sharing or protecting data are outside the scope of this chapter, in this section we provide a few key examples that are especially relevant to the context of data mining and business analytics.

We first consider some of the benefits of data sharing, as well as some of the costs associated with data protection.

Firms can capitalize in various ways on the data of current and potential customers. Detailed knowledge of a consumer's preferences and behavior can help firms better target their products or ads, lowering advertising costs (Blattberg and Deighton 1991), providing enhanced, personalized services (Acquisti and Varian 2005), increasing consumer retention and loyalty, but also enforcing profit-enhancing price discrimination (Varian 1985) (although the latter may not always be the case in presence of competition; Fudenberg and Tirole 2000). For instance, the granular targetability made possible by online advertising may increase revenues for marketers and merchants (according to Beales, 2010, the price of behaviorally targeted advertising is almost three times as much the price of untargeted advertising). Similarly, by aggregating consumer data, firms can forecast trends and predict individual preferences, leading to the ability to provide valuable product recommendations (Bennett and Lanning 2007), or improve or redesign services based on observed behavior. Furthermore, revenues from targeted consumers may allow firms to provide lower-cost versions of a product to other consumers, and support services provided to consumers at a price of zero – but in exchange for their data.

Indeed, some of the benefits data holders gain from data may get passed on, or shared with, data subjects themselves (Lenard and Rubin 2009; Goldfarb and Tucker 2010), in the form of free content or services (made possible by advertising or personal data trades), personalized services, reduced search costs, or more efficient interactions with merchants or their sites. Positive consumer externalities may also materialize. For instance, better consumer data may allow firms to target the right consumers, reducing the amount of marketing investment that gets wasted with consumers uninterested in the product, and potentially leading to lower product prices (see also Blattberg and Deighton 1991), or aggregation of web searches of many individuals could help detect disease outbreaks (Wilson and Brownstein 2009), or the aggregation of location data could be used to improve traffic conditions and reduce road congestion. In other words, the aggregation of private data could create a public good, with societal benefits accruing from big data.

One should note, however, that many of the benefits consumers can enjoin from data sharing may also be obtained without the disclosure of *personally identified* data. In other words, repeating benefits of “big data” while protecting privacy may not necessarily be contradictory goals: as further discussed in Section **Error! Reference source not found.**, advancements in privacy enhancing technologies suggest that there exist many shades of grey between the polar extremes of absolute sharing and complete protection of personal data; rather, it is possible to selectively protect or disclose different types of personal

information, and modulate their identifiability, in order to optimize privacy trade-offs for individuals and society as a whole. As a result, benefits from data may be gained also when data is protected, and the actual societal costs of privacy protection may turn to be limited. For instance, the privacy enhancing provisions of the Fair Credit Reporting Act did not raise as significant barriers to profitable uses of consumer data as critics of the Act feared before its passage (Gellman 2002); the possible reduction of ads effectiveness caused by regulation limiting behavioral targeting may simply be offset by using ads on sites with specific content, larger ads, or ads with interactive, video, or audio features (Goldfarb and Tucker 2010); and certain types of privacy regulation in healthcare may actually foster innovation in the form of higher probability of success of Health Innovation Exchanges (Adjerid et al. 2013).

As for the costs that come from privacy violations or disclosed data, they can be both tangible and intangible both data holders and data subjects alike.

From the perspective of the data subject, Calo (2011) distinguishes between subjective and objective privacy harms: the former derive from unwanted perceptions of observation; the latter consist of the unanticipated or coerced use of information concerning a person against that person. Hence, the former relate to the anticipation, and the latter to the consequences, of losing control over personal information. Subjective harms may include anxiety, embarrassment, or fear; the psychological discomfort associated with feeling surveilled; the embarrassment associated when sensitive information is exposed publicly; or the chilling effects of fearing one's personal life will be intruded. These harms may be hard to capture and evaluate in economic terms, and usually are not recognized by U.S. courts as *actual* damage (Romanosky and Acquisti 2009). Objective harms could be both immediate and tangible, and indirect and intangible. They could include the damages caused by identity theft, the efforts spent deleting (or avoiding) junk mail; the time spent dealing with annoying telemarketing; the higher prices one pays due to (adverse) price discrimination; but also the effects of profiling, segmentation, and discrimination. For instance, profiling could be used to nudge consumers towards products that may not enhance their well-being,² and information revealed on a social network may lead to job market discrimination (Acquisti and Fong 2012). In more general terms, as an individual's data is shared with other parties, those parties may gain a bargaining advantage in future transactions with that individual. For instance, while a consumer, thanks to behavioral advertising, may receive targeted ads for products she is actually interested in, other entities (such as marketers and merchants) will accumulate data about the consumer that may permit the creation of a detailed dossier of her preferences and tastes, and the prediction of her future behavior. As noted in Section Error! Reference source not found., microeconomic models predict that, in presence of myopic customers, this information will affect the allocation of surplus of future transactions, increasing the share of the data holder over that of the data

subject. Ultimately, the disclosure of personal data affects the balance of power between the data subject and the data holder.

Data holders can also, under certain conditions, bear costs from the misuse of consumers data. For instance, Romanosky and Acquisti (2009) note that, following a data breach, firms suffer in terms of consumer notification costs, fines, settlement costs, stock market losses, or loss of consumer's trust. However, it may often be the case that a firm could externalize the privacy costs of using consumer data, while internalizing much of the gains (Swire and Litan 1998).

Often, objective privacy harms are merely probabilistic: once data is revealed or intruded, it may or may not lead to the actual negative consequences we have just described. For instance, poor data handling practices by a consumer report firm may later cause a consumer's mortgage request to be wrongfully denied; or, the breach of a database containing consumers' credit cards may later lead to identity theft (Camp 2007). The metaphor of a 'blank check' has been used to refer to the uncertainty associated with privacy costs: disclosing personal information is like signing a blank check, that may never be cashed in – or perhaps cashed it at some unpredictable moment in time with an indeterminably low, or high, amount to pay. In economic terms, the damage from disclosed data are, in Knight (1921)'s terms, *ambiguous* and, up to a point, unknowable.³ Consider, in fact, that some privacy costs are high-probability events with negligible individual impact (for instance, spam); other costs are low-probability events with very significant adverse consequences (for instance, some of the costs associated with the more pernicious forms of identity theft). Because of this and their often intangible dimensions, privacy costs may be hard to assess and therefore also to act upon. Either because of low likelihood of occurrence, or limited perceived magnitude of damage, privacy costs may therefore dismissed as unimportant at the individual level – even when, in the aggregate, they may amount to significant societal damage, or a significant transfer of wealth from data subjects to others (including the data holders).

Do Consumers Value Privacy?

Farrell (2012) notes that privacy is both a final and an intermediate good: "[c]onsumers care about privacy in part for its own sake: many of us at least sometimes feel it's just icky to be watched and tracked. [...] Consumers also care about privacy in a more instrumental way. For instance, loss of privacy could identify a consumer as having a high willingness to pay for something, which can lead to being charged higher prices if the competitive and other conditions for price discrimination are present." In this section, we summarize a number of empirical investigations of consumers' privacy valuations. In the following section (Section **Error! Reference source not found.**), we examine the hurdles consumers face in making privacy decisions consistent with those valuations.

Numerous factors influence individuals' privacy concerns (Milberg et al. 1995), and therefore the mental 'privacy calculus' that individuals make when deciding whether to protect or disclose personal information (Laufer and Wolfe 1977; Culnan and Armstrong 1999; Dinev and Hart 2006). Researchers from diverse disciplines (such as economics, marketing, information systems, and computer science) have attempted to estimate empirically the value that, in this calculus, individuals assign to privacy and their personal data. The resulting findings suggest that privacy valuations are significantly context dependent. Furthermore, willingness to pay, or reservation prices, may not adequately capture the value of privacy for those individuals who simply do not feel they should have to pay to protect their privacy.

Huberman et al. (2005) used a second-price auction to estimate the price at which individuals were willing to publicly reveal personal information such as their weight. Individuals whose weight was more deviant from the perceived norm for the rest of the group were more likely to exhibit higher valuations. Wathieu and Friedman (2005) found that survey participants were more accepting of an organization sharing their personal information after having been explained the economic benefits of doing so. Cvrcek et al. (2006) reported large differentials across EU countries in the price EU citizens would accept to share mobile phone location data. Hann et al. (2007) focused on online privacy and, using a conjoint analysis, found that protection against errors, improper access, and secondary use of personal information was worth US\$30.49-44.62 among U.S. subjects. Rose (2005) found that, although most participants in a survey self-reported being very sensitive to privacy issues, less than half of them would be willing to pay roughly \$29 to have their privacy protected by means of property rights on personal information. Both Varian et al. (2005) and Png (2007) estimated U.S. consumers' implicit valuation of protection from telemarketers using data about the Do Not Call list adoptions. They found highly differing values, from a few cents to as much as \$30. Tsai et al. (2011) found that, when information about various merchants' privacy policies was made available to them in a compact and salient manner, subjects in an experiment were more likely to pay premia of roughly 50 cents to purchase products from more privacy protective merchants.

At the same time, various studies have highlighted a dichotomy between self professed privacy attitudes and actual self-revelatory behavior.

Tedeschi (2002) reported on a Jupiter Research study in which the overwhelming majority of surveyed online shoppers would give personal data to new shopping sites for the chance to win \$100. Spiekermann et al. (2001) found that even participants in an experiment who could be classified as privacy conscious and concerned were willing to trade privacy for convenience and discounts: differences across individuals in terms of reported concerns did not predict differences in self-revelatory behavior. Similar findings were obtained in different settings by Acquisti and Grossklags (2005) and Acquisti and Gross (2006). Coupled with the observation that businesses focused on providing privacy

enhancing applications have met difficulties in the marketplace (Brunk 2002), these results suggest a potential privacy paradox: people want privacy, but do not want to pay for it, and in fact are willing to disclose sensitive information for even small rewards (for an overview of this area, see Acquisti, 2004, and Acquisti and Grossklags, 2007). In fact, Acquisti et al. (2013) have recently presented an application of the endowment effect to the privacy domain: subjects who started an experiment from positions of greater privacy protection were found to be five times more likely than other subjects (who did not start with that protection) to forego money to preserve their privacy. These results illustrate the challenges with pinpointing exact valuations of personal data: consumers' privacy valuations are not only context dependent, but affected by numerous heuristics and biases (see Section **Error! Reference source not found.**), and so are individuals' decisions to share or to protect personal information (John et al. 2011). In addition, awareness of privacy risks (and potential solutions to privacy threats) may also significantly affect consumers' privacy choices valuations – which is why revealed preferences arguments (that rely on observing consumer's choices in the marketplace – for instance, in the case of privacy, their propensity to share data online or to use protecting technology) may not necessarily provide the fuller or clearer picture of what privacy is ultimately worth to individuals. We discuss some of those hurdles that affect privacy decision making and valuations in the following section.

Hurdles in Privacy Behavior

A stream of research investigating the so-called privacy paradox has focused on the hurdles that hamper individuals' privacy-sensitive decision making. If consumers act myopically, or not fully rational (in the neo-classical economic sense of utility-maximizing, Bayesian-updates agents who make use of all the information consumers available to them), then market equilibria may not in fact guarantee privacy protection. In fact, in absence of regulatory protection of consumers' data, firms will tend to extract the surplus generated in transaction in which consumers' data is used for price discrimination (Acquisti and Varian 2005; Taylor 2004a).

There is, indeed, evidence that consumers face known decision making hurdles when facing privacy trade-offs, such as (a) incomplete information, (b) bounded cognitive ability to process the available information, and (c) a number heuristics (or cognitive and behavioral biases) which lead to systematic deviations from theoretically rational decision making (sometimes, various combinations of these factors affect consumer decision making at the same time).

Consider, first, the problem of incomplete information. In many scenarios – such as those associated with behavioral monitoring and targeting – the consumer may not even realize the extent at which her behavior is being monitored and exploited. Furthermore, after an individual has released control on her personal information, she is in a position of

information asymmetry with respect to the party with whom she is transacting. In particular, the subject might not know if, when, and how often the information she has provided will be used. For example, a customer might not know how the merchant will use the information that she has just provided to the merchant through a website.

Furthermore, the ‘value’ itself of the individual’s information might be highly uncertain and variable. The subject and the parties she is interacting with may evaluate differently the same piece of information, and the specific environmental conditions or the nature of the transaction may affect the value of information in unpredictable ways. For example, a customer might not know what damage she will incur because of her personal information becoming known, she might not know how much profit others will make thanks to that information, or she might not know the benefits she will forego if her privacy is violated. To what, then, is the subject supposed to anchor the valuation of her personal data and its protection?

Second, findings from behavioral economics exhaustively document consumers’ inability to exhaustively consider the possible outcomes and risks of data disclosures, due to bounded rationality. Furthermore, the individual will often find herself in a weaker bargaining position than other parties she is interacting with (for instance, merchants). In many transactions, the individual is unable to negotiate a desired level of information protection; she rather faces take-it-or-leave-it offers of service in exchange for personal data.

Third, even if the consumer had access to complete information about all trade-offs associated with data sharing and data protection, she will suffer from cognitive and behavioral biases that are more intense in scenarios where preferences are more likely to be uncertain. One such example is that, if the expected negative payoff from privacy invasions could be estimated, some individuals might seek immediate gratification, discounting hyperbolically (Rabin and O’Donoghue 2000) future risks (for example of being subject to identity theft), and choosing to ignore the danger. Hence, because of asymmetric information, self-gratification bias, overconfidence, or various other forms of misrepresentation studied in the behavioral economic literature, individuals might choose not to protect their privacy *possibly* against their own best interest. They might be acting *myopically* when it comes to protecting their privacy even when they might be acting *strategically* (as rational agents) when bargaining for short-term advantages such as discounts (Acquisti 2004).

Consider, for instance, the case of data breaches. As discussed in Romanosky and Acquisti (2009), after being notified of a breach of her financial information, a consumer may not be able to identify the right course of action: should she, for instance, punish the financial firm that, due to faulty security controls, compromised her data, by changing to a competitor? While this may appear as a risk-reducing behavior, by doing so the consumer would have now disclosed her personal information to another firm – and

actually materially increased the probability that another future breach will involve her data. Furthermore, the cost of acting may be significant: calling the breached firm to obtain details about the breach and its consequences, notifying financial institutions of the occurred breach and of potentially compromised accounts, or subscribing to credit alert and insurance services, are all actions which carry perceived cognitive, transaction, and actual costs. Such costs may appear greater to the consumer than the perceived benefit from action. It could also be that, because of psychological habituation due to repeated instances of data breaches report in the media, the consumer may become desensitized to their effects – which counter the desired impact of notifications. Ultimately, the consumer may ‘rationally’ decide to remain ‘ignorant’ (following the Choicepoint breach, fewer than 10% of affected individuals availed themselves of the free credit protection and monitoring tools offered by Choicepoint; Romanosky and Acquisti 2009). This example suggests how nuanced and full of obstacles is the path that lead from consumer notification of privacy problem to her actually taking action to solve that problem.

Based on these hurdles, recent behavioral privacy research has questioned the validity and effectiveness of regimes based on transparency and control mechanism (also known as choice and notification; Brandimarte et al. 2013; Adjerid, Acquisti, Brandimarte, and Loewenstein, Adjerid et al.; Acquisti et al., 2013).⁴

An improved understanding of cognitive and behavioral biases that hamper privacy (and security) decision making, however, could also be exploited for normative purposes. Specifically, knowledge of those biases could be used to design technologies and policies that anticipate and counter those very biases (Acquisti 2009). Such technologies and policies would be informed by the growing body of behavioral economics research on soft or asymmetric paternalism (Loewenstein and Haisley 2008) as well as research on privacy and security usability. They may help consumers and societies achieve their desired balance between information protection and information sharing.

Technology, Regulation, and Market Forces

As noted in Acquisti (2010), progresses in computer science, statistics, or data mining have not only produced potentially privacy-eroding business analytics tools or big data technologies; they have also led to the development, over the past few decades, of privacy enhancing technologies which allow the protection of (certain) individual data simultaneously to the sharing, or analysis, or aggregate, de-identified, or non-sensitive identified data. Online activities and transactions for which privacy preserving correspondents exist include electronic payments (Chaum 1983), online communications (Chaum 1985), Internet browsing (Dingledine et al. 2004), credentials (Camenisch and Lysyanskaya 2001), or even online recommendations (Canny 2002). One of the most interesting direction of research relates to executing calculations in encrypted spaces (Gentry 2009), and whether these types of computations will make it possible to have

both privacy *and* big data, confidentiality *and* analytics. In the best-case scenario, the deployment of privacy enhancing technologies may result in a win–win for data holders and data subjects: certain data is protected (thereby avoiding costs associated with certain privacy intrusions), whereas other data gets shared, analyzed, and used (thereby enjoying the benefits and the value of data, big or small). Alternatively, the old economic adage that there is no free lunch may apply: whenever protection is applied to a dataset, the utility of that dataset is decreased (Duncan et al. 2001). The interesting economic question then becomes, whose utility will be adversely affected – or, in other words, who will bear the costs if privacy enhancing technologies become more popular in the age of big data: data subjects (whose benefits from business analytics and big data would shrink with the amount of information they share), data holders (who may face increasing costs associated with collecting and handling consumer data), or both?

Attempting to answer the above question remains an open research question. An additional and related open question, however, is whether, even if privacy enhancing technologies were found to increase data subjects' welfare more than they would adversely affect data holders' welfare, market forces alone would lead to the deployment and success of those technologies. While there is no lack of evidence online of both disclosure/publicity seeking and privacy seeking behavior, privacy enhancing technologies (as opposed to security technologies such as anti-viruses or firewalls) have not gained widespread adoption. Several reasons may explain this situation: on the consumers' side, a first obvious explanation is low consumer demand for privacy; however, other, more nuanced (and non-mutually exclusive) explanations include users' difficulties and costs in using privacy technologies (see Whitten and Tygar 1999), switching costs, as well as biases such as immediate gratification, which reduce demands for those products even by privacy sensitive consumers. On the data holders' side, in absence of regulatory intervention, or of clear evidence that privacy protection can act as a distinctive source of competitive advantage for a firm, it is unlikely that firms will incur the costs to transition to technologies that may, in the short run, limit their access to consumer data relative to their competitors.

The debate over the comparative economic advantages of regulation and self-regulation of privacy remains intense to this date. On the one hand, Gellman (2002) challenges the view that the unrestricted trafficking in personal information always benefits the consumer, and that privacy trade-offs may merely be evaluated on the basis of monetary costs and benefits. He concludes that an unregulated, privacy-invasive market in personal data can be costly for consumers. F. H. Cate (2002), Cate et al. (2003), Rubin and Lenard (2001), and Lenard and Rubin (2009), on the other hand, claim that legislative initiatives that restrict the amount of personal information available to business would actually penalize the consumers themselves: regulation should be undertaken only when a

given market for data is not functioning properly, and when the benefits of new measures outweigh their costs.

It may not be possible to resolve this debate using purely economic tools. Economic theory, as we have discussed above, has brought forward arguments both supporting the view that privacy protection *increases* economic efficiency, and that it *decreases* it. Empirically, the costs and benefits associated with the protection and revelation of consumers' data have not proven easily amenable to aggregation: First, as soon as one attempts an aggregate evaluation of the impact of privacy regulation, one faces the challenge of delimiting the problem: data breaches, identity theft, spam, profiling, or price discrimination are all examples of privacy problems, yet they comprise very different expected benefits and costs for the parties involved. Second, even within each scenario, it may be hard to statically measure at a point in time the aggregate costs and benefits of data protection and data sharing, since the benefits and costs of privacy happen over time (for instance, data revealed today may only damage the individual years from now). And third, in addition to measurable outcomes (such as the financial losses due to identity theft, or the opportunity costs of spam), other privacy invasions require an estimation of consumers' valuations of privacy. Furthermore, as we have noted elsewhere in this chapter, numerous of the benefits associated with data disclosure may, in fact, still be gained when data is protected. Evaluations and conclusions regarding the economic value of privacy and the optimal balance between disclosure and protection are, therefore, far from simple.

Acknowledgement The author would like to thank the editors, the anonymous reviewers, Veronica Marotta, and Laura Brandimarte for particularly insightful comments and suggestions. This chapter is partly based on previous work by the author, including Acquisti (2010) and Brandimarte and Acquisti (2012).

Notes

¹ This section is based on material previously discussed in Acquisti (2010, sec. 3).

² Some consumer data firms advertise databases with contacts to individuals suffering from various types of addiction, such as gambling (see <http://www.dmnews.com/media-one-gamblers-database/article/164172/>).

³ Knight (1921) distinguished between *risk* (the random outcomes of an event can be described with a known probability distribution) and *ambiguity* (those probabilities are unknown).

⁴ See also Chapter 2 of this volume, by Barocas and Nissenbaum.

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

References

- Acquisti, A. 2004. Privacy in electronic commerce and the economics of immediate gratification. In *Proceedings of the ACM Conference on Electronic Commerce (EC '04)*, pp. 21–29.
- Acquisti, A. 2009. Nudging privacy: The behavioral economics of personal information. *IEEE Security & Privacy* 7(6): 82–85.
- Acquisti, A. 2010. The economics of personal data and the economics of privacy. Background Paper for OECD Joint WPISP-WPIE Roundtable.
- Acquisti, A., I. Adjerid, and L. Brandimarte. 2013. Gone in 15 seconds: The limits of privacy transparency and control. *IEEE Security & Privacy* 11(4): 72–74.
- Acquisti, A., and C. Fong. 2012. An experiment in hiring discrimination via online social networks. In *Privacy Law Scholars Conference*.
- Acquisti, A., and R. Gross. 2006. Imagined communities: Awareness, information sharing, and privacy on the Facebook. In *Workshop on Privacy Enhancing Technologies (PET '06)*.
- Acquisti, A., and J. Grossklags. 2005. Privacy and rationality in individual decision making. *IEEE Security and Privacy* 3(1): 24–30.
- Acquisti, A., and J. Grossklags. 2007. What can behavioral economics teach us about privacy? In *Digital Privacy: Theory, Technologies and Practices*, ed. S. G. C. L. Alessandro Acquisti and Sabrina De Capitani di Vimercati, 363–377. Boca Raton, FL: Auerbach Publications.
- Acquisti, A., L. K. John, and G. Loewenstein. 2013. What is privacy worth? *Journal of Legal Studies* 42(2): 249–274.
- Acquisti, A., and H. R. Varian. 2005. Conditioning prices on purchase history. *Marketing Science* 24(3): 367–381.
- Adjerid, I., A. Acquisti, L. Brandimarte, and G. Loewenstein. Sleights of privacy: Framing, disclosures, and the limits of transparency.
- Adjerid, I., A. Acquisti, R. Padman, R. Telang, and J. Adler-Milstein. 2013. The impact of health disclosure laws on health information exchanges. In *NBER Workshop on the Economics of Digitization*.

- Akerlof, G. A. 1970. The market for 'lemons': Quality uncertainty and the market mechanism. *Quarterly Journal of Economics* 84(3): 488–500.
- Beales, H. 2010. The value of behavioral targeting. Network Advertising Initiative.
- Bennett, J., and S. Lanning. 2007. The Netflix prize. In *Proceedings of KDD Cup and Workshop*.
- Blattberg, R. C., and J. Deighton. 1991. Interactive marketing: Exploiting the age of addressability. *Sloan Management Review* 33(1): 5–14.
- Brandimarte, L., and A. Acquisti. 2012. The economics of privacy. In *Handbook of the Digital Economy*, ed. M. Peitz and J. Waldfogel. New York: Oxford University Press.
- Brandimarte, L., A. Acquisti, and G. Loewenstein. 2013. Misplaced confidences privacy and the control paradox. *Social Psychological and Personality Science* 4(3): 340–347.
- Brunk, B. D. 2002. Understanding the privacy space. *First Monday* 7(10).
- Calo, R. 2011. The boundaries of privacy harm. *Indiana Law Journal* 86.
- Camenisch, J., and A. Lysyanskaya. 2001. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Advances in Cryptology - EUROCRYPT '01*, LNCS 2045, 93–118. Heidelberg: Springer.
- Camp, L. J. 2007. *Economics of Identity Theft: Avoidance, Causes and Possible Cures*. New York: Springer.
- Canny, J. F. 2002. Collaborative filtering with privacy. In *IEEE Symposium on Security and Privacy*, 45–57.
- Cate, F. H. 2002. Principles for protecting privacy. *Cato Journal* 22(1): 33–57.
- Cate, F. H., R. E. Litan, M. Staten, and P. Wallison (2003). Financial privacy, consumer prosperity, and the public good: Maintaining the balance. Federal Trade Commission Workshop on Information Flows: The costs and benefits to consumers and businesses of the collection and use of consumer information.
- Chaum, D. 1983. Blind signatures for untraceable payments. In *Advances in Cryptology*, 199–203. New York: Plenum Press.
- Chaum, D. 1985. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM* 28(10): 1030–1044.
- Culnan, M. J., and P. K. Armstrong. 1999. Information privacy concerns, procedural fairness, and impersonal trust: An empirical investigation. *Organization Science* 10(1): 104–115.

- Cvrcek, D., M. Kumpost, V. Matyas, and G. Danezis. 2006. The value of location information. In *ACM Workshop on Privacy in the Electronic Society (WPES)*.
- Dinev, T., and P. Hart. 2006. An extended privacy calculus model for e-commerce transactions. *Information Systems Research* 17(1): 61–80.
- Dingledine, R., N. Mathewson, and P. Syverson. 2004. Tor: The second-generation onion router. In *Proc. 13th Conference on USENIX Security Symposium*, 13:21.
- Duncan, G. T., S. A. Keller-McNulty, and S. L. Stokes. 2001. Disclosure risk vs. data utility: The ru confidentiality map. In *Chance*.
- Farrell, J. 2012. Can privacy be just another good? *Journal on Telecommunications and High Technology Law* 10:251–445.
- Fudenberg, D., and J. Tirole. 2000. Customer poaching and brand switching. *RAND Journal of Economics*, 634–657.
- Gellman, R. 2002. Privacy, consumers, and costs - how the lack of privacy costs consumers and why business studies of privacy costs are biased and incomplete. March.
- Gentry, C. 2009. Fully homomorphic encryption using ideal lattices. In *Proc. 41st Annual ACM symposium on Theory of Computing*, 169–178.
- Goldfarb, A., and C. Tucker. 2010. Privacy regulation and online advertising. Available at SSRN: <http://ssrn.com/abstract=1600259>.
- Hann, I.-H., K.-L. Hui, T. S. Lee, and I. P. Png. 2007. Overcoming online information privacy concerns: An information processing theory approach. *Journal of Management Information Systems* 42(2): 13–42.
- Hirshleifer, J. 1971. The private and social value of information and the reward to inventive activity. *American Economic Review* 61(4): 561–574.
- Hirshleifer, J. 1980. Privacy: Its origins, function and future. *Journal of Legal Studies* 9, no. 4 (December): 649–664.
- Huberman, B. A., E. Adar, and L. R. Fine. 2005. Valuating privacy. *IEEE Security & Privacy* 3:22–25.
- John, L. K., A. Acquisti, and G. Loewenstein. 2011. Strangers on a plane: Context-dependent willingness to divulge sensitive information. *Journal of Consumer Research* 37(5): 858–873.
- Knight, F. 1921. *Risk, Uncertainty and Profit*. Boston: Hart, Schaffner & Marx; Houghton Mifflin.

- Laufer, R. S., and M. Wolfe. 1977. Privacy as a concept and a social issue: A multidimensional developmental theory. *Journal of Social Issues* 33(3): 22–42.
- Lenard, T. M., and P. H. Rubin. 2009. In defense of data: Information and the costs of privacy. Technology Policy Institute.
- Loewenstein, G., and E. Haisley. 2008. The economist as therapist: Methodological issues raised by light paternalism. In *Perspectives on the Future of Economics: Positive and Normative Foundations*, ed. A. Caplin and A. Schotter. New York: Oxford University Press.
- Milberg, S. J., S. J. Burke, H. J. Smith, and E. A. Kallman. 1995. Values, personal information privacy, and regulatory approaches. *Communications of the ACM* 38(12): 65–74.
- Noam, E. M. 1996. Privacy and self-regulation: Markets for electronic privacy. In *Privacy and Self-Regulation in the Information Age*. National Telecommunications and Information Administration.
- Png, I. 2007. On the value of privacy from telemarketing: Evidence from the 'Do Not Call' registry. Working Paper, National University of Singapore.
- Posner, R. A. 1981. The economics of privacy. *American Economic Review* 71, no. 2 (May): 405–409.
- Rabin, M., and T. O'Donoghue. 2000. The economics of immediate gratification. *Journal of Behavioral Decision Making* 13(2): 233–250.
- Romanosky, S., and A. Acquisti. 2009. Privacy costs and personal data protection: Economic and legal perspectives. *Berkeley Technology Law Journal* 24(3).
- Rubin, P. H., and T. M. Lenard. 2001. *Privacy and the Commercial Use of Personal Information*. Boston: Kluwer Academic Publishers.
- Solove, D. J. 2006. A taxonomy of privacy. *University of Pennsylvania Law Review* 154(3): 477.
- Spiekermann, S., J. Grossklags, and B. Berendt. 2001. E-privacy in 2nd generation e-commerce: Privacy preferences versus actual behavior. In *3rd ACM Conference on Electronic Commerce*.
- Stigler, G. J. 1980. An introduction to privacy in economics and politics. *Journal of Legal Studies* 9, no. 4 (December): 623–44.
- Swire, P. P., and R. E. Litan. 1998. *None of Your Business - World Data Flows, Electronic Commerce, and the European Privacy Directive*. Washington, DC:

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Brookings Institution Press.

- Taylor, C. R. 2004a. Consumer privacy and the market for customer information. *RAND Journal of Economics* 35(4): 631–651.
- Taylor, C. R. 2004b. Privacy and information acquisition in competitive markets. Technical report, Duke University, Economics Department, 03-10.
- Tedeschi, B. 2002. E-commerce report; everybody talks about online privacy, but few do anything about it. *New York Times*, June 3.
- Tsai, J. Y., S. Egelman, L. Cranor, and A. Acquisti. 2011. The effect of online privacy information on purchasing behavior: An experimental study. *Information Systems Research* 22(2): 254–268.
- Varian, H. 1985. Price discrimination and social welfare. *American Economic Review* 75(4): 870–875.
- Varian, H., F. Wallenberg, and G. Woroch. 2005. The demographics of the do-not-call list. *IEEE Security & Privacy* 3(1): 34–39.
- Varian, H. R. 1996. Economic aspects of personal privacy. In *Privacy and Self-Regulation in the Information Age*. National Telecommunications and Information Administration.
- Wathieu, L., and A. Friedman. 2005. An empirical approach to understanding privacy valuation. In *4th Workshop on the Economics of Information Security*.
- Whitten, A., and J. D. Tygar. 1999. Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In *8th USENIX Security Symposium*.
- Wilson, K., and J. Brownstein. 2009. Early detection of disease outbreaks using the Internet. *Canadian Medical Association Journal* 180(8): 829.

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Chapter 4

Changing the Rules: General Principles for Data Use and Analysis

Paul Ohm

Introduction

How do information privacy laws regulate the use of big data techniques, if at all? Do these laws strike an appropriate balance between allowing the benefits of big data and protecting individual privacy? If not, how might we amend or extend laws to better strike this balance?

This chapter attempts to answer questions like these. It builds on the first chapter of this volume, which focused primarily on legal rules governing the *collection* of data. This chapter will focus primarily on the law of the United States, although it will make comparisons to the laws of other jurisdictions, especially the European Union.

Most information privacy law focuses on collection or disclosure and not use. Once data has been legitimately obtained, few laws dictate what may be done with the information. The exceptions to this general pattern receive attention below; laws that govern use tend to focus on particular types of users, especially users that lawmakers have deemed owe obligations of confidentiality to data subjects. For example, law regulating the health and financial industries, industries that historically have evolved obligations of confidentiality, constrain not only collection and disclosure but also use.

This chapter argues that our current information privacy laws are failing to protect individuals from harm. The discussion focuses primarily on shortcomings in the law that relate to specific features of big data, although it also describes a few shortcomings that relate only tangentially to these features. All of these shortcomings expose some individuals to the risk of harm in certain circumstances. We need to develop ways to amend the laws to recalibrate the balance between analytics and risk of harm. Ultimately, the chapter proposes five general approaches for change.

Current State of the Law Governing the Use of Information

Privacy law tends to divide the activities of data analysis (or, to use the more recent term, data science) into three steps, each subject to its own rules with specific, tailored levels of coverage and burden. These steps are collection, use, and disclosure. The first chapter focused mostly on collection and some on disclosure; this chapter will tackle use.

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Simply put, most uses of data are not regulated. Many data analysis practices fall wholly outside any privacy law, at least in the United States. Those that are nominally governed by law tend to be restricted lightly, if at all. The few use restrictions that exist tend to build upon the Fair Information Practice Principles (FIPPs). This part offers a thumbnail sketch of a few example laws, but even as to these laws, the analysis is necessarily brief and incomplete. Space does not allow for a thorough, much less complete, survey of the law. The purpose of this chapter is not to offer legal advice, but rather it is meant to critique and consider the need to expand the law.

The Fair Information Practice Principles

The FIPPs were promulgated first in an influential report issued by an Advisory Committee to the Secretary of Health, Education, and Welfare (“HEW”) which suggested in 1973 five rules for protecting the privacy of individuals in record-keeping systems:

1. There must be no personal data record-keeping systems whose very existence is secret.
2. There must be a way for a person to find out what information about the person is in a record and how it is used.
3. There must be a way for a person to prevent information about the person that was obtained for one purpose from being used or made available for other purposes without the person’s consent.
4. There must be a way for a person to correct or amend a record of identifiable information about the person.
5. Any organization creating, maintaining, using, or disseminating records of identifiable personal data must assure the reliability of the data for their intended use and must take precautions to prevent misuses of the data.¹

The FIPPs have been embraced by many scholars and policymakers, and form the basis of numerous government regulations. Each regulation, however, seems to embrace a different form of the FIPPs, with some that look very different from this original set. For example, the HEW FIPPs greatly influenced the OECD’s *Guidelines on the Protection of Privacy and Transborder Flows of Personal Data*, an influential set of voluntary standards developed by members of the OECD, which specify eight principles: collection limitation, data quality, purpose specification, use limitation, security safeguards, openness, individual participation, and accountability.² The Federal Trade Commission has for many years focused on its own version of the FIPPs as an organizing principle for its privacy mandates. The FTC’s version proposes five requirements: notice, consent, access, data integrity, and enforcement.³

Legal scholar Fred Cate has noted that, at least since the OECD Guidelines, every example of the FIPPs has “reflect[ed] a distinct goal of data protection as empowering consumers to control information about themselves, as opposed to protecting individuals from uses of information about them that are unfair or harmful.”⁴ This manifests in particular on a focus on notice and choice, which Chapter 2 argues persuasively is insufficient for protecting privacy in big data contexts. According to Cate, “in the past two decades most FIPPS have been applied in practice to require primarily notice and, in some instances, choice,” citing FTC pronouncements and the HIPAA rules as examples of how completely notice and choice has taken hold in the United States.⁵

But notice and choice is not only an American phenomenon, argues Cate, as even many of the provisions of the EU Data Protection Directive “can be waived with consent,” notwithstanding assertions by EU officials that the “directive is not concerned with notice and consent.”⁶

Sectoral Privacy Protection

Privacy law in the United States is often referred to as “sectoral.”⁷ No privacy law sweeps broadly across many different industries in the way the EU Data Protection Directive does in Europe. Rather, privacy laws in this country tend to focus on particular sectors – usually differentiated by industry segments or the type of information held. Thus, the Health Information Portability and Accountability Act (HIPAA) regulates the privacy of information generated and stored by the health industry,⁸ the Family Educational Rights and Privacy Act (FERPA) protects the privacy of information in schools,⁹ and the Graham Leach Bliley Act (GLB) provides some protection for information stored by financial institutions.¹⁰

For the most part, privacy law in the United States focuses on the collection or disclosure of information, not on its use. Most U.S. privacy laws do not implement a strong form of the FIPP of purpose limitation – information may be collected (or not) and shared (or not) according to these rules but once information has been legitimately obtained consistent with these laws, it can then be used for any purpose by any person associated with the organization possessing the information.

There are some exceptions to this general trend. Some privacy laws do limit the uses to which legitimately held information can be put. In the discussion that follows, the attention placed on these relative outliers should not obscure the broader point: most privacy laws do not contain these types of constraints. Once legitimately held, all information is fair game. Consider three examples of laws that regulate use: The Privacy Act, the Fair Credit Reporting Act (FCRA), and HIPAA.

The sweeping name given to the Privacy Act may mislead given the law’s relatively narrow and specialized scope. It protects the privacy of information maintained in large government databases. Enacted in 1974 – a time of general distrust in government and

This is a preliminary version of the book Privacy, Big Data, and the Public Good: Frameworks for Engagement, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

emerging fear about the impact of computerization on liberty and privacy – it represents the strong influence of the FIPPs. Like most FIPPs-based approaches, it focuses much more on collection and disclosure and notice and choice than on dictating narrowly circumscribed rules for use. Information collected must be associated with a purpose, and that information can be disclosed within an organization only “to those officers and employees of the agency . . . who have a need for the record in the performance of their duties.”¹¹ In addition, the Act allows the maintenance of “only such information about an individual as is relevant and necessary to accomplish a purpose of the agency required to be accomplished by statute or by executive order of the President.”¹² Thus, government agencies are not supposed to collect information for which there is no clearly specified purpose.

The FCRA covers the activities of consumer reporting agencies, primarily the “Big Three” credit reporting agencies, Experian, TransUnion, and Equifax, but also any entity that “furnish[es] consumer reports to third parties” used for credit, insurance, or employment purposes.¹³ The Federal Trade Commission, which shares authority with many other state and federal agencies under the FCRA to enforce the law, has argued that the law applies broadly beyond the Big Three. For example, in early 2012, the FTC sent letters to some of those who marketed mobile “background screening” applications, warning them of their likely FCRA obligations.¹⁴ The law limits the uses of consumer reports by consumer reporting agencies to activities enumerated in a long “white list,” such as issuing credit or evaluating a potential employee.¹⁵ It forbids uses for other purposes.

HIPAA applies only to “covered entities.” Many researchers who handle health information are not governed by HIPAA because they work for organizations that are not covered. For covered entities, such as many health care providers, HIPAA limits the use of information to a long list of permitted uses, such as law enforcement purposes or to facilitate the donation of cadaveric organ donation.¹⁶ For general research purposes, HIPAA generally requires the prior authorization of the subject of the data, with three limited exceptions: Authorization is not required with permission of an Institutional Review Board or Privacy Board; for a purpose preparatory to research such as to assist preparing a research protocol; or involving only the information of decedents.¹⁷ HIPAA also allows research outside these exceptions on datasets that have been de-identified according to a very specific and stringent standard.¹⁸

Gaps in the Law

None of the laws cited above focus on the special attributes of big data. Instead, they regulate all uses of data, regardless of the size of the dataset or the techniques used. Generally speaking, this probably makes sense. We should not create laws that treat big

data research as a wholly different and independent endeavor. We should instead create privacy laws that cover specific contexts of appropriate size and scope.

But big data techniques place special pressures on current privacy law, pressures that will lead some to call for a narrow revision or maybe even wholesale retrenchment of some privacy laws. Consider two. First, big data thrives on surprising correlations and produces inferences and predictions that defy human understanding. These characteristics call seriously into question laws that rely on providing notice to and receiving consent from data subjects. How can you provide notice about the unpredictable and unexplainable? This will lead some to call for us to abandon the FIPPs of use specification and purpose limitation.

Second, big data techniques resist attempts to reduce privacy risks by adding noise to personal information. Laws, such as HIPAA, reward anonymization and other methods of obscuring data, but big data techniques can often restore that which is removed or obscured. Consider both of these pressures in greater depth. Because big data techniques produce surprising correlations, we increasingly will find it difficult to know that we are in the presence of risky data, the kind of data that will likely lead to privacy harm. Thinking about it from the regulator's point of view, big data's tendency to generate surprising correlations attacks at their core laws developed upon preconceived intuitions, for example privacy laws structured around the specification of "bad data lists," lists of the types or categories of information that require special handling. Bad data lists require a nearly omniscient regulator capable of distinguishing, *ex ante* and from the regulator's typical remove, good data from bad. Big data undermines this approach. Mathematician Rebecca Goldin has said something similar about what big data might do to laws meant to prevent race discrimination. As described by The Economist:

[R]acial discrimination against an applicant for a bank loan is illegal. But what if a computer model factors in the educational level of the applicant's mother, which in America is strongly correlated with race? And what if computers, just as they can predict an individual's susceptibility to a disease from other bits of information, can predict his predisposition to committing a crime?¹⁹

Although we have banned discrimination based on race, big data helps companies find a reasonable proxy for race. Following a similar mechanism, if a behavioral advertising law regulates only databases containing personally identifiable information, no matter how this term is defined, clever statisticians will find a way to infer the very same information they have been denied from seemingly unrelated data.

For example, suppose a study determines tomorrow that height and weight retain a surprising amount of identifying information. It would be a short-sided response to add

height and weight to the lists of types of information that had to be kept from researchers. Every list that prohibits the collection of A will be defeated once data owners realize that unregulated B and C can be used to derive A.

Similarly, the notice and choice at the heart of FIPPs cannot do enough to protect privacy in the age of big data. Big data succeeds by drawing inferences that confound expectations. A data scientist who does no more than confirm prior intuitions will soon be out of work. The best data scientists find results that are not only counter-intuitive but also sometimes governed by mysterious, opaque mechanisms. Big data empowers through surprise. Thus, a regime which depends solely on limited purpose, notice, and choice cannot do enough to protect against the unpredictability of tomorrow promises. In the same way, legal limits on purpose and use will have trouble addressing the problems of big data. An important principle found in many FIPPs is the principle of limited purpose. The HEW report demands that “There must be a way for a person to prevent information about the person that was obtained for one purpose from being used or made available for other purposes without the person’s consent.”²⁰ The OECD requires both a “Purpose Specification” and a “Use Limitation.”²¹ The EU Data Protection Directive similarly requires a “purpose limitation.”²²

It is difficult to square a purpose limitation (or even an obligation merely to specify a purpose) with big data. A big data analyst often cannot specify a purpose except at a very high level of abstraction – to “learn something about our customers,” or to “find the patterns hiding in the data.” This is why so many big data practitioners are loathe to delete old data; you never know what use we will find for it tomorrow! The very idea of big data seems anathema to a purpose limitation which would require that, using the language of the European Union’s Article 29 Working Group, “data should be processed for a specific purpose and subsequently used or further communicated only insofar as this is not incompatible with the purpose of the transfer.”²³

Toward a New Conceptualization of Using Law to Protect Privacy

If big data puts new pressures on information privacy, we may need to change information privacy law. I believe that change is urgently needed. Too many people are considering forms of research that escape regulatory scrutiny, and too much of this research might subject individuals and society to jarring shifts from the historical status quo and wreak devastating harm to individuals. In this part, I sketch the impending harm, and I propose some prescriptions to help us restore what we might lose.

Big Data Research Needs to Be Regulated More

We need to fill significant gaps in our privacy laws for data practices generally and for big data practices in particular. The current regulatory landscape exposes too many people to too many unjustifiable risks of significant harm. These harms impact not only

individuals but also groups and some of these harms extend across the entire society. The next, last part of this chapter will offer a roadmap for ways we might extend our present regulatory obligations, but first, consider the argument that change is necessary.

A touchstone for this work is Helen Nissenbaum's idea of privacy as contextual integrity.²⁴ Contextual integrity requires us, as the name suggests, to frame privacy problems and solutions in appropriately sized contexts as opposed to searching for comprehensive privacy solutions that cover all of society. The book in which this chapter appears focuses on a few narrow contexts: big data analytics, primarily (although not exclusively) conducted in traditional research institutions, with a special focus on the study of cities. Accordingly, I will focus my analysis on these contexts too, even though much of what I say will apply more broadly, for example to big data research conducted inside private corporations.

My argument for change builds also on two strands of the work of Danielle Citron. Citron has argued that individuals can suffer significant harm from the mishandling of information about them stored in massive databases.²⁵ She has also shined a spotlight on the way official decision making, particularly by the government, is increasingly becoming algorithmic.²⁶ In this second strand, she has argued that we face threats from the complexity and opacity of this shift, chronicling tales of people trapped with Kafka-esque bureaucracies abetted by the rise of new technology.²⁷

In my recent work, I have built upon Citron's arguments. I have highlighted the risk of significant harm inherent in massive databases of information, something I have called the "database of ruin."²⁸ I have discussed a method for assessing this risk, one borrowed from computer security, of building threat models designed to reveal the kind and intensity of harm people might be exposed to, risks not only of identity theft and invidious discrimination, but also traditional privacy harms such as stalking, harassment, and blackmail.²⁹ These threat models suggest the need to focus not only on external risks such as hackers and government surveillance but also internal risks such as graduate students "peeking" at sensitive information to satisfy idle curiosity or worse.³⁰ Internal threats like these are much more difficult to guard against, according to computer scientists.³¹

While Citron and I have focused primarily on "traditional" and individualistic harms of privacy, there is a rich body of literature focused on broader harms to both the individual and society. This work has been labeled the "New Privacy" movement by some.³² There are many strands to this work and not enough space to elaborate all of them fully. New Privacy scholars argue that society suffers under the panoptic effect of surveillance. Julie Cohen focuses on how people will be chilled from experimentation and play, and as a result stunted in their development as emergent individuals.³³ Neil Richards focuses in particular on the intellectual development of individuals, worrying most about data involving evidence of the intellectual products we consume: what we

read, listen to, and watch.³⁴ Paul Schwartz argues that society suffers when it is not afforded private spaces to engage in a deliberative democracy.³⁵ Others see privacy as essential to preventing imbalances in power.³⁶

Finally, I draw from the work of scholars and other commentators who worry about the dehumanizing effect of treating individuals merely as data. Daniel Solove contends that “the information in databases often fails to capture the texture of our lives. Rather than provide a nuanced portrait of our personalities, they capture the stereotypes and the brute facts of what we do without the reasons.”³⁷ Jaron Lanier decries what he calls “antihuman software design.”³⁸ Danielle Citron writes about the mistakes that come from substituting a human role in decision making.³⁹

Prescriptions

I offer five proposals for reform, at varying levels of generality. First, any rules we create or expand should be calibrated to the sensitivity of the information stored in the database, meaning according to the risk of significant harm to individuals or groups. Certain types of databases are riskier than others, and I offer some first-cut rules for making these distinctions. Second, previous rules based on distinctions between PII and non-PII no longer make sense and need to be abandoned. Third, we should build and police walls or gaps between different datasets. Fourth, we should remind data analysts repeatedly that the numbers they analyze represent the lives of people. These reminders should scale up with the sensitivity of information, meaning those who work with the most sensitive information should face constant, even uncomfortable, reminders. Fifth, and finally, researchers must constantly assess the ethics of new practices. These ethical conversations need to be imposed or at least incentivized by rule or law. The need for conversations about ethics are particularly required at the interface of traditional and new modes of research funding and organization, to counteract the likely jealousy that will be felt within traditional research centers (such as universities) for the relatively thin set of rules governing research being done in private corporations. Consider each of these prescriptions in greater depth in turn.

Sensitive Information Not all information can be used to cause harm. Over hundreds of years of legal evolution, we have begun to recognize some categories as especially likely to cause harm. These include health, sex, financial, and educational information, to name only a few. In other work, I have examined closely what makes a category of information sensitive, and I have concluded that there are at least four things.⁴⁰ First, sensitive information can be used to cause harm. Second, this risk of harm is substantial, not improbable nor speculative. Third, categories of sensitive information often involve relationships we consider confidential, such as doctor-patient or bank-

customer. Finally, categories of sensitive information tend to reflect majoritarian sentiments, protecting us from harms suffered by many.⁴¹

As we extend privacy law to cover currently unregulated forms of data analysis, we should focus on contexts involving sensitive information, including new categories of information not yet deemed sensitive but deserving of the label. I have pointed to three new candidates: precise geolocation, genomic information, and biometric information. We should write new privacy laws covering these (and other) non-regulated forms of information.⁴²

Ending the PII/non-PII Distinction Every privacy law that has ever been written has rewarded the anonymization of data. In fact, most privacy laws do not apply at all to data that has been stripped of identifiers. In the typical parlance, privacy law usually applies only to data containing personally identifiable information, or PII. We should abandon this mode of regulation, recognizing instead that even apparently anonymized information often contains enough residual data to re-link to individuals. Privacy laws should continue to apply even to data that has been de-identified, at least for the most sensitive forms of data. But this is not to say that de-identification cannot continue to play a role, because sensible de-identification can greatly reduce the risk of troubling outcomes. Rather than treating all purportedly anonymized data as completely regulated, we might reconfigure our privacy laws to act more like sliding scales, reducing the requirements of the rules for sufficiently de-identified information, along lines suggested by Paul Schwartz and Dan Solove.⁴³

Legislating Gaps We should use law to create gaps between contexts. Many theorists have written about the special problems of total surveillance. Julie Cohen in particular advocates the imposition of gaps between contexts, allowing people to carry out different experiments – to play – in order to learn and develop.⁴⁴

One way to do this is to declare some modes or categories of information off-limits from data processing. In some sense, this is what we have done through what I have called protected channel laws.⁴⁵ Protected channel laws deem some methods of surveillance specially protected. One example is the federal Wiretap Act, which declares it a felony to use a packet sniffer to acquire the contents of communications in transit across a telephone or computer network, absent some exception.⁴⁶

In the context of big data and research, because of protected channel laws like the Wiretap Act, we probably understand less about what people say on the telephone than we do what they say in stored media such as social networks. A sociologist simply cannot obtain a massive database of the recorded content of telephone calls, even if he has a good reason to want it, and even if he can describe profound benefits he might develop

from studying this kind of information. This inability should not be viewed as regrettable; it is instead a laudable example of using law to impose Cohen's gaps.

To some data scientists, gaps may seem anathema, contrary to core goals of big data, because they represent troubling blind spots. But if we believe in the importance of privacy gaps, we should try to change this attitude. A legally imposed privacy gap is a feature not a bug.

Even more importantly, we need to disabuse people of the idea that legally imposed gaps need to be highly tailored to prevent only narrowly defined, highly specific types of horrific privacy harm. As Cohen puts it, the goal of gap building is to enable unexpected advantage. “Evolving subjectivity, or the everyday practice of self, responds to the play-of-circumstances in unanticipated and fundamentally unpredictable ways. . . . [T]he play-of-circumstances operates as a potent engine of cultural dynamism, mediating both evolving subjectivity and evolving collectivity, and channeling them in unexpected ways.”⁴⁷ This means that the gaps we define will not be the product of hyper-rationalized decision making, but instead that it will be based on inexact, human-scale approaches. As Cohen puts it, “privacy consists in setting of limits precisely where logic would object to drawing lines.”⁴⁸ Or as I have said earlier, “we will be forced to carve out these gaps using machetes not scalpels.”⁴⁹

Reminding Researchers about Humanity Some of the greatest concerns about big data build upon fears about its dehumanizing effects. We worry that big data techniques will replace traditional, official modes of decision making about the lives of individuals. Even if these techniques lead to results that we may consider more efficient, we should worry that they allow too much distance between the decision makers – increasingly those who call themselves data scientists – and those whose lives they control. These fears are in part based on instrumental concerns: we worry about the unfairness of being trapped by the machine or in the machine. We worry about being victims of officious bureaucrats. But the concerns are not merely about instrumental effects. Even if data science can lead to results that are efficient, viewed through a utilitarian lens, they yet may not be better if they result in treating people as dehumanized widgets.

A prophylactic step we might take is to design mechanisms to constantly remind those who operate on information about the people whose lives they assess, direct, or divulge. We should consider forcing data scientists to think about the lives of those analyzed. We should use techniques like visceral notice⁵⁰ to send cues to data scientists that lives are being impacted. Perhaps more controversially, for studies on sensitive information, we should opt to make the very act of data analysis uncomfortable in subtle ways, to remind the analysts that they are operating on people.

I am informed in this admittedly unorthodox idea from my experience as a former employee of the federal government. Federal employees are subjected to stringent ethical

rules, for example, rules limiting the type and size of gifts they may accept. The purposes of these rules are primarily instrumental: they are intended to prevent bribery, blackmail, and fraud infecting the decision making of the government. But, in my personal experience, they serve a secondary, possibly unintended but still laudable, purpose: they remind government workers that it is they who serve the citizenry, not the other way around. The life of a federal worker can sometimes seem practically ascetic next to his or her corporate colleagues; I am reminded of a time when I was forced to decline a voucher to buy a cafeteria lunch at a meeting held on the campus of a Silicon Valley tech giant, even though everybody else was allowed to accept. There was never the possibility that a \$10 turkey sandwich would have caused me to abuse the public trust, but I was reminded when I was forced to refuse that my job carried special, important burdens that I would be unwise to forget.

I think it makes sense to try to replicate this experience, sometimes, for data scientists, particularly those working with sensitive information. Data scientists who access individual salary information, for example, or data containing past drug use or criminal convictions, should be reminded by their work conditions that they wield great power. We should place them within bare walls and under flickering fluorescent lighting rather than ply them with foosball tables and free soda. In fact, there may be precedent for this idea. Some information is considered so sensitive, the only people permitted to analyze it are forced to come to the data rather than the other way around. One example occurs in the use of “data rooms” to facilitate corporate mergers or acquisitions.⁵¹ These are secure and highly monitored rooms that potential acquirers can visit to examine sensitive documents without being able to retain verbatim copies. Access to information stored by government agencies is often subjected to similarly strict controls. Even though the point of these measures are for security rather than to remind the analyst of the humanity underneath the numbers, we might think of imposing conditions like these for this second reason too.

Developing Ethical Norms Finally, it is imperative that data scientists begin to debate, agree upon, and perhaps even codify ethical norms of behavior around responsible data science. Lawmakers and policymakers abhor an ethical vacuum, and if ever a serious, headline-grabbing violation of big data techniques should occur – the information equivalent of the Tuskegee Syphilis study – the first question outsiders will ask is whether the practice violated the relevant community’s internal ethical norms.

This prescription builds on the work of many others. Helen Nissenbaum situates norms as key to understanding contextual integrity (and determining when contextual integrity has been breached).⁵² Ryan Calo has called for learning from human subjects review in medical research for big data.⁵³

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

This prescription can take one of many different forms. At the very least, it encourages informal deliberation within communities of researchers. Better yet, groups can try to identify people or institutions that can convene others to discuss norms formally.

Similar efforts have been undertaken before. Notably, the Belmont Report established the modern baseline for biomedical and behavioral research and served as a precursor to today's Common Rule and the rise of institutional review boards.⁵⁴ Although not nearly as prominent, researchers in information and communication technology, under the auspices of the Department of Homeland Security, produced the Menlo Report, which attempted to provide ethical guidance for computer research involving human subjects.⁵⁵

Conclusion

Today, it is fair to say that a wide swath of research being done or proposed involving big data techniques is entirely unregulated, and what little work is regulated is regulated only with respect to how the data is collected or subsequently disclosed. Although some might cheer this state of affairs, complimenting the government for staying out of the way of research, I do not. Data analysis is poised to become a central tool of science, and in some fields it already has, and the results produced will become the basis for decision making. As we expand the reach and power and influence of data science, we must take steps to prevent harm, to ensure that this remains always a humanistic endeavor, and help people preserve their sense of power and autonomy.

Notes

¹ Secretary's Advisory Committee on Automated Personal Data Systems, *Records, Computers and the Rights of Citizens* (Washington, DC: U.S. Department of Health, Education and Welfare, 1973), 41-42.

² Organisation for Economic Co-operation and Development, *OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data* (1980), <http://www.oecd.org/internet/ieconomy/oecdguidelinesontheprotectiionofprivacyandtransborderflowsofpersonaldatal.htm>.

³ Federal Trade Commission, *Fair Information Practice Principles*, <http://www.ftc.gov/reports/privacy3/fairinfo.shtm>.

⁴ Fred Cate, "The Failure of the Fair Information Practice Principles," in *Consumer Protection in the Age of the Information Economy*, ed. Jane K. Winn. (Surrey, UK: Ashgate, 2006), 356.

⁵ Ibid., 357.

⁶ Ibid., 360.

⁷ Paul M. Schwartz, “Preemption and Privacy,” *Yale Law Journal* 118 (2009): 902.

⁸ Health Insurance Portability & Accountability Act (HIPAA), 45 C.F.R. §§ 164.501, et seq.

⁹ Family Educational Rights & Privacy Act, 20 U.S.C. § 1232g.

¹⁰ Graham-Leach-Bliley Act, 15 U.S.C. § 6801, et seq.

¹¹ 5 U.S.C. § 552a(b)(1).

¹² 5 U.S.C. § 552a(e)(1).

¹³ 15 U.S.C. § 1681a(d).

¹⁴ See <http://www.ftc.gov/opa/2012/02/mobileapps.shtm>.

¹⁵ 15 U.S.C. § 1681b.

¹⁶ See http://privacyruleandresearch.nih.gov/pdf/HIPAA_Privacy_Rule_Booklet.pdf

¹⁷ 45 C.F.R. § 164.512(i).

¹⁸ 45 C.F.R. § 164.514(e)(2).

¹⁹ “New Rules for Big Data,” *The Economist*, February 25, 2010.

²⁰ Secretary's Advisory Committee on Automated Personal Data Systems, *Records*.

²¹ Organisation for Economic Co-operation and Development, *OECD Guidelines*.

²² EU Data Protection Directive of 1995, Directive 95/46 EC of the European Parliament and the Council.

²³ Working Party on the Protection of Individuals with Regard to the Processing of Personal Data, “Working Document on Transfers of Personal Data to Third Countries: Applying Articles 25 and 26 of the EU Data Protection Directive,” DG XV D/5025/98 WP 12 (July 24, 1998).

²⁴ Helen Nissenbaum, *Privacy in Context: Technology, Policy, and the Integrity of Social Life* (Stanford, CA: Stanford University Press, 2009).

²⁵ Danielle Keats Citron, ‘Reservoirs of Danger: The Evolution of Public and Private Law at the Dawn of the Information Age,’ *California Law Review* 80 (2007): 241.

²⁶ Danielle Keats Citron, “Technological Due Process,” *Washington University Law Review* 85 (2008): 1249.

²⁷ Ibid. See also Daniel J. Solove, “Privacy and Power: Computer Databases and Metaphors for Information Privacy,” *Stanford Law Review* 53 (2001): 1393.

²⁸ Paul Ohm, “Broken Promises of Privacy: Responding to the Surprising Failure of Anonymization,” *UCLA Law Review* 57 (2010): 1701.

²⁹ Paul Ohm, “Sensitive Information?” forthcoming, manuscript on file.

³⁰ Ibid.

³¹ S. Stolfo et al., eds., *Insider Attack and Computer Security: Beyond the Hacker* (New York: Springer, 2008).

³² See Paul M. Schwartz and William M. Treanor, “The New Privacy,” *Michigan Law Review* 101 (2012): 2163–2181.

³³ Julie Cohen, *Configuring the Networked Self: Law, Code, and the Play of Everyday Practice* (New Haven, CT: Yale University Press, 2012).

³⁴ Neil Richards, “Intellectual Privacy,” *Texas Law Review* 87 (2008): 387.

³⁵ Paul Schwartz, “Internet Privacy and the State,” *Connecticut Law Review* 32 (2000): 815.

³⁶ Priscilla M. Regan, *Legislating Privacy: Technology, Social Values, and Public Policy* (Chapel Hill, NC: University of North Carolina Press, 1995).

³⁷ Solove, “Privacy and Power.”

³⁸ Jaron Lanier, *You Are Not a Gadget: A Manifesto* (New York: Knopf, 2010), 193.

³⁹ Citron, “Technological Due Process.”

⁴⁰ Ohm, “Sensitive Information.”

⁴¹ Ibid.

⁴² Ibid.

⁴³ Paul M. Schwartz and Daniel J. Solove, “The PII Problem: Privacy and a New Concept of Personally Identifiable Information,” *NYU Law Review* 86 (2011): 1814.

⁴⁴ Cohen, “Configuring the Networked Self.”

⁴⁵ Ohm, “Sensitive Information.”

⁴⁶ 18 U.S.C. § 2511.

⁴⁷ Cohen, “Configuring the Networked Self.”

⁴⁸ Ibid.

⁴⁹ Paul Ohm, “Mind the Gap,”

<http://www.concurringopinions.com/archives/2012/03/mind-the-gap.html>.

⁵⁰ Ryan Calo, “Against Notice Skepticism in Privacy (and Elsewhere),” 87 Notre Dame Law Review 87 (2012): 1027.

⁵¹ See <http://www.digitaldataroom.com/What-is-a-Data-Room.html>.

⁵² Nissenbaum, *Privacy in Context*.

⁵³ Ryan Calo, Consumer Subject Review Boards: A Thought Experiment, 66 Stan. L. Rev. Online 97 (2013).

⁵⁴ “Belmont Report: Ethical Principals and Guidelines for the Protection of Human Subjects of Research,” Report of the National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research, 44 Fed. Reg. 23,192 (April 18, 1979).

⁵⁵ “The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research” (“Menlo Report”), Department of Homeland Security, 76 Fed. Reg. 81,517 (December 28, 2011).

Chapter 5

Enabling Reproducibility in Big Data Research: Balancing Confidentiality and Scientific Transparency

Victoria Stodden

Introduction

The 21st century will be known as the century of *data*. Our society is making massive investments in data collection and storage, from sensors mounted on satellites down to detailed records of our most mundane supermarket purchases. Just as importantly, our reasoning about these data is recorded in software, in the scripts and code that analyze this digitally recorded world. The result is a deep digitization of scientific discovery and knowledge, and with the parallel development of the Internet as a pervasive digital communication mechanism we have powerful new ways of accessing and sharing this knowledge. The term *data* even has a new meaning. Gone are the days when scientific experiments were carefully planned prior to data collection. Now the abundance of readily available data creates an observational world in itself suggesting hypotheses and experiments to be carried out after collection, curation, and storage of the data has already occurred. We have departed from our old paradigm of data collection to resolve research questions – nowadays, we collect data simply *because we can*.

In this chapter I outline what this digitization means for the independent verification of scientific findings from these data, and how the current legal and regulatory structure helps and hinders the creation and communication of reliable scientific knowledge.¹ Federal mandates and laws regarding data disclosure, privacy, confidentiality, and ownership all influence the ability of researchers to produce openly available and reproducible research. Two guiding principles are suggested to accelerate research in the era of big data and bring the regulatory infrastructure in line with scientific norms: the Principle of Scientific Licensing and the Principle of Scientific Data and Code Sharing. These principles are then applied to show how intellectual property and privacy tort laws could better enable the generation of verifiable knowledge, facilitate research collaboration with industry and other proprietary interests through standardized research dissemination agreements, and give rise to dual licensing structures that distinguish between software patenting and licensing for industry use and open availability for open research. Two examples are presented to give a flavor of how access to data and code might be managed in the context of such constraints, including the establishment of ‘walled gardens’ for the validation of results derived from confidential data, and early

research agreements that could reconcile scientific and proprietary concerns in a research collaboration with industry partners.

Technological advances have complicated the data privacy discussion in at least two ways. First, when datasets are linked together, a richer set of information about a subject can result but so can an increased risk of a privacy violation. Linked data presents a challenging case for open scientific research, in that it may permit privacy violations from otherwise non-violating datasets. In this case privacy tort law is suggested as a viable remedy for privacy violations that arise from linking datasets.

Second, the subjects of studies are becoming more knowledgeable about privacy issues, and may wish to opt for a greater level of access to their contributed data than that established by traditional research infrastructures, such as Institutional Review Boards. For data collection and release that happens today, research subjects have very little say over the future openness of their data. A suggestion is made to permit individuals to share their own data with provisions regarding informed consent. For example, an enrollee in a clinical trial for a new Crohn's disease treatment may wish to permit other Crohn's researchers access to the data arising from her participation, perhaps in an effort to help research advance in an area about which she cares deeply. At the moment, this is not only nonstandard, but downstream data use is difficult for the participant to direct.

Ownership itself can be difficult to construe since many resources typically go into creating a useful dataset, from research scientists who design the experiment, to data collectors, to participants, to curators, to industry collaborators, to institutes and funding agencies that support the research, further complicating the discussion of data and code access. Data access becomes increasingly complex, underscoring the need for a broad understanding of the value of maximizing open access to research data and code.

Trust and Verify: Reliable Scientific Conclusions in the Era of Big Data

Scientific research is predicated on an understanding of scientific knowledge as a public good – this is the rationale underlying today's multibillion-dollar subsidies of scientific research through various federal and state agencies. The scientific view is not one of adding nuggets of truth to our collective understanding, but instead one of weighing evidence and assigning likelihoods to a finding's probability of being true. This creates a normative structure of skepticism among scientists: the burden is on the discovering scientist to convince others that what he or she has found is more likely to be correct than our previous understanding. The scientific method's central motivation is the *ubiquity of error* – the awareness that mistakes and self-delusion can creep in absolutely anywhere and that the scientist's effort is primarily expended in recognizing and rooting out error.

As a result, standards of scientific communication evolved to incorporate full disclosure of the methods and reasoning used to arrive at the proffered result.

The case for openness in science stems from Robert Boyle's exhortations in the 1660s for standards in scientific communication. He argued that enough information should be provided to allow others in the field to independently reproduce the finding, creating both the greatest chance of the accurate transmission of the new discoveries and also maximizing the likelihood that errors in the reasoning would be identified. Today, communication is changing because of the pervasive use of digital technology in research. Digital scholarly objects such as data and code have become essential for the effective communication of computational findings. Computations are frequently of such a complexity that an explanation sufficiently detailed to enable others to replicate the results is not possible in a typical scientific publication. A solution to this problem is to accompany the publication with the code and data that generated the results and communicate a *research compendium*.² However, the scientific community has not yet reached a stage where the communication of research compendia is standard.³ A number of delicate regulatory and policy changes are essential to catalyze both scientific advancement and the development of applications and discoveries outside academia by making the data and code associated with scientific discoveries broadly available.

Challenge 1: Intellectual Property Law and Access to Digital Scholarly Objects

The Intellectual Property Clause of the United States Constitution has been interpreted to confer two distinct powers, the first providing the basis for copyright law: Securing for a limited time a creator's exclusive right to their original work;⁴ and the second giving the basis for patent law: Endowing an inventor with a limited-term exclusive right to use their discoveries in exchange for disclosure of the invention. In this section the barrier copyright creates to open reproducible research will be discussed first, then the role of patent law in potentially obfuscating computational science.

Creators do not have to apply for copyright protection, as it adheres automatically when the original expression of the idea is rendered in fixed form. Many standard scientific activities, such as writing a computer script to filter a dataset or fit a statistical model, will produce copyrighted output, in this case the code written to implement these tasks. Building a new dataset through the original selection and arrangement of data will generate ownership rights through copyright for the dataset creator, to give another example.⁵ The default nature of copyright creates an *intellectual property* framework for scientific ideas at odds with longstanding scientific norms in two key ways.⁶ First, by preventing copying of the research work it can create a barrier to the legal reproduction and verification of results.⁷ Second, copyright also establishes rights for the author over

the creation of derivative works. Such a derivative work might be something as scientifically productive as, say, the application of a software script for data filtering to a new dataset, or the adaptation of existing simulation codes to a new area of research.

As computation becomes central to scientific investigation, copyright on code and data become barriers to the advancement of science. There is a copyright exception, titled *fair use*, which applies to “teaching (including multiple copies for classroom use), scholarship, or research”⁸ but this does not extend to the full research compendium including data, code, and research manuscript. In principle, a relatively straightforward solution to the barrier copyright imposes would be to broaden the fair use exception to include scientific research that takes place in research institutions such as universities or via federal research grants; however, this is extremely challenging in practice.⁹ Distinguishing legal fair use is not a clear exercise in any event, and an extension to research more broadly may still not sufficiently clarify rights. A more practical mechanism for realigning intellectual property rights with scientific norms is the Reproducible Research Standard (RRS), applying appropriate open licenses to remove restrictions on copying and reuse of the scientific work, as well as possibly adding an attribution requirement to elements of the research compendium. Components of the research compendium have different features that necessitate different licensing approaches and a principle for licensing scientific digital objects can guide choices:

Principle of Scientific Licensing *Legal encumbrances to the dissemination, sharing, use, and reuse of scientific research compendia should be minimized, and require a strong and compelling rationale before their application.*¹⁰

For media components of scientific work, the Reproducible Research Standard suggests the Creative Commons attribution license (CC BY), which frees the work for replication and re-use without prior author approval, with the condition that attribution must accompany any downstream use of the work.

Many licenses exist that allow authors to set conditions of use for their code. In scientific research code can consist of scripts that are essentially stylized text files (such as python or R scripts) or the code can have both a compiled binary form and a source representation (such as code written in C). Use of the CC BY license for code is discouraged by Creative Commons.¹¹ The Reproducible Research Standard suggests the Modified Berkeley Software Distribution (BSD) license, the MIT license, or the Apache 2.0 license, which permit the downstream use, copying, and distribution of either unmodified or modified source code, as long as the license accompanies any distributed code and the previous authors’ names are not used to promote modified downstream code.¹² The Modified BSD and MIT licenses differ in that the MIT license does not include a clause forbidding endorsement.¹³ The Apache 2.0 license differs in that it

permits the exercise of patent rights that would otherwise extend only to the original licensor, meaning that a patent license is granted for those patents needed for use of the code.¹⁴ The license further stipulates that the right to use the work without patent infringement will be lost if the downstream user of the code sues the licensor for patent infringement.

Collecting, cleaning, and preparing data for analysis can be a significant component of empirical scientific research. Copyright law in the United States forbids the copyrighting of ‘raw facts’ but original products derived from those facts can be copyrightable. In *Feist Publications, Inc. v. Rural Telephone Service*, the Court held that the *original* “selection and arrangement” of databases is copyrightable:¹⁵ the component falling under copyright must be original in that “copyright protection extends only to those components of the work that are original to the author, not to the facts themselves.”¹⁶ Attaching an attribution license to the original “selection and arrangement” of a database may encourage scientists to release the datasets they have created by providing a legal framework for attribution and reuse of the original selection and arrangement aspect of their work.¹⁷ Since the raw facts themselves are not copyrightable, such a license cannot be applied to the data themselves. The selection and arrangement may be implemented in code or described in a text file accompanying the dataset, either of which can be appropriately licensed. Data can however be released to the public domain by marking with the Creative Commons CC0 standard.¹⁸

This licensing structure that makes the total of the media, code, data components – the research compendium – available for reuse, in the public domain or with attribution, is labeled the *Reproducible Research Standard*.

Patent law is the second component of intellectual property law that affects the disclosure of scientific scholarly objects. In 1980 Congress enacted two laws, the Stevenson-Wydler Act and the Bayh-Dole Act, both intended to promote the commercial development of technologies arising from federally funded research. This was to be facilitated through licensing agreements between research entities, such as universities, and for-profit companies. The Bayh-Dole Act explicitly gave federal agency grantees and contractors, most notably universities and research institutions, title to government-funded inventions and charged them with using the patent system to disclose and commercialize inventions arising in their institution. In 2009 this author carried out a survey of computational scientists, in order to understand why they either shared or withheld the code and data associated with their published papers. In the survey one senior professor explained that he was not revealing his software because he was currently seeking a patent on the code.¹⁹ In fact, 40% of respondents cited patent seeking or other intellectual property constraints as a reason they were not sharing the code associated with published scientific results.²⁰ Rates of software patenting by academic institutions have been increasing over the last decade, posing a potentially serious

problem for scientific transparency and reproducibility.²¹ Instead of ready access to the code that generated published results, a researcher may be required to license access to the software through a university's technology transfer office, likely being prohibitively expensive for an academic scientist in both time and money. In December of 1999, the National Institutes of Health stated that

the use of patents and exclusive licenses is not the only, nor in some cases the most appropriate, means of implementing the [Bayh-Dole] Act. Where the subject invention is useful primarily as a research tool, inappropriate licensing practices are likely to thwart rather than promote utilization, commercialization, and public availability.²²

The federal funding agencies are without authority to issue regulations regarding patentable inventions, and the NIH viewpoint above does not appear to have been adopted by technology transfer offices at the university and institutional research level. A typical interpretation is that of Columbia University, where this author is employed, which follows: "The University claims, as it may fairly and rightfully do, the commercial rights in conceptions that result primarily from the use of its facilities or from the activity of members of its faculty while engaged in its service."²³ Not all universities make such an a priori claim to determine the patenting and licensing fate of research inventions. For example, Stanford University's *Research Policy Handbook* says that as a researcher, "I am free to place my inventions in the public domain as long as in so doing neither I nor Stanford violates the terms of any agreements that governed the work done."²⁴ The Bayh-Dole Act also grants agencies 'march-in' rights to obtain intellectual property (presumably to grant nonexclusive licenses, but not necessarily), but the process is long with multiple appeal opportunities. In July of 2013, however, in a letter to Francis Collins, head of the NIH, Senator Leahy recommended the use of march-in rights on patented breast cancer genetic research "to ensure greater access to genetic testing for breast and ovarian cancer."²⁵

Challenge 2: Scale, Confidentiality, and Proprietary Interests

Even without intellectual property law encumbrances to the dissemination of digital scholarly objects, other barriers can create obstacles to access. For example, the sheer size of many datasets may require specialized computational infrastructure to permit access, or scale itself can even prohibit access. For example, the July 2013 release of the Sloan Digital Sky Survey (SDSS) is 71.2 terabytes in size, making a conventional download of data to a personal laptop impossible.²⁶ The approach of the SDSS is to create different websites for different data types, and provide a variety of tools for access

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

including SkyServer SQL search, CasJobs, and Schema Browser, each with a different purpose in mind.²⁷ This infrastructure permits search and user-directed access to significantly smaller subsets of the entire database.

In some fields however even 70 terabytes would not seem large. CERN director general Rolf Heuer said in 2008 that, “[t]en or 20 years ago we might have been able to repeat an experiment. They were simpler, cheaper and on a smaller scale. Today that is not the case. So if we need to re-evaluate the data we collect to test a new theory, or adjust it to a new development, we are going to have to be able reuse it. That means we are going to need to save it as open data.”²⁸ In March of 2013, the CERN data center passed a storage milestone by exceeding 100 petabytes of data.²⁹ It is not clear how this can be made open data in the sense discussed in this chapter, as Director Heuer suggests. The traditional approaches to making data and code available seem intractable for such datasets at the present time. I use these examples to introduce a Principle of Scientific Data and Code Sharing:

Principle of Scientific Data and Code Sharing *Access to the data and methods associated with published scientific results should be maximized, only subject to clearly articulated restrictions such as: privacy or confidentiality concerns, legal barriers including intellectual property or HIPAA regulations, or technological or cost constraints.*

This principle can also be phrased as ‘Default to Open’, meaning that it takes compelling and convincing reasons, articulated in detail (i.e. the precise section of HIPAA that is restricting disclosure, or the part of intellectual property law that is creating a barrier) to close data and code from public access.^{30,31} A careful defense of any deviation from full openness will have the effect of maximizing the availability of data and code. A corollary effect is an uncovering of the reasons for not sharing data and presumably a greater understanding of the precise nature of legal barriers to disclosure and their appropriateness given the nature of the underlying data and code.³² Sequestering a dataset due to ‘confidentiality’, with no further justification, should no longer be acceptable practice.

The second corollary from the Principle of Scientific Data and Code Sharing is that it implies *levels* of access. Whether due to privacy concerns, technological barriers, or other sources, restrictions on data and code availability do not necessarily imply absolute barriers. In the case of CERN, internal research design mechanisms exist to make up for some of the shortcomings in openness of research data and the inability of independent groups to verify findings obtained from empirical data. Specifically, either independent research groups within CERN access the data from the collider and carry out the research in isolation from each other, or the same group will verify analyses using independent

toolsets.³³ Of crucial importance, these internal groups have access to the infrastructure and technologies needed to understand and analyze the data. In this case, there has been some openness of the data and the use of independent parallel research increases the chances of catching errors, all improvements over the more commonly seen research context where the data are accessed only by the original researchers and analyzed without any reported validation or verification cross-checks.

A second illustrative example originates from the Wandell Lab in the Psychology Department at Stanford University. Brian Wandell, the Isaac and Madeline Stein Family Professor of Psychology, has an MRI machine for his lab research. For the lifetime of the machine, each image has been carefully stored in a database with metadata including equipment settings, time, date, resolution, and other pertinent details of the experimental setup. The output image data are, however, subject to HIPAA regulations in that each image is a scan of a subject's brain and therefore privacy restrictions prevent these from being made publicly available. The Wandell Lab belongs to a consortium with several other research groups at different universities in California. In order to permit some potential verification of results based upon these images, there is no legal barrier to giving researchers within these authorized groups access to the database, and thereby creating the possibility for independent cross-checking of findings inside this 'walled garden'. While this does not achieve the same potential for finding errors as open release would (more eyes making more bugs increasingly shallow), it satisfies the Principle of Scientific Data and Code Sharing by maximizing access subject to the inherent legal constraints with which the data are endowed. Although the implementation details may differ for different data, understanding and developing infrastructure to facilitate these middle-ground data access platforms or walled gardens, will be essential for the reliability of results derived from confidential data.³⁴ One could also cast the CERN approach as a type of walled garden since it is characterized by independent research on the same question on closed data, carried out by different internal groups.

Another potential barrier to data and code release derives from collaboration with partners who may be unwilling to release the data and software that arise from the project, and may not be academic researchers bound by the same notions of scientific transparency. For example, industry research partners do not necessarily have the goal of contributing their research findings to the public good, but are frequent research collaborators with academics. A conflict can ensue, for example, at the point of publication when the academic partner wishes to publish the work in a journal or a conference proceedings that requires data and code disclosure, or when the researcher simply wishes to practice really reproducible research and make the data and code openly available.³⁵ One possible solution is to offer template agreements for data and code disclosure at the beginning of the collaboration, possibly through the institution's technology transfer office or through funding agency access policy.³⁶ Unfortunately the

issue of data and code access is often ignored until the point at which one party would like to make them available after the research has been completed.³⁷

When a patent is being sought on the software associated with the research, broader access can be achieved by implementing patent licensing terms that distinguish between commercial and research applications, in order to permit reuse and verification by researchers, while maintaining the incentives for commercialization and technology transfer provided by the Bayh-Dole Act. The Stanford Natural Language Processing Group for example uses such a dual licensing strategy. Their code is available for download by researchers under an open license and groups that intend commercial reuse must pay licensing fees.³⁸.

Challenge 3: Linked Data and Privacy Tort Law

Access to datasets necessarily means data with common fields can and will be linked. This is very important for scientific discovery as it enriches subject-level knowledge and opens new fields of inquiry, but it comes with risks such as revealing private information about individuals that the datasets in their isolated, unlinked form would not reveal. As has been widely reported, data release is now mandated for many government agencies through Data.gov. In 2009 Vivek Kundra, then–federal chief information officer,³⁹ was explicit – saying that, “the dream here is that you have a grad student, sifting through these datasets at three in the morning, who finds, at the intersection of multiple datasets, insight that we may not have seen, or developed a solution that we may not have thought of.” On February 22, 2013, the Office of Science and Technology Policy directed federal agencies with significant research budgets to remit plans to make data arising from this research openly available.⁴⁰ This includes academic research funded by the National Science Foundation and the National Institutes of Health, for example.

An instructive example about the privacy risks from data linking that Kundra describes comes from the release of genomic information. An individual’s genomic information could be uncovered by linking their relatives’ genomic information together, when this individual has not shared any of his or her genetic information directly. Recall, we carry 50% of the DNA from each of our parents and children, and an average of 50% from each of our siblings. Privacy risks could include, for example, an insurance company linking the genetic signature information to medical records data, possibly through a genetic diagnostic test that was performed, and then to other insurance claims, for individuals whose relatives had made their DNA available though they themselves did not.⁴¹ A number of cities are releasing data, for example public school performance data, social service agency visits, crime reports, and other municipal data, and there has been controversy over appropriate privacy protection for some of these data.⁴² Research that links these datasets may have laudable aims – better understanding the factors that help

students succeed in their education – but the risks to linking datasets can include privacy violations for individuals.

Much of the policy literature around privacy in digitally networked environments refers to corporate or government collected data used for commercial or policy ends.⁴³ Insufficient attention has been paid to the compelling need for access to data for the purposes of verification of data-driven research findings. This chapter does not advocate that the need for reproducibility should trump privacy rights, but instead that scientific integrity should be part of the discussion of access to big data, including middle ground solutions such as those discussed earlier in this chapter.

Traditional scientists in an academic setting are not the only ones making inferences from big data and linked data, as Chapters 6 and 7 in this volume show. The goal of better decision making is behind much of the current excitement surrounding big data, and supports the emergence of ‘evidence-based’ policy, medicine, practice, and management. For conclusions that enter the public sphere, it is not unreasonable to expect that the steps that generated the knowledge be disclosed to the maximal extent possible, including making the data they are based on available for inspection, and making the computer programs that carried out the data analysis available.

We cannot know how data released today, even data that all would agree carry no immediate privacy violations, could help bring about privacy violations when linked to other datasets in the future. These other datasets may not be released, or even imagined, today. It is impossible to guard completely against the risk of these types of future privacy violations. For this reason a tort-based approach to big data access and privacy is an important alternative to creating definitive guidelines to protect privacy in linked data. Perhaps not surprisingly, however, privacy tort law developed in the pre-digital age and is not a perfect fit with today’s notions of privacy protection and big data access.

Much of the current scholarly literature frames the online privacy violation question as protection again defamation or the release of private information by others, and does not explicitly consider the case of linked data. For example, privacy torts are often seen as redress for information made available online, without considering the case of harm from new information derived from combination of non-private sources. This can happen in the case of data linking, as described above, but differs in that a privacy violation can be wholly inadvertent and unforeseen, and may not be individually felt but can affect an entire class of people (those in the dataset). This, along with persistence of privacy-violating information on the web, changes the traditional notion of an individual right to privacy.⁴⁴ In current privacy tort law one must establish that the offender intended to commit a privacy invasion,⁴⁵ that the conduct was “highly offensive to the reasonable person,” and that the information revealed was sufficiently private.⁴⁶

Current privacy tort law protects against emotional, reputational, and proprietary injuries caused by any of: a public disclosure of private facts; an intrusion on seclusion; a

depiction of another in a false light, or an appropriation of another's image for commercial enrichment.⁴⁷ Articulating privacy rights in big data and linked data founders on accountability since it is unlike securing private (real) property or a landlord ensuring his or her building is secured.⁴⁸ Potential privacy violations deriving from linked data cannot always be foreseen at the time of data release. The Principle of Scientific Data and Code Sharing frames a possible way forward: research data that does not carry any immediate privacy violations should be released (and otherwise released in a way that makes the data maximally available for independent reproducibility purposes that safeguards privacy); linked datasets should either be released or the methods to link the datasets should be released with caveats to check for new privacy violations; and if privacy violations still arise, redress could be sought through the tort system. If tort law responds in a way that matches our normative expectations regarding privacy in data, this will permit a body of law to grow around big data that protects privacy. In order for this to be effective, a broadening of tort law beyond the four types of privacy-violating behaviors needs to occur. Harms arising from the release of private information derived from data, and from linked data, could be included in the taxonomy of privacy torts. These may not be intentioned or foreseeable harms, and may potentially be mass torts as datasets with confidentiality violations are likely to contain records on a large number of people. The issue of liability and responsibility for privacy violations becomes more complex than in the past, and there may be chilling effects on the part of institutions and funding agencies with regard to open data. Finally, making code and data available is not costless as databases and access software can cost a considerable amount of money, and innovative middle-ground solutions that may be project specific can add to that cost.⁴⁹

Research data poses yet another unique challenge to privacy law. Many research collaborations exist across international boundaries, and it is common for some members of a research team to be more heavily involved with the associated data than other members. Access to data on the Internet is not generally restricted by country and enforcing privacy violations across international borders poses a considerable challenge for scientific research. Data and code must be made available to maximally permit verification, subject to privacy and other barriers, and these data may be accessible from anywhere in the world through the Internet. Privacy violations from linked data can thus occur in countries with more stringent privacy standards though the release of the data may occur in a country that does not have a mechanism for legal redress of privacy violations.

Challenge 4: Changing Notions of Data Ownership and Agency

The notion of a data owner is a rapidly changing concept as many entities contribute to dataset creation, increasing the complexity of the data-sharing issue. Data is collected

both by people and by automated systems such as sensor arrays, and goes through myriad processing in the course of information extraction. Different entities may carry out data cleaning and filtering, data curation and warehousing, facilitation of data access, recombination of datasets to create novel databases, or preservation and provenance through repositories and institutions – each possibly creating intellectual property and ownership rights in the data. There is a similar story for research code, as it evolves through different applications and extensions by different people and becomes an amalgam of many contributions. The open release of data and code means untangling ownership and tracking contributions. Versions of code and data are vitally important for reproducibility – as code is modified, even as bugs are fixed, or data are extended, corrected, or combined, it is important to track which instantiation produced which scientific findings.

There is a new source of potential ownership as well. Subjects in a study can feel a sense of ownership over information about themselves, including medical descriptions or choices they have made. It is becoming increasingly the case that study participants wish to direct the level of access to data about themselves and traditional notions of privacy protection may not match their desires. Some data owners would prefer that data about themselves, that might traditionally be considered worthy of privacy protection such as medical data or data resulting from clinical trials participation, should be made more fully available.⁵⁰ As noted in a World Economic Forum Report, “[o]ne of the missing elements of the dialogue around personal data has been how to effectively engage the individual and give them a voice and tools to express choice and control over how data about them are used.”⁵¹ ⁵² Traditional mechanisms, such as the Institutional Review Board or national laboratory policy, may be overprotecting individuals at the expense of research progress if they are not taking individual agency into account.

These changing notions of ownership can impede sharing, if permission from multiple parties is required to grant open access, or to relinquish data, or even to simply participate in the development of infrastructure to support access. A careful assessment of ownership and contributions to dataset development will inform liability, in the case of breaches of privacy. While some of this assessment and tracking is done today for some datasets, for the majority of datasets there is very little provenance available and little clarity regarding ownership rights.

Conclusion

The goal of this chapter is to bring the consideration of scientific research needs to the discussion around data disclosure and big data. These needs comprise a variety of issues, but a primary one is the need for independent verification of results, for reproducibility from the original data using the original code. This chapter asserts two principles to guide

policy thinking in this area: the **Principle of Scientific Licensing**, that legal encumbrances to the dissemination, sharing, use, and re-use of scientific research compendia should be minimized, and require a strong and compelling rationale before their application; and the **Principle of Scientific Data and Code Sharing**, that access to the data and methods associated with published scientific results should be maximized, only subject to clearly articulated restrictions interpreted in the most minimally restrictive way, including intellectual property or HIPAA restrictions, or technological or cost constraints.

The chapter outlines intellectual property barriers to the open release of research data and code, and proposes open licensing solutions. Templatized sharing agreements are suggested to guide data and code release at the beginning of collaboration with industry partners who may have a different agenda to the open sharing of data and code that arise from the research. The chapter also argues for dual licensing of patented research code: license fees for commercial reuse, and open availability for academic research purposes. To address privacy and confidentiality in sharing there must be a move to maximize openness in the face of these concerns. Sharing within a group of authorized researchers in the field, or with scientists who have sought permission, can create a ‘walled garden’ that, while inferior to open sharing, can still obtain some of the properties and benefits of independent verification that is possible from public access. ‘Middle-ground’ platforms such as walled gardens are possible solutions to maximize the reliability of scientific findings in the face of privacy and confidentiality concerns.

The linking of open data sets is framed as an open-ended threat to privacy. Individuals may be identified through the linking of otherwise non-identifiable data. Since these linkages cannot, by definition, be foreseen and are of enormous benefit to research and innovation, the use of privacy tort law is suggested both to remedy harm caused by such privacy violations and to craft a body of case law that follows norms around digital data sharing.

Finally, privacy can be an overly restrictive concept, both legally and as a guiding principle for policy. Data ownership can be difficult to construe since many resources can create a useful dataset, and individuals may prefer to release what might be considered private information by some. In the structure of data collection and release today, such individuals have very little say over the future openness of their data. A sense of agency should be actively restored to permit individuals to share data.

Some of the concern about open data stems from the potential promulgation of misinformation as well as perceived privacy risks. In previous work I have labeled that concern ‘Taleb’s Criticism’.⁵³ In a 2008 essay, Taleb worries about the dangers that can result from people using statistical methodology without having a clear understanding of the techniques.⁵⁴ An example of Taleb’s Criticism appeared on UCSF’s EVA website, a repository of programs for automatic protein structure prediction.⁵⁵ The UCSF

researchers refuse to release their code publicly because, as they state on their website, "[w]e are seriously concerned about the 'negative' aspect of the freedom of the Web being that any newcomer can spend a day and hack out a program that predicts 3D structure, put it on the web, and it will be used." However, an analogy can be made to early free speech discussions that encouraged open dialog. In a well-known quote Justice Brandeis elucidated this point in *Whitney v. California* (1927), writing that "If there be time to expose through discussion the falsehood and fallacies, to avert the evil by the processes of education, the remedy to be applied is more speech, not enforced silence." In the open data discussion this principle can be interpreted to favor a deepening of the dialog surrounding research, which is in keeping with scientific norms of skepticism and the identification of errors. In the case of the protein structure software, the code remains closed and a black box in the process of generating research results.⁵⁶

Increasing the proportion of verifiable published computational science will stem from changes in four areas: funding agency policy, journal publication policies, institutional research policies, and the attitudes of scientific societies and researchers themselves. Although there have been significant recent advances from each of these four stakeholder groups, changing established scientific dissemination practices is a collective action problem. Data and code sharing places additional burdens on all these groups, from curation and preparation through to hosting and maintenance, which go largely unrewarded in scientific careers and advancement. These burdens can be substantial for all stakeholders in terms of cost, time, and resources. However, the stakes are high. Reliability of the results of our investments in scientific research, the acceleration of scientific progress, and the increased availability of scientific knowledge are some of the gains as we begin to recognize the importance of data and code access to computational science.

Acknowledgement I would like to thank two anonymous and extraordinarily helpful reviewers. This research was supported by Alfred P. Sloan Foundation award number PG004545 "Facilitating Transparency in Scientific Publishing" and NSF award number 1153384 "EAGER: Policy Design for Reproducibility and Data Sharing in Computational Science."

Notes

¹ Because of the wide scope of data considered in this article, the term *computational science* is used in a very broad sense, as any computational analysis of data. See V. Stodden, "Resolving Irreproducibility in Empirical and Computational Research," *IMS*

Bulletin, November 2013, for different interpretations of reproducibility for different types of scientific research.

² R. Gentleman and D. Temple Lang, “Statistical Analyses and Reproducible Research,” Bioconductor Working Series, 2004. Available at <http://biostats.bepress.com/bioconductor/paper2/>.

³ D. Donoho, A. Maleki, I. Ur Rahman, M. Shahram, and V. Stodden, “Reproducible Research in Computational Harmonic Analysis,” *Computing in Science and Engineering* 11, no. 1 (2009): 8–18. Available at <http://www.computer.org/cSDL/mags/cs/2009/01/mcs2009010008-abs.html>.

⁴ For a discussion of the Copyright Act of 1976 see e.g. Pam Samuelson, “Preliminary Thoughts on Copyright Reform Project,” *Utah Law Review* 2007 (3): 551–571. Available at <http://people.ischool.berkeley.edu/~pam/papers.html>.

⁵ See *Feist Publications Inc. v. Rural Tel. Service Co.*, 499 U.S. 340 (1991) at 363–364.

⁶ For a detailed discussion of copyright law and its impact on scientific innovation see V. Stodden, “Enabling Reproducible Research: Licensing for Scientific Innovation,” *International Journal for Communications Law and Policy*, no. 13 (Winter 2008–09). Available at http://www.ijclp.net/issue_13.html.

⁷ See V. Stodden “The Legal Framework for Reproducible Scientific Research: Licensing and Copyright,” *Computing in Science and Engineering* 11, no. 1 (2009): 35–40.

⁸ U.S. 17 Sec. 107.

⁹ This idea was suggested in P. David, “The Economic Logic of ‘Open Science’ and the Balance between Private Property Rights and the Public Domain in Scientific Data and Information: A Primer.” Available at <http://ideas.repec.org/p/wpa/wuwpdc/0502006.html>. For an analysis of the difficulty of an expansion of the fair use exception to include digital scholarly objects such as data see J. H. Reichman and R. L. Okediji, “When Copyright Law and Science Collide: Empowering Digitally Integrated Research Methods on a Global Scale,” *Minnesota Law Review* 96 (2012): 1362–1480. Available at http://scholarship.law.duke.edu/faculty_scholarship/267.

¹⁰ A research *compendium* refers to the triple of the research article, and the code and data that underlies its results. See Gentleman and Temple Lang, “Statistical Analyses and Reproducible Research.”

¹¹ See “Can I Apply a Creative Commons License to Software?” <http://wiki.creativecommons.org/FAQ>.

¹² <http://opensource.org/licenses/bsd-license.php>.

¹³ <http://opensource.org/licenses/mit-license.php>.

¹⁴ <http://www.apache.org/licenses/LICENSE-2.0>.

¹⁵ Miriam Bitton, “A New Outlook on the Economic Dimension of the Database Protection Debate,” *IDEA: The Journal of Law and Technology* 47, no. 2 (2006): 93–169. Available at <http://ssrn.com/abstract=1802770>.

¹⁶ *Feist v. Rural*, 340. The full quote is “Although a compilation of facts may possess the requisite originality because the author typically chooses which facts to include, in what order to place them, and how to arrange the data so that readers may use them effectively, copyright protection extends only to those components of the work that are original to the author, not to the facts themselves. . . . As a constitutional matter, copyright protects only those elements of a work that possess more than de minimis quantum of creativity. Rural’s white pages, limited to basic subscriber information and arranged alphabetically, fall short of the mark. As a statutory matter, 17 U.S.C. Sec. 101 does not afford protection from copying to a collection of facts that are selected, coordinated, and arranged in a way that utterly lacks originality. Given that some works must fail, we cannot imagine a more likely candidate. Indeed, were we to hold that Rural’s white pages pass muster, it is hard to believe that any collection of facts could fail.”

¹⁷ See A. Kamperman Sanders, “Limits to Database Protection: Fair Use and Scientific Research Exemptions,” *Research Policy* 35 (July 2006): 859, for a discussion of the international and WIPO statements of the legal status of databases.

¹⁸ For details on the CC0 protocol see <http://creativecommons.org/licenses/by/0.0/>.

¹⁹ V. Stodden, “The Scientific Method in Practice: Reproducibility in the Computational Sciences,” MIT Sloan School Working Paper 4773-10, 2010. Available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1550193.

²⁰ Ibid.

²¹ V. Stodden and I. Reich, “Software Patents as a Barrier to Scientific Transparency: An Unexpected Consequence of Bayh-Dole,” Conference on Empirical Legal Studies, 2012. Available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2149717.

²² See National Institutes of Health, “Principles for Recipients of NIH Research Grants and Contracts on Obtaining and Disseminating Biomedical Research Resources: Request for Comments.” Available at http://www.ott.nih.gov/policy/rt_guide.html.

²³ See Columbia University, *Faculty Handbook*, Appendix D: “Statement of Policy on Proprietary Rights in the Intellectual Products of Faculty Activity.” Available at <http://www.columbia.edu/cu/vpaa/handbook/appendixd.html> (accessed August 21, 2013).

²⁴ See Stanford University, *Research Policy Handbook*. Available at http://doresearch.stanford.edu/sites/default/files/documents/RPH%208.1_SU18_Patent%20and%20Copyright%20Agreement%20for%20Personnel%20at%20Stanford.pdf (accessed August 21, 2013).

²⁵ “Leahy Urges Action to Ensure Access to Affordable Life-Saving Diagnostic Tests for Breast and Ovarian Cancer” (press release). See <http://www.leahy.senate.gov/press/leahy-urges-action-to-ensure-access-to-affordable-life-saving-diagnostic-tests-for-breast-and-ovarian-cancer> (accessed August 21, 2013).

²⁶ See <http://www.sdss3.org/dr10/>.

²⁷ See http://www.sdss3.org/dr10/data_access/, including http://skyserver.sdss3.org/dr10/en/help/docs/sql_help.aspx, <http://skyserver.sdss3.org/CasJobs/>, and <http://skyserver.sdss3.org/dr10/en/help/browser/browser.aspx> (accessed August 23, 2013).

²⁸ “In Search of the Big Bang,” *Computer Weekly*, August 2008. Available at <http://www.computerweekly.com/feature/In-search-of-the-Big-Bang> (accessed August 23, 2013).

²⁹ “CERN Data Centre Passes 100 Petabytes,” *CERN Courier*, March 28, 2013. Available at <http://cerncourier.com/cws/article/cern/52730> (accessed August 23, 2013). 100 petabytes is about 100 million gigabytes or 100,000 terabytes of data. This is equivalent to approximately 1500 copies of the Sloan Digital Sky Survey.

³⁰ See D. H. Bailey, J. Borwein, and V. Stodden, “Set the Default to ‘Open’,” *Notices of the American Mathematical Society*, June/July 2013, available at <http://www.ams.org/notices/201306/rnoti-p679.pdf>, and V. Stodden, J. Borwein, and D. H. Bailey, “‘Setting the Default to Reproducible’ in Computational Science Research,” *SIAM News*, June 3, 2013, available at <http://www.siam.org/news/news.php?id=2078>.

³¹ For a complete discussion of HIPAA, see Chapters 1 (Strandburg) and 4 (Ohm) in this volume.

³² Some of these barriers were elucidated through a survey of the machine learning community in 2009. See Stodden, “The Scientific Method in Practice.”

³³ E.g. “All results quoted in this paper are validated by using two independent sets of software tools. ... In addition, many cross checks were done between the independent combination tools of CMS and ATLAS in terms of reproducibility for a large set of test scenarios” (from <http://cds.cern.ch/record/1376643/files/HIG-11-022-pas.pdf>).

³⁴ Reichman and Uhlir proposed that contractual rules governing data sharing, for example providing licensing terms or compensating creators, create a knowledge “semi-commons.” A ‘semi-commons’ can exist through data pooling and thus sharing the burden of warehousing and supporting access infrastructure and tools, in exchange for increased access to the data. However, the concept of the ‘walled garden’ is slightly different in this example in that authorized independent researchers are given full access to the resources for verification and/or reuse purposes thereby mimicking open data as fully as possible under the privacy constraints inherent in the data. J. H. Reichman and Paul F. Uhlir, “A Contractually Reconstructed Research Commons for Scientific Data in a Highly Protectionist Intellectual Property Environment,” *Law and Contemporary Problems* 66 (Winter 2003): 315–462. Available at <http://scholarship.law.duke.edu/lcp/vol66/iss1/12>.

³⁵ For an assessment of the reach of data and code disclosure requirements by journals, see V. Stodden, P. Guo, and Z. Ma, “Toward Reproducible Computational Research: An Empirical Analysis of Data and Code Policy Adoption by Journals.” *PLoS ONE* 8, no. 6 (2013). Available at <http://www.plosone.org/article/info%3Adoi%2F10.1371%2Fjournal.pone.0067111>.

³⁶ For a further discussion of such template agreements, see V. Stodden, “Innovation and Growth through Open Access to Scientific Research: Three Ideas for High-Impact Rule Changes,” in *Rules for Growth: Promoting Innovation and Growth through Legal Reform* (Kansas City, MO: Kauffman Foundation, 2011). Available at http://www.kauffman.org/~/media/kauffman_org/research%20reports%20and%20covers/2011/02/rulesforgrowth.pdf.

³⁷ Of course, researchers in private sector for-profit firms are not the only potential collaborators who may have a different set of intentions regarding data and code availability. Academic researchers themselves may wish to build a start-up around the scholarly objects deriving from their research, for example. In a survey conducted by the author in 2009, one senior academic wrote he would not share his code because he intended to start a company around it. See Stodden, “The Scientific Method in Practice.”

³⁸ See <http://nlp.stanford.edu/software/>.

³⁹ See http://www.whitehouse.gov/the_press_office/President-Obama-Names-Vivek-Kundra-Chief-Information-Officer/ (accessed September 1, 2013).

⁴⁰ See “Expanding Public Access to the Results of Federally Funded Research,” <http://www.whitehouse.gov/blog/2013/02/22/expanding-public-access-results-federally-funded-research> (accessed September 1, 2013).

⁴¹ For other examples see e.g. J. E. Wiley and G. Mineau, “Biomedical Databases: Protecting Privacy and Promoting Research,” *Trends in Biotechnology* 21, no. 3 (March 2003): 113–116. Available at <http://www.sciencedirect.com/science/article/pii/S0167779902000392>.

⁴² See e.g. <https://data.cityofchicago.org/> and <http://schoolcuts.org>, <https://nycopendata.socrata.com/> and <https://data.ny.gov/> (state level), and <https://data.sfgov.org/>. The Family Educational Rights and Privacy Act (FERPA) attempts to address this with a notion of student privacy; see <http://www.ed.gov/policy/gen/guid/fpcos/ferpa/index.html>. The State of Oklahoma recently passed a bill, the Student DATA Act, to protect student school performance data; see <http://newsok.com/oklahoma-gov.-mary-fallin-signs-student-privacy-bill/article/3851642>. In New York State a case was filed in 2013 to prevent a third party from accessing student data without parental consent; see <http://online.wsj.com/article/AP0d716701df9f4c129986a28a15165b4d.html>.

⁴³ See e.g. the report from the World Economic Forum, “Unlocking the Value of Personal Data: From Collection to Usage,” (Geneva, 2013). Available at <http://www.weforum.org/reports/unlocking-value-personal-data-collection-usage>.

⁴⁴ See e.g. D. Citron, “Mainstreaming Privacy Torts,” *California Law Review* 98 (2010): 1805–1852.

⁴⁵ See e.g. *McCormick v. Haley*, 307 N.E.2d 34, 38 (Ohio Ct. App. 1973).

⁴⁶ Restatement (Second) of Torts § 652B (1977).

⁴⁷ See Citron, 1809.

⁴⁸ Citron; e.g. *Kline v. 1500 Massachusetts Ave. Apartment Corp.*, 439 F.2d 477, 480–81 (D.C. Cir. 1970), holding the landlord liable for a poorly secured building when tenants were physically beaten by criminals.

⁴⁹ See e.g. F. Berman and V. Cerf, “Who Will Pay for Public Access to Research Data?” *Science* 341, no. 6146 (2013): 616–617. Available at <http://www.sciencemag.org/content/341/6146/616.summary>.

⁵⁰ Individuals may direct their data to be used for research purposes only, or to be placed in the public domain for broad reuse, for example. See e.g. Consent to Research,

<http://weconsent.us>, which supports data owner agency and informed consent for data sharing beyond traditional privacy protection.

⁵¹ World Economic Forum, 12.

⁵² Some restrictions on subject agency exist; see e.g. *Moore v. Regents of University of California* 51 Cal.3d 120 (Supreme Court of California July 9, 1990). This case dealt with ownership over physical human tissue, and not digital data, but the tissue could be interpreted as providing data for scientific experiments and research, in a role similar to that of data. See also the National Institutes of Health efforts to continue research access to the Henrietta Lacks cell line, taking into account Lacks family privacy concerns. E. Callaway, “Deal Done over HeLa Cell Line,” *Nature News*, August 7, 2013. Available at <http://www.nature.com/news/deal-done-over-hela-cell-line-1.13511>.

⁵³ V. Stodden, “Optimal Information Disclosure Levels: Data.gov and ‘Taleb’s Criticism’,” <http://blog.stodden.net/2009/09/27/optimal-information-disclosure-levels-datagov-and-talebs-criticism/>.

⁵⁴ N. Taleb, “The Fourth Quadrant: A Map of the Limits of Statistics,” http://www.edge.org/3rd_culture/taleb08/taleb08_index.html (accessed September 1, 2013).

⁵⁵ See <http://eva.compbio.ucsf.edu/~eva/doc/concept.html> (accessed September 1, 2013).

⁵⁶ See e.g. A. Morin, J. Urban, P. D. Adams, I. Foster, A. Sali, D. Baker, and P. Sliz, “Shining Light into Black Boxes,” *Science* 336, no. 6078 (2012): 159–160. Available at <http://www.sciencemag.org/content/336/6078/159.summary>.

Part II

Practical Framework

The essays in this section make powerful arguments for the value of data in the public sector. We are all aware of the value to the private sector; indeed, the value of many of large companies in the US, like Google, Facebook and Yahoo, lie in their access to large datasets on individual behavior, and their ability to turn data into privately held information. Yet the experience of the authors demonstrates that the gap between vision and reality in the public sector is large, for many reasons. The authors identify new approaches that can enable the public sector custodians to combine and use data, and new approaches to enable researcher access so that data can be turned into publicly held information. A major leitmotif in each chapter is of course, trust.

What is the vision? An illustrative, but not exhaustive, list identified by the authors of the potential and actual value of big data range from simply better, targeted, city management to reduced taxpayer cost and burden, from great transparency and less corruption to greater economic growth, and indeed to addressing problems of epidemics, climate change and pollution. Interestingly, as Elias points out, the European Commission recognized the value as far back as 1950, when the European Convention on Human rights noted '*There shall be no interference by a public authority with the exercise of this right [to privacy] except such as in accordance with the law and is necessary in a democratic society in the interest of national security, public safety or the economic well-being of the country, for the protection of disorder or crime, for the protection of health or morals, or the protection of the rights and freedoms of others'* (Elias, this volume) Indeed, as Greenwood et al. point out in an evocative phrase, data can be seen as the oil in the new economy and we should build towards providing the appropriate business, legal and technical (BLT) infrastructure to facilitate its use.

The reality is quite different. In practice, as Goerge points out, there are many practical decisions to be made, including what data to access, how to build capacity to use and present the data, and how to keep data secure. And the challenges include the fact that public attorneys and data custodians are often reluctant to provide data access because of the unclear legal framework and the downside risk associated with privacy breaches. Many of the problems could be addressed with sufficient funding, but the primary challenge identified by both Goerge and Koonin/Holland is building the trust necessary to provide access.

Closing the gap between the vision and the reality is the practical thrust of most of the chapters. The most important task is building trust. Elias notes that a very useful UK survey provides a roadmap. It found that the key elements to build trust include identifying the legal status of those bodies holding data; developing agreed and common standards covering data security and the authentication of potential research users;

developing public support for the use for research of de-identified personal information; and creating a coordinated governance structure for all activities associated with access, linking and sharing personal information.

A logical set of next steps, then, is to move from artisanal approaches to protecting privacy to a much more systematic approach. The authors provide a set of illustrative suggestions which very much mirror the UK survey. Greenwood et al. propose a set of regulatory standards and financial incentives to entice owners to share data (very much in the spirit of the Acquisti chapter earlier in the book). They explicitly discuss BLT rules that can be developed by companies and governments. They propose using big data itself to keep track of user permissions for each piece of data and act as a legal contract. Most specifically, they propose building an open Personal Data Store (openPDS) personal cloud trust network and Living Informed Consent, where the user is entitled to know what data is being collected, by what entities and put in charge of sharing authorizations. Landwehr is less sanguine about the adoption of such technologies. He argues for analysis on encrypted files or building systems in which information flow, rather than access control is used to enforce policies. Wilbanks proposes portable legal consent framework, which is a commons-based approach that can be used to recruit individuals who understand the risks and benefits of data analysis and use.

All of these approaches must be built to be scalable for big data. If public is to know what is being done to their data, and users are to know the analytical properties of the data, it is critically important to track data provenance – and even more importantly, information flows. This is very difficult territory indeed, as Landwehr points out. Provenance information has been characterized formally as an acyclic directed graph; such graphs get complex very fast, yet tracing changes is necessary to both replication and validation of scientific results. Most applications in the public sector are not designed to assure users or data providers that big data sets are accessed according to prescribed policies; hence in the near future, unless a sustained effort is put in place to build applications to code, the only approaches are likely to be manual and individuals will need to trust researchers. This, in turn, raises a major problem, because, as noted by Wilbanks, the reality is that informed consent terms and procedures are written by non experts, with fields of study very different from reidentification.

In sum, the vision of big data for the public good can be achieved – the authors provide evidence from cities from Chicago to New York and in areas from health and the environment to public safety. But if the vision is to be delivered in a large scale fashion, the authors in this chapter also make it clear that the public sector must make substantial investments to build on the infrastructure ideas identified by the authors in this chapter, as well as others. If big data are the oil of the new economy, it will be necessary to build the data equivalent of interstate highways.

Chapter 6

The Value of Big Data for Urban Science

Steven E. Koonin and Michael J. Holland

Introduction

The past two decades have seen rapid advances in sensors, database technologies, search engines, data mining, machine learning, statistics, distributed computing, visualization and modeling and simulation. These technologies, which collectively underpin ‘big data’, are allowing organizations to acquire, transmit, store, and analyze all manner of data in greater volume, with greater velocity and of greater variety. Cisco, the multinational manufacturer of networking equipment, estimates that by 2017 there will be three networked devices for every person on the globe.¹ The ‘instrumenting of society’ that is taking place as these technologies are widely deployed is producing data streams of unprecedented granularity, coverage, and timeliness.

The tsunami of data is increasingly impacting the commercial and academic spheres. A decade ago, it was news that Walmart was using predictive analytics to anticipate inventory needs in the face of upcoming severe weather events.² Today, retail (inventory management), advertising (online recommendation engines), insurance (improved stratification of risk), finance (investment strategy, fraud detection), real estate, entertainment, and political campaigns routinely acquire, integrate, and analyze large amounts of societal data to improve their performance. Scientific research is also seeing the rise of big data technologies. Large federated databases are now an important asset in physics, astronomy, the earth sciences, and biology. The social sciences are beginning to grapple with the implications of this transformation.³ The traditional data paradigm of social science relies upon surveys and experiments, both qualitative and quantitative, as well as exploitation of administrative records created for non-research purposes. Well-designed surveys generate representative data from comparatively small samples, and the best administrative datasets provide high-quality data covering a total population of interest. The opportunity now presents to understand how these traditional tools can be complemented by large volumes of ‘organic’ data that are being generated as a natural part of a modern, technologically advanced society.⁴ Depending upon how sampling errors, coverage errors, and biases are accounted for, we believe the combination can yield new insights into human behavior and social norms.

Governments too are exploring whether making their data more open can help them to become more participatory, decentralized, and agile institutions able to address

problems faster and more successfully on behalf of their citizens. As a result, open government data portals are becoming common in the United States at the federal, state and local levels.⁵ To seize these opportunities, agencies will need to build their own internal capacity for data analytics as well as make judicious use of the expertise of their vendor communities if they are to deliver services more efficiently, increase the precision and accuracy of enforcement actions, set more informed policies, or more effectively plan infrastructure improvements. Not only can administrative, regulatory, and enforcement agencies benefit from improved data analytics, but statistical agencies are looking for additional tools to help them fulfill their obligation to produce accurate national, state, or local statistics (while cautious given that harms resulting from disclosures can seriously impact participants' willingness to participate in surveys).⁶ Citizens too are interested in urban data to ensure government transparency and accountability as well as to enhance their local government's opportunities to improve urban living.⁷

Recognizing the economic value of government data beyond the usual arguments for increasing government transparency and efficiency, the Obama administration issued in May 2013 an executive order to make information generated and stored by the federal government more open and accessible with an explicit goal of fueling entrepreneurship, innovation, and scientific discovery.⁸ Putting urban data in the hands of citizens has the potential to improve governance and participation, but data in the hands of entrepreneurs and corporations can also stimulate the development of new products and services. Climate Corporation, a start-up that was acquired in late 2013 by Monsanto for about \$1 billion, combines 30 years of weather data, 60 years of crop yield data, and 14 terabytes of soil data – all from government agencies – for such uses as research and pricing crop insurance.⁹ A recent study by the Knight Foundation, which has supported activities at the nexus of technology, civic innovation, open government, and citizen engagement, found that these 'civic tech' firms in the U.S. garnered more than \$430 million from private sector investors and foundations between January 2011 and May 2013.¹⁰ Start-ups facilitating peer-to-peer sharing attracted the most private sector investment, while start-ups facilitating access, transparency, and usability of government data attracted the most foundation investment. In an analysis of the worldwide value of open data for both government and industry, the McKinsey Global Institute estimates that open data could enable more than \$3 trillion in additional value annually across the seven domains it analyzed: education, transportation, consumer products, electricity, oil and gas, health care, and consumer finance;¹¹ that \$3 trillion is 3.4% of the estimated 2013 gross world product of US\$85 trillion.¹²

At the Center for Urban Science & Progress, our goal is to collect and analyze data that will allow us to characterize and quantify the 'pulse of the city'. We are not alone in believing that a new science of cities is emerging, with an understanding of how scaling laws and scientific simulations can apply to transportation systems, energy use, economic

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

activity, health, innovation, and the full range of urban activities.^{13,14,15} In this chapter, we first address the motivations for this new urban science by defining a broad set of systems of interest and describe a data taxonomy as we see them applying to our field of study. We take note where we see particular municipal interests in these data flows. We then discuss some of the technical issues where we see big data differing from traditional data analyses for urban issues, and we close with a small set of non-technical issues that we believe warrant attention if cities are to fully realize the potential of big data analytics.

Urban Science

Given the trend towards more data and increasing availability of open data, it is not a fantasy to ask “if you could know anything about a city, what do you want to know?” understanding that local governments have responsibility for education; fire; police; delivery of human services; operation of public works like streets, sewers, and solid waste and storm water management; urban planning and zoning; fostering local economic development; and development of parks and recreational opportunities to improve the quality of life. Cities deliver services to their citizens through infrastructure and through processes. We want to know how those systems operate, how they interact, and how they can be optimized.

There are three classes of urban systems about which data must be acquired.

The Infrastructure Major questions about urban infrastructure focus on its extent, condition, and performance under varying scenarios of use. We need to know the condition of the built infrastructure: Are the bridge joints corroding? Can we find the leaky pipes? Which pavement resists excessive wear from heavy vehicles? We need to understand the operation of the infrastructure: How is traffic flowing? Is the electrical grid balanced? Is building energy efficiency performing as expected?

The Environment Major questions about the urban environment focus on the sources and fates of pollutants, the health burdens those pollutants place on vulnerable subpopulations, and the vitality of natural systems facing demands for environmental services. We need to understand whether a city’s river can support recreational uses such as fishing and rowing when simultaneously allowing for nearby industrial uses. In addition to the usual meteorological and pollution variables of interest, we need to understand the full range of environmental factors, such as noise, that influence people’s day-to-day experience of the city.

The People Major questions about urban populations focus on the interactions of people with each other and institutions, their interactions as organizations, as well as their interactions with the built and natural environments. Cities are built by and for people

and so cannot be understood without studying the people: their movement, health status, economic activities, how they communicate, their opinions, etc.

Yet it is this ever finer temporal and spatial granularity of data about individuals and the increasing power of informatic tools to combine and mine these streams of data that stoke concerns about privacy and data access, particularly when these tools are in the hands of individuals or organizations whose interests are not perceived as being aligned with those of the data subjects. Further development of both technical tools and administrative controls that can assure privacy and security of potentially massive data flows are necessary precursors to the deeper scientific study of cities.

An Urban Data Taxonomy

From an urban science perspective, data can be thought of as falling into four broad categories according to how it is generated: transactional data, *in situ* sensor data, remote sensor data, and citizen science data. Privacy concerns can arise not only over how the data is generated, but also as a result of where the data is generated, collected, contributed and how it is correlated with other data – whether by government, private sector institutions, or individuals. We will comment on these differences, but we do not want to suggest our treatment is comprehensive.

Transactional Data The first category of urban data is the traditional transactional data – the text and numerical records – that agencies and commercial entities generate in their routine course of business. These data sources are the familiar records such as permits, tax records, public health and land use data in the public sector or sales, inventory, and customer records in the private sector that social scientists have been exploiting for decades, if not centuries. Text and numerical records can be aggregated at the city level (census, statistical bureaus), at the firm or neighborhood level (census blocks, tracts, neighborhoods), or at the individual level (retail sales records, surveys). As commerce, government, and many individual activities migrate to the digital sphere, the available volume of data is growing and the vast majority of it is ‘born digital’.

For municipal governments, a major opportunity lies in extracting the full value of the traditional transactional data already in their possession. As an example, the City of New York was able to prioritize the 300 or so daily complaints about illegal housing conversions so that enforcement actions focus on those that posed the highest risk of deaths to occupants and first responders as a result of fire or structural collapse. By pulling together existing information on foreclosures, tax liens, building complaints, sanitation violations and building wall violations from multiple departments, the City increased its rate of issuing orders to vacate unsafe properties from less than 3% of onsite inspections to well over 40%.¹⁶ Developing IT architectures and interagency agreements

that allow data analysis systems to operate seamlessly across disparate agency datasets is a significant enabling technical and organizational challenge for the field.

In Situ Sensor Data Next, data collected from the local environment immediately around a sensor or scanner is the most rapidly growing category of data relevant to the interests of urban science. Enabled by progressively cheaper microprocessors and wireless communications, engineers are rapidly developing methods to instrument infrastructure and the environment or extract people's movement from commonly used personal electronic devices, such as cell phones.¹⁷ The expanding 'internet of things' enabled by the ease of scanning barcodes or QR codes and the plummeting price of RFID tags will only accelerate the stream of data related to object identity, location, and time of last movement. Questions of ownership of such data streams are discussed by Stodden in Chapter 5 of this volume.

For municipal governments, who have a more complete toolkit for influencing the local built environment than do either federal or state governments, a major opportunity lies in understanding with increasing spatial and temporal resolution how their urban infrastructure is being used. As an example, researchers used three-week-long mobile phone billing records generated by 360,000 San Francisco Bay Area users (6.56% of the population, from one carrier) to measure transient origin–destination matrices for vehicles.¹⁸ Their dataset, which is two orders of magnitude larger in terms of population and time of observation than the most recent surveys, allowed them to allocate conclusively the major traffic flows on congested roads to a very few geographical sources of drivers. This suggests that traffic engineers should focus their efforts to modify commuting behavior on just those few driver sources, rather than implementing measures seeking to change behavior within the full commuting region.

Beyond fixed *in situ* sensors to record light levels, temperature, loading, pollution, etc., personal sensors that record location, activity, and physiology are becoming available. Detailed personal time series data are starting to be voluntarily made public by athletes using Fitbit activity monitors or by those in the quantified-self communities. Newly emerging applications of portable, unobtrusive assistive health care technologies for monitoring those with physical or cognitive impairments raise privacy concerns, but also present new opportunities for municipalities to improve the provision of human services.¹⁹ Social media streams, such as Facebook, Twitter, and Foursquare, may be considered a specialized subset of this data category, particularly when postings of activity or sentiment are geocoded. The privacy concerns presented by social media streams have been adequately commented upon by other authors in this volume, except to say that citizens, by virtue of their far more frequent interaction with local government compared to state or national governments and greater feeling of control, may be more

willing to share information with local agencies if they see improved services in exchange.

Remote Sensor Data Cameras and other synoptic sensors are a rich new source of data relevant to urban science. There is an ongoing proliferation of video cameras at points of commerce and automatic teller machines and at portals for pedestrians and vehicles. Despite an estimated 30 million cameras in public spaces in the United States, very little of the video collected is routinely analyzed, other than as needed for post-event forensics. Traffic scene surveillance for congestion and license plate monitoring may be the major exceptions. Rapid automated analysis of camera feeds is computationally challenging, but computer vision enabled by unsupervised machine learning is beginning to open up new opportunities, with real-time labeling of objects in natural scenes possible.²⁰ Privacy concerns precipitated by the ease of unauthorized discovery of webcams are of regulatory interest in the United States. The Federal Trade Commission settled a case in 2013 against TRENDnet, which sold its Internet-connected SecurView cameras for purposes ranging from home security to baby monitoring, after hackers posted live feeds of nearly 700 consumer cameras on the Internet, showing activities such as babies asleep in their cribs and children playing in their homes.²¹

For municipalities, their business improvement districts (BIDs) may be among the earliest, most enthusiastic adopters, aside from police departments, of facial recognition software and other video feeds for monitoring activities in city centers. BIDs provide supplemental services, such as cleaning streets, providing security, making capital improvements, constructing pedestrian and streetscape enhancements, and marketing the area.²² They are non-profit or quasi-governmental entities authorized by local governments to which businesses within that district's boundaries pay an additional tax in order to fund projects. For some urban science studies, BIDs may be data sources as rich as those of city agencies.

Persistent remote sensing also offers new possibilities for urban science. While transient remote sensing of urban features from satellites or aircraft is well established,²³ persistent observation from urban vantage points is an intriguing possibility. Instrumentation on a tall building in an urban center can ‘see’, modulo shadowing, tens of thousands of buildings within a 10 kilometer radius, without the mass, volume, power, or data rate constraints of airborne platforms. As an example, varying sampling rates in the visible spectrum allows for the study of a range of phenomena. At low sampling rates, one can watch new lighting technologies penetrate a city and correlate what is known about early adopters or lagging adopters from municipal permitting databases to tease out the behavioral and financial components of energy-efficient lighting technology diffusion. At very high sampling rates, transients observable in the lights might provide a measure of other plug loads that would only be accessible with expensive submetering.

This is a preliminary version of the book Privacy, Big Data, and the Public Good: Frameworks for Engagement, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Moderate sampling rates can observe daily behavioral information. Visible, infrared, hyperspectral, and radar imagery are all phenomenologies to be explored for urban scenes, as are RADAR and LIDAR (Light Detection and Ranging).

The synoptic and persistent coverage of such modalities, together with their relatively easy and low-cost operation, may offer a useful complement to *in situ* sensing. Privacy issues can be addressed, in part, by careful design of the spatial resolution of the collected images. Clearly, unexpected monitoring from a public vantage point raises issues of data collection, which Strandburg addresses in Chapter 1 of this volume. Information collections of which individuals are unaware have far greater potential to disturb once those collections are revealed. Particular care needs to be taken in the design, approval, and socialization of remote-sensing campaigns.

Citizen Science Data Participatory (crowd-sourced) data streams are a potentially important tool of urban science.²⁴ There is a long history of successful citizen science where amateur scientists have made significant contributions. One of the longest running is the Audubon Christmas Bird Count, a repository of early winter bird observations recorded since 1900 that has been used by academic researchers and federal, state, and local wildlife and land planning agencies.²⁵ The last decade has seen a huge expansion in the sorts of scientific endeavor where non-professionals can contribute, thanks to the extraordinary development of information technology. Activities have moved beyond donation of spare computing resources, such as SETI@home for analysis of radio telescope data, to the participatory sensing of environment phenomena, such as noise.²⁶ Mobile participatory platforms collecting a variety of location, photos, and text inputs will require many of the tools of big data to fully exploit their output.

For municipal governments, citizen science activities can provide data at a geospatial resolution unobtainable with tools normally available to agencies. One example is participatory urban tree inventories using mobile platforms, such as OpenTreeMap, which allows volunteers to input information (location, species ID, photos) for individual trees. Tree-level data is more useful from a forestry management perspective than the aggregate tree canopy coverage commonly available from overhead imaging techniques. While measurement errors have been studied for manually collected field data,²⁷ understanding the validity of data streams from these new tools is evolving. Privacy issues related to any personally identifiable information about volunteers need to be carefully considered in the design of the mobile application,²⁸ and campaigns that permit data collection in the United States by children under age 13 can require compliance with the Children's Online Privacy Protection Act of 1998.

How Is Big Data Different?

Aside from their origins, traditional microdata resulting from censuses, sample surveys, administrative records, and statistical modeling differ from big data in several technical aspects as noted by Capps and Wright.²⁹ Much of the usual microdata encompass records numbering in the hundreds of millions, while big datasets are many orders of magnitude greater. The computational challenges associated with massive data management are substantially different from those for static datasets in terms of scale and throughput. Technical advances are required to scale data infrastructure for curation, analytics, visualization, machine learning, data mining, as well as modeling and simulation to keep up with the volume and speed of data.³⁰

Official statistics and datasets tend towards periodic cycles of input, analysis, and release – a corporation’s quarterly earnings report or the Bureau of Labor Statistics’ Employment Situation Summary on the first Friday of every month – while much of the data relevant to urban science flows continuously. Many government agencies or corporations would like to analyze that data in real time for operational reasons. Traditional microdata, including surveys, tends to be labor intensive, subject to human error, and costly in their collection, while big data are often born digital and seem relatively cheap by comparison.³¹

Surveys, which form the foundation of official statistics, “are the result of careful data collection design with clearly defined uses, while big data come with unknowns (e.g. uses are less clear, data are less understood, data are of unknown quality, and representativeness is largely unknown).”³² Capps and Wright also note that with respect to surveys, response assumes permission to use. Big data, as with much traditional administrative data, come as byproducts of other primary activities without asking explicitly and thus without any assumed permission to use beyond uses compatible with the purpose for which the data was collected. In fact, one might argue that the exploitation of data originally acquired for another purpose is a hallmark of big data in an urban science setting given the potential scale of data generated and held by municipalities.

Somewhat counterintuitively, sheer scale is one of the few characteristics of big data that can help limit some – though clearly not all – privacy or confidentiality issues. Massive datasets at the petascale and above are challenging to transfer, since the high-capacity, wide-scale communications networks required are extremely expensive to maintain. In physics and astronomy where such datasets are common, analyses are sent to the data rather than propagating copies of data for independent use. As a result, disclosure risk measures need only be implemented for the training data used to develop estimation procedures and for the final results prior to transmission back to the analyst. Subsets of massive data collections do remain vulnerable to unauthorized and undetected copying, but that risk is improbable for the full collection. In the urban science arena, data from sensor networks and simulation data could reach such scales.

Privacy Risks The value of any large urban dataset is enhanced through its association with other data. Observations are linked through location and time, as well as through entity (person, firm, vehicle, structure). The power of such linkage in producing new information is significant and increasingly well recognized. For example, knowing an individual's ZIP code localizes that person to 1 in 30,000 (the average population of a ZIP code).³³ Linking a ZIP code with a birthdate reduces the pool to approximately one in 80, while further connecting gender and year-of-birth are sufficient, on average, to uniquely specify an individual.

However, data mining, data linking, and statistical analyses are not the only source of risk presented by big data technologies. Reliance upon distributed or cloud computing resources can create additional privacy risk including risks from unencrypted intermediate datasets resident in the cloud,³⁴ but other authors in this volume, particularly Landwehr in Chapter 10, will address security issues in detail. Tools for visualizing massive datasets are becoming increasingly powerful, allowing users to explore datasets with hundreds of millions of records interactively in real time as a recently developed TaxiViz tool demonstrates.³⁵ Data for 540 million trips can be interrogated graphically in real time. Each trip record consists of: trip id, taxi id, driver id, pickup location, dropoff location, pickup date and time, dropoff date and time, traveled distance, fare amount, tip amount, and toll amount. Taxi and driver ids were anonymized to avoid the linking of records to the actual taxi medallion and taxi driver's license. Users can draw arbitrarily small regions of interest onto the underlying map, defining a region of trip origin and a region of trip destination. The visualization tool then shows taxi rides meeting those origin–destination criteria. Applied to a passenger who regularly catches a cab in front of their house or apartment and is dropped off at their place of work, such a tool could easily allow detailed analysis of any variation in that regular pattern. Applying traditional disclosure prevention procedures, such as suppressing cells in statistical tables based upon survey data, is not straightforward in an exploratory data visualization tool. Depending upon exposure risks of the particular data involved, the software engineering required to limit allowable queries could be quite sophisticated. Techniques such as parallel coordinates are beginning to be explored as a method to allow privacy-preserving data visualizations.³⁶

Realizing the Value of Urban Data

Agencies, businesses, and researchers are better able to turn the deluge that is big data into useful information and understanding when access to data is at its most open. The Open Knowledge Foundation sets out a vision that "a piece of data or content is open if anyone is free to use, reuse, and redistribute it – subject only, at most, to the requirement to attribute and/or share-alike."³⁷ In practice, there are multiple situations where issues of

safety, security, liability, confidentiality, or proprietary concerns limit the realization of that vision. In this section, we identify a few steps we believe municipalities and the data science community can take to facilitate the use of public data.

To correlate data, it must be brought together. But organizational barriers, within and between, institutions abound. As Wilbanks notes in Chapter 11, in his discussion of frameworks for sharing data, use is frequently “governed by a hodge-podge of contractual instruments” flowing from the data collection. Where data is not open by default, the transaction costs, particularly in length of time to reach closure, arising from negotiation of data transfer agreements with multiple agencies and bureaus can be substantial. Municipal governments can facilitate the usability of their public data by developing standardized data-sharing agreements appropriate to the laws and regulations of their jurisdictions in consultation with interested individuals, civic organizations, and businesses. In addition, municipalities can design their data systems to enable sharing by building internal IT interfaces as if they were citizen-facing external interfaces.³⁸ Doing so makes it easy to open data to external connections, whether to the research community, the hacker community, or technology vendors, once the decision to do so is made.

Once brought together, tools exist for managing disclosure risk, which Karr and Reiter discuss in Chapter 13. Many of these have been developed by statistical agencies and in settings, such as education, medicine, and financial services, where statutory mandates act as a driver. Methods continued to be developed for estimating re-identification risks in these specialized settings.³⁹ A current challenge is to adapt those tools so that data scientists can implement them on their broader range of datasets without excessive computational penalties. We should also recognize that all disclosure does not have equal potential for harm nor are individuals and organizations uniform in their tolerance of disclosure. If the risk estimation tools can be developed, it would be helpful to use consistent, transparent language for communicating disclosure risks akin to what the climate community uses to make distinctions between levels of confidence in scientific understanding and the likelihoods of a specific result, e.g. ‘very high confidence’ means a statement has at least a 9 out of 10 chance of being correct, or ‘extremely unlikely’ means a less than 5% probability of the outcome.⁴⁰

Finally, work by the information science, management information systems, and e-government research communities has documented barriers to value creation from open data platforms, among them are problems of diverse user needs and capabilities, the limitations of internally oriented data management techniques, untested assumptions about information content and accuracy, and issues associated with information quality and fitness for use.⁴¹ Fortunately, there are signs that municipalities are building their own capacity for data science, with groups established in multiple cities. Prominent examples include the Mayor’s Office of Data Analytics and the Center for Innovation

through Data Intelligence in New York City and a Mayor's Office of New Urban Mechanics in both Boston and Philadelphia, but Chief Information Officers in cities as large as Chicago and as small as Asheville, NC, are taking steps to develop data science capacities. Having capable, in-house data scientists who can demonstrate to their fellow civil servants the value big data has for solving practical problems may be one of most significant steps any municipal government can take in breaking down the barriers to value creation.

Conclusions

In closing, we wholeheartedly agree with the National Research Council's Committee on the Analysis of Massive Data assessment that "it is natural to be optimistic about the prospects" for big data.⁴² We believe the tools of big data combined with increasingly open data will improve our scientific understanding of the cities that will be home to 67% of the world's population by 2050⁴³ and could contribute as much as \$3 trillion annually to world economic growth. But those benefits are not foregone conclusions. Concerns over privacy precipitated by these developing big data technologies will lead to a reassessment, and possibly a rebalancing, of access as uses evolve and benefits to society at large are weighed against costs to individuals. Even public records (property, criminal, court, birth, death, marriage, divorce records, licenses, deeds, mortgages, corporate records, business registration) open for hundreds of years may have access restricted or some of the personally identifiable information they contain suppressed given that those records, now issued in electronic format, have become accessible and transmissible in ways that were never previously possible or expected.⁴⁴

However, we should note that not all potential threats to realizing the value of big data are privacy related. Quantification often brings unexamined power and prestige to public policy debates,⁴⁵ so caution in the interpretive power of data analyses is crucial, given the real potential for harm in some cases.⁴⁶ In the late 1960s, New York City Mayor John Lindsay hired consultants from the RAND Corporation to help modernize municipal service delivery and achieve budget savings. RAND recommended an overhaul of fire station locations and the number of engines responding to fires, based on flawed firefighter response time data. When fire broke out in the Bronx, firefighters were unable to respond in time, and fires ended up burning out of control.

For those of us who are interested improving our scientific understanding of how cities operate, our goal should be not just relevant research but impact, but we need to approach this goal with a degree of humility. And so, urban data scientists need to be aware continually of the context from which data comes, the context in which analyses are used to make decisions, and the context within which privacy concerns are balanced.

Notes

¹ “Cisco Visual Networking Index: Forecast and Methodology, 2012–2017,” May 29, 2013. Available at http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf.

² Constance L. Hays, “What Wal-Mart Knows About Customers’ Habits,” *The New York Times*, November 14, 2004.

³ G. King, “Ensuring the Data-Rich Future of the Social Sciences,” *Science* 331, no. 6018 (2011): 719–721, doi: 10.1126/science.1197872.

⁴ Robert M. Groves, “Three eras of survey research,” *Public Opinion Quarterly* 75, no. 5 (2011): 861–871, doi: 10.1093/poq/nfr057.

⁵ For CUSP, the main open government data portals are <http://www.data.gov/> for federal data, <https://data.ny.gov/> for New York State data, and <https://nycopendata.socrata.com/> for New York City data. Data.gov tracks countries with national open data sites that provide access to machine-readable data (<http://www.data.gov/opendatasites>). This is not solely a phenomenon of Western developed countries. Africa has at least two open data efforts: Open Data for Africa (<http://opendataforafrica.org/>) supported by the African Development Bank Group and Africa Open Data (<http://africaopendata.org/>) developed with support from the non-governmental sector.

⁶ Mick P. Couper, Eleanor Singer, Frederick G. Conrad, and Robert M. Groves, “Experimental Studies of Disclosure Risk, Disclosure Harm, Topic Sensitivity, and Survey Participation,” *Journal of Official Statistics* 26, no. 2 (2010): 287–300.

⁷ Datakind (<http://www.datakind.org/>) and Code for America (<http://codeforamerica.org/>) are examples of non-governmental organizations that seek to engage data scientists in projects for the public and non-profit sector that will lead to better decision making and greater social impact.

⁸ Executive Order 13642, *Making Open and Machine Readable the New Default for Government Information*, 78 FR 28111, May 14, 2013. Office of Management and Budget Memorandum M-13-13, *Open Data Policy – Managing Information as an Asset*, May 9, 2013. Available at <http://www.whitehouse.gov/sites/default/files/omb/memoranda/2013/m-13-13.pdf>.

⁹ David Kesmodel, “Monsanto to Buy Climate Corp. for \$930 Million,” *Wall Street Journal*, October 2, 2013; and Quentin Hardy, “Big Data in the Dirt (and the Cloud),” *The New York Times*, October 11, 2011.

¹⁰ Mayur Patel, Jon Sotsky, Sean Gourley, and Daniel Houghton, “The Emergence of Civic Tech: Investments in a Growing Field,” The John S. and James L. Knight Foundation, December 4, 2013. Available at http://www.knightfoundation.org/media/uploads/publication_pdfs/knight-civic-tech.pdf (accessed December 27, 2013).

¹¹ James Manyika, Michael Chui, Diana Farrell, Steve Van Kuiken, Peter Groves, and Elizabeth Almasi Doshi, *Open Data: Unlocking Innovation and Performance with Liquid Information* (McKinsey Global Institute, October 2013).

¹² *The World Factbook 2013-14* (Washington, DC: Central Intelligence Agency, 2013). Calculation based upon 2012 gross world product of US\$84.97 trillion (purchasing power parity), inflated by 2.9%.

¹³ M. Batty, K. W. Axhausen, F. Giannotti, A. Pozdnoukhov, A. Bazzani, M. Wachowicz, G. Ouzounis, and Y. Portugali, “Smart Cities of the Future,” *European Physical Journal – Special Topics* 214 (2012): 481–518, doi:10.1140/epjst/e2012-01703-3.

¹⁴ Luís M. A. Bettencourt, José Lobo, Dirk Helbing, Christian Kühnert, and Geoffrey B. West, “Growth, Innovation, Scaling, and the Pace of Life in Cities,” *PNAS* 104, no. 17 (2007): 7301–7306, doi:10.1073/pnas.0610172104.

¹⁵ L. Bettencourt, J. Lobo, and D. Strumsky, “Invention in the City: Increasing Returns to Patenting as a Scaling Function of Metropolitan Size,” *Research Policy* 36 (2007): 107–120, doi:10.1016/j.respol.2006.09.026.

¹⁶ Office of the Mayor, “Mayor Bloomberg and Speaker Quinn Announce New Approach to Target Most Dangerous Illegally Converted Apartments” (press release), PR-193-11, The City of New York, June 7, 2011. Available at <http://www.nyc.gov/cgi-bin/misc/pfprinter.cgi?action=print&sitename=OM&p=1390075778000>.

¹⁷ Marta C. Gonzalez, Cesar A. Hidalgo, and Albert-Laszlo Barabasi, “Understanding Individual Human Mobility Patterns,” *Nature* 453, no. 5 (2008): 779–782, doi:10.1038/nature06958.

¹⁸ P. Wang, T. Hunter, A. M. Bayen, K. Schechtner, and M. C. Gonzalez, “Understanding Road Usage Patterns in Urban Areas,” *Scientific Reports* 2 (2012): article 1001, doi:10.1038/srep01001.

¹⁹ T. Giannetsos, T. Dimitriou, and N. R. Prasad, "People-centric Sensing in Assistive Healthcare: Privacy Challenges and Directions," *Security and Communication Networks* 4 (2011): 1295–1307, doi:10.1002/sec.313.

²⁰ Clément Farabet, Camille Couprie, Laurent Najman, and Yann LeCun, "Learning Hierarchical Features for Scene Labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, no. 8 (2013): 1915–1929, doi:10.1109/TPAMI.2012.231.

²¹ Federal Trade Commission, "Marketer of Internet-Connected Home Security Video Cameras Settles FTC Charges It Failed to Protect Consumers' Privacy" (press release), September 4, 2013. Available at <http://www.ftc.gov/opa/2013/09/trendnet.shtm> (accessed December 30, 2013).

²² Richard Briffault, "A Government for Our Time? Business Improvement Districts and Urban Governance," *Columbia Law Review* 99, no. 2 (1999): 365–477.

²³ Xiaojun Yang, *Urban Remote Sensing: Monitoring, Synthesis and Modeling in the Urban Environment* (Hoboken, NJ: Wiley-Blackwell, 2011), doi:10.1002/9780470979563.

²⁴ S. Buckingham Shum et al., "Towards a Global Participatory Platform," *European Physical Journal – Special Topics* 214 (2012): 109–152, doi:10.1140/epjst/e2012-01690-3.

²⁵ Erica H. Dunn et al., "Enhancing the Scientific Value of the Christmas Bird Count," *The Auk* 122 (2005): 338–346.

²⁶ Nicolas Maisonneuve, Matthias Stevens, and Bartek Ochab, "Participatory Noise Pollution Monitoring using Mobile Phones," *Information Polity* 15 (2010): 51–71, doi:10.3233/IP-2010-0200.

²⁷ Nathalie Butt, Eleanor Slade, Jill Thompson, Yadvinder Malhi, and Terhi Riutta, "Quantifying the Sampling Error in Tree Census Measurements by Volunteers and Its Effect on Carbon Stock Estimates," *Ecological Applications* 23, no. 4 (2013): 936–943, doi:10.1890/11-2059.1.

²⁸ Salil S. Kanhere, "Participatory Sensing: Crowdsourcing Data from Mobile Smartphones in Urban Spaces," in *Distributed Computing and Internet Technology*, 19–26 (Berlin: Springer, 2013).

²⁹ C. Capps and T. Wright, "Toward a Vision: Official Statistics and Big Data," *Amstat News*, August 1, 2013. Available at

<http://magazine.amstat.org/blog/2013/08/01/official-statistics/> (accessed September 19, 2013).

³⁰ National Research Council, *Frontiers in Massive Data Analysis* (Washington, DC: The National Academies Press, 2013). Available at http://www.nap.edu/catalog.php?record_id=18374.

³¹ Capps and T. Wright, "Toward a Vision: Official Statistics and Big Data."

³² Ibid.

³³ Latanya Sweeney, "K-anonymity: A Model for Protecting Privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, no. 5 (2002): 557–570, doi:10.1142/S0218488502001648.

³⁴ Xuyun Zhang, Chang Liu, Surya Nepal, Suraj Pandey, and Jinjun Chen, "A Privacy Leakage Upper-Bound Constraint Based Approach for Cost-Effective Privacy Preserving of Intermediate Datasets in Cloud," *IEEE Transactions on Parallel and Distributed Systems* 24, no. 6 (2013): 1192–1202, doi:10.1142/S0218488502001648.

³⁵ N. Ferreira, J. Poco, H. T. Vo, J. Freire, and C. T. Silva, "Visual Exploration of Big Spatio-Temporal Urban Data: A Study of New York City Taxi Trips," *IEEE Transactions on Visualization and Computer Graphics* 19, no. 12 (2013): 2149–2158, doi:10.1109/TVCG.2013.226.

³⁶ Aritra Dasgupta and Robert Kosara, "Privacy-Preserving Data Visualization Using Parallel Coordinates," in *Proc. Visualization and Data Analysis (VDA)*, 78680O-1–78680O-12 (International Society for Optics and Photonics, 2011), doi:10.1117/12.872635.

³⁷ Open Knowledge Foundation, "Open Definition." Available at <http://opendefinition.org/od/> (accessed on January 5, 2014).

³⁸ Michael Chui, Diana Farrell, and Steve Van Ku, "Generating Economic Value through Open Data," in *Beyond Transparency: Open Data and the Future of Civic Innovation*, ed. Brett Goldstein and Lauren Dyson (San Francisco, CA: Code for America Press, 2013), 169.

³⁹ Fida Kamal Dankar, Khaled El Emam, Angelica Neisa, and Tyson Roffey, "Estimating the Re-identification Risk of Clinical Data Sets," *BMC Medical Informatics & Decision Making* 12, no. 1 (2012): 66–80, doi:10.1186/1472-6947-12-66.

⁴⁰ IPCC 2007, *Climate Change 2007: Synthesis Report. Contribution of Working Groups I, II and III to the Fourth Assessment Report of the Intergovernmental Panel on Climate*

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Change, ed. Core Writing Team, R. K. Pachauri, and A. Reisinger (Geneva: IPCC, 2007), Appendix II, p. 83.

⁴¹ S. S. Dawes and N. Helbig, “Information Strategies for Open Government: Challenges and Prospects for Deriving Public Value from Government Transparency,” in *Electronic Government*, ed. M. A. Wimmer et al., Lecture Notes in Computer Science 6228 (Berlin: Springer, 2010), 50–60, doi:10.1007/978-3-642-14799-9_5.

⁴² National Research Council, 2.

⁴³ *World Population Prospects: The 2010 Revision, Volume I: Comprehensive Tables*, ST/ESA/SER.A/313 (United Nations, Department of Economics and Social Affairs, Population Division, 2011).

⁴⁴ D. R. Jones, “Protecting the Treasure: An Assessment of State Court Rules and Policies for Access to Online Civil Court Records,” *Drake Law Review* 61 (2013): 375.

⁴⁵ Theodore M. Porter, *Trust in Numbers: The Pursuit of Objectivity in Science and Public Life* (Princeton, NJ: Princeton University Press, 1996).

⁴⁶ Joe Flood, *The Fires: How a Computer Formula, Big Ideas, and The Best of Intentions Burned Down New York City—and Determined the Future of Cities* (New York: Riverhead Books, 2010).

Chapter 7

Data for the Public Good: Challenges and Barriers

Robert M. Goerge

Introduction

Comprehensive, high-quality, multidimensional data has the potential to improve the services cities provide, as it does with the best private service-providing businesses. City officials, politicians, and stakeholders require data to (1) inform decisions that demonstrate service effectiveness, (2) determine which services should be targeted in a geographic area, and (3) utilize limited resources to best serve residents and businesses.

Administrative data is now ubiquitous in government agencies concerned with health, education, social services, criminal justice, and employment. Local government has primarily used this data to count cases and support budget making within the programs for which the data is collected. Data linked across programs, where individuals and families can be tracked with multiple data sources either cross-sectionally or longitudinally, is rare. Both data scientists and the public sector currently have an excellent opportunity to use the big data of government to improve the quality and quantity of analyses to improve service delivery. This chapter describes an effort in one place to use the administrative data collected in the public sector to have an impact by informing city leadership.

Privacy rules and regulations and bureaucratic silos often prevent city officials from obtaining and using data to address some of their most intractable problems. This chapter also addresses the barriers to sharing and acquiring information. How can municipalities unlock the potential value of data and harness the analytic resources that are generally in short supply, access data not in their control that would enhance their data-driven capacities, and comprehensively address the range of education, health, employment, and crime-related issues for which they are responsible? Officials must ultimately have better information to maximize their utilization of limited resources to improve the well-being of their residents and the effectiveness of the organizations (health care, schools, social service agencies and police departments) that serve them. This chapter will primarily focus on a successful Chicago effort, and also refer to others around the country that offer strategies for cities to make data-driven decisions more common.

All trends and forecasts suggest that cities globally will become even more important socially, economically, and politically.¹ Income inequality is increasing and occurs to the greatest extent and closest proximity in cities.² While poverty in the United States is increasing more quickly in suburban areas, the vast majority of the poor still live in cities.

Cities will need to provide a range of health and social services while those with higher incomes will pay the cost of those services through taxes. As cities grow and, as the resources available to them become scarcer, they need to increase both service efficiency and effectiveness.³ The bankruptcy of Detroit and the dire fiscal situation of many other cities and school districts across the nation highlight the extremes of these trends.⁴

Cities must find ways to educate children, train unemployed adults, and keep residents safe and healthy, while at the same time supporting large and small businesses, the non-profit sector, and faith-based service organizations. The public sector needs to learn ‘what works’ locally, in specific neighborhoods, with specific populations, and what services, programs, or interventions need to be modified or discontinued. In order to make those determinations, high-quality data (which is often confidential and sensitive) combined from multiple sources is key.

Current Open Data Initiatives and Need for Confidential Data for Decision Making

In order to address the needs of cities, their residents, and non-profit and business sectors, confidential and sensitive data must be used and shared to create databases that can fuel better service and analysis. Data and tools needed to drive the development of 21st-century housing and commercial developments are increasingly available through open data portals in cities around the world (Koonin and Holland, Chapter 6 in this volume).⁵ These current open data initiatives utilize information that is generally *not* confidential.

The fields of engineering and urban design are already using data-driven models to revamp and create new neighborhoods. Barcelona, San Francisco, Chicago, and other cities are demonstrating how the use of new tools and data enables the design of neighborhoods that are environmentally friendly and sustainable.⁶ In Chicago, “LakeSim will connect existing urban design tools with scientific computer models to create detailed simulations relevant to large-scale development. Instead of planning separately for different pieces of the infrastructure, the framework will allow developers to simulate how various designs influence the environment, transportation, and business spheres in Chicago Lakeside under a range of possible scenarios, over hundreds of acres and decades of time.”⁷ In most cases, however, the data that is used to fuel these efforts is not confidential, because it is overwhelmingly about places and the information that is needed about the population is often in aggregate form.

The data that addresses what human services city residents would benefit from is difficult to acquire. To address and target specific social issues, personally identifiable information (PII) is needed. Given national and international concern about privacy, the key is to access PII without disclosing the identities of individuals. The challenge for cities – government and its private sector partners – is how to use data (primarily

administrative, but also social media and any other private source such as utility data) on characteristics and needs of individuals and families to provide better services and to support analysis that leads to better evaluation, planning, and monitoring of city functions and program outcomes. Furthermore, cities must make complex decisions about (1) what data to develop and access from counties, states, the federal government, and private sources; (2) how to develop the capacity to use data; (3) how to present data and be transparent; and (4) how best to keep data secure so that individuals and organizations are protected.

The range of administrative data collected by cities is considerable and includes but is not limited to crime activity, school outcomes, public health care and surveillance, early care and education, workforce development, after-school activities, tax payments, and receipt of human services. Linking these data with place-based assets, such as businesses, homes, community services, transportation, and emergency services, provides the comprehensive data infrastructure that can inform and improve the city's fulfillment of its responsibilities to its residents.

Cities Must Capitalize on Their Investment in Collecting Data

Public sector staff, whether teachers, police officers, public health workers, or staff of contracted providers, spend an enormous amount of time collecting data, described above, that is entered into computerized systems. A return on this investment of time and money can be realized if what is often rounding error in a city budget is spent to create data warehouses that can be the basis for the analysis of the services provided.

Two major urban areas have taken important steps that are impossible to roll back because of the direct benefits that have been seen in serving individuals and families. These efforts have gone hand-in-hand with creating data warehouses that support decision making by city leaders. Unfortunately, few cities have the type of systems described below.

The Allegheny County Department of Human Services in Pennsylvania (Pittsburgh) operates a data warehouse that includes child welfare, behavioral health, public school, welfare program, corrections, housing, and aging services.⁸ Front-line workers are able to access information on individuals or families from other agencies to inform their information gathering or service provision. Data is also used to conduct system-wide analyses to support research, strategic planning, needs assessment, and program evaluation. There are tight controls over who sees what data based on what front-line workers need to know.

New York City created Worker Connect which links data on a single individual family or household across child protective services, juvenile justice, aging, mental health, criminal justice, health and hospitals, and homeless services. Worker Connect gives front-line agency staff access to real-time data about individuals that allows them to

enroll and process cases faster, provide quicker referrals, and spend more time on casework.⁹ This system also facilitates studies such as the NYC/Cleveland/Chicago study of foster children entering the juvenile justice system described in the next section.

Legal, Bureaucratic, Jurisdictional, and Technical Barriers to Creating Better Data

Just as the data systems in New York City and Allegheny County can have benefits for an individual's well-being, analysis of the entirety of the data can have a positive impact on the well-being of subpopulations of vulnerable individuals and families. Employing data scientists and subject matter experts, whether inside or outside of government, with the right analytic tools and data, can have a significant impact on the decision making around schools, health care, employment, and public safety. Confidential data on individual/family service experiences, employment, and criminal records is the major resource needed by these experts to conduct the research and analysis that builds evidence. Without PII about people (names, birthdates, home addresses) and places (addresses), data linkage at the person or place level is impossible, which leaves decision makers with single-program datasets that are significantly less useful. The following are the barriers to combining data and sharing it with those who can generate the evidence needed to address cities' toughest problems.

Legal Barriers There are a variety and potentially increasing number of federal, state, and municipal laws designed to protect individuals from harm that data disclosure might cause (Strandburg, Chapter 1, and Ohm, Chapter 4, in this volume).¹⁰ While these laws usually contain provisions allowing conditional data sharing, public attorneys are often reluctant to approve the use of data that might appear to violate the law or incur additional risk for their clients (mayors or agency heads). Educating public sector attorneys on what the actual law is concerning the conditions under which confidential data can be shared is seldom done, except perhaps by other attorneys whose clients need to access data.

Although little research has been done regarding harm that disclosure of protected data actually causes, there is strong public sentiment about protecting sensitive data on individuals – unless someone has been arrested for a crime.¹¹ There are instances in which individuals have been misidentified because of incorrect data or unauthorized use of data. This has occurred in homeland security and law enforcement at all levels of government. Clearly, cities must keep the data they collect secure, while at the same time making the data available to selected users for constructive purposes.

Bureaucratic Barriers Sometimes government leaders and staff are reluctant to share their data with other public sector agencies. This often results in ‘federated’ or virtual

databases, where each dataset remains with the responsible agency and data is only combined in highly regulated ways where each party can control the outcome. In this case, data are not actually integrated into one database. This added layer of bureaucracy, unless managed extremely well, adds significant additional burden to any effort that requires data from multiple agencies, since it requires multiple permissions and actual bringing the data together in one place for analysis purposes. This model is truly an example of how data still exists in silos that do not facilitate its use.

In many instances, government leaders feel safer giving their data to an external party (a consultant or an academic organization) than to a fellow agency leader within the same jurisdiction. Lack of data sharing between two public sector agencies is often a symptom of lack of trust between the agencies. In this situation, the external party can link data from multiple agencies for the benefit of these agencies, without any one agency having another's data.

Political Barriers Elected leaders find it difficult to fund research when services are being cut. Perhaps over time, the benefits of data-driven decision making in improving services will become evident to the public. Then, our elected officials will be able to allocate tax dollars to research and analysis without fear of special interest group backlash. Fortunately, private philanthropy and monies from the federal government can often be used to support these important efforts.

Jurisdictional Barriers In order to have simple indicator data as well as comprehensive microdata on individuals to guide programmatic and fiscal decision making, cities often need to enter into agreements with state and/or county agencies to access what they require to create comprehensive data. City school districts, criminal courts, jails, prisons, and public housing are often not under a mayor's jurisdiction. Also, data from welfare programs (TANF, SNAP, Medicaid, LIHEAP) are not typically available to cities unless cities are also counties in county-administered states (e.g. NYC, San Francisco). So, to obtain welfare data on residents, cities must employ mechanisms and state agency agreements (sometimes established and sometimes not) to access data. Many agreements are cumbersome and need to be renewed on a yearly basis, which creates an opportunity to negotiate the terms – but also takes time as the paperwork moves slowly through the bureaucracy. As funding becomes increasingly scarce, government agencies are requiring payment for data – even between two government agencies! The need to support data efforts within agencies has led to fee-for-service in the data-sharing arena.

Technical Barriers The technical barriers to integrating data are declining over time as tools are being developed in both the proprietary and open source space at a rapid rate.

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Clearly there is fast turnover in the computing infrastructure, both hardware and software, applications, dashboards, and other tools used to process, analyze, and visualize data. The challenge will be for cities to decide where and how to deploy their data resources and personnel so that data is continually available, but is also continually upgraded. Furthermore, as computational power becomes less expensive and more open software is available, government has the potential to maintain state-of-the-art data infrastructures.

Given the salary discrepancies between the public and private sectors, it is often difficult to prevent the turnover of computing staff. Training staff who have both data science and policy acumen and retaining them is also important if cities are to take advantage of the data they collect. Universities must prepare the workforce of the future so that there is sufficient technical expertise available.¹² In addition, strong relationships with external organizations, such as university research centers, can provide institutional memory so that new officials and staff can maintain the practices of data-driven decision making. The next section provides an example of such a research center.

Chicago: A Case Study

This chapter continues with an example that shows how a set of stakeholding organizations took steps to build data resources that met the needs of policymakers and helped inform not only local decision makers, but key actors outside of the city, in state capitals and Washington, about what is needed to address a few of the key social problems nationwide.

Informational Needs of Policymakers

The Integrated Database on Child and Family Programs (IDB) in Illinois is one approach to addressing the data needs described above. The database is the oldest of its kind in the country, and has been continually maintained by Chapin Hall at the University of Chicago by the author and his colleagues since the mid-1980s.¹³ The primary purpose of the IDB project is to inform policymakers of the circumstances of children and youth in Illinois, to provide evaluative data, and to conduct research that leads to improved policy and programs for vulnerable children and youth. In order to report on all individuals, the purpose has always been to collect data on the entire populations of individuals in public sector programs, as opposed to tracking samples. (This has resulted in the IDB holding data that the state and city no longer have available to them.)

The IDB was begun three decades ago by a group of state, foundation, and academic leaders who believed strongly that administrative data has important value in social research. In particular, the goal is to improve the decisions that public sector employees – from front-line workers through agency leadership – make about vulnerable children and adults and ultimately improve the lives of Illinois children and families.

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Chapin Hall, in partnership with state, county, and city agencies, has brought together and linked a wide range of datasets at the individual, family, and case levels to achieve what was described in the previous paragraph through research and analysis. These include data from over a dozen state and city agencies and nearly all social programs, with records for over 10 million individuals. Program data includes maternal and child health data, Women, Infants and Children Nutritional Program, Supplemental Nutritional Assistance Program and Food Stamps, Temporary Assistance to Needy Families and Aid to Families with Dependent Children, Medicaid, abuse and neglect reports, child welfare services, juvenile justice, adult incarcerations, arrests, employment and earnings (Unemployment Insurance wage data), mental health services, alcohol and substance abuse treatment, child care subsidies, and special education, and data on early childhood programs (e.g. Head Start), K–12 student information, and postsecondary education in Chicago. These data have been linked at the individual level in most cases since at least 1990. Data is multigenerational, which means that children, parents, and often grandparents in the data are linked. Most of these data have addresses available and are thereby able to be geocoded and spatially analyzed.

Over time, the IDB has expanded to include adults and families in public sector programs. The city of Chicago, as both the economic engine and home of families with the most severe problems in Illinois, has been a focus of the research undertaken by most users of the IDB. For example, the K–12 data includes well over 80% of the school-age children in Chicago since 1991.

Maintenance Challenge

All the barriers described at the beginning of this chapter were experienced by Chapin Hall in building the IDB. (This author has literally been told, “There’s a new sheriff in town and nobody is getting this data.”) Nevertheless, the IDB continues to have the key ingredients for success including (1) strong support from executive leadership across government agencies, (2) a long and growing track record of success, (3) no data breaches, (4) lawyers and laws usually on the side of data sharing, and (5) independent funding for most of the work.

For the data to have impact, a long-term relationship between the public officials and the researchers that is mutually beneficial is paramount. The researchers benefit by understanding the problems better and being able to communicate their findings to officials in a manner that can be used in practical ways. Public officials also keep researchers ‘honest’ in that policy or programmatic recommendations that sound good in journal articles may be simply not practical. This reality check increases the integrity of the academic mission of building knowledge. Public officials benefit from having expertise that they cannot always buy and, through researchers, who are often around longer than the officials, can access the institutional memory around the data activities of

a city. Because there was strong support from officials above the “new sheriff,” data continue to flow without pause.

The next section addresses how Chapin Hall has used and uses the data. We continue to overcome and manage barriers to maintain the effort and conduct research projects and analyses for government agencies. This is necessary because the laws, rules, and context for government activity change constantly and affect the construction of the IDB. Appendix B contains additional examples of efforts to promote data-driven decision making in cities that face many of the same barriers but often address them in different ways.

Uses of the Integrated Data Base

Researchers at Chapin Hall, the University of Chicago, and many other institutions have used the IDB for evaluations, dissertations, and other studies leading to peer-reviewed journal articles and formal reports to policymakers. Research ranges from purely descriptive to using data in randomized control trials. In addition, the IDB has been used in multistate and multicity studies. Parts of the IDB were linked to the U.S. Census Bureau’s American Community Survey data, a representative sample of the entire population, in four states so that families who did and did not participate in particular government programs could be studied.¹⁴ The following sections describe how the IDB has been used in other studies to inform policy and practice.

Families in Multiple Systems

While states and cities collect data for ongoing program management, they often lack the capacity to transform these data into information that aids policy decisions and program development. For example, in 2008, Illinois state officials hypothesized that a minority of families accounted for the majority of service costs, but they could not identify the characteristics of these families and quantify these costs *across agencies and programs*. Illinois has little information on where to target their interventions to both alleviate problems now and prevent problems in the future because the state does not have the capacity to reconstruct family service histories.

Chapin Hall was asked by the governor’s office to conduct a study that would clearly identify families in multiple systems and multiple programs with the aim of understanding the costs associated with all program utilization. The programs that we analyzed were those with the highest per diem costs, including foster care, adult and juvenile incarceration, mental health services, and substance abuse treatment. Even with the support of the governor, it was necessary to convince each of five state agency directors that the value of the project justified the effort that their staff (primarily lawyers) would have to expend in making it happen. The necessity also arose because the governor’s office had no research funds and required the agencies to pay for an equal part of the research. Permission and funding was secured from five state agencies, and one

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

additional agency declined. While the data from the declining agency (birth certificates) would have been beneficial, it would have added only demographic data and did not affect the core research questions.

The results of the analysis allow the state to target resources geographically and individually.¹⁵ The study revealed that 23% of extended families being served in multiple systems (health, mental health, criminal and juvenile justice, and child welfare) account for 64% of the service intervention resources *and* utilize 86% of the funding resources. Spatial analysis showed Chicago has the densest geographic pockets of these families, but there are also areas in smaller urban areas around the state that have high rates. For example, there are over 10 census tracts where more than two-thirds of all children live in families participating in multiple systems. These data are crucial for targeting service delivery and focusing on the small number of multisystem families who are utilizing the majority of resources.

Chicago Public Housing Transformation

From 2004 to 2007, Chapin Hall and other researchers in Chicago met with all child- and family-serving agencies in Chicago to address the impact of public housing transformation on the children and families affected by it.¹⁶ The transformation involved razing high-rise public housing structures and providing support for families to move to other parts of the city. These meetings occurred roughly every six weeks, and as questions arose, Chapin Hall staff analyzed demographic and geographic data in the IDB to best describe service utilization. In particular, the IDB analyses resulted directly in multiple instances of two city agencies partnering to address identified problems. The data showed that certain schools and specific communities would need additional resources to serve the particular educational and emotional needs of children newly displaced to their community.

A key factor in bringing all of the agencies to the table and gaining their participation was a pledge that all the information Chapin Hall researchers produced would remain confidential for planning purposes – and that nothing would be published without the written approval of the agencies providing data for a particular analysis. This exemplifies the concern that public officials have about opening up research that may prompt additional demands on them. Appendix A summarizes a set of analyses and actions that resulted from the discussions among agency leadership and researchers during this project.

Three-City Study of Foster Children Entering the Juvenile Justice System

Children who receive child welfare services are at risk for later delinquency and involvement with the juvenile justice system. Individuals who become involved in both systems appear to have increasingly complex needs, but may be less likely to receive comprehensive, coordinated care because of agency boundaries. Understanding the

This is a preliminary version of the book Privacy, Big Data, and the Public Good: Frameworks for Engagement, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

pathway from child maltreatment to participation in the juvenile justice system can help cities address one of their most vexing problems –how youth become perpetrators of crime and violent behavior.

The Three-City Study of Foster Care Children Entering the Juvenile Justice System Project consisted of separate, but highly comparable, analyses of administrative data from three large urban localities: Cook County in Illinois, Cuyahoga County in Ohio, and New York City in New York. The analyses pursued three goals: (1) determine how many children who experience out-of-home foster care placements become involved in the juvenile justice system; (2) understand individual characteristics (sex, race/ethnicity) and child welfare involvement (placement type, number of spells, age of first child welfare placement) that may distinguish children in foster care who are later involved in the juvenile justice system from those who are not; and (3) compare patterns of results across the three locations to suggest common and distinct trends. The main challenge in this project was a technical one: to make the data comparable across the three sites, so that valid comparisons could be made. The lack of comparability or knowledge of comparability is a major impediment to multisite studies.

Population and Poverty Estimates and the Inadequacy of Census Data for Cities

The IDB has also been used to create information to substitute for data that was available for many decades from the Census Bureau. For decades, cities have relied on decennial census data for good measures (albeit 10 years apart) of their population and housing characteristics at the block level. In the last decade, the Census Bureau began collecting long-form data on a continual rolling-sample basis across the country. The current American Community Survey (ACS) only provides 5-year averages at the census tract level for the data that was received by cities at the block or block-group level for each decennial census. The administrative data in the IDB and place-level ACS data made it possible to build statistical models to calculate census tract population and poverty estimates for 0–18-year-olds in Chicago. City officials require this data to make informed service resource allocation decisions, which affect school openings and closings, child care slot distribution, senior citizen programs, and many other services based on population characteristics and density.

The federal government and states are well served by the ACS and other Census Bureau data, because they make decisions using state- and county-level data, which are available annually. Much has been written and much said about this issue and some, including commercial firms, have taken to marketing their own estimates of population using available data in combination with administrative data that states and private sector firms maintain.¹⁷ Unfortunately, it is generally true that few data sources are consistent

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

over time as data-collecting agencies rarely take into consideration all the uses of their data by others.

Supply and Demand for Early Care and Education

Leaders in Chicago agencies who are responsible for managing early childhood programs and combining funding streams to provide high-quality care assert that the availability of one type of program in a community is dependent on the number of other programs in that community. This is the result not only of the number of eligible children, but also the complexities of federal and state regulations and the availability of funding for each program. Chapin Hall researchers combined data from the population and poverty estimates described above with data from the IDB, including child care licensing data, and Head Start, pre-K, and child care utilization, to develop an ongoing analysis of the supply and demand for early care and education programs in Chicago at the census tract level.¹⁸ Since the attempt is made to keep this data up to date, the challenge is to get data from the data providers in a timely fashion. As resources become increasingly scarce, even if the data is collected, it is often a burden for agencies to extract it for analytic purposes or to share it with other organizations. Computer professionals are seen as non-essential personnel in some public sector agencies. Therefore, there is often significant turnover in who manages the data.

Using Census Survey Data Linked with Administrative Data

In many ways, linking population data with administrative data provides an ideal data resource for cities seeking to understand who does and does not use their services and what the outcomes are for each group. By integrating individual-level U.S. Census Bureau survey data with individual-level social program administrative data from Illinois, Maryland, Texas, and Minnesota, Chapin Hall and colleagues at three other universities and the Census Bureau developed a comprehensive model of employment support program eligibility and take-up.¹⁹ The resulting dataset was a representative sample of all families in these states combined with administrative data on child care subsidies, income maintenance, employment, and nutritional programs at the individual and family levels. Researchers examined individual, family, household, and neighborhood characteristics that affected program take-up; the conditions under which low-income families utilize these programs; and the effect of participation in these programs on employment.

Development of Necessary Data Warehouses

Cities need to move beyond their role as data collectors and become data integrators, stewards, and users. Their decades as mere data collectors have shown no significant return. The role of data steward means delivering data to individuals, either within or

outside city government, who can use it to provide policymakers with the information needed to best serve city residents.

Cities have proven they can do this when there is a compelling need. The question is whether it can be sustained. Chicago city staff combined numerous sources of data by location in real time to address the security challenge during the NATO summit in 2012.²⁰ Much of the data was confidential and never shared outside those agencies involved in the NATO summit security. Most of the data was place based, but some included confidential identifying information. Data on crime incidents, 911 and 311 reports, building conditions, municipal vehicle locations, transportation and traffic, Twitter streams, and other place-based data were also included in order to have information ready in case of a crisis. Similarly, other cities are making tremendous efforts in the public safety arena, with efforts like CompStat showing promise in addressing particular types of crime in specific locations.

It is possible that the new emphasis on data-driven decision making and evidence-based practice (including predictive analytics in policing and experimental efforts in education) will allow cities to build the data infrastructure and skill sets needed to use and sustain it with less effort than was needed when some current efforts first started decades ago. Executive orders and city ordinances may be necessary, but in most cases, creating better data is already allowed by the law. The following are the core ingredients in empowering cities with the data they need.

Sustainability

Sustainability is a major challenge when elected officials change in cities. New leadership needs to be educated about the importance of existing data efforts, and new relationships based on mutual trust and need must be formed. In the worst case, efforts begun during one administration are discontinued in the next. Transition to a new administration does not necessarily mean an interruption in the data flow, however. When strong relationships with middle-level staff and legal agreements are in place, data sharing and utilization can continue without pause. Often, and ideally, data delivery, with the proper legal permissions, is a hands-off process that occurs whether or not anybody is watching.

Strong Leadership

Most mayors and politicians are risk averse, and will not risk a potential class-action lawsuit or bad publicity around privacy. However, mayors and agency directors have the power to promote data sharing between and among city agencies. Mayor Bloomberg in New York and Mayor Emmanuel in Chicago both exemplify the positive impact strong leaders can have on data-driven city management and decision making.

“Where there's a will, there's a way.” If an organization that has data wants to provide it to another organization for the purpose of improving services, there is usually a legal

way to share the data. Although there are many potential legal and bureaucratic barriers to data sharing, if strong leaders (influential over the use of data resources) want data sharing to happen, it will most likely occur. A recent GAO report provides some examples of barriers that are perceived rather than actual.²¹ For the most part, the legal and regulatory framework does not ultimately prevent data sharing. Sufficient discretion is given to public sector organizations that the attorneys who write the contracts can find a way to accommodate data use and sharing either inside or outside of government.

Building Strong Relationships

Trust between organizations can often take years to develop, through personal connections and organizational relationships based on civic standing, reputation, and experience. The process often involves many meetings, discussions, and negotiations. It may also involve refuting the fear that sharing data with external organizations may lead to data being indiscriminately disseminated.

Some collaborations across the country have had great success in linking confidential data because they have built enduring relationships. For example, participants in the University of Pennsylvania's Actionable Intelligence for Social Policy represent a group of cities and states that have combined datasets on individuals across multiple public sector programs.²² In all cases, these efforts have depended on the sharing of personally identifiable information attached to service records to build comprehensive histories of problems, disabilities, assets, and service receipt of individuals and families. These state and local efforts have built incrementally over time both relationships and data resources to address questions important to policymakers while contributing to the knowledge base in multiple fields of social science.

Notes

¹ Bruce Bruce and Jennifer Bradley, *The Metropolitan Revolution: How Cities and Metros Are Fixing Our Broken Politics and Fragile Economy* (Washington, DC: Brookings Institution, 2013).

² Lisa Gennetian, Jens Ludwig, Thomas McDade, and Lisa Sanbonmatsu, "Why Concentrated Poverty Matters," *Pathways*, Spring 2013, 10–13, http://www.stanford.edu/group/scspi/_media/pdf/pathways/spring_2013/Pathways_Spring_2013_Gennetian_Ludwig_McDade_Sanbonmatsu.pdf.

³ "U.S. Cities Growing Faster than Suburbs," *Real Time Economics* (Wall Street Journal blog), May 23, 2013, <http://blogs.wsj.com/economics/2013/05/23/u-s-cities-growing-faster-than-suburbs/>.

⁴ For insight into school district bankruptcies, see Kristi L. Bowman, “Before School Districts Go Broke: A Proposal for Federal Reform,” *University of Cincinnati Law Review* 79 (2011): 895.

⁵ See <http://www.data.gov/opendatasites> for a list of cities all over the world that have open data sites.

⁶ See <http://sustainablecitiesrcn.files.wordpress.com/2013/11/urban-ecodesign-in-the-city-of-barcelona-quaderns-temes-de-disseny-elisava-2012.pdf>;
<http://urbanccd.org/articles/computation-enabled-design-chicago-lakeside-development>;
<http://www.sf-planning.org/index.aspx?page=3051>.

⁷ See <http://worldlandscapearchitect.com/lakesim-prototype-connects-urban-planning-with-scientific-models-for-large-scale-development/#.UqUQdGRDuGI>.

⁸ See

<http://www.allegenycounty.us/uploadedFiles/DHS/about/DataWarehouseHistory.pdf>;
“Human Services: Sustained and Coordinated Efforts Could Facilitate Data Sharing while Protecting Privacy,” GAO-13-106 (Washington, DC: U.S. Government Accountability Office, February 8, 2013).

⁹ Isidore Sobkowski and Roy S. Freedman, “The Evolution of Worker Connect: A Case Study of a System of Systems,” *Journal of Technology in Human Services* 31, no. 2 (2013): 129–155, doi:10.1080/15228835.2013.772010.

¹⁰ Somin Sen Gupta, “No U.S. Action, So States Move on Privacy Law,” *The New York Times*, October 31, 2013, www.nytimes.com/2013/10/31/technology/no-us-action-so-states-move-on-privacy-law.html?_r=0; “Human Services: Sustained and Coordinated Efforts.”

¹¹ Even the identities of perpetrators of abuse or neglect against their own children are not disclosed, except in heinous cases or murder.

¹² E.g. see the University of Chicago’s Master of Science in Computational Analysis and Public Policy, <http://capp.sites.uchicago.edu/>.

¹³ Chapin Hall at the University of Chicago has, since its inception in 1985 as a research and policy center, focused on a mission of improving the well-being of children and youth, families, and their communities. This is done through policy research – by developing and testing new ideas, generating and analyzing information, and examining policies, programs, and practices across a wide range of service systems and organizations. Primary colleagues instrumental in the creation of the IDB include Lucy Bilaver, Bong Joo Lee, John Van Voorhis, Mairead Reidy, and Nila Barnes.

¹⁴ Bruce Meyer and R. Goerge, "Errors in Survey Reporting and Imputation and Their Effects on Estimates of Food Stamp Program Participation," Working Paper 11-14 (Washington, DC: U.S. Census Bureau, Center for Economic Studies, 2011).

¹⁵ R. Goerge, C. Smithgall, R. Seshadri, and P. Ballard, "Illinois Families and Their Use of Multiple Service Systems," Issue Brief (Chicago, IL: Chapin Hall at the University of Chicago, 2010).

¹⁶ Agencies that participated in this project were (*state agencies*) Illinois Department of Children and Family Services, Illinois Department of Corrections, Illinois Department of Employment Security, Illinois Department of Healthcare and Family Services, Illinois Department of Human Services; (*local agencies*) Chicago Department of Children and Youth Services, Chicago Department of Human Services, Chicago Department of Public Health, Chicago Housing Authority, Chicago Public Schools, Cook County Juvenile Court, Mayor's Office of Budget and Management; (*University of Chicago*) Chapin Hall Center for Children, Consortium on Chicago School Research, Office of Community and Government Affairs, School of Social Service Administration.

¹⁷ For a perspective on using the ACS, see <http://www.prb.org/Publications/Articles/2009/2010censustestimony.aspx>; Geolytics is an example of a commercial firm that provides yearly estimates of population at the block level.

¹⁸ R. Goerge, J. Dilts, D. Yang, M. Wassermann, and A. Clary, "Chicago Children and Youth 1990–2010: Changing Population Trends and Their Implications" (Chicago: Chapin Hall Center for Children at the University of Chicago, 2007).

¹⁹ R. Goerge, A. Harris, L. Bilaver, K. Franzetta, M. Reidy, D. Schexnayder, D. Schroeder, J. Staveley, J. L. Kreader, S. Obenski, R. Prevost, M. Berning, and D. Resnick, "Employment Outcomes for Low-Income Families Receiving Child Care Subsidies in Illinois, Maryland, and Texas" (Chicago: Chapin Hall at the University of Chicago, 2009).

²⁰ See <http://datasmart.ash.harvard.edu/news/article/chicagos-windygrid-taking-situational-awareness-to-a-new-level-259>.

²¹ "Highlights of a Forum: Data Analytics for Oversight and Law Enforcement," GAO-13-680SP (Washington, DC: U.S. Government Accountability Office, July 15, 2013), <http://www.gao.gov/products/GAO-13-680SP>.

²² See <http://wwwaisp.upenn.edu/>.

Appendix A

The data Chapin Hall presented during the project's first two years led to a number of specific actions. These included efforts to (1) improve educational opportunities for Chicago Housing Authority (CHA) children, (2) increase Head Start enrollment among CHA children, (3) improve labor market outcomes of CHA parents, and (4) increase CHA children's enrollment in All Kids. Each of these efforts is described briefly below.

Improve Educational Opportunities for CHA Children

- Contrary to the perceptions of some school administrators, the data Chapin Hall presented indicated that relatively few schools had experienced a large influx of children whose families had relocated because of the Plan for Transformation. However, relocated children did comprise more than 5% of the student population at a small number of schools.
- CHA representatives met with the principals of some of those schools to discuss ways of improving the academic performance of the relocated children.
- Some of those schools participated in a pilot program that aimed to increase coordination between the schools and the CHA Service Connector.
- Chapin Hall also found that CHA children were half as likely to be enrolled in selective, magnet, or charter schools as their non-CHA peers, and that CHA children whose families had relocated continued to be concentrated in underperforming schools even after relocation.
- As a result, CHA and Chicago Public Schools (CPS) began to work together to increase the percentage of CHA children enrolled in selective, magnet, or charter schools, and to encourage CHA parents to consider the performance of the schools their children will be attending when deciding where to relocate.

Increase Head Start Enrollment among CHA Children

- Chapin Hall's analysis showed that only one-third of CHA three-year-olds and 44% of CHA four-year-olds were enrolled in an early childhood program in fall 2005.
- The Chicago Department of Children and Youth Services (DCYS) responded by targeting seven community areas into which a significant number of CHA families had relocated, with the goal of enrolling all CHA three- and four-year-olds in those communities in Head Start by September 2006.

This is a preliminary version of the book Privacy, Big Data, and the Public Good: Frameworks for Engagement, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Improve Labor Market Outcomes of CHA Parents

- Chapin Hall's analysis of Unemployment Insurance wage data from the Illinois Department of Employment Security (IDES) suggested that CHA household heads fall into one of three groups: approximately half have no earnings from employment in any given year; one quarter work sporadically; and one quarter are regularly employed, but generally not earning enough to escape poverty.
- CHA used that information to develop a new employment initiative that involved collaboration between case managers from the Service Connector and the Illinois Department of Human Services and included different interventions for each of the three groups.
- CHA reported high take-up rates in the two communities where the initiative was piloted.
- Service Connector and IDES agreed to discuss how precertification of CHA residents and relocatees, who are categorically eligible for employer tax credits, could increase their attractiveness to potential employers.

Increase CHA Children's Enrollment in All Kids

- Chapin Hall presented data indicating that a significant percentage of CHA children were not enrolled in Medicaid or Kid Care and that these children were probably not covered by insurance provided by a parent's employer.
- Department of Healthcare and Family Services responded by targeting CHA developments and community areas with the most non-enrolled children for All Kids enrollment.
- To facilitate this process, some Service Connector case managers were trained to enroll CHA children in All Kids.

Appendix B

University of Chicago Crime Lab

- *Opportunity:* Test a promising program to reduce youth violence and improve school outcomes through randomized control trial (RCT).
- *Challenge:* (1) Getting consent to link administrative data to the control group individuals, which is challenging because it never results in high enough consent rates for sufficient power to discern differences between control and treatment groups. (2) With consent, researchers could link administrative data on arrests and

school outcomes to treatment and control group individuals. Obtaining school and police data is difficult because of legal restrictions and organizational burden.

- *Solution:* (1) University IRB allowed only treatment group to provide consent for use of their administrative data. Control group did not need to provide consent. Potential benefit outweighed potential harm to control group. (2) One agency had to be taught how to do record linkage and a second agency allowed researchers to receive confidential data to link treatment and control groups.
- *Gain:* Added to the evidence base on what interventions reduced violent behavior and increased school graduation. Showed how to work with university IRB to allow RCT to be implemented in a rigorous manner.

SF Youth Database

- *Opportunity:* Learn about at-risk youth to serve them better.
- *Challenge:* (1) Have three agency directors agree to share their data; (2) actually produce data so that linkage can be done; (3) analyze data so that learning can occur to change practice and policy.
- *Solution:* (1) Agency directors did agree to share data; (2) through cumbersome methods of physically sharing data, data was linked; (3) consultants were engaged to analyze data to facilitate learning.
- *Gain:* Learned how much multiagency youth became involved in criminal justice system and the need to intervene earlier with these youth.

NYC Impact of Shelter Use and Housing Placement on Homeless Adults

- *Opportunity:* Change how homeless adults were served.
- *Challenge:* (1) Acquire data on adults using NYC shelters; (2) match to national death registry.
- *Solution:* (1) Obtained data by working with NYC agencies; (2) matched individuals in shelters to death records despite some data challenges.
- *Gain:* Learned that prompt resolution of homelessness may contribute to reduced mortality.

(See Stephen Metraux, Nicholas Eng, Jay Bainbridge, and Dennis P. Culhane, “The Impact of Shelter Use and Housing Placement on Mortality Hazard for Unaccompanied Adults and Adults in Family Households Entering New York City Shelters: 1990–2002,” *Journal of Urban Health: Bulletin of the New York Academy of Medicine* 88 (2011): 1091–1104.)

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Chicago Data Dictionary

- *Opportunity:* Learn what data is available in City of Chicago databases.
- *Challenge:* (1) List databases/systems/datafiles in City of Chicago code and sister agencies; (2) develop a tool to store and manipulate the data to drive future data use and decision making.
- *Solution:* (1) Passed a city ordinance to conduct this work, contracted with research organization with experience doing this, aggregated existing lists of databases, met with agency personnel to identify contents of databases; (2) built data dictionary application to store all of this information.
- *Gain:* For the first time, there is a process in the city for all information about the data that it collects and is responsible for that can be accessed on a public-facing website.

Chapter 8

A European Perspective

Peter Elias

Introduction

Chapters 6 and 7 have highlighted the research potential of large datasets, particularly those which derive from administrative systems, monitoring devices and customer databases, outlining the legal, ethical and practical issues involved in gaining access to and linking between such data for research with potentially wide social and economic value. While the focus in these earlier chapters has been primarily upon the development of these data as research resources within the United States, many of the legal and ethical issues outlined have wider relevance. Here the focus shifts to an examination of a number of these matters from a European perspective.

There is a clearly stated ambition within Europe is to create a research environment (the European Research Area, ERA¹) in which research interests are promoted via cross-border access to microdata free from legal constraints and other obstacles which form impediments to this ambition such as the languages used² and differences in the scientific culture of research. This chapter explores the legal obstacles to wider cross-border data access for researchers based in different European countries and illustrates with examples how these are being addressed. The chapter is presented in two parts. The first part gives an historical overview of the progress that has been made across Europe to develop a harmonised approach to legislation designed to provide individuals and organisations with what has become known as the ‘right to privacy’. It charts the immediate post war efforts to establish the right to a private life and traces the impact this has had in terms of the use for research of electronic records containing personal information. In so doing an attempt is made not simply to highlight the forces that have helped shape the new legislation that the European Union (EU) is about to introduce, but to address the question as to what constitutes ‘misuse of personal data’ in terms of privacy concerns and to gauge how important this is for European citizens.

The second part examines the impact that these legislative developments have had and are continuing to have on cross-border access to microdata for research purposes. It identifies the tension between the ambitions of the EU to create a European Research Area within which research communities gain access to and share data across national boundaries and the desire within the EU to have a harmonised legislative framework which provides protection from misuse of personal information. How do these competing aims impact upon research plans and what mechanisms are being introduced to facilitate

cross-border access to personal data for research? What part has the research community played in helping to negotiate the legal labyrinths that hinder cross-border access to and sharing of personal data? To shed light on these questions two developments are described which illustrate the approaches that are being developed. The first of these focuses upon personal microdata which form ‘official statistics’³ and shows how academics have been working closely with national statistical agencies to overcome legal obstacles and promote cross-border research access to de-identified micro records. The second describes an initiative being undertaken within one EU member state, the United Kingdom, to promote wider access to and linkage between administrative data held by government departments and agencies operating under different national jurisdictions *within* one country. This example indicates the complex nature of the processes that are required to maintain the balance between the protection of privacy on the one hand and, on the other, the need for good research access to personal information in order to address important research issues.

The chapter concludes by looking forward, identifying the further work that needs to be done at the European level to achieve better transnational access to data held by government agencies, the private sector and the academic community whilst providing adequate safeguards through which individuals can retain protection of their right to a private life.

The Evolution of a ‘Right to Privacy’ across Europe (1948–1981)

At the end of the Second World War many European countries had experienced destruction on a massive scale and most were financially impoverished. Keen to avoid any repetition of the punitive post-war conditions that had given rise to the growth of militaristic fascist movements after the First World War, the U.S. Marshall Plan of 1947 provided a huge macroeconomic injection, reflecting the desire of the United States to see the formation of a more integrated Europe. This in turn meshed with ideas about the formation of a ‘United States of Europe’ which were growing among Western European countries – a political and economic alliance which could form a counterweight to the post war dominance of the USSR across Eastern Europe.

The *International Committee of the Movements for European Unity* was established in 1946. As an umbrella organisation, this committee facilitated discussions between a number of countries (Belgium, France, the Netherlands, Luxembourg and the United Kingdom) which led to a congress held in 1948 calling for political and economic union between the states of Europe. France and Belgium pushed for the creation of a federal Europe, whereas the United Kingdom expressed its wishes to see a consultative framework established for cooperation at the economic level. As a compromise between these two positions the Council of Europe⁴ was formed in 1949, tasking its members to

This is a preliminary version of the book Privacy, Big Data, and the Public Good: Frameworks for Engagement, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

achieve a greater unity ... for the purpose of safeguarding and realising the ideals and principles which are their common heritage and facilitating their economic and social progress.

(Article 1 of the Statute)

One of the most important steps taken by the Council of Europe was the development of the European Convention on Human Rights. Drawing heavily upon the Universal Declaration of Human Rights, which was adopted by the UN General Assembly in 1948, the 1950 European Convention on Human Rights (ECHR), introduced the concept of the right to privacy. Article 8 of the ECHR states that

Everyone has the right to respect for his private and family life, his home and his correspondence

(ECHR: Article 8.1)

and

There shall be no interference by a public authority with the exercise of this right except such as is in accordance with the law and is necessary in a democratic society in the interest of national security, public safety or the economic well-being of the country, for the protection of disorder or crime, for the protection of health or morals, or the protection of the rights and freedoms of others.

(ECHR: Article 8.2)

What distinguished the European Declaration on Human Rights from that of the United Nations was the setting up of the European Court of Human Rights, giving individuals the standing to file legal claims for the restitution of their rights. This, in turn, has established a body of case law surrounding the interpretation of the 'right to respect for private and family life'. Initially these cases covered issues such as the legality of activities by the state to intercept phone conversations and the monitoring by employers of their employees' relationships with others.⁵ However, the transformation in electronic communication which commenced in the late 1970s, together with the widespread adoption of electronic processing of personal information, caused some European countries to re-examine the interpretation of the 'right to a private life' and to consider the need for safeguards specific to the increasing use of electronic data. In the UK a committee was set up 'to consider whether legislation was needed to protect individuals and organisations from intrusions into their personal privacy'.⁶ In its response to this committee's report, the UK government concluded that

This is a preliminary version of the book Privacy, Big Data, and the Public Good: Frameworks for Engagement, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

those who use computers to handle personal information ... can no longer remain the sole judges of whether their own systems adequately safeguard privacy (Cmnd 5353, 1975).⁷

By 1978, and following the report of a further committee,⁸ moves were afoot to create a Data Protection Authority which would have responsibility for drafting new codes, breaches of which would be criminal offences. These moves were echoed at the European level by the Council of Europe and more widely within the OECD. Early in 1978 the OECD established a group of experts, tasked to elaborate a set of principles governing the protection of personal data. Working in close collaboration with the Council of Europe, seven basic principles of data protection were defined. These were:

1. There should be limits to the collection of personal data, which should be collected by fair and lawful means and, where possible, with the consent of the data subject.
2. Personal data should be relevant to the purpose for which they are required, should be accurate, complete and up-to-date.
3. The purpose for which personal data are required should be specified not later than at the time of collection.
4. Personal data should not be disclosed or used for purposes other than that for which they were collected, except with the consent of data subjects.
5. Personal data should be protected by reasonable security safeguards against unauthorised access, loss, destruction, modification or disclosure.
6. Means should be established to facilitate the existence and nature of personal data and the identity and residence of the data controller.
7. Data subjects should have the right to gain access to their data, to challenge such data, to request erasure and to have the right to challenge any denial of these rights.⁹

By 1980 the Council of Europe had proposed a *Convention for the Protection of Individuals with regard to the Automatic Processing of Personal Data*.¹⁰ This convention reflected the seven basic principles agreed by OECD member countries. Additionally the convention specified that data holders should take appropriate security measures against accidental or unauthorised destruction of data as well as unauthorised access, alteration or dissemination. Importantly, the convention had a specific chapter on transborder flows of personal data, stating that

A party shall not, for the sole purpose of the protection of privacy, prohibit or subject to special authorisation transborder flows of personal data going to the territory of another part.

(Chapter 111 Article 12)

This is a preliminary version of the book Privacy, Big Data, and the Public Good: Frameworks for Engagement, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

An important derogation from this provision relates to the situation where national legislation covers and protects specific types of personal data and there is no equivalent legislative protection in the country to which data might flow.

Neither the OECD nor the Council of Europe has legal authority to impose these conditions within national legislation. Signatories to the Council of Europe convention were agreeing to a provision that they would implement domestic legislation to realise its principles. However, there was no guarantee that this would happen in a consistent way across Europe. Nonetheless, the Convention, which adopted by most of the Council of Europe countries,¹¹ represented a landmark in establishing a framework designed to safeguard an individual's right to privacy in the electronic age. Passed in 1981, those who drafted the convention could have had little idea of the rapid pace of technological change that lay ahead.

The Right to Privacy and the European Union (1982–2013)

Throughout the 1980s the European Commission, the body that forms the administrative heart of the European Economic Community (now the European Union), was slow to pick up on the principles enunciated by the OECD and the Council of Europe. In part this reflected the fact that the European Economic Community (EEC) was concerned more with trade and economic considerations. By 1985 the Council of Europe's *Convention for the Protection of Individuals with regard to the Automatic Processing of Personal Data* came into effect, but its adoption among the EEC signatory countries was uneven. Recognising this, and in line with its widening remit to cover more than trade and other economic objectives, the EEC published a draft data protection directive in 1990.

European Union Directives and Regulations

It is important to clarify the difference between two legal instruments, *directives* and *regulations* that have been available to the EEC and now the EU. A *directive* places a legal requirement on member states to achieve a specific result. It does not specify how the result should be achieved; leaving countries with a degree of leeway with respect to the national legislation that might be required to achieve the stated aim of the directive. A *regulation* does not require member states to pass legislation to achieve its aim. It is a legal instrument that applies to all member states and which takes precedence over relevant national legislation.

This is a preliminary version of the book Privacy, Big Data, and the Public Good: Frameworks for Engagement, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

The European Union (EU), formed when the twelve Member States of the European Economic Community (EEC) signed the Treaty of Maastricht in 1995, wished to ensure that the principles enunciated in the Council of Europe Convention should become enshrined in the articles establishing the EU. After a considerable period of consultation within and across member states of the EEC, the EU Data Protection Directive¹² was passed in 1995. While helpful in providing the framework for national legislation, the directive did not legally bind Member States in a direct way. Under the principle known as ‘transposition’, each country should find ways to implement a directive such that its minimum conditions are met. Additionally, an individual does not have recourse to legal remedies if they consider that their privacy has been breached unless such remedies exist within national legislation.

The 1995 directive has been hugely influential in shaping the legislation in EU countries with regard to the implementation of the principles set out by the OECD and the Council of Europe some 15 years earlier. As new countries have been admitted to the EU, they are required to ensure that the aims of the 1995 directive are met. The directive additionally gives citizens the right to access their personal data and to request it to be removed from processing if incomplete or inaccurate.

Despite its influence, the European Commission concluded that, given the immense technological changes that had occurred since it was passed, the directive should be reformed and its operation strengthened by positioning it more directly within EU legislation through the creation of a new regulation.

The EU General Data Protection Regulation

A proposed new law, the General Data Protection Regulation (GDPR), was published by the European Commission in January 2012, aiming for its adoption by 2014 and implementation over the following two years. In addition to the fact that this moves data protection from a set of requirements to be achieved via national legislation to a legal requirement superseding national legislation, the other major changes from the preceding directive cover:

- the scope of the legislation (it will apply to all organisations and individuals based outside the EU if they process personal data of EU residents);
- the establishment of national data protection authorities to be coordinated by a European Data Protection Board;
- the need for consent (valid consent for data to be collected and the purposes for which it will be used must be explicit rather than implicit);

- the right to be forgotten (personal data must be removed from use if consent is withdrawn).

While much remains to be clarified before the proposed regulation is passed by the European Parliament¹³ it is clear that the need for consent could have a serious impact upon the collection, access to and sharing of research data. Some indication of these concerns can be gathered from documents prepared by those organisations with research interests (refs.). Of particular concern here is Article 83 of the proposed regulation, which is termed a derogation – an exception to the requirements of the proposed regulation regarding the need for explicit consent for data collection and from the restrictions on the processing of data which are deemed ‘sensitive’.¹⁴ For this derogation to apply, specific conditions are to be placed on the processing of personal data for ‘historical, statistical and scientific research purposes’:

Within the limits of this Regulation, personal data may be processed for historical, statistical or scientific research purposes only if:

- (a) *these purposes cannot be otherwise fulfilled by processing data which does not permit or no longer permit the identification of the data subject*
- (b) *data enabling the attribution of information to an identified or identifiable data subject is kept separately from the other information as long as these purposes can be fulfilled in this manner.*

(Proposed General Data Protection Regulation, Article 83(1), 2012)

Additionally, the proposed regulation would empower the European Commission to further specify the criteria and requirements associated with the processing of personal data for historical, statistical or scientific research purposes. The regulation provides for the imposition of sanctions upon those individuals or organisations that fail to comply with the new Europe-wide legislation. Unintentional breaches of the regulation could lead to a warning from data protection authorities and/or regular audits of data protection practices. Stiffer penalties can be imposed (of up to €100m or 5% of turnover in the case of organisations). Penal sanctions that would be directly applicable to individuals fall outside the competence of the European Union.

Many amendments have been submitted to the European Commission seeking to remove this derogation or to limit it only to personal data concerning the health status of the data subject. If adopted, such amendments would limit research based on personal data to those data for which data subjects had given personal consent. For research on a wide range of topics in the social, economic, behavioural and medical sciences, analysis

of large linked datasets is an essential part of the research process. Even if consent had been obtained at the time data were originally collected, the reuse of such data for a new purpose would require consent to be obtained again.

Finally, it should be noted that the GDPR does not oblige data controllers to provide access to personal information for scientific research purposes. Such access would be legal, but cannot be claimed.

The Views of EU Citizens

For legislation to work effectively it must reflect an established need for a legal instrument and it must have widespread support for the outcome it seeks to achieve. In 2010, and as part of the preparatory work to inform the development of the General Data Protection Regulation, the European Commission organised a survey on attitudes to and awareness of personal privacy. More than 25,000 European aged 15 and over were interviewed¹⁵ across 27 member states. Among many areas which were explored, questions were asked about the following:

- the nature of information considered as personal;
- perceived necessity of disclosing personal information;
- knowledge of regulations governing the processing of personal data.

Regarding the responses to categories of data deemed personal, the EU average responses showed that financial and medical information are ranked higher than other categories of data, with about three quarters placing these two categories top of the list. There were large variations between countries, with respondents from North West Europe typically stating that more than three quarters regarded financial information as personal compared with just over 40% in Poland and Romania. Similar variation was found for the view that medical information was personal.

Three quarters of Europeans interviewed agreed that disclosing personal information is an increasing part of modern life. Again, respondents in North West Europe were more inclined to this view than those of central and southern Europe. Unsurprisingly, only one third of Europeans are aware of the existence of a national public authority protecting their rights regarding their personal data.

Figure 1 illustrates the wide variation that exists in the public perception of the extent to which different organisations are trusted to protect personal information.

While the survey contained no specific questions relating to the use for research of their personal data, it is clear from the responses to questions about who collects personal data that there is a reasonably high degree of trust in health and medical institutions and national public authorities to protect personal information. Given that these organisations

are a major source of personal data with research value, it is incumbent upon them to maintain (and possibly increase further) this trust.

Some member states have also undertaken research into the public attitudes towards access to and linkage between personal information. A UK report for the Wellcome Trust was based upon qualitative research designed to understand the general public's attitudes to different types of personal data and data linking.¹⁶ The research looked at whether health data are viewed differently from other types of data, and what are the perceived risks and benefits, to self and society, of linking different kinds of data for research purposes and other purposes. The conclusions of this small scale study echo the findings from the Europe-wide survey conducted some years earlier, namely that if the potential benefits of the research could be made clear to data subjects, and if the risks of any harm arising to them from inappropriate use of linked data were minimised, there was general approval for research access to and linkage between personal data.

What Constitutes 'Misuse of Personal Data'?

For research purposes it is vital that access to and linkage between personal information should not be restricted by the new European legislation. Researchers rarely need to know the identities of those who are the subjects of their research, working usually with what are termed 'de-identified data'. But linkage between different datasets does require that information, which is unique to individuals, is available in the data sources to be linked. Data security is key to the maintenance of public trust in the use of personal data and security measures must be maintained in order to avoid misuse. It is important therefore to identify what constitutes a 'misuse of personal information' in the context of research. There is no legal definition of research misuse of personal data,¹⁷ and it has been left to data controllers to establish their own rules over what constitutes misuse. Typically, these rules relate to inadvertent or deliberate disclosure of identities, unauthorised data use, and the failure to maintain data security. More specifically, they cover:

- placing within the public domain information which reveals (or has the potential to reveal) the identities of data subjects;
- using personal information from data subjects for some purpose other than research which has been planned and agreed with data controllers;
- failing to maintain personal information in a secure condition, such that others who do not have access to such data for research purposes may gain access.

It should be noted that these conditions do not make any statement about the purpose of the research nor of the 'fitness for purpose' of the data to be used for research and the quality and accuracy of the research findings stemming from the research. These are

This is a preliminary version of the book Privacy, Big Data, and the Public Good: Frameworks for Engagement, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

matters which should be regulated by data controllers and may be specific to local conditions and cultural sensitivities or which will be revealed by peer review of research findings.

There is, as yet, no clear framework for the elaboration of these rules in a consistent manner across Europe. As a corollary, there is no agreement on the nature of the penalties to be imposed as a result of a breach of these rules.

Where Do We Now Stand?

Since the late 1940s, European countries and Europe-wide institutions have been concerned to protect individual privacy, particularly insofar as such privacy relates to the electronic processing of personal data. Through various conventions, guidelines, directives and now a regulation, the countries of Europe have been moving gradually towards a harmonised and consistent approach towards the application of a right to privacy. In so doing, a balance has been sought between the adoption of policies, procedures and laws which provide this right, and the value to society of research which makes use of personal data to address issues which are of public benefit. For the study of a wide range of issues including: health-related matters; the distribution of social and economic disadvantage; educational progress; environmental conditions; and political developments, access to relevant personal data is vital for research. A major step forward is now underway to harmonise European legislation on data protection. This need for a balance is recognised in the proposed European regulation but the derogation which facilitates such use of personal data by freeing research use from the need for consent is under attack from a range of interest groups and lobbyists who see this as the ‘thin end of a wedge’ – a set of clauses which allows for the processing of personal data without the consent of data subjects and for research purposes which may be spurious. While these concerns are understandable, it is up to the research communities, their funders and their policy beneficiaries to illustrate how these concerns will be addressed and the concomitant risks to individual privacy minimised. The next section of this chapter presents two examples that illustrate how this can be achieved.

Overcoming Obstacles to Data Access and Sharing

The preceding sections of this chapter have focussed specifically upon the development of the legal definition of a ‘right to privacy’ across Europe and give some indication of the restrictions that this can place upon the research use of personal data. Currently, it is national legislation and the national interpretation of the EU concept of ‘data protection’ that creates the biggest obstacle to cross-border data sharing. For some countries of the EU, such as Germany with its federal structure, the ambitions of federally-funded research efforts to conduct comparative research within the country and between states

(Länder) is restricted by the legal power of the states relative to the federal government.¹⁸ Similarly, within the United Kingdom, the devolution of power in some administrative areas from the UK government to the devolved nations (Scotland, Wales, and Northern Ireland) has created data sharing issues. Given that problems can exist within countries, it will come as no surprise to find that EU institutions face difficulties in permitting transnational access for research as the next section will demonstrate.

Cross-Border Access to Microdata Designated as Official Statistics

Nowhere is this tension between the need for research access to personal data and national privacy laws more evident than at the Statistical Offices of the European Communities (Eurostat). While member states are required to provide harmonised statistical data to Eurostat, which are then made publically available in aggregate form, access to the national official microdata records supplied to Eurostat remains subject to national laws. This, in turn, led to a situation where official microdata records that may be generally available for research in one country could not be released by Eurostat without the consent of all countries.

Under considerable pressure from research communities across the EU, resolution of this problem was high on the Eurostat agenda for many years. In 2002 a Commission Regulation¹⁹ came into force which permitted ‘authorised bodies’ (e.g. universities and recognised research institutes) to have access to anonymised microdata from a limited number of EU-wide surveys, subject to approval by all national statistical authorities. While this did provide an avenue for cross-border research access, the process of gaining access was slow and expensive, often taking more than 6 months from initial application and with no guarantee of a successful outcome. This regulation was repealed in 2013 with a new regulation²⁰ designed to improve the speed and efficiency of cross-border access to official microdata records for research purposes. Under this new regulation, Eurostat has responsibility for overseeing the operation of any proposed cross-border data access request. The national statistical authorities that provided the data to Eurostat are consulted on each research proposal submitted by researchers interested in access to EU confidential personal information. If the statistical authority of one country does not accept a particular research proposal, the data for this country is then excluded from the cross-national database to be supplied for the stated research purpose. The decision of one country now has no influence on the access to other countries data.

To take advantage of the changing EU landscape with respect to cross-border access to official microdata, **Data without Boundaries** (DwB) was set up as an Integrating Activity.²¹ With a four-year budget covering the period 2011–2015, DwB is an initiative led primarily by academic research groups to overcome the legal and other barriers to cross-border research based upon access to microdata records held by national statistical institutes. There are three main strands to the work plan for DwB. First, it encourages

transnational access to data held at national statistical institutes by funding the travel and subsistence costs for researchers and paying any usage charges they may face. Second, it is promoting the development of a network of remote access centres, through which access can be given to authorised researchers without them having to travel to the place where national official microdata records are stored. Remote access is provided by various means, including the use of thin client technology, submission of analytical code by email and the use of encryption devices. Third, it has a range of additional activities (standards development, training workshops, etc.) all of which are designed to promote wider use of European official microdata for research.

While these developments are to be welcomed, it is unlikely that an access facility operated by Eurostat, providing remote access to official EU microdata by European researchers will become a realistic proposition in the short run. Nevertheless, the hope is that, as some countries develop remote access to data held in their national statistical institutes (e.g. as is now the case in the Netherlands and the United Kingdom), other countries will perceive the benefits that ensue in terms of the research it promotes.

Overcoming National Obstacles to Research Data Access and Sharing

As the experience of Data without Boundaries has revealed, the major obstacles to transnational data access often lie at the national level. However, it is not simply a question of the need for revised national legislation, or for a new European regulation to be implemented. National issues can be tricky and complex as will be illustrated by recent developments within the United Kingdom to improve research access to and linkage between public sector administrative datasets.

Administrative data in the UK, as in most other European countries and the USA, are derived from a range of activities which constitute the everyday functioning of public agencies. Raising tax revenues from individuals and businesses and from the sale of goods, paying social security benefits and pensions, recording educational progress, monitoring judicial systems all provide an ‘electronic trail’ that has significant research potential given that the data already exist, are usually population wide and are continuous through time. While individual government departments and agencies may use such data for performance monitoring or service delivery, they are often underutilised as research resources. Attempts to gain access to such data for research purposes have sometimes been denied, especially where requests have been made to link data held by different departments.

In the area of medical statistics, personal data which are the by-product of an administrative system are now being linked to generate new and powerful research resources. While progress has been slow to digitise the UK’s National Health Service, as new national datasets started to become available the health research community acted quickly to establish the legal and ethical framework for access to and linkage between

pseudonymised health records. Based on this experience, and recognising the vast research potential of other categories of administrative data, a group of research funding agencies convened a workshop in May 2011 to review progress made across the biomedical sciences in respect of the research use of data derived from administrative systems in health services (e.g. hospital admissions, GP practice visit, patient prescribing data, etc.) and to consider the much slower rate of progress in areas of wider socio-economic interest (incomes, social security, education, housing, environmental conditions, etc.). Given the lack of progress in the latter area the workshop advocated that a Taskforce should be set up to detail the problems and to propose solutions.

In its report, the Administrative Data Taskforce²² identified the major obstacles surrounding access to, linking between and sharing personal information for research purposes. These were:

- the legal status of those bodies holding data;
- the lack of agreed and common standards covering data security and the authentication of potential research users;
- the need for public support for the use for research of de-identified personal information;
- the need for a coordinated governance structure for all activities associated with access, linking and sharing personal information.

The report of the Taskforce was welcomed by the UK Government, which allocated sufficient funds to allow for a Research Data Centre to be established in each of the four countries of the UK, coordinated and supported by a new national service (the Administrative Data Research Service) and with the formation of a UK-wide governance structure currently under construction. Importantly, a legal team was set up to draft new legislation which would permit data sharing and linkage by those bodies which currently faced legal barriers to such activities.

This example demonstrates how complex obstacles to data sharing can be overcome if there is sufficient impetus from the scientific community. An important element of this impetus comes from demonstration of the research value of new forms of data and the public benefit that derives from the research they underpin.

EU Privacy Legislation and Research – Looking Ahead

For the past two decades European countries have made concerted efforts to prevent the misuse of personal information held in electronic formats. However, the approach that was adopted has led to a situation where there are varying degrees of protection in different countries. Coupled with the technological changes that have taken place over

this same period, such as cloud storage and processing of data, the development of high speed data networks and the flow of personal data through social media, the European Commission has taken the bold step of proposing the introduction of a new legal instrument that would supersede all national legislation, thereby providing a common legal framework for the protection of personal information. The proposed new law, the General Data Protection Regulation, is making steady progress through the European institutions and is likely to be passed by the European Parliament and adopted by the European Council in 2014. Fundamental to the new law is the notion that individuals should give consent for their personal data to be processed. Recognising that this could have serious consequences for research, a derogation has been included in the legislation which will permit the processing of personal information for research purposes if consent is impractical or infeasible.

In the widespread public consultation associated with the introduction of the new legislation, concerns have surfaced about the way this derogation will operate. What is classed as research? What if researchers fail to protect the information they are processing even if it has been de-identified? In other words, will this derogation provide a loophole, allowing the processing of personal information in ways which, although legal, could cause reputational or physical harm to individuals? The fact that amendments to the new law have been put forward to strike out this derogation indicates that some unease exists among sections of the European citizenry. A survey of the adult population across 27 countries indicates that although there is a reasonable degree of trust among the population over the extent to which different institutions can be trusted to hold their personal data in a secure manner, for data collected by some institutions, notably those in the private sector organisations, more than half the population does not trust them to hold data securely. In this situation, research use of such data may give rise to public concern about the nature of the derogation for research in the new law.

Looking ahead, it is likely that the next few years are likely to be difficult for researchers who wish to build significant new research resources from the growing volumes of 'big data' that can be linked via personal identifying information (e.g. official statistical sources, private sector databases, charitable bodies) especially where such data may cross national boundaries. Those countries that traditionally have had a cautious approach to research access to administrative data (e.g. Germany) are unlikely to modify their practices following adoption of the new law. Conversely, those countries that have a more open approach (e.g. Netherlands) may now have to reconsider how they work as the European Commission develops the protocols that will sit alongside the new legislation – protocols that will define with more clarity how the derogation for research will be operated.

The uncertainties that will prevail at the European level should not stop individual member states from moving ahead with ambitious plans to link large data sets held by the

public and private sectors. Foremost in this respect is the work currently being undertaken within the UK to facilitate linkage at the individual level between comprehensive health records, tax and social security data, education records and criminal justice records. The goal within the UK is to establish mechanisms that allow researchers to build large-scale longitudinal databases on individuals and organisations using detailed information on individuals that already exists within different public and private sector bodies. This work is driven by the research community, working closely with data controllers to ensure that the steps taken to move towards this goal are legal, ethical, well-governed and capable of producing benefits for the well-being of the citizenry.

Translating the actions of any single member state into Europe-wide procedures which allow cross-border access to large linked datasets is going to prove slow and cumbersome, even though the new General Data Protection Regulation is designed to improve efficiency in this respect. However, the time is right for the development of ethical guidelines for governance of the research use of data, particularly where such data arise from administrative systems, customer databases, monitoring devices or communications and where transnational access for research is proposed. While the research community must take a clear lead in this area,²³ the OECD, European Commission and Council of Europe have shown previously that they can coordinate the resources and expertise needed to develop such guidelines and that they have the collective voice required for their implementation.

Notes

¹ The European Research Area is described by the European Commission as ‘a unified research area open to the world, based on the internal market, in which researchers, scientific knowledge and technology circulate freely’. European Commission, Proposal for a Regulation of the European Parliament and of the Council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation), Brussels COM 2012 011, 2012.

² The European Union has 24 official and working languages.

³ Data that are collected via national statistical agencies, often via legal instruments that require individuals and organisations to provide the information that is requested.

⁴ Not to be confused with the *Council of the European Union*, also informally known as the *EU Council*, which is where national ministers from each EU country meet to adopt laws and coordinate policies, and the *European Council*, another EU institution, where EU leaders meet around 4 times a year to discuss the EU’s political priorities. The **Council of Europe** is not an EU institution.

⁵ See e.g. Niemetz v. Germany and Capland v. UK.

⁶ Known as the report of the Younger Committee. UK House of Lords, Committee on Privacy, Report of the Committee on Privacy, Kenneth Younger, chair (Home Office, Cmnd 5012, H. M. Stationery Office, 1972).

⁷ Some indication of the scale of change that was under way comes from a survey conducted as part of the work of the Younger Committee. In April 1971 it was estimated that the total number of computers in the UK in use or an order for all purposes was 6,075. By 1995 it was estimated that at least half a million data users were obliged to register under the 1984 Data Protection Act (see n.12).

⁸ The Lindop Report on Data Protection (Cmnd 7341, H. M. Stationery Office, 1978).

⁹ OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data.

¹⁰ ‘Automatic processing’ is defined in the Convention as ‘the following operations if carried out in whole or in part by automated means: storage of data, carrying out of logical and/or arithmetic operations on those data, their alteration, erasure, retrieval or dissemination’ (Article 2, Chapter 1 European Convention on Personal Data Protection).

¹¹ In the UK it led to passage of the 1984 Data Protection Act.

¹² Officially known as Directive 95/56/EC ‘on the protection of individuals with regard to the processing of personal data and on the free movement of such data’.

¹³ For an overview of the more than 3,000 amendments which have been submitted in the committees involved with the regulation, plus analysis of the weight of opinion towards strengthening or weakening the regulation, see <http://lobbyplag.eu/lp>.

¹⁴ Data that relate to issues such as sexual orientation, religious and political beliefs.

¹⁵ A regionally stratified sample of household addresses was used for sampling purposes. The individual selected for interview was on the ‘nearest birthday’ rule. TNS Opinion and Social, *Special Eurobarometer 359: Attitudes on Data Protection and Electronic Identity in the European Union* (Brussels, 2011).

¹⁶ Wellcome Trust, *Summary Report of Qualitative Research into Public Attitudes to Personal Data and Linking Personal Data* (London: Wellcome Trust, 2013). Available at http://www.wellcome.ac.uk/stellent/groups/corporatesite/@msh_grants/documents/web_document/wtp053205.pdf.

¹⁷ As opposed to a ‘breach of personal data processing’, which is defined in European law and relates to the requirement of data controllers to prevent a breach of security leading

to accidental or unlawful destruction, loss, unauthorised disclosure of access to personal information (Article 2(i) of e-privacy Directive 2002/58/EC).

¹⁸ As an example, data collected within the National Educational Panel Study, a multi-cohort longitudinal study of educational progress, may not be used for cross-state comparative research that identifies specific states.

¹⁹ Commission Regulation (EC) No 831/2002 on Community Statistics, concerning access to confidential data for scientific purposes.

²⁰ Commission Regulation (EU) No 557/2013 as regards access to confidential data for scientific purposes.

²¹ The formal name for projects that help to integrate research resources and researchers across the European Union.

²² The UK Economic and Social Research Council, the UK Medical Research Council and the Wellcome Trust (ref).

²³ See e.g. Elias and Entwistle (2013).

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Chapter 9

The New Deal on Data: A Framework for Institutional Controls

Daniel Greenwood, Arkadiusz Stopczynski, Brian Sweatt,
Thomas Hardjono, and Alex Pentland

Introduction

In order to realize the promise of a Big Data society and to reduce the potential risk to individuals, institutions are updating the operational frameworks which govern the business, legal, and technical dimensions of their internal organizations. In this chapter we outline ways to support the emergence of such a society within the framework of the *New Deal on Data*, and describe future directions for research and development.

In our view, the traditional control points relied on as part of corporate governance, management oversight, legal compliance, and enterprise architecture must evolve and expand to match operational frameworks for big data. These controls must support and reflect greater user control over personal data, as well as large-scale interoperability for data sharing between and among institutions. The core capabilities of these controls should include responsive rule-based systems governance and fine-grained authorizations for distributed rights management.

The New Realities of Living in a Big Data Society

Building an infrastructure that sustains a healthy, safe, and efficient society is, in part, a scientific and engineering challenge which dates back to the 1800s when the Industrial Revolution spurred rapid urban growth. That growth created new social and environmental problems. The remedy then was to build centralized networks that delivered clean water and safe food, enabled commerce, removed waste, provided energy, facilitated transportation, and offered access to centralized health care, police, and educational services. These networks formed the backbone of society as we know it today.

These century-old solutions are, however, becoming increasingly obsolete and inefficient. We now face the challenges of global warming, uncertain energy, water, and food supplies, and a rising population and urbanization that will add 350 million people to the urban population by 2025 in China alone.¹ The new challenge is how to build an infrastructure that enables cities to be energy efficient, have secure food and water supplies, be protected from pandemics, and to have better governance. Big data can

enable us to achieve such goals. Rather than static systems separated by function – water, food, waste, transport, education, energy – we can instead regard the systems as dynamic, data-driven networks. Instead of focusing only on access and distribution, we need networked and self-regulating systems, driven by the needs and preferences of citizens – a ‘nervous system’ that maintains the stability of government, energy, and public health systems around the globe. A *control* framework should be established which enables data to be captured about different situations, those observations to be combined with models of demand and dynamic reaction, and the resulting predictions to be used to tune the nervous system to match those needs and preferences.

The engine driving this nervous system is big data: the newly ubiquitous digital data now available about so many aspects of human life. We can analyze patterns of human activity within the digital breadcrumbs we all leave behind as we move through the world: call records, credit card transactions, GPS location fixes, among others.² These data, which record actual activity, may be very different from what we put on Facebook or Twitter; our postings there are what we choose to tell people, edited according to the standards of the day and filtered to match the persona we are building. Although mining social networks can give great insight into human nature,³ the value is limited for operational purposes.⁴

The process of analyzing the patterns within these digital breadcrumbs is called ‘reality mining.’⁵ The Human Dynamics research group at MIT found that these patterns can be used to tell us if we are likely to get diabetes,⁶ or whether we are the sort of person who will pay back loans.⁷ By analyzing them across many people, we are discovering that we can begin to explain many things – crashes, revolutions, bubbles – that previously appeared unpredictable.⁸ For this reason, the magazine *MIT Technology Review* named our development of reality mining one of the 10 technologies that will change the world.⁹

The New Deal on Data

The digital breadcrumbs we leave behind are clues to who we are, what we do, and what we want. This makes personal data – data about individuals – immensely valuable, both for public good and for private companies. As the European Consumer Commissioner, Meglena Kuneva, said recently, “Personal data is the new oil of the Internet and the new currency of the digital world.”¹⁰ The ability to see details of so many interactions is also immensely powerful and can be used for good or for ill. Therefore, protecting personal privacy and freedom is critical to our future success as a society. We need to enable more data sharing for the public good; at the same time, we need to do a much better job of protecting the privacy of individuals.

A successful data-driven society must be able to guarantee that our data will not be abused – perhaps especially that government will not abuse the power conferred by access to such fine-grained data. There are many ways in which abuses might be directly

targeted – from imposing higher insurance rates based on individual shopping history,¹¹ to creating problems for the entire society, by limiting user choices and enclosing users in information bubbles.¹² To achieve the potential for a new society, we require the *New Deal on Data*, which describes workable guarantees that the data needed for public good are readily available while at the same time protecting the citizenry.¹³

The key insight behind the New Deal on Data is that our data are worth more when shared. Aggregate data – averaged, combined across population, and often distilled to high-level features – can be used to inform improvements in systems such as public health, transportation, and government. For instance, we have demonstrated that data about the way we behave and where we go can be used to minimize the spread of infectious disease.¹⁴ Our research has also shown how digital breadcrumbs can be used to track the spread of influenza from person to person on an individual level. And the public good can be served as a result: if we can see it, we can also stop it. Similarly, if we are worried about global warming, shared, aggregated data can reveal how patterns of mobility relate to productivity.¹⁵ This, in turn, equips us to design cities that are more productive and, at the same time, more energy efficient. However, to obtain these results and make a greener world, we must be able to see people moving around; this depends on having many people willing to contribute their data, if only anonymously and in aggregate. In addition, the Big Data transformation can help society find efficient means of governance by providing tools to analyze and understand what needs to be done, and to reach consensus on how to do it. This goes beyond simply creating more communication platforms; the assumption that more interaction between users will produce better decisions may be very misleading. Although in recent years we have seen impressive uses of social networks for better organization in society, for example during political protests,¹⁶ we are far from even starting to reach consensus about the big problems: epidemics, climate change, pollution – big data can help us achieve such goals.

However, to enable the sharing of personal data and experiences, we need secure technology and regulation that allows individuals to safely and conveniently share personal information with each other, with corporations, and with government. Consequently, the heart of the New Deal on Data must be to provide both regulatory standards and financial incentives enticing owners to share data, while at the same time serving the interests of individuals and society at large. We must promote greater idea flow among individuals, not just within corporations or government departments.

Unfortunately, today most personal data are siloed in private companies and therefore largely unavailable. Private organizations collect the vast majority of personal data in the form of mobility patterns, financial transactions, and phone and Internet communications. These data must not remain the exclusive domain of private companies, because they are then less likely to contribute to the common good; private organizations must be key players in the New Deal on Data. Likewise, these data should not become the exclusive

domain of the government. The entities who should be empowered to share and make decisions about their data are the people themselves: users, participants, citizens. We can involve both experts and use the wisdom of crowds – users themselves interested in improving society.

Personal Data: Emergence of a New Asset Class

One of the first steps to promoting liquidity in land and commodity markets is to guarantee ownership rights so that people can safely buy and sell. Similarly, a first step toward creating more ideas and greater flow of ideas – idea liquidity – is to define ownership rights. The only politically viable course is to give individual citizens key rights over data that are about them, the type of rights that have undergirded the European Union's Privacy Directive since 1995.¹⁷ We need to recognize personal data as a valuable asset of the individual, which can be given to companies and government in return for services.

We can draw the definition of ownership from English common law on ownership rights of possession, use, and disposal:

- *You have the right to possess data about yourself.* Regardless of what entity collects the data, the data belong to you, and you can access your data at any time. Data collectors thus play a role akin to a bank, managing data on behalf of their ‘customers’.
- *You have the right to full control over the use of your data.* The terms of use must be opt in and clearly explained in plain language. If you are not happy with the way a company uses your data, you can remove the data, just as you would close your account with a bank that is not providing satisfactory service.
- *You have the right to dispose of or distribute your data.* You have the option to have data about you destroyed or redeployed elsewhere.

Individual rights to personal data must be balanced with the need of corporations and governments to use certain data-account activity, billing information, and the like to run their day-to-day operations. The New Deal on Data therefore gives individuals the right to possess, control, and dispose of copies of these required operational data, along with copies of the incidental data collected about the individual, such as location and similar context. These ownership rights are not exactly the same as literal ownership under modern law; the practical effect is that disputes are resolved in a different, simpler manner than would be the case for land ownership disputes, for example.

In 2007, one author (AP) first proposed the New Deal on Data to the World Economic Forum.¹⁸ Since then, this idea has run through various discussions and

eventually helped to shape the 2012 Consumer Data Bill of Rights in the United States, along with a matching declaration on Personal Data Rights in the European Union.

The World Economic Forum (WEF) echoed the European Consumer Commissioner Meglena Kuneva in dubbing personal data the ‘new oil’ or new resource of the 21st century.¹⁹ The ‘personal data sector’ of the economy today is in its infancy, its state akin to the oil industry during the late 1890s. Productive collaboration between government (building the state-owned freeways), the private sector (mining and refining oil, building automobiles), and the citizens (the user-base of these services) allowed developed nations to expand their economies by creating new markets adjacent to the automobile and oil industries.

If personal data, as the new oil, is to reach its global economic potential, productive collaboration is needed between all stakeholders in the establishment of a *personal data ecosystem*. A number of fundamental uncertainties exist, however, about privacy, property, global governance, human rights – essentially about who should benefit from the products and services built on personal data.²⁰ The rapid rate of technological change and commercialization in the use of personal data is undermining end-user confidence and trust.

The current personal data ecosystem is feudal, fragmented, and inefficient. Too much leverage is currently accorded to service providers that enroll and register end-users. Their siloed repositories of personal data exemplify the fragmentation of the ecosystem, containing data of varying qualities; some are attributes of persons that are unverified, while others represent higher quality data that have been cross-correlated with other data points of the end-user. For many individuals, the risks and liabilities of the current ecosystem exceed the economic returns. Besides not having the infrastructure and tools to manage personal data, many end-users simply do not see the benefit of fully participating. Personal privacy concerns are thus addressed inadequately at best, or simply overlooked in the majority of cases. Current technologies and laws fall short of providing the legal and technical infrastructure needed to support a well-functioning digital economy.

Recently, we have seen the challenges, but also the feasibility of opening private big data. In the Data for Development (D4D) Challenge (<http://www.d4d.orange.com>), the telecommunication operator Orange opened access to a large dataset of call detail records from the Ivory Coast. Working with the data as part of a challenge, teams of researchers came up with life-changing insights for the country. For example, one team developed a model for how disease spreads in the country and demonstrated that information campaigns based on one-to-one phone conversations among members of social groups can be an effective countermeasure.²¹ Data release must be carefully done, however; as we have seen in several cases, such as the Netflix Prize privacy disaster²² and other similar privacy breaches,²³ true anonymization is extremely hard – recent research by de

Montjoye et al. and others^{24,25} has shown that even though human beings are highly predictable, we are also unique. Having access to one dataset may be enough to uniquely fingerprint someone based on just a few data points, and this fingerprint can be used to discover their true identity. In releasing and analyzing the D4D data, the privacy of the people who generated the data was protected not only by technical means, such as removal of personally identifiable information (PII), but also by legal means, with the researchers signing an agreement that they would not use the data for re-identification or other nefarious purposes. Opening data from the silos by publishing static datasets – collected at some point and unchanging – is important, but it is only the first step. We can do even more when data is available in real time and can become part of a society's nervous system. Epidemics can be monitored and prevented in real time,²⁶ underperforming students can be helped, and people with health risks can be treated before they get sick.²⁷

The report of the World Economic Forum²⁸ suggests a way forward by identifying useful areas on which to focus efforts:

- *Alignment of key stakeholders* Citizens, the private sector, and the public sector need to work in support of one another. Efforts such as NSTIC²⁹ in the United States – albeit still in its infancy – represent a promising direction for global collaboration.
- *Viewing 'data as money'* There needs to be a new mindset, in which an individual's personal data items are viewed and treated in the same way as their money. These personal data items would reside in an 'account' (like a bank account) where they would be controlled, managed, exchanged, and accounted for just as personal banking services operate today.
- *End-user centricity* All entities in the ecosystem need to recognize end-users as vital and independent stakeholders in the co-creation and exchange of services and experiences. Efforts such as the User Managed Access (UMA) initiative³⁰ provide examples of system design that are user-centric and managed by the user.

Enforcing the New Deal on Data

How can we enforce this New Deal? The threat of legal action is important, but not sufficient; if you cannot see abuses, you cannot prosecute them. Enforcement can be addressed significantly without prosecution or public statute or regulation. In many fields, companies and governments rely on rules governing common business, legal, and technical (BLT) practices to create effective self-organization and enforcement. This approach holds promise as a method by which institutional controls can form a reliable operational framework for big data, privacy, and access.

One current best practice is a system of data sharing called a 'trust network', a combination of networked computers and legal rules defining and governing expectations

regarding data. For personal data, these networks of technical and legal rules keep track of user permissions for each piece of data and act as a legal contract, specifying what happens in case of a violation. For example, in a trust network all personal data can have attached labels specifying where the data come from and what they can and cannot be used for. These labels are exactly matched by the terms in the legal contracts between all of the participants, stating penalties for not obeying them. The rules can – and often do – reference or require audits of relevant systems and data use, demonstrating how traditional internal controls can be leveraged as part of the transition to more novel trust models. A well-designed trust network, elegantly integrating computer and legal rules, allows automatic auditing of data use and allows individuals to change their permissions and withdraw data.

The mechanism for establishing and operating a trust network is to create system rules for the applications, service providers, data, and the users themselves. System rules are sometimes called ‘operating regulations’ in the credit card context, ‘trust frameworks’ in the identity federation context, or ‘trading partner agreements’ in a supply value chain context. Several multiparty shared architectural and contractual rules create binding obligations and enforceable expectations on all participants in scalable networks. Furthermore, the design of the system rules allows participants to be widely distributed across heterogeneous business ownership boundaries, legal governance structures, and technical security domains. However, the parties need not conform in all or even most aspects of their basic roles, relationships, and activities in order to connect to a trust network. Cross-domain trusted systems must – by their nature – focus enforceable rules narrowly on commonly agreed items in order for that network to achieve its purpose.

For example, institutions participating in credit card and automated clearing house networks are subject to profoundly different sets of regulations, business practices, economic conditions, and social expectations. The network rules focus on the topmost agreed items affecting interoperability, reciprocity, risk, and revenue allocation. The knowledge that fundamental rules are subject to enforcement action is one of the foundations of trust and a motivation to prevent or address violations before they trigger penalties. A clear example of this approach can be found in the Visa Operating Rules, which cover a vast global real-time network of parties agreeing to rules governing their roles in the system as merchants, banks, transaction processors, individual or business card holders, and other key system roles.

Such rules have made the interbank money transfer system among the safest systems in the world and the backbone for daily exchanges of trillions of dollars, but until recently those were only for the ‘big guys’.³¹ To give individuals a similarly safe method of managing personal data, the Human Dynamics group at MIT, in partnership with the Institute for Data Driven Design (co-founded by John Clippinger and one author (AP)) have helped to build an open Personal Data Store (openPDS).³² The openPDS is a

consumer version of a personal cloud trust network now being tested with a variety of industry and government partners. The aim is to make sharing personal data as safe and secure as transferring money between banks.

When dealing with data intended to be accessible over networks – whether big, personal, or otherwise – the traditional container of an institution makes less and less sense. Institutional controls apply, by definition, to some type of institutional entity such as a business, governmental, or religious organization. A synopsis of all the BLT facts and circumstances surrounding big data is necessary in order to know what access, confidentiality, and other expectations exist; the relevant contextual aspects of big data at one institution are often profoundly different from those at another. As more and more organizations use and rely on big data, a single formula for institutional controls will not work for increasingly heterogeneous BLT environments.

The capacity to apply appropriate methods of enforcement for a trust network depends on clear understanding and agreement among the parties about the purpose of the system and the respective roles or expectations of those connecting as participants. Therefore, some contextual anchor is needed to have a clear basis for establishing an operational framework and institutional controls appropriate for big data.

Transitioning End-User Assent Practices

The way users grant authorization to share their data is not a trivial matter. The flow of personal information such as location data, purchases, and health records can be very complex. Every tweet, geotagged picture, phone call, or purchase with credit card provides the user's location not only to the primary service, but also to all the applications and services that have been authorized to access and reuse these data. The authorization may come from the end-user or be granted by the collecting service, based on umbrella terms of service that cover reuse of the data. Implementation of such flows was a crucial part of the Web 2.0 revolution, realized with RESTful APIs, mash-ups, and authorization-based access. The way personal data travels between services has arguably become too complex for a user to handle and manage.

Increasing the range of data controlled by the user and the granularity of this control is meaningless if it cannot be exercised in an informed way. For many years, a poor model has been provided by End User License Agreements (EULAs), long incomprehensible texts that are accepted blindly by users trusting they have not agreed to anything that could harm them. The process of granting meaningful authorization cannot be too complex, as it would prevent a user from understanding her decisions. At the same time, it cannot be too simplistic, as it may not sufficiently convey the weight of the privacy-related decisions it captures. It is a challenge in itself to build end-user assent systems that allow users to understand and adjust their privacy settings.

This gap between the interface – single click – and the effect can render data ownership meaningless; one click may wrench people and their data into systems and rules that are antithetical to fair information practices, as is prevalent with today's end-user licenses in cloud services or applications. Managing the long-term tensions fueled by 'old deal' systems operating simultaneously with the New Deal is an important design and migration challenge during the transition to a Big Data economy. During this transition and after the New Deal on Data is no longer new, personal data must continue to flow in order to be useful. Protecting the data of people outside of directly user-controlled domains is very hard without a combination of cost-effective and useful business practices, legal rules, and technical solutions.

We envision 'living informed consent', where the user is entitled to know what data is being collected about her by which entities, empowered to understand the implications of data sharing, and finally put in charge of the sharing authorizations. We suggest that readers ask themselves a question: *Which services know which city I am in today?* Google? Apple? Twitter? Amazon? Facebook? Flickr? Some app I authorized a few years ago to access my Facebook check-ins and have since forgotten about? This is an example of a fundamental question related to user privacy and assent, and yet finding an accurate answer can be surprisingly difficult in today's ecosystem. We can hope that most services treat data responsibly and according to user authorizations. In the complex network of data flows, however, it is relatively easy for data to leak to careless or malicious services.³³ We need to build solutions that help users to make well-informed decisions about data sharing in this environment.

Big Data and Personal Data Institutional Controls

The concept of 'institutional controls' refers to safeguards and protections implemented through legal, policy, governance, and other measures that are not solely technical, engineering, or mechanical. Institutional controls in the context of big data can perhaps best be understood by examining how such controls have been applied to other domains, most prevalently in the field of environmental regulation. A good example of how this concept supports and reflects the goals and objectives of environmental regulation can be found in the policy documents of the Environmental Protection Agency (EPA), which gives the following definition in its Institutional Controls Glossary:

Institutional Controls – Non-engineering measures intended to affect human activities in such a way as to prevent or reduce exposure to hazardous substances. They are almost always used in conjunction with, or as a supplement to, other measures such as waste treatment or containment. There are four categories of institutional controls: governmental controls; proprietary controls; enforcement tools; and informational devices.³⁴

The concept of an ‘institutional control boundary’ is especially clarifying and powerful when applied to the networked and digital boundaries of an institution. In the context of Florida’s environmental regulation, the phrase is applied when a property owner’s risk management and clean-up responsibilities extend beyond the area defined by the physical property boundary. For example, a recent University of Florida report on clean-up target levels (CTLs) states, “in some rare situations, the institutional control boundary at which default CTLs must be met can extend beyond the site property boundary.”³⁵

When institutional controls apply to “separately owned neighboring properties” a number of possibilities arise that are very relevant to management of personal data across legal, business, and other systemic boundaries. Requiring the party responsible for site clean-up to use “best efforts” to attain agreement from the neighboring owners to institute the relevant institutional controls is perhaps the most direct and least prescriptive approach. When direct negotiated agreement is unsuccessful, then use of third-party neutrals to resolve disagreements regarding institutional controls can be required. If necessary, environmental regulation can force the acquisition of neighboring land by compelling the party responsible to purchase the other property or by purchase of the property directly by the EPA.³⁶

In the context of big data, institutional controls are seldom, if ever, imposed through government regulatory frameworks such as are seen in environmental waste management oversight by the EPA.³⁷ Rather, institutions applying measures constituting institutional controls in the big data and related information technology and enterprise architecture contexts will typically employ governance safeguards, business practices, legal contracts, technical security, reporting, and audit programs and various risk management measures.

Inevitably, institutional controls for big data will have to operate effectively across institutional boundaries, just as environmental waste management must sometimes be applied across real property boundaries and may subject multiple different owners to enforcement actions corresponding to the applicable controls. Short of government regulation, the use of system rules as a general model is one widely understood, accepted, and efficient method for defining, agreeing, and enforcing institutional and other controls across BLT domains of ownership, governance, and operation.

Following on from the World Economic Forum’s recommendation to treat personal data stores in the manner of bank accounts,³⁸ a number of infrastructure improvements need to be realized if the personal data ecosystem is to flourish and deliver new economic opportunities:

- *New global data provenance network* In order for personal data stores to be treated like bank accounts, origin information regarding data items coming into the data store

must be maintained.³⁹ In other words, the provenance of all data items must be accounted for by the IT infrastructure on which the personal data store operates. The databases must then be interconnected in order to provide a resilient, scalable platform for audit and accounting systems to track and reconcile the movement of personal data from different data stores.

- *Trust network for computational law* For trust to be established between parties who wish to exchange personal data, some degree of ‘computational law’ technology may have to be integrated into the design of personal data systems. This technology should not only verify terms of contracts (e.g. terms of data use) against user-defined policies but also have mechanisms built in to ensure non-repudiation of entities who have accepted these digital contracts. Efforts such as the UMA initiative are beginning to bring better evidentiary proof and enforceability of contracts into technical protocol flows.⁴⁰
- *Development of institutional controls for digital institutions* Currently, a number of proposals for the creation of virtual currencies (e.g. BitCoin,⁴¹ Ven⁴²) have underlying systems with the potential to evolve into self-governing ‘digital institutions’.⁴³ Such systems and the institutions that operate on them will necessitate the development of a new paradigm to understand aspects of institutional control within their context.

Scenarios of Use in Context

Developing frameworks for big data that effectively balance economic, legal, security, and other interests requires an understanding of the relevant context and applicable scenarios within which the data exists.

A sound starting point from which to establish the applicable scenarios of use is to enumerate the institutions involved with a given set of big data, and develop a description of how or why they hold, access, or otherwise intermediate the data. Although big data straddles multiple BLT boundaries, one or more institutions are typically able to, or in some situations required to, manage and control the data. The public good referred to in the title of this book can be articulated as design requirements or even as certification criteria applicable to those institutions that operate the systems through which the big data is computed or flows.

It may be also be necessary to narrowly define certain aspects of the scenario in which the data exist in order to establish the basic ownership, control, and other expectations of the key parties. For example, describing a transaction as a financial exchange may not provide enough relevant detail to reveal the rights, obligations, or other outcomes reasonably expected by the individuals and organizations involved. The sale of used cars via an app, the conduct of a counseling session via Google Hangout, and the earning of a master’s degree via an online university all represent scenarios in which

the use case of a financial exchange takes place. However, each of these scenarios occurs in a context that is easily identifiable: the sale of goods and deeper access to financial information if the car is financed; the practice of therapy by a licensed professional accessing and creating confidential mental health data; or e-learning services and protected educational records and possibly deeper financial information if the program is funded by scholarship or loans. The scenarios can also identify the key elements necessary to establish existing consumer rights – the people (a consumer and a used car dealer), the transaction (purchase of a used car), the data (sales and title data, finance information, etc.), and the systems (the third-party app and its relevant services or functions, state DMV services, credit card and bank services, etc.). The rights established by relevant state lemon laws, the Uniform Commercial Code, and other applicable rules will determine when duties arise or are terminated, what must be promised, what can be repudiated, by whom data must be kept secure, and other requirements or constraints on the use of personal data and big data. These and other factors differ when a transaction that seems identical operates within a different scenario, and even scenarios will differ depending on which contexts apply. The following four elements are critical for defining high-level goals and objectives:

1. Who are the *people* in the scenario (e.g. who are the parties involved and what are their respective roles and relationships)?
2. What are the relevant *interactions* (e.g. what transactions or other actions are conducted by or with the people involved)?
3. What are the relevant *data* and datasets (e.g. what types of data are created, stored, computed, transmitted, modified, or deleted)?
4. What are the relevant *systems* (e.g. what services or other software are used by the people, for the transactions, or with the data)?

Inspired by common law, the New Deal on Data sets out general principles of ownership that both guide and inform basic relationships and expectations. However, the dynamic bundle of recombinant rights and responsibilities constituting ‘ownership’ interests in personal data and expectations pertaining to big data vary significantly from context to context, and even from one scenario to another within a given general context. Institutional controls and other system safeguards are important methods to ensure that there are context-appropriate outcomes that are consistent with clearly applicable system scenarios as well as the contours and foundations for a greater public good. The New Deal on Data can be achieved in part by sets of institutional controls involving governance, business, legal, and technical aspects of big data and interoperating systems. Reference scenarios can be used to reveal signature features of the New Deal on Data in

various contexts and can serve as anchors in evaluating what institutional controls are well aligned to achieve a balance of economic, privacy, and other interests.

The types of requirements and rules governing participation by individuals and organizations in trust networks vary depending on the facts and circumstances of the transactions, data types, relevant roles of people, and other factors. Antecedent but relevant networks such as credit card systems, trading partner systems, and exchange networks are instructive not only for their many common elements but also as important examples of how vastly different they are from one another in their contexts, scenarios, legal obligations, business models, technical processes, and other signature patterns. Trust networks that are formed to help manage big data in ways that appropriately respect personal data rights and other broader interests will similarly succeed to the extent they can tolerate or promote a wide degree of heterogeneity among participants for BLT matters that need not be uniform or directly harmonized. In some situations, new business models and contexts will emerge that require fresh thinking and novel combinations of roles or types of relationships among transacting parties. In these cases, understanding the actual context and scenarios is critical in customizing acceptable and sustainable BLT rules and systems. Example scenarios can describe deeper fact-based situations and circumstances in the context of social science research involving personal data and big data.⁴⁴ The roles of people, their interactions, the use of data, and the design of the corresponding systems reflect and support the New Deal on Data in ways that deliberately provide greater immediate value to stakeholders than is typically expected.

The New Deal on Data is designed to provide good value to anyone creating, using, or benefiting from personal data, but the vision need not be adopted in its entirety before its value becomes apparent. Its principles can be adopted on a large scale in increments – an economic sector, transaction type, or data type at a time. Adopting the New Deal on Data in successive phases helps to address typical objections to change based on cost, disruption, or overregulation. Policy incentives can further address these objections, for example by allowing safe harbor protections for organizations operating under the rules of a trust network.

Predesigned use cases can provide benchmarks for determining whether given uses of personal data are consistent with measurable criteria. Such criteria can be used to establish compliance with the rules of a trust network and for certification by government for the right to safe harbor or other protections. Because the New Deal on Data is rooted in common law and the social compact, the appropriate set of rights and expectations covering privacy and other personal data interests can be enumerated, debated, and agreed upon in ways that fit the given use cases.

Conclusions

Society today faces unprecedented challenges and meeting them will require access to personal data, so we can understand how society works, how we move around, what makes us productive, and how everything from ideas to diseases spread. The insights must be actionable and available in real time, thus engaging the population, creating the nervous system of the society. In this chapter we have reviewed how big data collected in institutional contexts can be used for the public good. In many cases, although the data needed to create a better society has already been collected, it sits in the closed silos of companies and governments. We have described how the silos can be opened using well-designed and carefully implemented sets of institutional controls, covering business, legal, and technical dimensions. The framework for doing this – the New Deal on Data – postulates that the primary driver of change must be recognizing that ownership of personal data rests with the people that data is about. This ownership – the right to use, transfer, and remove the data – ensures that the data is available for the public good, while at the same time protecting the privacy of citizens.

The New Deal on Data is still new. We have described here our efforts to understand the technical means of its implementation, the legal framework around it, its business ramifications, and the direct value of the greater access to data that it enables. It is clear that companies must play the major role in implementing the New Deal, incentivized by business opportunities, guided by legislation, and pressured by demands from users. Only with such orchestration will it be possible to modernize the current system of data ownership and put immense quantities and capabilities of collected personal data to good use.

Notes

¹ Jonathan Woetzel et al., “Preparing for China’s Urban Billion” (McKinsey Global Institute, March 2009), http://www.mckinsey.com/insights/urbanization/preparing_for_urban_billion_in_china.

² David Lazer, Alex Sandy Pentland, Lada Adamic, Sinan Aral, Albert Laszlo Barabasi, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, and Myron Gutmann, “Life in the Network: The Coming Age of Computational Social Science,” *Science* 323 (2009): 721–723.

³ Sinan Aral and Dylan Walker, “Identifying Influential And Susceptible Members Of Social Networks,” *Science* 337 (2012): 337–341; Alan Mislove, Sune Lehmann, Yong-Yeol Ahn, Jukka-Pekka Onnela, and J. Niels Rosenquist, *Pulse of the Nation: U.S. Mood throughout the Day Inferred from Twitter* (website), <http://www.ccs.neu.edu/home/amislove/twittermood/> (accessed November 22, 2013); Jessica Vitak, Paul Zube, Andrew Smock, Caleb T. Carr, Nicole Ellison, and Cliff

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Lampe, "It's Complicated: Facebook Users' Political Participation in the 2008 Election," *Cyberpsychology, Behavior, and Social Networking* 14 (2011): 107–114.

⁴ Alexis Madrigal, "Dark Social: We Have the Whole History of the Web Wrong," *The Atlantic*, October 12, 2013, <http://www.theatlantic.com/technology/archive/2012/10/dark-social-we-have-the-whole-history-of-the-web-wrong/263523/>.

⁵ Nathan Eagle and Alex Pentland, "Reality Mining: Sensing Complex Social Systems," *Personal and Ubiquitous Computing* 10 (2006): 255–268; Alex Pentland, "Reality Mining of Mobile Communications: Toward a New Deal on Data," *The Global Information Technology Report 2008-2009* (Geneva: World Economic Forum, 2009), 75–80.

⁶ Alex Pentland, David Lazer, Devon Brewer, and Tracy Heibeck, "Using Reality Mining to Improve Public Health and Medicine," *Studies in Health Technology and Informatics* 149 (2009): 93–102.

⁷ Vivek K. Singh, Laura Freeman, Bruno Lepri, and Alex Sandy Pentland, "Classifying Spending Behavior using Socio-Mobile Data," *HUMAN* 2 (2013): 99–111.

⁸ Wei Pan, Yaniv Altshuler, and Alex Sandy Pentland, "Decoding Social Influence and the Wisdom of the Crowd in Financial Trading Network," in *2012 International Conference on Privacy, Security, Risk and Trust (PASSAT), and 2012 International Conference on Social Computing (SocialCom)*, 203–209.

⁹ Kate Greene, "Reality Mining," *MIT Technology Review*, March/April 2008, <http://pubs.media.mit.edu/pubs/papers/tr10pdfdownload.pdf>.

¹⁰ Meglena Kuneva, European Consumer Commissioner, "Keynote Speech," in *Roundtable on Online Data Collection, Targeting and Profiling*, March 31, 2009, http://europa.eu/rapid/press-release_SPEECH-09-156_en.htm.

¹¹ Kim Gittleson, "How Big Data Is Changing The Cost Of Insurance," *BBC News*, November 14, 2013, <http://www.bbc.co.uk/news/business-24941415>.

¹² Aniko Hannak, Piotr Sapiezynski, Kakhki Arash Molavi, Balachander Krishnamurthy, David Lazer, Alan Mislove, and Christo Wilson, "Measuring Personalization of Web Search," in *Proc. 22nd International Conference on World Wide Web (WWW 2013)*, 527–538.

¹³ Pentland, "Reality Mining of Mobile Communications."

¹⁴ Anmol Madan, Manuel Cebrian, David Lazer, and Alex Pentland, "Social Sensing for Epidemiological Behavior Change," in *Proc. 12th ACM International Conference on*

Ubiquitous Computing (Ubicomp 2010), 291–300; Pentland et al. “Using Reality Mining to Improve Public Health and Medicine.”

¹⁵ Wei Pan, Gourab Ghoshal, Coco Krumme, Manuel Cebrian, and Alex Pentland, “Urban Characteristics Attributable to Density-Driven Tie Formation,” *Nature Communications* 4 (2013): article 1961.

¹⁶ Lev Grossman, “Iran Protests: Twitter, the Medium of the Movement,” *Time Magazine*, June 17, 2009; Ellen Barry, “Protests in Moldova Explode, with Help of Twitter,” *The New York Times*, April 8, 2009.

¹⁷ “Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of Such Data,” *Official Journal* L281 (November 23, 1995): 31–50.

¹⁸ World Economic Forum, “Personal Data: The Emergence of a New Asset Class,” January 2011, <http://www.weforum.org/reports/personal-data-emergence-new-asset-class>

¹⁹ Ibid.

²⁰ Ibid.

²¹ Antonio Lima, Manlio De Domenico, Veljko Pejovic, and Mirco Musolesi, “Exploiting Cellular Data for Disease Containment and Information Campaign Strategies in Country-Wide Epidemics,” School of Computer Science Technical Report CSR-13-01, University of Birmingham, May 2013.

²² Arvind Narayanan and Vitaly Shmatikov, “Robust De-Anonymization of Large Sparse Datasets,” in *Proc. 2008 IEEE Symposium on Security and Privacy (SP)*, 111–125.

²³ Latanya Sweeney, “Simple Demographics Often Identify People Uniquely,” Data Privacy Working Paper 3, Carnegie Mellon University, Pittsburgh, 2000.

²⁴ de Montjoye, Yves-Alexandre, Samuel S. Wang, Alex Pentland, “On the Trusted Use of Large-Scale Personal Data,” *IEEE Data Engineering Bulletin* 35, no. 4 (2012): 5–8.

²⁵ Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-Laszlo Barabasi, “Limits of Predictability in Human Mobility,” *Science* 327 (2010): 1018–1021.

²⁶ Pentland et al., “Using Reality Mining to Improve Public Health and Medicine.”

²⁷ David Tacconi, Oscar Mayora, Paul Lukowicz, Bert Arnrich, Cornelia Setz, Gerhard Troster, and Christian Haring, “Activity and Emotion Recognition to Support Early

Diagnosis of Psychiatric Diseases,” in *Proc. 2nd International ICST Conference on Pervasive Computing Technologies for Healthcare*, 100–102.

²⁸ World Economic Forum, “Personal Data.”

²⁹ The White House, “National Strategy for Trusted Identities in Cyberspace: Enhancing Online Choice, Efficiency, Security, and Privacy,” Washington, DC, April 2011, http://www.whitehouse.gov/sites/default/files/rss_viewer/NSTICstrategy_041511.pdf.

³⁰ Thomas Hardjono, “User-Managed Access UMA Profile of OAuth2.0,” Internet draft, 2013, <http://docs.kantarainitiative.org/uma/draft-uma-core.html>.

³¹ A Creative Commons licensed example set of integrated business and technical system rules for the institutional use of personal data stores is available at <https://github.com/HumanDynamics/SystemRules>.

³² See <http://openPDS.media.mit.edu> for project information and <https://github.com/HumanDynamics/openPDS> for the open source code.

³³ Nick Bilton, “Girls around Me: An App Takes Creepy to a New Level,” *The New York Times, Bits* (blog), March 30, 2012, <http://bits.blogs.nytimes.com/2012/03/30/girls-around-me-ios-app-takes-creepy-to-a-new-level>.

³⁴ U.S. Environmental Protection Agency, RCRA Corrective Action Program, “Institutional Controls Glossary,” Washington, DC, 2007, <http://www.epa.gov/epawaste/hazard/correctiveaction/resources/guidance/ics/glossary1.pdf>.

³⁵ University of Florida, Center for Environmental & Human Toxicology, “Development of Cleanup Target Levels (CTLs) for Chapter 62-777, F.A.C.,” Technical report, Florida Department of Environmental Protection, Division of Waste Management, February 2005, [http://www.dep.state.fl.us/waste/quick_topics/publications/wc/FinalGuidanceDocumentsFlowCharts_April2005/TechnicalReport2FinalFeb2005\(Final3-28-05\).pdf](http://www.dep.state.fl.us/waste/quick_topics/publications/wc/FinalGuidanceDocumentsFlowCharts_April2005/TechnicalReport2FinalFeb2005(Final3-28-05).pdf).

³⁶ U.S. Environmental Protection Agency, “Institutional Controls: A Guide to Planning, Implementing, Maintaining, and Enforcing Institutional Controls at Contaminated Sites,” OSWER 9355.0-89, Washington, DC, December 2012, <http://www.epa.gov/superfund/policy/ic/guide/Final%20PIME%20Guidance%20December%202012.pdf>.

³⁷ Ralph A. DeMeo and Sarah Meyer Doar, “Restrictive Covenants as Institutional Controls for Remediated Sites: Worth the Effort?” *The Florida Bar Journal* 85, no. 2 (February 2011); Florida Department of Environmental Protection, Division of Waste

Management, “Institutional Controls Procedures Guidance,” Tallahassee, June 2012, http://www.dep.state.fl.us/waste/quick_topics/publications/wc/csf/icpg.pdf; University of Florida, “Development of Cleanup Target Levels.”

³⁸ World Economic Forum, “Personal Data.”

³⁹ Thomas Hardjono, Daniel Greenwood, and Alex Pentland, “Towards a Trustworthy Digital Infrastructure for Core Identities and Personal Data Stores,” in *Proc. ID360 Conference on Identity*, 2013.

⁴⁰ Hardjono, “User-Managed Access UMA Profile of OAuth2.0”; Eve Maler and Thomas Hardjono, “Binding Obligations on User-Managed Access (UMA) Participants,” Internet draft, 2013, <http://docs.kantarainitiative.org/uma/draft-uma-trust.html>.

⁴¹ Simon Barber, Xavier Boyen, Elaine Shi, and Ersin Uzun, “Bitter to Better – How to Make Bitcoin a Better Currency,” in *Proc. Financial Cryptography and Data Security Conference (2012)*, LNCS 7397, 399–414.

⁴² Stan Stalnaker, “About [Ven Currency],” <http://www.ven.vc> (accessed January 16, 2014).

⁴³ Thomas Hardjono, Patrick Deegan, and John Clippinger, “On the Design of Trustworthy Compute Frameworks for Self-Organizing Digital Institutions,” in *Proc. 16th International Conference on Human-Computer Interaction (2014)*, forthcoming; Lazer et al., “Life in the Network.”

⁴⁴ See e.g. the study SensibleDTU (<https://www.sensible.dtu.dk/?lang=en>). This study of 1,000 freshman students at the Technical University of Denmark gives students mobile phones in order to study their networks and social behavior during an important change in their lives. It uses not only data collected from the mobile phones (such as location, Bluetooth-based proximity, and call and sms logs), but also from social networks and questionnaires filled out by participants.

Chapter 10

Engineered Controls for Dealing with Big Data

Carl Landwehr

Introduction

It is one thing for a patient to trust a physician with a handwritten record that is expected to stay in the doctor's office. It's quite another for the patient to consent to place their comprehensive electronic health record in a repository that may be open to researchers anywhere on the planet. The potentially great payoffs from (for example) being able to find a set of similar patients who have suffered from the same condition as oneself and to review their treatment choices and outcomes will likely be unavailable unless people can be persuaded that their individual data will be handled properly in such a system.

Agreeing on an effective set of institutional controls (see Chapter 9) is an essential prerequisite, but equally important is the question of whether the agreed upon policies can be enforced by controls engineered into the system. Without sound technical enforcement, incidents of abuse, misuse, theft of data, and even invalid scientific conclusions based on undetectably altered data can be expected. While technical controls can limit the occurrence of such incidents substantially, some will inevitably occur. When they do, the ability of the system to support accountability will be crucial, so that abusers can be properly identified and penalized and systems can be appropriately reinforced or amended.

Questions to ask about the engineered controls include:

- How are legitimate system users identified and authenticated?
- What mechanisms are employed to distinguish classes of users and to limit their actions to those authorized by the relevant policies?
- What mechanisms limit the authorities of system administrators?
- How is the system software installed, configured, and maintained? How are user and administrator actions logged?
- Can the logs be effectively monitored for policy violations?
- When policy violations are detected, what mechanisms can be used to identify violators and hold them to account?

Finally, the usability of the engineered controls – their impact on system users – must be considered. Time and again, users have demonstrated that they will find ways around controls that they see as needlessly complex or strict and that impede them from their primary goals in using the system.

Threats to Big Data: Accidental, Malicious

Threats to ‘big data’ sets can come from several directions. Not least of these are the threats of accidental damage or loss, for example from device failure, physical loss of a mobile device (laptop, tablet, memory stick), physical damage to a device from fire, flood, or simply dropping it, if it’s small enough to carry.

The general motives for intentional compromise of datasets or systems are relatively few, but they can be manifested in many ways. Financial gain is likely to be the strongest motive for compromise; desire for fame has motivated some groups and individuals from the early days of the Internet; desire for revenge (against an employer or a lover, for example, but also against a government organization or a research colleague) can be a strong motivator; and finally simple curiosity can lead to attempts to defeat security and confidentiality controls.

The nature of the data, and what motivated individuals might glean from them, is significant in assessing threat. For example, a large set of medical records from which obviously identifying information has been removed might not seem to be of much interest to those beyond the research community. Yet if the records concerned a sexually transmitted disease or drug addiction, and if a celebrity suspected of having that condition was treated at a facility that contributed to the dataset, someone might exert considerable effort to re-identify the data.

The custodian needs to consider carefully both the value of the raw data and what kinds of inferences might be drawn from them by users or abusers. This exercise will help identify what might motivate particular individuals or groups to attack the system. Chapters 1–5 in this volume provide considerable guidance about legal and policy issues; Chapters 9 and 11 identify mechanisms that may facilitate research access to the data; Chapters 13 and 14 identify ways to limit unwanted disclosures. The custodian needs to pay even closer attention to those with administrative access to datasets, because they will likely have access that is unrestricted, or much less restricted, than the researchers to whom data may be released under particular controls. If they have the ability to disable or corrupt logs, they can also damage accountability.

Vulnerabilities: Accidental, Engineered

Nearly all of today’s cyberattacks take advantage of latent vulnerabilities in software that has been legitimately installed on the user’s system. Although it is possible to write rigorous specifications¹ for software and even to prove that an implementation conforms

to those specifications, this is a costly process and one that is infeasible to apply to large and complex software systems. Commercially produced software, even software that has undergone standard testing regimes and is considered of good quality, will typically include a few bugs per thousand lines of code, at minimum. Only a fraction of these will be exploitable security vulnerabilities, but with the total size of codebases for both operating systems and applications in the range of hundreds of kilobytes to thousands of megabytes, it's unrealistic to think that there will not be exploitable vulnerabilities in the platforms used to store and process large datasets. A typical exploit will take advantage of an accidental flaw in the software to gain a foothold and then download a larger payload that provides the infrastructure attackers use to achieve whatever objectives they may have – for example, altering sensitive data, sending sensitive data back to a place where the attacker can retrieve it, or denying access to the data to legitimate users.

It is also possible for vulnerabilities to be engineered into operating systems, compilers, network support, and applications by those responsible for developing or maintaining them. For example, a developer may decide to leave a ‘back door’ into a software component that can later be used to perform remote testing or maintenance. Such an engineered-in ‘feature’ can be exploited by others if they learn of its existence.

A more subtle form of vulnerability that may enable the illicit transmission of information is referred to as a ‘covert channel’, or more commonly today as a ‘side channel’. When, as a side effect of a computation, some shared resource (such as a file lock) is manipulated, and that manipulation is visible outside of the immediate context, that manipulation may be used to transmit information. For example, the power consumed by a chip as it computes a cryptographic algorithm, if monitored, may expose the key being used in the computation (Kocher et al. 1999). While these channels have rarely been used to compromise large datasets, it is nearly impossible to build a system entirely free of them.

Today’s software is also largely built from components. An application programmer will naturally look for libraries and components that will reduce the new code that must be developed. Whatever code is incorporated in this way brings with it whatever vulnerabilities have been accidentally left in it or purposely engineered into it. Further, flaws may arise in the interactions of components that individually behave as intended.

The question of whether ‘open source’ or ‘closed source’ software is more or less likely to contain exploitable vulnerabilities has been hotly debated over the years. Having the source code available for anyone to inspect and analyze seems to be an advantage, but only if people with the appropriate expertise and access to the appropriate tools actually do the inspection and analysis. Evidence to date is that qualified people rarely carry out such tasks without being paid.

When the source is closed, typically some commercial entity has a financial interest in it. The owner of the software has incentives to assure its quality (including freedom

from bugs and vulnerabilities) because if others find those bugs and vulnerabilities after the software is released, the owner may still have to fix them and the product's reputation will suffer. Nevertheless, experience shows that owners, even when informed of security issues in their products, do not necessarily act promptly to address them. Moreover, software owners *could* hide all sorts of things in delivered software should they wish to. Of course, if malicious software were found in a commercial product that seemed to have been placed there purposely by the developer, the company might very well lose its customers in short order, so there would be a strong incentive to make any such hidden features appear to be accidentally introduced.

Strategies

Given that there will be vulnerabilities and there may be threats against them, what is the right strategy for dealing with them? We will discuss some mechanisms for implementing strategies in subsequent sections, but what is the right overall approach?

The Canadian Conservation Institute, advising on controlling biological infestations in a museum environment, identifies an approach with five stages: Avoid, Block, Detect, Respond, Recover/Treat (Strang 1996). Approaches developed for dealing with cyber 'infestations' have much in common:

- Treasury Board of Canada (TBC 2006): Prevention, Detection, Response, and Recovery
- Microsoft (Microsoft 2010): Protect, Detect, Respond, Recover
- NIST (NIST 2013): Know, Prevent, Detect, Respond, Recover

The addition of 'know' in the NIST framework emphasizes that one must first have an inventory of what is to be protected. This requirement seems particularly germane for a big data environment. Depending on the policies to be enforced, what is to be protected may include not only the datasets themselves but also the use made of them – the sets of queries posed and the responses provided, for example. The identity of the querier may also need to be either publicly recorded or protected from view, according to policy.

If policy violations could be prevented with certainty through technical controls, the need for detection, response, and recovery would be greatly reduced, but unfortunately today they can't be. Detection of intentional breaches of computer system security is quite challenging. Usually the perpetrator is not interested in publicizing the breach, and data can be copied and removed while leaving the original completely unaltered – the money seems still to be in the vault, so to speak. Verizon reported that 70% of disclosed data breach incidents reported in 2012 were discovered by third parties, not by the custodians of the data (Verizon 2013, 53). Detection often happens only when the stolen

material is used in some public way, to purchase goods or to generate publicity, for example, which may be considerably after the actual theft.

Response often means conducting a post mortem to discover exactly how the data were stolen (or in the case of an integrity attack, how they were modified) and addressing the problems exposed in order to avoid a recurrence. A key aspect of response is to establish accountability for the incident. This may lead to prosecution of the perpetrator but also may affect personnel charged with maintaining the confidentiality of the data sets. Mechanisms for establishing accountability are crucial to deterring both negligence and malice.

Recovery involves restoring the integrity of the system following an incident. The most basic technical support for recovery is maintenance of backup copies of both data and software used to process it. The recovery process must assure the integrity of the restored backup copy. Cryptographic hashes, digital signatures, and public reference copies may be used for this purpose. The complexity of the recovery process depends on the subtlety and sophistication of the attack. If data are stolen, they may simply have been copied and there is no need to recover the data *per se* at all. But if they were stolen through a piece of software maliciously inserted into the system (malware), that malware needs to be located and removed. If the malware is present in backup copies of the software, restoring the system to a previous state may not in fact eliminate the source of the breach, so care must be taken in this process.

Technical Controls

In this section, we review the kinds of technical controls available to implement policies controlling access to data and for providing accountability when access is granted. We consider the role of encryption in these controls as well as currently available mechanisms to assure software integrity.

Identification and Authentication

As we have already discussed, maintaining accountability for the use of a dataset is one of the strongest tools we have to assure the subjects of the data that their information will not be abused. To maintain accountability, we need to be able to link human beings with activities that take place in the computer, and that linkage starts with having users identify themselves and authenticating that the claimed identity is correct. The simplest and most commonly used form of this technology, and one of the most easily abused, is the user ID (for identification) and password (for authentication). Some recent research documents in some detail how people really use passwords and introduces a metric for the difficulty of guessing passwords (Bonneau 2012). Despite the desires of many, it seems unlikely that passwords will disappear soon, so it's important for security administrators to realize the implications of imposing constraints on users' choices of

passwords. In general, it appears better to allow (and encourage) users to choose relatively long passwords and *not* to put constraints on the content (e.g., at least one digit, at least one special character, no words in the dictionary) instead of trying to increase the entropy of relatively short passwords (Weir et al. 2010).

More promising is the increasing use of two-factor authentication schemes, in which the user enters not only a password (something the user *knows*) but also something to indicate that he possesses some physical token (something the user *has*) as well. For many years, the prevalent way of achieving two-factor authentication was to employ a commercially provided token that would generate a new short pseudo-random challenge number every minute or so. The user's system would track which token was assigned to a particular user and could compute what challenge would appear on that token during a specific time interval. The cost of such a scheme included both paying for the commercial token and associated server-side software plus the cost to the user (in time, space, and weight) of carrying and using the token. More recently, several schemes have been developed that take advantage of the fact that most users today carry mobile phones. In this case, the system merely needs to know the user's mobile phone number and it can transmit (via audio or text message) a challenge number over that medium. The user then keys that number into the system, proving that whoever has typed in the initial user ID and password also has access to the user's mobile phone.

Biometrics can provide another form of two-factor authentication. A biometric (fingerprint, iris pattern, or facial image) is generally hard (though not necessarily impossible) to spoof, but of course a camera or other device must be available to read the biometric and the reader needs a trustworthy communication path to the biometric authentication database. Recording the observed biometric can also improve accountability.

Once the user is authenticated, there is also the question of whether another user may take over for her – the professor logs in and the graduate student takes over for her, for example. There are schemes for *continuous authentication* designed to deal with this sort of problem, but in general they are either irritating (requiring the user to re-authenticate periodically) or they require some additional mechanism such as an active token. Biometrics that a user provides as a side effect of using the system (typing characteristics, facial images, gait) are the subject of research; success could provide more usable continuous authentication.

Also relevant to this discussion are the continuing efforts to develop federated identity systems. The idea is to allow users to establish strong authentication with one of several identity providers and then have that identity forwarded to others as needed, avoiding re-authentication and providing a 'single sign-on' capability. Efforts include the earlier Liberty Alliance, and current Shibboleth and OpenID (Birrell and Schneider 2013). It seems likely that some form of federated identity management will be widespread

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

within a few years. Risks of federation include the possibility that a weak or sabotaged provider may forward incorrect authentication data or it may be unavailable when needed.

Access Control and Information Flow Policies

Without a security or confidentiality policy, there can be no violations, only surprises. The chapters in Part I of this volume address policy considerations from many vantage points. This section is concerned with the tools available to state and enforce policies mechanically within computer systems.

Access control policies are generally stated in terms of *subjects*, *objects*, and *access modes*. In the physical world, subjects are people and objects may be the records kept in an office environment, and there may be rules about which individuals, or which job functions, are allowed to read and update various documents or fields within documents. In a computer system, a subject is typically a process, which is a program in execution in the computer, and the subject is identified with a user who has been authenticated.

Objects are typically files or other containers of information that a process may need to access in order to complete its computations, and the access mode may be read, write, or execute, so a specific subject may be authorized to read, write, or execute a particular object. The set of all allowed triples (subject, object, access mode) defines a security policy for the system. Although no one ever writes a security policy that way, because it would be far too detailed and time consuming, people do write policies that determine what kinds of access different classes of users require (or that the programs invoked by users require). The *least privilege principle* states that a user (or a process) should only be granted access to those resources (files, programs) that it needs in order to do its job and no others. In practice, this principle is loosely observed, because managing very fine-grained access control is cumbersome. The general notion in an access control policy is that higher level security and confidentiality policies are enforced on the basis of which subjects are allowed to read, write, or execute which objects (Landwehr 1981).

An alternative, and perhaps more intuitive, way to express policies is in terms of *information flow*. An information flow policy specifies the allowed and prohibited flows among subjects and objects in a computer system. In an access control policy, if one wishes to assert that subject A should not be able to read the information in object B, it is insufficient simply to ban A from ever having permission to read B, since the data contained in B might be copied (by some other subject) into object C, which A might be able to read. Stating the constraint in terms of information flow restrictions is straightforward: information in object B is not allowed to flow to subject A, through any path.

Role-based Access Control Many applications have distinct sets of privileges associated with job functions. For example, a bank teller may require the ability to update

information in bank accounts, while a bank vice president might not require that ability but would require the ability to generate summary information across many accounts. To simplify management of sets of privileges, the notion of ‘role-based access controls’ was developed (Sandhu 1996; Landwehr et al. 1984).

Usage Control With the recognition that policies often limit the use of data to particular purposes, the notion of ‘usage controls’ is being explored (Park and Sandhu 2004). Restrictions on the use of collected data are frequently phrased in terms of the ‘purpose’ of the collection; the data may be used for some purpose or purposes, but not others. Deducing the purpose behind a particular access request is difficult, but some current research aims at the automated enforcement of such policies or the automated detection of potential violations of such policies (Tschantz et al. 2012).

Prevention: Policy Enforcement via Reference Monitors

One of the foundational papers in computer security developed the notion of a ‘reference monitor’ to enforce security policies (Anderson 1972). The reference monitor (RM) is a component that checks every reference made by a subject to an object to determine whether it is permitted under the current security policy. To assure policy enforcement, the RM must be correct, tamperproof, and non-bypassable; that is, the RM must be small and simple enough that the correctness of its enforcement is clear (‘correct’), it must not be possible to alter the RM (‘tamperproof’), and it must not be possible for subjects to make references that don’t go through the RM (‘non-bypassable’). The cost of placing a check on every reference that a computer makes to memory was well understood to be infeasible; rather the idea was that when, for example, a program processed a request to read or write a new file, the RM would be invoked to check that the request was consistent with the current set of permissions. If so, the file would be opened and the program would be free to read (or write, or execute, or some combination of these) without further interference. Even so, commercial systems were not generally built around this concept, and attempts to build RM-based systems generally did run into performance and other technical issues. Nevertheless the concept of a reference monitor as an idealization of enforcement mechanisms remains useful. In practice, for reasons of efficiency and modular extensibility, security checks tend to be decentralized, resulting in a distributed RM. Although distribution of checks makes it more difficult to assure that the RM has the three properties required, it has been shown that the same classes of policies enforceable by centralized RMs can be enforced by RMs that make checks ‘in-line’, that is, distributed throughout the system (Schneider 2000).

Enforcement of information flow policies has taken more time to mature; it turns out that RMs can’t necessarily enforce such policies directly, because whether information flows from a variable x to a variable y can depend not only on the current execution path

but on the set of alternative execution paths *not* taken, which are not visible to the RM (for a detailed exposition, see McLean 1994; Schneider 2000). Compilers have been developed that can accept information flow policies as input and assure that only permitted flows will occur in the code they generate. To assure an information flow policy is maintained system-wide is a more substantial challenge since typically systems are composed of many components and applications, and assuring that the flow policies are properly enforced may require either clever architectures or substantial reconstruction of components. Research continues in this area.

RMs are designed to prevent violations of policy before they occur, and so they address the ‘prevent’ portion of an overall strategy for securing a computer system. Other measures are needed to deal with detection, response, and recovery.

Cryptography and Its Applications

Today’s cryptography can be used to protect both information in transit (communications) and information at rest (stored in files or databases, for example). Publicly available cryptographic algorithms can prevent even the strongest attacker from decrypting enciphered data unless the encryption key is compromised. Thus cryptography transforms the problem of protecting a large amount of data into the problem of protecting a much smaller amount of data – the encryption key.

Two fundamental kinds of cryptographic algorithms are in use today: *symmetric* algorithms, in which the same key is used for encryption and decryption, and *asymmetric* algorithms (also known as public key algorithms) in which one key is used for encryption and a different key is used for decryption. The development and gradual deployment of public key cryptography has been a major advance in the past 30 years, facilitating secure communications and electronic commerce on the Internet. Secure hash algorithms, which can generate a short encrypted digest of a long (or short) string of bits, provide a mechanism that enables a program to verify that a particular piece of data has not been altered. This mechanism is the basis for many useful functions such as digital signatures and assuring integrity and provenance of data.

As already noted, cryptography in itself offers a way to transform one kind of problem into another. While this can be very helpful, the problems remain of generating and managing the keys essential to the process and of implementing the cryptographic algorithms correctly. As the strength of publicly available cryptography has increased, the focus of attacks has moved to exploiting errors in keying, in the cryptographic protocols used to generate and communicate keys among separated parties, and in implementations of these protocols and algorithms. In fact, going back to World War II and earlier, errors in the use of the cryptographic algorithms, rather than weaknesses in the algorithms themselves, have been a common cause of significant compromises (Kahn 1996). This fact highlights again the more general issue of *usability* of security controls

in all sorts of contexts. Often the users of systems themselves will find the security measures designed into or added onto a system sufficiently inconvenient that they will find a way around them, whether it be writing down and sharing passwords or reusing the same cryptographic key for different messages. System designers ignore this lesson at the peril of the overall security of their systems.

Media Encryption Today, cryptography can help users of large datasets in a few specific ways. Sadly, it is not uncommon to see reports of lost laptops or other devices that, despite their small physical size, may hold very large volumes of data. If sensitive data are held on any such portable device, they should be encrypted in such a manner that they are not readable by whoever finds (or steals) the device. Many hard drives and even flash drives today provide built-in encryption processors so that the entire drive is automatically protected from simple theft. The user will be responsible for providing a key to the device in order to unlock it. Of course, in order to process the data, they must normally be decrypted, and even though the permanent files on the external hard drive or flash drive may be updated and re-encrypted, in most situations caches and temporary files may be created and remain on the host machine unless precautions are taken that those are encrypted as well.

Encryption for Fine-Grained Sharing Media encryption and file encryption provide for only relatively coarse-grained sharing of information. The development of attribute-based encryption (Goyal et al. 2006; Piretti et al. 2006) and more recently functional encryption (Boneh et al. 2011) enable much finer grained sharing policies to be enforced. In functional encryption, the data are encrypted once. A user with certain attributes will have an individual key that will enable her to learn a specific function of the encrypted data (which might be specific fields of a record, for example), but nothing else. Research is still advancing in this area, and attempts are underway to explore practical applications (Akinyele et al. 2011).

Computing on Encrypted Data What if you could encrypt data, perform any algorithm you wish on the encrypted data, and then decrypt the data in such a way that the decrypted data provided the results of the algorithm? In 2009, Gentry proved that this sequence is theoretically possible, though the scheme he used to demonstrate it would not be nearly efficient enough to be practical (Gentry 2009). If this scheme (called fully homomorphic encryption, FHE) were to become practical, it would enable many interesting ways of distributing trust – for example, the processor on which the results were computed would be unable to compromise any of the data used to compute the results, or the result itself. Cryptographers have been hard at work to find practical schemes to realize the promise of FHE. Functional encryption can be seen as one such

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

example: it limits the function computed in the encrypted domain in specific ways. DARPA's PROCEED (Programming Computation on Encrypted Data) program is pushing in precisely this direction (DARPA 2010) and IARPA's SPAR (Security and Privacy Assurance Research; IARPA 2011) program has been pursuing related goals in the context of database access.

Software Integrity

The analysis of big datasets will involve programs of many sorts. Nearly all of these will have been created by people other than the person doing the analysis. How can the analyst be sure that the program being executed is the one intended? Put another way, how can one be sure of the software configuration of one's computer as a whole?

One control engineered for this purpose creates what is known as a 'root of trust' in the machine. The idea is that, starting with the point at which the machine is powered up and the initial firmware begins to load the operating system, the integrity of each new layer of software loaded into the system should have its integrity checked before it starts to run. Checking integrity in this case amounts to checking a secure hash of the bit string corresponding to the software load of the particular program. This does nothing to assure that the software is correct or lacks exploitable vulnerabilities, but it does assure that the software hasn't been altered since the secure hash was computed (presumably by the developer). Once the software begins operation, exploitable vulnerabilities can still be triggered and cause undesired effects until the next time the checksums are recomputed and checked.

To facilitate this sort of checking, a computing industry consortium (the Trusted Computing Group) developed specifications for hardware, called a Trusted Platform Module (TPM) starting in the early 2000s, and the microprocessors at the heart of many computers now incorporate TPM functions. Although the initial versions of the technology would have required the entire software load to be checked, later versions allow a 'dynamic root of trust' to be established in a way that simplifies initiation of new application environments. This technology also provides a way for one platform to attest to other, remote platforms that it is operating with a specific, known software configuration. If one is using a remote computing resource (as in 'cloud computing') and desires to verify that the cloud environment is the one expected, this technology can help. Although widely deployed, this hardware is largely unused. Microsoft's BitLocker software is an exception.

Data Provenance

Knowing where the data in a dataset came from and how they have been transformed is critically important for scientific research. Engineered controls for assuring data provenance have gained increasing attention over the past decade (Buneman et al. 2001;

Muniswamy-Reddy et al. 2006), but what is in use seems primarily to be manual methods, overlaid on database management systems in some cases.

Provenance information has been characterized formally as an acyclic directed graph in which the leaves are data items and the nodes are other data and processes on which the present value of the data depends. For data that are updated frequently from a number of sources, the provenance graph could clearly become quite large.² On the other hand, this is exactly the information needed to resolve some scientific disputes.

This definition of provenance is closely related to what we discussed earlier as ‘information flow’. Information flow is a more comprehensive idea because it aims to record implicit flows of information (flow that occurs when a variable is left unchanged because of the state of a condition that might have caused it to change, but didn’t). Provenance deals only with explicit flows.

One might also observe that the TPM technology described above actually aims to establish (or perhaps to assure) the provenance of the software running on a particular platform at a particular time. Since the software is simply a bit string from the TPM’s perspective, it could just as well be a file or database, and so the same technology could be applied to assure that the program processes a specific, intended dataset.³ However, it would not help in tracing and recording changes made to the dataset in execution; it could at best record the final state by computing an encrypted checksum for the new version.

For some data, the provenance information itself might be sensitive – consider the provenance of a health record involving a sexually transmitted disease, for example. In this case, access controls will need to be applied to the provenance data. Conversely, some researchers have proposed using the provenance data as a security control (Ni et al. 2009; Martin et al. 2012).

The past few years have seen increasing efforts to investigate how provenance might be handled in a cloud computing context (Muniswamy-Reddy et al. 2010; Abbadi and Lyle 2012).

Detection and Recovery

Detecting when data have been incorrectly modified using an authorized mechanism, either accidentally or maliciously, requires logging; otherwise there is no way to detect the change. Further, the log must be available for review and it must *be* reviewed, and the log itself must be protected against accidental or malicious modification. This may sound like a recursive problem – if we can’t protect the data, how can we protect the log? But mechanisms can be applied to the log that would not work for data in general. A solution does require care in organizing the logging system and in administering the privileges of system users.

Key questions to be addressed in organizing a log or audit trail for a user of large datasets include the following.

- What events must be logged? For example, one might simply log the event that a researcher checks out a dataset and checks it back in later, or one might log every access to an online dataset. The latter approach may permit faster detection of problems (assuming the log is monitored) but will result in a great deal of low-level data to be archived, and using low-level events to reconstruct higher level actions that may reflect the behavior of an attacker is a significant effort in itself.
- How can we assure all relevant events are logged? In effect, a reference monitor for the data is needed; the log file records its actions.
- How is the initiator of the logged event identified? Accountability for actions is a key control in any system. In the heart of a computer system, a process (a program in execution) is typically identified as the initiator of a specific action. It must be possible to trace back from the identity of a process to a responsible human or accountability is lost.
- How is the log itself protected from corruption? Write-once optical media are still used in some specialized applications today, but it is much more common to send these records to one or more remote archive providers. The archive facilities evolve with technology for providing highly reliable long-term storage. An attacker, having penetrated a target system and wishing to cover her tracks, must either modify or disable local logging facilities, inhibit communications, or penetrate the archive sites as well.
- How is the log protected from compromise? The log itself is likely to contain sensitive data. For example, in a multiuser system it will incorporate actions from many different users, and no individual user may have the right to read log entries generated by others.
- Where will the log be kept? To avoid having a physical failure corrupt both the data and the log, they can be separated either on different devices in a system or the log data might be streamed to an offsite location.

Some researchers have proposed using query logs as a means for deriving user intent in accessing data as a way of enforcing purpose-based controls (Tschantz et al. 2012; Gunter et al. 2011). Access patterns present in the logs may be analyzed (using machine learning techniques, perhaps) to see whether they correspond to proper or improper usage patterns.

For static datasets, recovery from damage will simply be restoration of the original set. An offline backup will suffice in such simple cases. Where the dataset is dynamic, along with the logging processes just considered, and possibly integrated with them, there must be processes for creating backup files that can be restored. Changes made since the most recent backup may be recovered from the log files.

This is a preliminary version of the book Privacy, Big Data, and the Public Good: Frameworks for Engagement, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Very similar considerations apply to the backup files: they need to be kept in a protected location, access to them needs to be controlled, and they need to be protected from corruption and compromise. Frequently the most cost-effective way to compromise system data has been to steal a backup tape. Good practice dictates that all the backup tapes be encrypted and that the owner of the data, rather than the administrator of the backup facility, manage the keys.

Disposal

How long should data be kept? Some scientific data, such as experimental results obtained in the laboratory or observations of the natural world such as climate records, should be accessible for the indefinite future, and provenance is a key concern. Some filtered and abstracted datasets, such as released census records, fall in the same category. On the other hand, some confidential raw data have lifetimes specified by corporate or governmental policies. If data are collected or authorized for a specific use, it may be appropriate to delete them once that use has been made. What measures are available to be sure data are destroyed at the end of their intended life?

For data stored on computers, the main concern is more often losing data rather than failing to destroy it. Consequently, a great deal of attention may be paid to providing redundant backup copies, as discussed above. When one wishes to destroy the data, not only the primary copy, but also all of the backup copies, need to be dealt with. Further, as everyone should know by now, simply invoking the ‘delete’ function for a file doesn’t typically delete anything at all; it most likely simply links the file into a ‘deleted file’ data structure, from which it may be removed again if the user invokes an ‘undelete’ function. Even when the file is ‘permanently’ deleted, the storage space it occupied is unlikely to be overwritten until some other file is allocated that space and actually stores something in it. Further, hardware devices often have mechanisms to deal with hardware faults so that if a fault occurs in a particular region of storage, the data are copied to a new block and the faulty block is removed from regular use. But the data in the faulty block may still be readable through special interfaces, and if they could contain important information, such as a cryptographic key, someone may exert the effort needed to recover it.

In earlier days, most archives and backups were kept on magnetic tape and could be erased directly by exposing the storage media to sufficiently strong and varying magnetic fields. Traces of data left behind after some effort to erase it are known as *remanence*, and guidelines have been produced to prescribe, for example, how many times a particular medium must be randomly overwritten before all traces of remanence are removed. Today there is a much wider range of storage media available and many (optical storage on CD-ROMs, for example) are not magnetically erasable and indeed not easily erased by any means. Even if they become unreadable for normal use, data may

well be recovered from them if additional technical means are brought to bear. Computer hardware that turns up for resale has frequently turned out to include significant and readily available traces of the past owner's activities (Garfinkel 2013) and indeed computer forensic investigation frequently depends on such traces. Physical destruction of the media remains a very real approach to assuring data destruction.

The cleanest approach seems to be to encrypt all of the copies under strong cryptographic algorithms and to keep the keys separately from the data. In this case, erasing the keys used in the encryption is equivalent to erasing the records, at least until the cryptographic scheme is broken. Many enterprises reduce the risk of losing data stored on lost or stolen smartphones in this way. All data on the smartphone is stored encrypted. A master encryption key is stored on the phone as well, but the phone will respond to a properly encoded 'remote wipe' command by zeroing the key, thereby effectively erasing all the data on the phone without requiring that the data be rewritten at all, or that the authorized user physically possess the phone. Disk drives and flash drives are available that support bulk encryption of all data on the drive; some of these provide a means to destroy or remove the key when the device is recycled. But this approach comes with the (real but decreasing) computational cost of encryption and decryption and the complexity of managing the keys.

NIST has produced useful guidelines on media sanitization that go into much more detail than is possible here; according to this guidance, overwriting a record stored on a magnetic hard drive one time is usually sufficient to render it unreadable without extraordinary effort, but this of course does not deal with the failed blocks problem. A draft revision (NIST 2012) incorporates discussions of 'cryptographic erase' and new types of media.

Future challenges for big data users in this area will probably come from their use of a multiplicity of devices, including mobile devices, to access data that they may later wish to expunge, and through the use of cloud resources to store and process datasets.

Big Data and the Cloud

The notion of a computing utility – first described by the creators of Multics in the mid-1960s (Corbató and Vyssotsky 1965) and in particular its goals of convenient remote access, continuous operation, and flexible expansion or contraction of resources – is being realized today in the 'cloud'⁴ computing that Amazon, Microsoft, Apple, Google, Rackspace, and others now provide. The technology of virtualization of resources underlies cloud computing services. Each cloud user can be provided an environment that seems to be a dedicated, real computer, remotely accessed. In fact, this environment is simulated by a virtual machine; many virtual machines share very large pools of processors and storage that are managed by the cloud provider. Thus what appears a

dedicated and modularly expandable resource to the cloud user is really a highly shared resource.

Researchers of big data might use cloud resources in several ways: for example, as archival storage for data and analytic results, as a source of computing horsepower to conduct analyses with software developed by the researcher, or as a source of application software services to analyze datasets stored in the cloud or elsewhere. Some threats require additional attention in public cloud environments: users are forced to rely on networked communications, they must depend on the competence of remote personnel acting for different management and possibly other alternative legal constraints, and they share resources with other tenants of unknown trustworthiness or intent. A more comprehensive list of cloud threat classes can be found in the Defense Science Board's report on the digital cloud (DSB 2013).

The ease of use that makes cloud computing attractive can mask some concerns that researchers need to consider. Issues to consider include the following.

For computing, how strong are the walls between different clients? In general, cloud providers provide 'multitenancy' – serving a large number of clients on exactly the same pieces of hardware – rather than 'sole occupancy'. The separation of different tenants is provided by the software (typically known as a hypervisor) that multiplexes the hardware. Though some attacks have been demonstrated that enable a user to break the hypervisor's separation (King et al. 2006) and others have shown that it's possible for data to be leaked from one user to another through relatively sophisticated signaling schemes using side channels (Wu et al. 2012), the risks a typical researcher takes in adopting a cloud service rather than, say, a shared, large-scale computing system at a research university do not seem to be great. In fact, universities are frequently turning to cloud providers to operate their e-mail and sometimes other computing resources.

For archival storage: where geographically do the data reside? Cloud providers typically have facilities at a variety of locations, including in different countries, both for reliability and availability, so that if one location has a problem, other locations can take over, and for economic reasons, if power is less expensive in one place than another. Policy may forbid that some data from crossing international borders or traversing certain networks. Some cloud providers may have service agreements that recognize such limitations, but *caveat emptor*.⁵

If a client uses the cloud just for storage, it may encrypt all the data before sending them to the provider and then the provider will be unable to compromise or undetectably alter them. However, the customer will also be unable to use the computing resources of the cloud provider to analyze the data unless they are first decrypted (unless computing with encrypted data, mentioned earlier, becomes a reality, in which case the cloud computing resource could be exploited without increasing the trust required in the provider).

Cloud vendors also provide services for ‘private clouds’ to address clients’ concerns about multitenancy, among other things. For example, Rackspace markets a private cloud service in which hardware infrastructure is dedicated to a client. Amazon’s Virtual Private Cloud (VPC) provides a ‘logically isolated’ portion of Amazon Web Services (AWS) and can be augmented with a Hardware Security Module (HSM) on which the client can store sensitive keys to be used to protect data stored on his AWS VPC. The HSM communicates over a Virtual Private Network (VPN) to the VPC; Amazon has no access to the client’s keys. These augmented services frequently add cost.

One of the major cloud applications for analyzing big data is Hadoop, an open source implementation based on Google’s MapReduce software (Vavilapalli et al. 2013). Hadoop uses HBase, modeled on Google’s BigTable storage system. These systems enable massive computing resources to be brought to bear on problems involving very large datasets, but they generally lack access control on the datasets. The Accumulo system, originally developed at the U.S. National Security Agency on top of Hadoop and subsequently open-sourced, provides a distributed key/value store at large scale but also incorporates cell-level access controls, lacking in HBase and BigTable, so that a computation can be limited to data for which the process has the correct authorizations (Accumulo 2013).

Cloud service providers are likely to see commercial benefit in providing flexible and reasonably strong mechanisms for protecting their clients’ data from other clients who may be curious about it and for enforcing policies on geolocation of data. Their infrastructures will be subject to the same kinds of vulnerabilities found in virtually all software systems today, but they may have more expert operators available to configure and manage their software systems than are available from smaller providers. For further discussion of cloud security and privacy issues, see Samarati and De Capitani di Vimercati (2010) and De Capitani di Vimercati et al. (2012).

Conclusions

Fundamental computing concepts for engineered controls on access to data and on information flows are reasonably well developed. They are perhaps not so widely deployed as they might be. The result is that for the next several years at least, an individual responsible for assuring that big datasets are accessed according to prescribed policies will need to rely on mechanisms that may be built into applications or may be lacking entirely. If they are built into applications, those applications will be layered over programming language libraries, operating systems, and perhaps hypervisors that will probably be reasonably reliable but will undoubtedly be penetrable. If they are not built into the applications, the individual will have to trust researchers who have access to the datasets to use them properly. In either case, as long as the data to be protected are not seen as of high value to an attacker, the kinds of problems likely to occur will mostly be

accidental. Nevertheless, accidental disclosures of (for example) masses of location data about citizens' movements in a city could create a considerable stir. Establishing accountability for the event will be crucial; hence the custodians of the dataset should pay particular attention to assuring they have reasonable logs and audit trails in place and that those mechanisms are monitored on a regular basis.

Areas of research that could change the picture in the future include advances in practical cryptographic solutions to computing on encrypted data, which could reduce the need to trust hardware and system software. Advances in methods for building systems in which information flow, rather than access control, is the basis for policy enforcement could also open the door for better enforcement of comprehensible policies.

Notes

¹ Assuring that the specifications correctly capture the system requirements is itself a difficult and inherently manual task.

² Note that *complete* provenance for a particular data item would include not only the programs that read and write the data, but also the provenance data for those programs themselves – version numbers, etc. – and also data about the compilers and other software used in creating them.

³ Note that this approach places trust in the correct functioning of the TPM mechanisms.

⁴ The cloud metaphor seems to have arisen from graphics in which a computing network was represented by an abstract cloud, to avoid the complexity of drawing all the lines. It is not a very helpful metaphor except perhaps in the sense that the cloud interface obscures the details behind it.

⁵ The global reach and accessibility of Internet-connected hosts makes geographic location irrelevant in relation to the probability of remote cyberattacks.

References

- Abbadi, Imad M., and John Lyle. 2011. Challenges for Provenance in Cloud Computing. In *Proc. TaPP '11, 3rd USENIX Workshop on the Theory and Practice of Provenance*, June. Available at

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
• in the book Cambridge University Press.

- http://www.usenix.org/events/tapp11/tech/final_files/Abbadi.pdf (accessed December 28, 2013).
- Accumulo. 2013. Apache Accumulo User Manual 1.5. Available at http://accumulo.apache.org/1.5/accumulo_user_manual.html (accessed December 28, 2013).
- Akinyele, J. A., M. W. Pagano, M. D. Green, C. U. Lehmann, Z. N. J. Peterson, and A. D. Rubin. 2011. Securing Electronic Medical Records Using Attribute-Based Encryption on Mobile Devices. In *ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices*, Chicago, October. Available at <http://sharps.org/wp-content/uploads/AKINYELE-CCS.pdf> (accessed December 28, 2013).
- Anderson, James P. 1972. *Computer Security Technology Planning Study*. ESD-TR-73-51, vol II, ESD/AFSC, Hanscom AFB, Bedford, MA, October. Available at <http://seclab.cs.ucdavis.edu/projects/history/papers/ande72.pdf> (accessed December 18, 2013).
- Birrell, Eleanor, and Fred B. Schneider. 2013. Federated Identity Management Systems: A Privacy-based Characterization. *IEEE Security & Privacy Magazine* 11, no. 5 (September–October): 36–48.
- Boneh, Dan, Amit Sahai, and Brent Waters. 2011. Functional Encryption: Definitions and Challenges. In *Proc. IACR 8th Theory of Cryptography Conference 2011*, LNCS 6597, 253–257. Heidelberg: Springer.
- Bonneau, Joseph. 2012. The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords. In *Proc. 2012 IEEE Symposium on Security and Privacy*, 538–552.
- Buneman, Peter, Sanjeev Khanna, and Wang-Chiew Tan. 2001. Why and Where: A Characterization of Data Provenance. In *Proc. International Conference on Database Theory (ICOT) 2001*, 316–330. Heidelberg: Springer.
- Corbato, Fernando J., and Victor A. Vyssotsky. 1965. Introduction and Overview of the MULTICS System. In *Proc. AFIPS Fall Joint Computer Conference 1965*, 185–197. Available at <http://www.multicians.org/fjcc1.html> (accessed December 28, 2013).
- DARPA (Defense Advanced Research Projects Agency). 2010. Broad Agency Announcement (BAA) Programming Computation on Encrypted Data (PROCEED). July. Available at

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

<https://www.fbo.gov/utils/view?id=11be1516746ea13def0e82984d39f59b> (accessed December 28, 2013).

- De Capitani di Vimercati, Sabrina, Sara Foresti, and Pierangela Samarati. 2012. Managing and Accessing Data in the Cloud: Privacy Risks and Approaches. In *Proc. 7th International Conference on Risk and Security of Internet and Systems (CRISIS)*, 1–9.
- DSB (U.S. Department of Defense, Defense Science Board). 2013. *Task Force Report: Cyber Security and Reliability in a Digital Cloud*. OUSD AT&L, January. Available at <http://www.acq.osd.mil/dsb/reports/CyberCloud.pdf> (accessed December 28, 2013).
- Garfinkel, Simson. 2013. Digital Forensics. *American Scientist* 101, no. 5 (September–October): 370ff.
- Gentry, Craig. 2009. Fully Homomorphic Encryption Using Ideal Lattices. In *Proc. ACM Symposium on Theory of Computing (STOC) 2009*, 169–178.
- Goyal, V., A. Sahai, O. Pandey, and B. Waters. 2006. Attribute-based Encryption for Fine-Grained Access Control of Encrypted Data. In *ACM Conference on Computer and Communications Security 2006*, 89–98.
- Gunter, Carl A., David M. Leibovitz, and Bradley Malin. 2011. Experience-based Access Management: A Life-Cycle Framework for Identity and Access Management Systems. *IEEE Security & Privacy Magazine* 9, no. 5 (September–October): 48–55.
- IARPA (Intelligence Advanced Research Projects Activity). 2011. Security and Privacy Assurance Research (SPAR) Program. IARPA-BAA-11-01. Available at <https://www.fbo.gov/utils/view?id=920029a5107a9974c2e379324a1dcc4e> (accessed December 28, 2013).
- Kahn, David. 1996. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet*, revised edition. New York: Scribner.
- King, S. T., P. M. Chen, Y.-M. Wang, C. Verbowski, H. J. Wang, and J. R. Lorch. 2006. SubVirt: Implementing Malware with Virtual Machines. In *Proc. 2006 IEEE Symposium on Security and Privacy*, 314–327.
- Kocher, Paul, Joshua Jaffe, and Benjamin Jun. 1999. Differential power analysis. In *Advances in Cryptology—CRYPTO'99*, 388–397. Heidelberg: Springer.

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

- Landwehr, Carl E. 1981. Formal Models for Computer Security. *ACM Computing Surveys* 13, no. 3 (September): 247–278.
- Landwehr, Carl E., Constance L. Heitmeyer, and John D. McLean. 1984. A Security Model for Military Message Systems. *ACM Transactions on Computer Systems* 2, no. 3 (August): 198–222.
- Martin, Andrew, John Lyle, and Cornelius Namiluko. 2012. Provenance as a Security Control. In *TaPP '12, Proc. 4th UNIX Workshop on Theory and Practice of Provenance*, June. Available at <https://www.usenix.org/system/files/conference/tapp12/tapp12-final17.pdf> (accessed December 28, 2013).
- McLean, John D. 1994. A General Theory of Composition for Trace Sets Closed under Selective Interleaving Functions. In *Proc. 1994 IEEE Symposium on Security and Privacy*, 79–93.
- Microsoft. 2010. Cybersecurity for Open Government, June. Available at <http://download.microsoft.com/download/1/1/F/11F98312-8E4C-4277-AF3F-B276FD6978DA/CyberSecurityWhitePaper.pdf> (accessed December 27, 2013).
- Muniswamy-Reddy, K., D. Holland, U. Braun, and M. Seltzer. 2006. Provenance-Aware Storage Systems. In *Proc. 2006 USENIX Annual Technical Conference*, June. Available at <http://www.eecs.harvard.edu/margo/papers/usenix06-pass/paper.pdf> (accessed December 28, 2013).
- Muniswamy-Reddy, K.-K., P. Macko, and M. Seltzer. 2010. Provenance for the Cloud. In *Proc. FAST '10, 8th USENIX Conference on File and Storage Technologies*. Available at www.usenix.org/events/fast10/tech/full_papers/muniswamy-reddy.pdf (accessed December 28, 2013).
- NIST (U.S. National Institutes of Standards and Technology). 2012. *Guidelines for Media Sanitization*, by Richard Kissel, Matthew Scholl, Steven Skolochenko, and Xing Li. Draft NIST Special Publication 800-88, Revision 1. Available at http://csrc.nist.gov/publications/drafts/800-88-rev1/sp800_88_r1_draft.pdf (accessed December 28, 2013).
- NIST (U.S. National Institutes of Standards and Technology). 2013. *Improving Critical Infrastructure Cybersecurity, Executive Order 13636: Preliminary Cybersecurity Framework*, October. Available at <http://www.nist.gov/itl/upload/preliminary-cybersecurity-framework.pdf> (accessed December 27, 2013).

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

- Ni, Qun, Shouhuai Xu, Elisa Bertino, Ravi Sandhu, and Weili Han. 2009. An Access Control Language for a General Provenance Model. In *Proc. Secure Data Management (SDM) 2009*, LNCS 5779, 68–88. Heidelberg: Springer.
- Park, Jaehong, and Ravi Sandhu. 2004. The UCON_{ABC} Usage Control Model. *ACM Transactions on Information and System Security* 7, no. 1 (February): 128–174.
- Piretti, Matthew, Patrick Traynor, Patrick McDaniel, and Brent Waters. 2006. Secure Attribute-Based Systems. In *ACM Conference on Computer and Communications Security 2006*, 99–112. Available at <http://www.patrickmcdaniel.org/pubs/ccs06b.pdf> (accessed December 28, 2013).
- Samarati, Pierangela, and Sabrina De Capitani di Vimercati. 2010. Data Protection in Outsourcing Scenarios: Issues and Directions. In *Proc. ACM AsiaCCS 2010*. Available at spdp.di.unimi.it/papers/sd-asiaccs10.pdf (accessed December 28, 2013).
- Sandhu, Ravi. 1996. Role-based Access Control Models. *IEEE Computer* 29, no. 2 (February): 38–47.
- Schneider, Fred B. 2000. Enforceable Security Policies. *ACM Transactions on Information and System Security* 3, no. 1 (February): 30–50.
- Strang, Thomas J. K. 1996. Preventing Infestations: Control Strategies and Detection Methods. Canadian Conservation Institute. *CCI Notes 3/1*. Available at http://www.cci-icc.gc.ca/publications/notes/3-1_e.pdf (accessed December 17, 2013).
- TBC (Treasury Board of Canada). 2006. *Operational Security Standard, Management of Information Technology Security (MITS)*, Sec. 16–18. Available at <http://www.tbs-sct.gc.ca/pol/doc-eng.aspx?id=12328§ion=text> (accessed December 17, 2013).
- Tschantz, Michael C., Anupam Datta, and Jeannette M. Wing. 2012. Formalizing and Enforcing Purpose Restrictions in Privacy Policies. In *Proc. 2012 IEEE Symposium on Security and Privacy*, May.
- Vavilapalli, V. K., A. C. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans, T. Graves, J. Lowe, H. Shah, S. Seth, B. Saha, C. Curino, O. O’Malley, S. Radia, B. Reed, and E. Baldeschwieler. 2013. Apache Hadoop YARN: Yet Another Resource Negotiator. In *Proc. ACM Symposium on Cloud Computing*, October. Available at <http://www.socc2013.org/home/program/a5-vavilapalli.pdf?attredirects=0> (accessed December 28, 2013).

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Verizon, Inc. 2013. *2013 Data Breach Investigations Report*, April. Available at http://www.verizonenterprise.com/resources/reports/rp_data-breach-investigations-report-2013_en_xg.pdf (accessed December 29, 2013).

Weir, Matt, Sudhir Aggarwal, Michael Collins, and Henry Stern. 2010. Testing Metrics for Password Creation Policies Using Large Sets of Revealed Passwords. In *Proc. ACM Conference on Computer and Communications Security*.

Wu, Zhenyu, Xu Zhang, and Haining Wang. 2012. Whispers in the Hyper-Space: High-Speed Covert Channel Attacks in the Cloud. In *Proc. USENIX Security Conference 2012*. Available at <https://www.usenix.org/system/files/conference/usenixsecurity12/sec12-final97.pdf> (accessed December 28, 2013).

Chapter 11

Portable Approaches to Informed Consent and Open Data

John Wilbanks

Introduction

What frameworks are available to permit data reuse? How can legal and technical systems be structured to allow people to donate their data to science? What are appropriate methods for repurposing traditional consent forms so that user-donated data can be gathered, de-identified, and syndicated for use in computational research environments?

This chapter will examine how traditional frameworks for permitting data reuse have been left behind by the mix of advanced techniques for re-identification and cheap technologies for the creation of data about individuals. Existing systems typically depend on the idea that de-identification is robust and stable, despite significant¹ evidence² that re-identification is regularly possible on at least some portion of a de-identified cohort. The promise that privacy can always be protected, that data can always be de-identified or made anonymous, is at odds with many of the emerging realities of our world.³

At issue here is a real risk to scientific progress. If privacy concerns block the redistribution of data on which scientific and policy conclusions are based, then those conclusions will be difficult to justify to the public who must understand them. We must find a balance between our ability to make and produce identifiable data, the known failure rates of de-identification systems, and our need for policy and technology supported by ‘good’ data. If we cannot find this balance we risk a tragedy of the data commons in which the justifications for social, scientific, and political actions are available only to a select few.⁴

Approaches and frameworks that are emerging to deal with this reality tend to fall along two contours. One uses technological and organizational systems to ‘create’ privacy where it has been eroded while allowing data reuse. This approach draws on encryption and boundary organizations to manage privacy on behalf of individuals. The second applies an approach of ‘radical honesty’ towards data contribution by acknowledging up front the tension between anonymization and utility, and the difficulty of true de-identification. It draws on the traditions of beneficence and utility as well as autonomy in informed consent to create reusable and redistributable open data, and leverages cloud-based systems to facilitate storage, collaborative reuse, and analysis of data.

Traditional Approaches Falling Behind

Most data collected commercially in the United States today lacks direct protection under the law. It is instead governed by a hodge-podge of contractual instruments such as terms of use, implemented by long and rarely read ‘click-through’ agreements on phones, tablets, and laptops. The parties gathering the data typically attempt to minimize the ability of the person about whom data is being gathered to comprehend the scope of data, and its usage, through a mixture of sharp design and obscure legal jargon.⁵

However, several kinds of data do receive direct legislative and/or executive protection for commercial and private use at the national level: primarily educational, financial, and health records. This chapter will focus on the last of the three.

Health records and health data – the two are often thought of separately, although they are rapidly aligning and in many cases merging – receive distinct privacy protections under the Health Insurance Portability and Accountability Act (HIPAA) passed in 1996. HIPAA contains a specific Privacy Rule regulating the use and disclosure of ‘protected health information’ (PHI) held by ‘covered entities’ involved in the administration and payment of health care. PHI is any information held about an individual by a covered entity about health status, payment, treatment, and other related information.

HIPAA lays out a specific set of kinds of data that are explicitly PHI: names, geographic indicators, dates, phone, fax, e-mail, Social Security numbers, medical record numbers, health insurance beneficiary numbers, account numbers, license numbers, vehicle identifiers (i.e. VIN or license plate), device identifiers, URLs, IP addresses, biometric identifiers, full face images, and “any other” unique number, characteristic, or code except the code used by the investigator to code the data. It is possible to distribute data about an individual, such as clinical information or treatment information, if these 18 identifiers are removed (this creates a ‘safe harbor’ in which the data is considered by law to be de-identified) or if the overall data, including PHI, has been certified by an expert to be technically de-identified to an extent where the possibility of re-identification is very small.

These regulations, however, ignore a fact of data that for years was well known to experts in the field but poorly known outside: if there is either enough direct data (such as clinical information) or indirect data (such as metadata or data emerging from ‘non-health’ devices) that is collected and shared, even the absence of the 18 identifiers is unlikely to protect against re-identification approaches by experts. The tension is not unique to health but perhaps is most acute: we need to know a lot about an individual to properly make use of that individual’s data in a scientific research context, but precisely by knowing a lot about the individual, we degrade the ability to guarantee that individual’s anonymity.

As an example, suppose we were attempting to understand why it is that a cancer drug fails to achieve its outcome in 75% of those who take it. This is unfortunately

common, and being able to know in advance which 25% will actually benefit from dosage, versus which will experience only toxic side effects, would be beneficial both to the patient and to the insurance system that pays for failed dosages just as it pays for successful ones.

But it requires a deep dive into those individuals' data: what are their unique genetic variations, which might provide clues as to how individual genomes affect drug response? What are their environments and lifestyles, which might provide clues as to how diet and exercise affect drug response? What kind of data is coming out of their classic clinical sampling, which might provide early signals if the drug is or isn't working? Taken together these data points provide precisely the kind of longitudinal health data that can power strong analytic models and pinpoint clusters of predictive information – but at the same time, can allow someone else to vector in on the identities of the individuals in the study.

High quantities of varied data tied to individuals will make this failure worse. As we begin to connect mobile devices to health through applications, connected hardware, fitness trackers, and even wearable electronics, it is not hard to imagine a world in which health insurance companies and mobile providers form partnerships to sense when a person is in line at McDonald's and send them a text message suggesting a day at the gym – perhaps even offering a coupon. When our grocery store loyalty cards are connected in turn to our fitness trackers, it will be very hard to hide.

But these at least are decisions that we have the power to change, or at least obscure from view. Open operating systems on mobile devices have the potential to re-create some forms of privacy in this space. But as our genomes and our medical records converge, as they inevitably will, there will be permanent records of the diseases and syndromes to which we are more or less susceptible.⁶ And those records are indeed almost perfectly identifiable – our genomes are far more accurate ways to precisely identify us than even our fingerprints or our credit card numbers, which are in turn each far more unique to us than our Social Security card numbers.

The law is most concerned with whether or not John Wilbanks has been uniquely identified by a health provider. But this is irrelevant when an entire network of providers outside the traditional health system knows that my credit card has been used at a grocery to buy pizza, that my phone spends five times as many minutes in fast food restaurants as it does in athletic environments, or that my online shopping habits are yielding larger and larger pants sizes. There is only one person for whom all of those can be true, and he's probably at risk for diabetes and metabolic syndrome.

It is precisely this network-aggregated, metadata-centric approach that has become so notorious through the revelations of domestic data capture at the National Security Agency in 2013 and before. This approach falls well outside the boundaries of health legislation or regulation, but can just as easily be used to infer health information as the

data that is traditionally known to be about health. Indeed, even social media postings and photographs contain data that can be converted to health data.⁷ Simply put, our legislative and regulatory systems that attempt to guarantee health privacy have been overwhelmed by the technocratic infrastructure that suffuses our daily lives, and are unlikely to catch up in the near future.

The deepest irony is that the protections are strongest and the regulations most effective at the institutions where health and life sciences research take place: the covered entities. Start-up companies, telecommunications providers, and others are almost entirely unaffected as they gather metadata and actual data from which health, and identity, can be inferred. Research data that is often federally funded or involves data from federally funded research becomes subject to the laws and regulations for government data management – which is significantly more complex, but is not necessarily able to prevent re-identification attacks.

The Database of Genotypes and Phenotypes (dbGaP) was founded in 2006 and is maintained by the U.S. National Institutes of Health. It was developed to archive and distribute the results of studies that have investigated the interaction of individual genetic variations (the genotype) with the observable health and traits of those individuals (the phenotype). These studies hold the promise of decoding which minute differences in genetic makeup affect whether or not a cancer drug will work before it is given, or what kinds of mutations mean a rapid acceleration of a disease such as Parkinson's or Alzheimer's.⁸

Recipients of U.S. NIH funding related to genome-wide association studies and whole genome sequencing studies are expected to deposit the results in dbGaP, based on policies first promulgated in 2007 and updated in 2009:

Consistent with the NIH mission to improve public health through research and the longstanding NIH policy to make data publicly available from the research activities that it funds, the NIH has concluded that the full value of sequence-based genomic data can best be realized by making the sequence, as well as other genomic and phenotype datasets derived from large-scale studies, available as broadly as possible to a wide range of scientific investigators.⁹

The idea behind dbGaP was to support two levels of data access: ‘pooled’ or ‘aggregated’ data that is not granular to the individual level (analogous to ZIP-code-level statistics in census data) in a public layer and individual study information in a more controlled layer. The controlled-layer data would still be de-identified according to HIPAA, vetted by data access committees, and released only to authorized investigators who agreed to the terms and conditions of use. These data are available behind a firewall and can be accessed after application for use by investigators.

The first two years of the database saw nearly 500 complete downloads of the total open layer of data.¹⁰ But in late 2008, a paper was published that demonstrated a feasible re-identification method:¹¹ how to plausibly identify at least some individuals inside large sets of aggregate genomic information that had been de-identified according to the laws and regulations. The NIH quickly moved to shift aggregate genomic data from the open layer of dbGaP to the controlled layer. Follow-on studies revealed that re-identification vectors were possible.¹²

The reason this is a problem is that the studies involved did not contemplate the risks of re-identification on a platform like dbGaP. The data came online under a dizzying variety of informed consent terms and procedures, which typically focused on the study itself rather than data sharing and were written by investigators whose field is health – not re-identification. In particular, the forms failed to notify participants of the risk of participation in a study destined for online archives.

This is a real problem. Our ability to understand how minute individual variations in genetics affect health will depend on having the right to perform research on individual variations. But our data governance systems are behind the technology curves of genetic data generation, data distribution over the internet, and re-identification.

We must acknowledge that participation in genetic studies carries a risk of identification, one that increases with the volume of data generated about an individual and as time improves the tools for re-identification – often in fields far removed from the genome. Thus, we face a real test of our capacity to design and implement new approaches to individual-level data governance in health, ones that both facilitate the acceleration of knowledge creation but also provide honest, transparent guidance to study participants.

Technological Control Frameworks

Given the complexity of HIPAA and other federal approaches to privacy, as well as the varying degrees of health data protection across the 50 states of the United States, many attempts to increase personal data privacy while facilitating sharing do not attempt to change legislation, harmonize policy, or otherwise engage in government. They instead attempt to encode a more contemporary approach to data gathering and sharing inside technological, structural, organizational, legal, and normative *frameworks* that work inside existing laws and attempt to leverage those laws to reach the goal of reconciling research with privacy. These frameworks have in common the idea of harm prevention – of preventing improper, harmful, or bad uses of data – through *implementation of controls*.

The concept of control as the guiding principle of these frameworks is essential to their design and implementation. Control can be implemented via software, contract, terms of use, intellectual property, liability, economics, or other means. Each attempts to

regulate the allowable behaviors of data users to protect the privacy of individuals whose data is being used.

These frameworks are discussed in more detail elsewhere in this volume, but it is worthwhile to quickly examine some of the most relevant systems in the context of health and to look at their advantages and disadvantages depending on the kind of data being controlled.

One of the most popular control frameworks is ‘differential privacy’, which provides access to statistical databases for queries while using algorithmic methods to reduce the odds of re-identification of records in the database itself. The goal is to respond to queries to the database as accurately as possible without compromising identity. The advantages of this approach are several: the approaches are well known and tested, the addition or subtraction of a single individual’s data is unlikely to affect the accuracy of a query response (allowing for withdrawal of an individual from a clinical study, for example, without significantly compromising the overall queries already run), and there is good research on how to protect against excessive distortion of the data.¹³

The disadvantages, however, are real. The connection between accuracy and identifiability creates a documented tension between the statistical addition of ‘noise’ to data to increase the difficulty of identification and the actual utility of the database. When the data concern movie reviews, this noise is innocuous. When the data are genetic variations – the individual mutations that make each of us unique – then the addition of noise to the data creates real problems. Trust that the underlying data accurately represent the genomic profile of the individuals is essential to the analysis, and thus adding noise via ‘fake’ mutations to the system to make identifiability harder in turn reduces the research utility of the data. Indeed, the entire point of a genetic study looking for health outcomes is in many ways to use accurate genetic data tied to outcomes in order to find patterns of individual mutations correlated to outcomes, and statistical approaches to differential privacy very likely would confuse the issue. As one author notes, standard anonymization techniques are not applicable to genomic information as it is the “ultimate identity code.”¹⁴

A technological framework often proposed to solve the same problem is homomorphic encryption (HE).¹⁵ HE enables predetermined queries to run on encrypted data that obtain, in turn, an encrypted result. That result can then be decrypted in a way that exactly matches the results that the data user would have seen if running the query on non-encrypted data. This would allow a user to run genome-wide queries on many genomes, find the individual variations, but not know from which genomes the variation patterns came.

A key advantage to HE is that it allows for one person to run the query but not be able to decrypt the answer (allowing for an expert, or an expert system, to execute encrypted queries on behalf of a less skilled operator, who would then be the only one

able to read the result). This is a real advantage in health and life sciences research, as the researchers working on these kinds of data are very unlikely to have sufficient training to manage complex encryption systems on their own. Another advantage of HE is its innate compatibility with cloud architectures to which large-scale data processing is rapidly moving.

A key disadvantage is speed. HE is inherently slow and gets slower the more information that is encoded in the ciphertext. Placing a single whole-sequence genome into HE would create serious performance problems, much less placing tens of thousands of genomes (as required for performing significantly powerful analysis). HE is also comparatively novel as a technology and lacks widespread support from vendors or expertise in implementation.

A third technical framework depends on distributed storage of the data. Private, sensitive information is stored across multiple databases held by multiple entities – for example, part of an individual’s genome would be held in Boston, while another part of the same individual’s genome would be held in New York. Each of the two parties would have incomplete, semi-random data. But when a technical framework integrates the data, ‘sensitive’ values may be extracted. This is known as ‘secret sharing’¹⁶ and may be analogized to the peer-to-peer approaches for sharing content on the web such as Bit Torrent.

A key advantage of secret sharing is that predefined queries can run on the shared data without transmission of the data itself, without complex or slow encryption, and without adding statistical noise to the information. Another advantage is the successful deployment of the secret sharing approach in auctions¹⁷ and financial markets,¹⁸ which increases the availability and support of tools for its implementation.

Organizational Structures for Control

Technical frameworks for privacy, on their own, are typically not sufficient to create a controlled environment for sharing. Organizations must operate the frameworks to ensure they are running correctly, watch for violations, punish transgressions, and perform other functions. The various technological frameworks map to some organizational structures better than others. Of the technologies noted here, only secret sharing, with its inherent distributed nature, aligns well with a non-organizational control structure.

Differential privacy maps well to a marketplace paradigm, in which data-sharing arrangements are structured as markets that can either ‘hedge’ an individual against harm by taking a stake in the market or even directly receive financial benefits in the event of data usage harms. This creates an interesting model for organizations, both for-profit and not-for-profit, to operate data marketplaces where individuals choose to store their data and have it be bought and sold for use inside the framework run by the organization.

Another structural paradigm that is implied is one we know well: banking. If data is an economic asset (as the market idea recognizes, and as it is often treated in the daily world of commerce and social media) then it is an asset in which an individual might wish to invest in hope of a return. Health record banking¹⁹ is a well-developed concept, and related models such as land trusts, conservation trusts, and development trusts are each being explored for technology-mediated, data-centric health collaboration.

There are also some established business models for access to sensitive data. One is the liability regime, in which a data holder provides access to data that may (or may not) be identifiable, but the data user must agree to punitive terms and conditions that govern the kinds of queries that may be run, restrict data that may be downloaded, and outline penalties for failure to comply. PatientsLikeMe is a good example of this, though the model is a longstanding and established one in data provision companies. Another is the services model, in which security and privacy are maintained by the data aggregator holding all data internally, and performing fee-for-service research as a consulting service rather than allowing access to the data directly.

In all of these structural-organizational frameworks, the organization would install terms of use around any entry into the market, presumably through contracts. Users would be verified and subject to some financial liability if they breach the contract by running queries they are not allowed to run, or attempting re-identification attacks, or by somehow rebuilding elements of the raw data and republishing them elsewhere.

The key weakness in any of these models is the inability to lock down the data should the data somehow ‘leak’ in violation of contractual restrictions or technical restrictions. The organization in charge of data aggregation would clearly have the right to sue the transgressor, but the copy that is then ‘in the wild’ would be nearly impossible to track down and protect.

This reflects a structural element of the intellectual property regime around data, which is inherently less able to lock down reuse copyrights and patents. Data is considered by most jurisdictions to fall under the regime of a fact, rather than a creative work or an invention, and thus sits in the public domain by default. This is a powerful and important default status, as it prevents facts from being owned, laws of nature from being enclosed, and ensconces them in the commons of the mind. But it also makes it hard to license *conditionally* – which is the root of open source software and Creative Commons licensing.

Conditional licensing is the use of a contract to note that reuse is allowed, but only if certain conditions are met, such as attribution, or non-commercial use, or even that downstream works must be relicensed under the same terms. The key is that in copyright, such terms are easily enforceable. The conditions follow the song, or the software. But these types of conditions for data don’t travel with the data as it is propagated across a network.²⁰ The public domain intellectual property status of data thus has a sharp tang in

the privacy sphere: data are unlikely to receive the kinds of protections that can be used to take down a dataset that was private but has been made public either by accident or by malice.

Commons-based Frameworks: Open Consent

The number of foundations, endowments, and other non-profit groups investing in health IT has exploded over the past ten years. Small, nimble groups formed by patients and large foundations created by technology entrepreneurs have joined traditional funders such as state and federal governments, creating a significant overall increase in funding to basic life sciences and translational health research. At the same time, a dizzying variety of data began to flood the market. We can use mobile devices to track our heart rate, blood pressure, weight, physical activity, gym visits, sleep, and more. Physicians are under pressure to complete the transition to electronic health records. Companies are aggressively moving to provide services ranging from genotyping to self-tracking to community disease management.

In the midst of this, expectations remain that some form of privacy or anonymity should be the goal, even if it is next to impossible to guarantee.²¹ The frameworks explored in the previous section attempt to address the imbalance between that goal and the reality of identification. But it is also possible that studies can create different ideas, or locate participants who are less worried about privacy than about advancing health and science.

To the extent these expectations do exist in health studies, they have emerged from the methods used to enroll participants: the documents and processes used to obtain informed consent. The NIH's data-sharing policy explicitly calls out protecting privacy and confidentiality as critical, which creates in turn a direct implication of a control-based framework. And thus as the ability to capture information explodes, and the cost of capture drops, the consent structures have remained resolutely dogged in assuring people their privacy will be protected – even though that assurance most definitely cannot be kept for all.

It is important that scientists recognize this, and find a solution. Given the wealth of clinical and genetic information now collected in a clinical trial it is becoming apparent that there are a variety of secondary uses of clinical trial data that could greatly enhance the rate of scientific progress in a variety of ways not foreseen by the original study developers. The same is true of epidemiological and/or observational studies.

This is particularly true of genetic data: the American Society of Human Genetics stated they are “acutely aware that the most accurate individual identifier is the DNA sequence itself or its surrogate here, genotypes across the genome. It is clear that these available genotypes alone, available on tens to hundreds of thousands of individuals in

the repository, are more accurate identifiers than demographic variables alone; the combination is an accurate and unique identifier.”²²

But a response has emerged to this problem, one that embeds the reuse of the information as a higher goal than the guarantee of privacy or the prevention of re-identification. These are often called ‘open’ consents, but the consent is simply part of a large *commons-based framework* intended to share data, rather than to control its use.

Commons-based frameworks for reuse attempt to recruit individuals who have not only benefited from the explosions in investment and technology, but who also understand the risks and uncertainties of making their data available for reuse. The key complexity comes from the uncertainty – from the ‘unknown unknowns’ that may emerge as risks downstream, years after data is made available for reuse. This is consistent with the complexity of the data itself, which is beyond most of our attempts to comprehend as individuals, as well as with the unknown benefits of reusable health data. We simply do not yet understand the system that is emerging well enough to precisely assess either its true benefits or true risks over the long term. Commons-based frameworks must live with this fundamental uncertainty, and participants must understand this uncertainty enough to provide informed consent for data redistribution and reuse.

‘Open’ consent was first developed by the founders of the Personal Genome Project (PGP) at Harvard Medical School. The PGP study asks participants to post health information (both as records and as interviews), performs whole genome sequencing, and creates an immortalized cell line that is available under liberal terms and at a low cost from a cell culture repository. PGP has the potential to create a data resource where a user might identify a promising set of variations tied to outcomes computationally, then order the cells and test the hypothesis *in vitro* within days.

The study’s model of consent starts not from the premise of preventing harm, or controlling use, but instead from the idea that participants in genotype–phenotype studies must understand that the data collected not only can be, but is intended to be accessed, shared, and linked to other sets of information. PGP participants must complete a thorough and, for many, difficult consent protocol proving they understand these intentions as well as the risks involved in participation, the bulk of which remain unknown or unknowable at the moment of consent.

Privacy is not guaranteed in the PGP, and identifiability is called out as a possibility. Participants can withdraw, and their data will be taken down from the study’s servers, but no guarantees are made that their data is gone from the web: if it has been downloaded and redistributed, the study can no longer control its presence or its use.

Commons-based Frameworks: Portable Consent

Inspired by the PGP's groundbreaking work, but desiring a simpler process to create informed consent, Sage Bionetworks began work on a 'portable' approach to informed consent, called Portable Legal Consent or PLC.

Portable Legal Consent was developed as a tool to allow patients to tell the doctors, researchers, and companies experimenting on them that they, the patients, have rights with respect to the data generated from their bodies. PLC states that what the patient desires is for the data to be shared broadly in the public domain, to serve scientific progress as a whole, regardless of the particular individual or institution that makes the breakthrough.

PLC emerged from the 2011 Sage Bionetworks Commons Congress, where a working group focused on the need for standardized approaches to privacy and patient empowerment. It became clear that two approaches were needed: one to populate computational platforms with individual-level data that can be used to perform computational research with as few barriers as possible, and one to empower patients to take control of their own data. The first approach became PLC, while the second approach informs the PEER platform hosted by the Genetic Alliance.

The legal inspirations for PLC were the informed consent process developed by the Personal Genome Project (from which it draws both ideas, and even some text in its FAQ). PLC also drew on the idea of 'human readable' interfaces to complex legal documents that Creative Commons pioneered. PLC embedded the idea that study investigators should disclose risks about research inside the idea that data should be something that can be remixed, to allow unexpected discoveries to emerge from the combination of earlier studies by later scientists.

If a person completed the PLC process, she had an informed consent that traveled with her from one upload of data into an environment that allowed many studies – that is portable from a research perspective, and that she controls. PLC meant that the data she chose to upload into a common genomics repository would be able to support a broad range of genomic and health research without the unintended fragmentation of data created by traditional consent systems.

Participants in the study enrolled via a web-based 'wizard' that extracted the key elements of the consent – the obligations placed on researchers (and the significant limitations of those obligations), the freedoms granted by participants to data users, the risks involved – into a clear, layperson-friendly structure. Participants completed the wizard and watched a short video before indicating consent and being presented with the consent form to sign. All text, software, multimedia, user interface designs, and related systems were provided as public domain documents under the CC0 public domain waiver so that study designers outside Sage Bionetworks may take up and remix the elements of PLC as they wish.

The wizard that served as a consent tutorial was, by the numbers, effective at communicating the risks and benefits of data sharing based on the survey data, though it is very likely that the population who enrolled was self-selected to bias in favor of pre-existing comprehension and risk tolerance. Concepts that were essential to the study, such as ‘re-identification’, received plain-language definitions submitted for ethical review and approval that appeared on mouse-over by the user, a method shown to increase comprehension in online environments.²³

Participants were required to tick checkboxes next to key statements (such as agreeing to allow redistribution of their data) to indicate assent to individual elements of the study rather than a single digital signature on a single legal form. The statements were grouped into pages indicating key freedoms granted to researchers, and key risks understood by participants, and included a page where participants who felt uncomfortable could exit the tutorial before being presented with the opportunity to sign any binding consent documents or upload files. Only after passing through all the pages and specifically indicating desire to sign the form was a participant presented with a digital version of a traditional informed consent document. Every word on every screen, including the script of the video, was approved by an independent Institutional Review Board (IRB).

After signing the form, participants proceeded to upload data files including electronic health records, genomes and genotypes, lifestyle data from mobile devices, and anything else they found relevant to their health. The data were to be de-identified and syndicated to Sage’s Synapse platform, a technology designed to facilitate computational reuse and collaboration on complex health and biological data. But the heterogeneity of the formats of data, particularly the use of large image files (including photographs) of medical records, made de-identification virtually impossible without rekeying the data by hand, and thus data were not syndicated by default to researchers.

PLC was completely voluntary – one need not enroll, and if enrolled, one need not upload any data. Several hundred enrolled and uploaded at least one data file. A survey was run concurrently of users and indicated strong comprehension of the risks and conditions of the study – preliminary data indicate that more than 90% of those completing both the consent process and the survey understood key issues around data upload, study withdrawal, and permissions granted to data users.

The PLC study assigned a unique identifier to every enrolled participant (the author’s is Individual 1418165 / syn1418165), and PLC was IRB-approved to send emails to the unique identifiers to ask follow-up questions, recruit participants for follow-on studies, and more. This constituted a ‘re-mail’ process that resolves between the ID and the e-mail address held apart from the syndicated data as well as some interface designs that protect against recipients accidentally revealing personal information in replies.

However, PLC's bioethical promise to engage people in research was greater than the scientific power it enabled during the study lifetime. As a variety of federal and international projects move to make health data more standardized and 'liquid', as well as to create positive rights to access health data, the situation will change – but for now, very few people have meaningful health data about themselves, and only a small percentage of those have that data in a computationally useable format. PDFs of health records, or scanned images, constituted the vast majority of donated data, but these are of little use in a computational collaboration.

Thus it is important to connect the lessons of PLC and other privacy-compatible research engagement systems to emerging policy and advocacy movements that increase personal access to health data. As standardized clinical trial data sharing moves into practice and standard electronic medical record formats become the norm, the scientific power of data donation will begin to match the ethical power of engagement in consent processes like PLC.

Portable Legal Consent 2.0

Our goal with future versions of PLC is to leverage the desire to participate in new forms of clinical study to generate not just patient engagement, but the creation of commonly pooled resources that are scientifically useful as well. The capital markets see data primarily as an economic asset, not as a research one, and may well be able to find value even in the kinds of heterogenous data donated through something like PLC 1.0 – indeed, treating it as metadata for clinical trial recruitment or even drugs marketing.

But the social value we hope to create comes from the data as a vector to connect groups of patients with groups of experts who can analyze the data and are committed to returning results to patients. Thus, future versions focus on making PLC compatible with many different sorts of clinical studies, whether initiated by patients themselves or by more traditional investigators, on Sage Bionetworks' BRIDGE patient engagement platform.

First, modularity is an essential element of forthcoming versions of PLC. There are multiple approaches to making consent, or privacy separate from consent, modular – most focus on differential approaches that indicate one use is acceptable while another is not. In PLC, the overwhelming response has been not a desire for differential consent that allows only academic use, or consent only for a certain field of use; instead participants are far more concerned about the *kind of data* they have and how they would like to donate it. Thus, modularization of PLC will proceed as a function of data classes rather than a complex set of differential permissions.

This has multiple benefits. The most important is that it allows a far simpler consent experience for many users. The most complex aspects of the existing consent process revolve around the donation of genomic sequence information, which is highly

identifiable no matter how much de-identification we perform around the metadata for the file. This is also the kind of data that is most ambiguous in terms of possible future harms, while simultaneously the kind of data that the fewest people have.

By breaking sequence data apart from other data classes the consent experience can be radically shortened and simplified for most participants. Only if a participant has sequence data and wishes to donate it would she move through the module related to sequences. Other benefits include the ability to rapidly repurpose user interface and legal code for other studies that are gathering data whose classes are represented in PLC 2.0, and the ability to begin applying automatic de-identification processes to data.

Feedback indicates at least five desired data classes:

- Genetic sequence
- Clinical information
- Medical record
- Patient-reported outcomes
- Personal sensor data (including but not limited to mobile device)

It is important to note that many of these data classes have fuzzy borders. Medical records contain clinical information and are moving to expand to the capture of personal sensor data and genetic sequence. Patient-reported outcomes also may contain clinical information and personal sensor data.

The choice of these labels for the modules does not ignore this reality but reflects instead a choice to focus on the file formats associated with each kind of data. It is easy to know by file format if one is dealing with a sequence, or an electronic medical record, or the output from a mobile application or fitness device, and that makes it easier to algorithmically assign consent modules and/or de-identification processes to that file.

Second, we are moving to tie the consent forms themselves more tightly to the system that qualifies users of the data under consent. In PLC 1.0, users of the data need only have a Google account or otherwise verify their identity in a lightweight form. In 2014, users inside Sage Bionetworks' Synapse system will also have to pass a comprehension test to access certain features, including the ability to access data available under liberal terms and use programmatic tools such as the API. This shift recognizes that the risk–benefit relationship in the consent process is not solely the responsibility of the data donor, but also of the data user.

Conclusion

Emerging methods for data generation increasingly fall outside traditional legal frameworks for health data protections. More and more, citizens are able to generate data about themselves directly, whether by purchasing it as a consumer service, installing applications on their phones and computers, wearing devices, or more. This data has

enormous scientific value but currently sits well outside the legal and regulatory frameworks typically associated with science. Whatever systems emerge for data reuse must be extensible and flexible enough to integrate with the data that is to come, not just the data we have today. And the likely outcome is not a single monoculture, but a diverse ecosystem that features technical, organizational, structural, and commons-based approaches as design choices available to study investigators, data users, and citizens.

In the short term, the most likely benefit of that ecosystem will be a greater ability to understand how specific changes from one genotype to the next affect health outcomes – so that clinicians understand why one person responds well to a drug, but another does not. In the long term, anyone who can analyze the data will have the capacity to become a genomic explorer whether they have lots of money/funding or not. The biggest impacts of that change are by definition hard to describe right now, just as it was hard at the dawn of the Internet to imagine the modern Web, or at the dawn of the Web to imagine the smartphone revolution.

Notes

¹ K. El Emam, E. Jonker, L. Arbuckle, and B. Malin, “A Systematic Review of Re-Identification Attacks on Health Data,” *PLoS ONE* 6 (12): e28071.

² C. Christine Porter, “De-Identified Data and Third Party Data Mining: The Risk of Re-Identification of Personal Information,” *Shidler Journal of Law, Commerce and Technology* 5 (September 23, 2008): 3.

³ Paul Ohm, “Broken Promises of Privacy: Responding to the Surprising Failure of Anonymization” (August 13, 2009); *UCLA Law Review* 57 (2010): 1701; University of Colorado Law Legal Studies Research Paper No. 9-12.

⁴ Jane R. Bambauer, “Tragedy of the Data Commons,” *Harvard Journal of Law and Technology* 25 (2011): 1–67.

⁵ Avi Charkham, “5 Design Tricks Facebook Uses to Affect Your Privacy Decisions,” *TechCrunch*, August 25, 2012, <http://techcrunch.com/2012/08/25/5-design-tricks-facebook-uses-to-affect-your-privacy-decisions/>.

⁶ Among many others, see R. Sachidanandam, D. Weissman, S. C. Schmidt, J. M. Kakol, L. D. Stein, G. Marth, S. Sherry, J. C. Mullikin, B. J. Mortimore, D. L. Willey, S. E. Hunt, C. G. Cole, P. C. Coggill, C. M. Rice, Z. Ning, J. Rogers, D. R. Bentley, P. Y. Kwok, E. R. Mardis, R. T. Yeh, B. Schultz, L. Cook, R. Davenport, M. Dante, L. Fulton, L. Hillier, R. H. Waterston, J. D. McPherson, B. Gilman, S. Schaffner, W. J. Van Etten, D. Reich, J. Higgins, M. J. Daly, B. Blumenstiel, J. Baldwin, N. Stange-Thomann, M. C. Zody, L. Linton, E. S. Lander, D. Altshuler, and International SNP Map Working Group,

“A Map of Human Genome Sequence Variation Containing 1.42 Million Single Nucleotide Polymorphisms,” *Nature* 409, no. 6822 (February 15, 2001): 928–933.

⁷ See e.g. “We Eat Less Healthy than We Think” at <http://data.massivehealth.com/#infographic/perception> – data derived from a social application that takes pictures of foods and ratings by more than 500,000 users.

⁸ This section owes an enormous debt to Dan Vorhaus’s online writing, available at the *Genomics Law Report* at <http://www.genomicslawreport.com/index.php/author/dvorhaus/>.

⁹ See U.S. National Institutes of Health, “Notice on Development of Data Sharing Policy for Sequence and Related Genomic Data,” Notice No. NOT-HG-10-006, release date October 19, 2009. At the time of writing, the policy is under review yet again, in part because of privacy issues emerging since the last update. See “Input on the Draft NIH Genomic Data Sharing Policy,” NIH Notice No. NOT-OD-14-018, release date September 27, 2013.

¹⁰ Catherine Clabby, “DNA Research Commons Scaled Back,” *American Scientist* 97, no. 2 (March–April 2009): 113.

¹¹ Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V. Pearson, Dietrich A. Stephan, Stanley F. Nelson, and David W. Craig, “Resolving Individuals Contributing Trace Amounts of DNA to Highly Complex Mixtures Using High-Density SNP Genotyping Microarrays,” *PLOS Genetics* 4, no. 8 (August 29, 2008): e1000167.

¹² Kevin B. Jacobs, Meredith Yeager, Sholom Wacholder, David Craig, Peter Kraft, David J. Hunter, Justin Paschal, Teri A. Manolio, Margaret Tucker, Robert N. Hoover, Gilles D. Thomas, Stephen J. Chanock, and Nilanjan Chatterjee, “A New Statistic and Its Power to Infer Membership in a Genome-Wide Association Study using Genotype Frequencies,” *Nature Genetics* 41 (2009): 1253–1257.

¹³ Khaled El Emam and Fida Kamal Dankar, “Protecting Privacy Using k-Anonymity,” *Journal of the American Medical Informatics Association* 15, no. 5 (2008): 627–637.

¹⁴ Liina Kamm, Dan Bogdanov, Sven Laur, and Jaak Vilo, “A New Way to Protect Privacy in Large-Scale Genome-Wide Association Studies,” *Bioinformatics* 29, no. 7 (April 1, 2013): 886–893.

¹⁵ Daniele Micciancio, “Technical Perspective: A First Glimpse of Cryptography’s Holy Grail,” *Communications of the ACM* 53, no. 3 (2010): 96.

¹⁶ *Ibid.*, 14.

¹⁷ P. Bogetoft et al., “Secure Multiparty Computation Goes Live,” in *Proc. Financial Cryptography 2009*, ed. R. Dingledine and P. Golle, Lecture Notes in Computer Science 5628 (Heidelberg: Springer, 2009), 325–343.

¹⁸ D. Bogdanov et al., “Deploying Secure Multi-party Computation for Financial Data Analysis,” in *Proc. Financial Cryptography 2012*, Lecture Notes in Computer Science 7397 (Heidelberg: Springer, 2012), 57–64.

¹⁹ Health Record Banking Alliance, “National Infrastructure for HIE Using Personally Controlled Records,” White paper, January 2013. Available at <http://www.healthbanking.org/docs/HRBA%20Architecture%20White%20Paper%20Jan%202013.pdf>.

²⁰ Thinh Nguyen, Science Commons, “Freedom to Research: Keeping Scientific Data Open, Accessible, and Interoperable.” Available at <http://sciencecommons.org/wp-content/uploads/freedom-to-research.pdf>.

²¹ Lee Rainie, Sara Kiesler, Ruogu Kang, and Mary Madden, “Anonymity, Privacy, and Security Online. A Report of the Pew Internet and American Life Project,” September 5, 2013, <http://www.pewinternet.org/Reports/2013/Anonymity-online/Summary-of-Findings.aspx>.

²² American Society of Human Genetics, “ASHG Response to NIH on Genome-Wide Association Studies,” Policy Statement, November 30, 2006, http://www.ashg.org/pdf/policy/ASHG_PS_November2006.pdf.

²³ P. Antonenko and D. Niederhauser, “The Influence of Leads on Cognitive Load and Learning in a Hypertext Environment,” *Computers in Human Behavior* 26, no. 2 (March 2010): 140–150.

Part III

Statistical Framework

If big data are to be used for the public good, the inference that is drawn from them must be valid for different, targeted, populations. In order for that to occur, statisticians have to access the data that they may understand the data generating process, whether the assumptions of their statistical model are met and what relevant information is included or excluded. It is clear from the earlier chapters that the utility of big data depends on being able to study small groups in real time, using new data analytics techniques, like machine learning or data mining. These demands pose real challenges for anonymization and statistical analysis. The essays in this section identify the issues, spell out the statistical framework for both analysis and data release and outline the key issues for future research.

A major theme of the essays is that neither the data generating process nor the data collection process is well understood for big data. As Kreuter and Peng argue, almost also statistical experience is based on survey data, and over time statisticians have parsed the sources of error neatly into a total survey error framework. But the data generating process of many data streams – like administrative data or big data -, is less transparent and not under the control of the researcher; access is critical to building that understanding. So, continuous effort will be needed to develop standards of transparency in the collection of data. Same is needed in – what they call “back end” - any linkage, data preparation and processing, analysis, and reporting -to ensure reproducibility . Kreuter and Peng are pointing out, that the research in linkage and matching needs to be expanded, because it not only enriches possible analysis, it helps to evaluate the quality of the linked sources. Another major theme is that while there is a long history of providing access to survey data, with a large number of different strategies for reducing the risk, that experience is not very useful for providing access to big data. When it comes to big data, big data carry greater disclosure risks than the typical survey sample. At the core of the problem is that the assessment of risk is based on assumptions about what a possible intruder knows about the released data (response knowledge), how easily someone can be identified (uniqueness) and whether that intruder is malicious. In the case of big data, a – maybe large - undefined number of people know the identities in the data (response knowledge); it is unknown how many might be malicious. Big data also captures information about many more individuals and can include many more variables, so nearly everyone in the data is unique in the population (uniqueness).

The chapters begin to parse out the key elements which need to be studied in this context. The first is the content: Karr and Reiter distinguish what must be protected from what might be protected. The second is the people: Karr and Reiter treat analysts,

who use the data for statistical analysis, differently from intruders, who try to reidentify individuals. The third is definitions: Karr and Reiter think that big data may change attitudes about privacy, and change what information is considered as sensitive (“former times”: salary vs. “now”: medical records, DNA sequences). Dwork goes a step further with this argument, because “big data mandates a mathematically rigorous theory of privacy, a theory amenable to measure – and minimize – cumulative privacy, as data are analyzed, re-analyzed, shared, and linked”. She points out, that we need a definition of privacy, equipped with a measure of privacy loss. Data usage and so research with micro data should be accompanied by publication of the amount of privacy loss, that is, its privacy ‘price’. A possible definition of privacy has to take into account that researchers want to learn useful facts from the data they analyze. In her view, it does not matter if someone is in the data, because a generalized result may affect someone, who is not in the database: “that’s the heart of differential privacy”.

Randomization of the original data is one key element for preventing privacy. As Dwork points out “the property of being differential private depends ONLY on the data protection algorithm – something the data curator – “good guy” – controls.” The preferred solution for anonymizing data is to generate redacted (synthetic) data. Although some examples of solutions are presented, all authors agree, that methods for generating massive synthetic databases merit further research. It is impossible to simultaneously preserve privacy and to release a synthetic dataset that answers too many research questions with much accuracy.

Non statistical approaches could draw on the experience dissemination practice on administrative or complex, combined data (like linked employer employee data). The authors outline indirect ways in which these kinds of data are available for researchers. The main access paths to those kinds of data are remote access solutions or safe centers, where restricted and controlled access is possible. For many big datasets, confidentiality risks of disseminating data may be so high that it is nearly impossible to share unrestricted-use microdata. The ideal scenario is that big data are held by trusted or trustworthy institutions(National Academies, 2014). As Karr and Reiter note, the data access “model of the future will be to take the analysis to the data rather than the data to the analyst or the analyst to the data. Big data is too big to take to the users.”

To overcome the dilemma of preventing privacy and make detailed empirical research possible, Karr/Reiter are in favor of a big data access as an integrated system including:

1. Unrestricted access to highly redacted data followed with
2. Means for approved researchers to access the confidential data via remote access solutions, glued together by

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

3. Verification servers that allows users to assess the quality of their inferences with the redacted data so as to be more efficient with their use (if necessary) of the remote data access.

Or in the words of Dwork: different privacy requires a new way of interacting with data, in which the analyst access data only through a privacy mechanism, and in which accuracy and privacy are improved by minimizing the viewing of intermediate results. But this is a foreign concept to data analysts. A major challenge will be to convince researchers working on big data to agree to these kinds of scenarios, both because big data are much more complex as the data we have worked before and the knowledge of the data generating process matters. Research is necessary to ensure that the access to big data will preserve the advantages of big data – so that the act of data redaction does not negatively affect data volume, the time taken to redact data does not significantly affect velocity, and there are minimal consequences on data variety in terms of the ability to keep all different data sources in a redacted dataset.

Dwork's conclusion is clarity itself in terms of establishing the future research agenda. By far the hardest problem is addressing the social challenges of a changing world, in which highly detailed research datasets are expected to be shared and reused, linked and analyzed, for knowledge that may or may not benefit the subject. Complexity of this type requires a rigorous theory of privacy and its loss. Other fields – economics, ethics, policy – cannot be brought to bear without a currency, or measure of privacy, with which to work. In this connected world, we cannot discuss trade-offs between privacy and statistical utility without a measure that captures cumulative harm.

Chapter 12

Extracting Information from Big Data: A Privacy and Confidentiality Perspective

Frauke Kreuter and Roger Peng

Introduction

Big data pose several interesting and new challenges to statisticians and others who want to extract information from data. As Robert Groves pointedly commented, the era is “appropriately called Big Data as opposed to Big Information,”¹ because there is a lot of work for analysts before information can be gained from “auxiliary traces of some process that is going on in the society.” The analytic challenges most often discussed are those related to three of the Vs that are used to characterize big data.² The *volume* of truly massive data requires expansion of processing techniques that match modern hardware infrastructure, cloud computing with appropriate optimization mechanisms, and re-engineering of storage systems.³ The *velocity* of the data calls for algorithms that allow learning and updating on a continuous basis, and of course the computing infrastructure to do so. Finally, the *variety* of the data structures requires statistical methods that more easily allow for the combination of different data types collected at different levels, sometimes with a temporal and geographic structure.

However, when it comes to *privacy* and *confidentiality*, the challenges of extracting (meaningful) information from big data are in our view similar to those associated with data of much smaller size, surveys being one example. For any statistician or quantitative working (social) scientist there are two main concerns when extracting information from data, which we summarize here as concerns about *measurement* and concerns about *inference*. Both of these aspects can be implicated by privacy and confidentiality concerns.

Measurement By questions of *measurement* we mean: Do the data contain the right key variables and all covariates to answer the research question? Are there any unobserved variables that confound the analysis when uncontrolled for? Is there systematic measurement error in the measures? What is the best unit of analysis on which the measures are taken?

Inference By questions of *inference* we mean: Do we understand the sampling process? Are all units we need in the analysis? Are certain units systematically missing?

Do some units appear multiple times? Do we have all measures on all units that we need?
Whom do the units represent?

In settings where data collection (big or small) is *designed* to answer specific scientific research questions, the answers to the questions above are (usually) relatively easy to determine or, better yet, are taken into account when the data are collected. But as soon as data are used for purposes other than those for which they were collected, or when the *data-generating process* breaks down – as is unfortunately often the case, both with data collected for experimental purposes and with data from sample surveys and censuses – this is no longer the case. Neither is it the case for most big data sources.

There are many reasons for breakdowns in the data-generating process of traditional data as well as big data; actual and perceived privacy and confidentiality threats can be one of them, though for traditional data usually more on the measurement side than on the inference side. This might be very different with big data. Irrespective of the size of the data at hand, researchers have to understand the data-generating process in order to determine whether statistical model assumptions are met and to know whether all relevant information is indeed included in the data for a given research question. Here is where we see a key difference between data commonly referred to as big data and more traditional data sources.

For traditional data, methodology has been developed to overcome problems in the data-generating process. Among survey methodologists, a guiding principle for detecting problems is the total survey error framework,⁴ and statistical methods for weighting, calibration, and other forms of adjustment⁵ are commonly used to mitigate errors in the survey process. Likewise for ‘broken’ experimental data, techniques like propensity score adjustment and principal stratification are widely used to fix flaws in the data-generating process.⁶ Some of this methodology is likely transferable to big data, but some of it might need to be tweaked. Most likely solutions will lie in a combination of traditionally designed data and big data. However, such solutions of *data linkage and information integration* are themselves threatened by concerns about privacy and confidentiality.

Data-Generating Processes

When we look closely, it is the data-generating process that differentiates big data from more traditional types of data, such as censuses, sample surveys, and experimental data. Even with more traditional data, we would, at least once in a while, see data of large volume (e.g. the U.S. census), data of high velocity (e.g. Nielsen TV meter),⁷ and data with a wide variety of different variables and a complex structure (e.g. National Health and Nutrition Examination Survey).⁸ What is different is that the data collection for all three of these examples was specifically *designed* with a research question in mind.

In the case of designed data collection, the research questions are translated into concepts that need to be measured (e.g. employment), measurement instruments are developed to best capture these concepts (e.g. people that did any work for pay or profit during the survey week, or those with a job but not working during the survey week because of illness, vacation, leave, etc.),⁹ populations are defined for which measures should be taken (e.g. U.S. civilian non-institutionalized population 16 years and older),¹⁰ and (usually) samples are drawn to make data collection more efficient. If used for official statistics these design features are transparent and made public. For other data products the Transparency Initiative of the American Association for Public Opinion Research¹¹ now has guidelines in place to match such transparency. A slightly different approach is taken by ESOMAR but also with the goal of creating transparency for the data-generating process.¹²

It is this transparency that allows an evaluation of the quality and usefulness of the data for a given research question – for example, knowledge about people who were sampled to be in the survey but could not be reached, or decided not to participate. If the error mechanism is known, researchers can evaluate potential bias. Statisticians express such bias, for example for an estimated mean, as a function of the rate of response and the difference between respondents and nonrespondents or, if response is seen as a stochastic process, as $\text{Bias}(\bar{y}_r) = \frac{\sigma_{yp}}{\bar{p}}$ where σ_{yp} is the covariance between y and the propensity to (in this case) respond to the survey, p , and \bar{p} is the mean propensity over the sample. If, for example, sampled cases with privacy concerns are less likely to participate in a survey¹³ but those privacy reasons are unrelated to the topic of the survey, then no bias will appear due to (this) breakdown in the process.¹⁴

In contrast, the data-generating process of many data streams is less transparent and much less under the control of the researcher. Many big data streams arise through systems that create data as a byproduct of human behavior, usually in interaction with a computer. Credit card transactions, loyalty cards, Twitter posts, Facebook entries, cellphone usage, all those behaviors leave electronic traces that arise *organically*.¹⁵ It is this difference between *designed* and *organic* data that poses extra challenges to the extraction of information.

Note, some big data have features of designed data, such as environmental monitors which are built to measure specific types of pollution. Their locations are carefully selected and their measurement times are either selected by design or continuous, removing any ‘organic’ element from the data-generating process. However, even in this example, the data collected by environmental monitoring agencies are often used for purposes for which they were not designed. The most common example perhaps is estimation of the health impacts of ambient air pollution.

The use of data for other than the designed purpose is also reflected in a comment by Horrigan, who characterizes big data as “nonsampled data, characterized by the creation of databases from electronic sources whose primary purpose is something other than statistical inference.”¹⁶ In designed data (ideally) all measures taken have to fulfill a purpose with respect to the research question (and a related statistic of interest), whereas with big data, the data fulfill a purpose for a particular action (i.e. the credit card payment, the reception of a phone call, or the expression of ‘liking’), but not for a particular research question unless the research question is a mere measure of the number of actions of that particular type.

The use of organic data for other than their immediate purpose within the system in which they arise is not peculiar to the big data area. For decades researchers have used administrative data in statistical analyses. Administrative data are the result of government or other administrative processes; prime examples of data often increasingly used for statistical analyses are tax data,¹⁷ or data produced through social security notifications.¹⁸ The variables contained in those data are usually those necessary for administering a program or administrative task, and they measure what is necessary to evaluate program-relevant features. When these data are used to extract information that go beyond the description of the program, researchers often quickly realize that the measures do not match the concepts they try to measure, and that often not all cases of interest are captured through these administrative data. For example, in the social security notifications that form the basis of the German administrative data used in a study of workplace heterogeneity and wage inequality,¹⁹ education is known to have many missing values or conflicting data within a person across employment spells, and certain groups such as civil servants and the self-employed are not included in the database at all.

Another mechanism by which big data often arise is the merging of multiple databases, studies, surveys, or datasets. Computing power today has allowed ‘megadatabases’ to be formed to address new and interesting questions using data in novel ways. However, two key concerns when merging takes place is that (1) now multiple datasets are being used for purposes for which they were not designed and (2) data that once retained a subject’s privacy may no longer do so when they are merged with another database.

Challenge: Measurement

There is much to say about possible measurement problems when using big data, for example challenges in text mining of Twitter or Facebook feeds to distinguish between sarcasm, irony, humor, actions vs. intentions, and so forth. Some recent examples have shown promise in solutions to automate such ‘feature selection’.²⁰ Here we focus on those measurement problems that are likely induced by privacy, confidentiality, and

related issues: (1) misreporting in anticipation of perceived data use and (2) lack of model-relevant variables.

We know from survey research that people misreport or underreport socially undesirable behavior and illegal activities to avoid unfavorable impressions and legal consequences should the information be disclosed and they overreport desirable behavior such as voting.²¹ Preliminary research reports reasonably accurate reflections of user characteristics on social network sites,²² though honesty is suspected to vary across personality types.²³ Not much is known about how reported attitudes and behaviors vary across networks, topics and (perceived) use of the data, and the associated *actual and perceived privacy* issues. In survey data collection self-administered modes have been shown to mitigate effects of social desirability compared to interviewer-administered modes.²⁴ But even if the mode is similar to those used in self-administered surveys (Web and text messages), the difference here is that the primary purpose of generating these data includes an intended readership. Also it is possible that what is shared is of good quality, but what matters most is not shared. Likewise a *privacy-related observation issue* can be the sharing/posting/reporting of information from people other than oneself. This trade-off between observation and non-observation error in big data deserves attention. It would be a sad state of affairs if increased research use of social media or other big data sources changes the very nature of these sites.

A second feature of many big data sources that can easily create measurement problems is the fact that while they are ‘case rich’ they tend to be ‘variable poor’.²⁵ The Billion Prices Project,²⁶ for example, can give timely and detailed information if the purpose is only to measure prices of consumer goods, but as soon as the research interest shifts to understanding how price changes lead to shifts in household expenditures a single source of big data will no longer suffice.²⁷ To perform household-level analysis consumption and purchase data need to be linked up at a household level, which might be doable if confidential credit card billing information from websites such as Amazon and Paypal are shared, though this obviously would require a data infrastructure that is currently not in place (see Chapter 11 in this volume, by Wilbanks).

A recent project at the University of Michigan uses Twitter data to predict initial claims for unemployment insurance using the University of Michigan Social Media Job Loss Index.²⁸ The prediction is based on a factor analysis of social media messages mentioning job loss and related outcomes.²⁹ As part of this project, auxiliary information was linked to Twitter messages via geographic location information, a powerful tool but not without its own *threats to privacy*. O’Neil and Schutt also point to the usefulness of location data for the correct interpretation of timestamp data by determining the user’s time zone.³⁰ Such information and other additional covariates can also be used to make

sense of outliers that would otherwise be removed from analyses, a common strategy in dealing with these types of data.³¹

Challenge: Inference

One interesting feature of big data is that because there is so much data, it is easy to overlook what is missing. At first glance one might think that there is no problem of inference with big data. After all, some say, with a constant stream of data and the technological capacity to capture all the data, sampling is no longer needed because the population can be analyzed instead.³² But what is the population? If Amazon is interested in the correlation between purchases of two different books on amazon.com, they can of course analyze the population and no inference problem appears. But if, for example, Twitter is used to measure health or political attitudes, the problem of inference is much harder and starts with a specification of the unit of analysis. Paul and Dredze reported good success with population-level metrics in their analysis of public health, but are hampered by the irregularity of Twitter postings when trying to answer questions on frequency of health problems, re-occurrence of health problems, and the like.³³

As with any data source, researchers have to ask themselves whether the data-generating process is appropriate for the research question and captures the population to which they are trying to make inference. In statistical terms, the challenge is to assess undercoverage, overcoverage, multiplicity, as well as other forms of missing data and (here) their relationship to privacy and confidentiality.

Undercoverage

Whether or not one captures all available data (at a given time point) or just a sample of the data, the dataset from which one draws cases (the frame) is almost never perfect. It is not hard to imagine that big data taken from the Internet suffer from *undercoverage*, from the exclusion of units in the target population (those without Internet access). According to estimates from the Current Population Survey, the Internet penetration rate among U.S. households was still below 80% in 2011,³⁴ and almost 30% of individuals report no Internet use anywhere (irrespective of access to a computer at home or elsewhere). Likewise, an estimated 30% of American adults do not have a credit card,³⁵ which means again about a third of the population is not represented in data from commercial vendors.

The proportion of people with Internet access who could be covered (if they all used a particular service that is accessed for web scraping) is, however, only one consideration and translates into *coverage bias* only when there is a sizable difference between those covered (e.g. by having Internet access or a credit card) and those uncovered. Coverage bias can be expressed as follows:³⁶

$$\text{bias}(\bar{Y}_c) = \frac{N_{uc}}{N_{pop}} [\bar{Y}_c - \bar{Y}_{uc}],$$

with N_{uc} reflecting the number of people who are undercovered, N_{pop} the total number of people in the population, \bar{Y}_c the mean for the covered units, and \bar{Y}_{uc} the mean for the undercovered units. Unfortunately, as can be seen from this equation, while the ratio of undercovered people to the population stays the same across all statistics derived from a given data source, the difference in means between covered and undercovered units can vary from variable to variable.

Several studies list potential indicators of bias when comparing those without Internet access to those with access.³⁷ In a German study, persons without Internet access are found to be less educated and slightly older than those with access.³⁸ In a U.S. study, Internet users are more likely to be in good health, to have health insurance, and to exercise regularly,³⁹ and for the elderly in the United States, significant differences between Internet users and non-users on financial and health variables were detected.⁴⁰ If we look at specific big data sources both the undercoverage rate and the potential for undercoverage bias is even bigger; in 2013 only 13% of the U.S. online population actively tweets⁴¹ and nearly half of Twitter users are under age 35 and only 2% are 65 or older.⁴²

Privacy and confidentiality concerns contribute to undercoverage, both at the person level but also at the level of individual postings or transactions.⁴³ Survey results from Google showed a strong correlation between privacy concerns and low engagement.⁴⁴ The concerns mentioned in the survey can be grouped into those about identity theft in the financial world, those about unwanted spam and solicitations in the digital world, as well as concerns about offline harm, stalkers, and employment risk in the physical world. A study of a selected group of college students found personality characteristics to be predictive of Facebook postings, with people who score high on the openness factor having less strict privacy settings, which may cause them to be susceptible to privacy attacks.⁴⁵ The individual perception of the risks of online participation can probably be characterized in terms of cost–utility equations, just like participation in surveys.⁴⁶ For many users the utility of connecting with friends, receiving and spreading information, and searching and purchasing online outweighs the costs and risks, but it remains interesting to see whether the recent privacy debates will change the equation for more people.⁴⁷

Overcoverage and Multiplicity

Overcoverage refers to the inclusion of units in the data that are not members of the target population, while multiplicity refers to multiple appearances of a single person. On Facebook one can, for example, find pages for dead people or animals, and errors in

geocodes⁴⁸ or disguised IP addresses might erroneously identify users as being part of a geographically defined population of interest when they are not. Similarly, Paul and Dredze observed in their analysis of Twitter data that “some tweets, particularly those about more serious illnesses like cancer, were in regards to family members rather than the user. This could throw off geographic statistics if the subject of a tweet lives in a different location than the author of the tweet.”⁴⁹

Multiplicity can occur for several reasons. Some people might open several Twitter accounts and Facebook pages, or use different credit cards to keep different aspects of their lives separate or to ease accounting. In simple scrapings of online data, those people would be included multiple times, which is no problem if a transaction is the unit of analysis but can be a problem if inference is made to a defined set of people. Depending on the research question, de-duplication is a problem most likely only solvable through data linkage or integration with other forms of additional identifying information, *creating again possible threats to privacy*.

Other Missing Data

Aside from undercoverage, data can be missing for a variety of other reasons. Infrequent postings by Facebook users would be an example of missing data. In principle those people are covered, but information on them is missing. Whether background information is missing because users choose to not reveal it, or because the data stream does not include these kinds of information, for analyses that try to explain behaviors or are interested in subgroup analyses, the lack of demographic and other background variables can be a big hindrance.⁵⁰

In a presentation at the 2013 FedCasic Conference, Link reported that about one-third of demographic information is missing from Facebook users.⁵¹ A study by Kosinski, Stillwell, and Graepel tried to infer demographic information from digital records of behavior – Facebook Likes – for 58,000 Facebook volunteers. They classified African Americans and Caucasian Americans correctly in 95% of cases, and males and females in 93% of cases, “suggesting that patterns of online behavior as expressed by Likes significantly differ between those groups, allowing for nearly perfect classification.”⁵² Good prediction accuracy was achieved for more private information such as relationship status (65%), substance use (73%), and sexual orientation (88% for males, 75% for females).

While it is unknown how well the prediction models would perform for those who did not volunteer to participate in the study, these results raise important concerns about the possibility of keeping information *purposefully private*. In the words of the authors:

[T]he predictability of individual attributes from digital records of behavior may have considerable negative implications, because it can easily be applied to

large numbers of people without obtaining their individual consent and without them noticing. Commercial companies, governmental institutions, or even one's Facebook friends could use software to infer attributes such as intelligence, sexual orientation, or political views that an individual may not have intended to share. One can imagine situations in which such predictions, even if incorrect, could pose a threat to an individual's well-being, freedom, or even life.⁵³

Other forms of missing data appear on an event level. Here too privacy concerns can be the explanation. Consider for example the use of cellphone data to analyze traffic patterns. Users of cellphones carry their devices with them and thus leave a trace in the data files of everywhere they go. However, some users might turn their devices off when they take trips that should not be known to anybody.

In his keynote lecture at the 2013 conference of the European Survey Research Association,⁵⁴ Couper points out that the behavior of people is likely to change based on *concerns about privacy*, and that this concern is likely to grow if the collection and use of big data become more broadly known. While some report behavioral changes in privacy and tracking settings⁵⁵ as result of concerns about data use, to our knowledge no data has been released by owners of major social network sites about the effects of increased awareness of data use. More research is needed on how to address issues of consent to data capturing and data use (see the discussion by Baracas and Nissenbaum in Chapter 2 of this volume). Given that the data generation process spans several countries, an exchange with professional organizations across the globe might be useful (see Chapter 8 in this volume, by Elias).

Solutions and Challenges: Data Linkage and Information Integration

Many of the measurement and inference problems – whether they occur because of privacy and confidentiality concerns or for other reasons – can be addressed through data linkage and other forms of information integration. Data linkage in particular has its own *privacy and confidentiality challenges*, but the other methods also have implications related to privacy and confidentiality.

In surveys, a variety of methods⁵⁶ that all fall under the heading of 'calibration' can correct for coverage and nonresponse errors by using auxiliary data. Auxiliary data can be any information that is available for the target population, either for each individual population unit or in aggregate form. In a household survey, counts of persons in groups defined by age, race/ethnicity, and gender may be published from a census or from population projections that are treated as highly accurate.⁵⁷ A necessary requirement for using those techniques is the *availability of comparable content and units of analysis* to link with information from more trusted sources. For example, if variables used for weighting are based on respondents' answers during an interview, then prior to data

collection care should be taken that the questions asked in the survey match those of the benchmarking source. Even for demographic variables, although this seems like a straightforward task, it is often not that simple. The current U.S. census, for example, allows for multiple-race categories, a feature many surveys or social network sites don't share. Non-probability surveys have started using such methods to attempt to make population estimates.⁵⁸ For some big data sources this would require cooperation with companies to obtain sufficient access to common variables, which can raise issues of confidentiality pledges given by the respective data collectors.

Another solution increasingly popular among surveys severely challenged by falling response rates, hard-to-reach populations, and stagnant survey budgets⁵⁹ is use of or linkage to administrative data to gain information about coverage bias, nonresponse bias, or measurement error in the survey data. Likewise, policymakers ask for evidence-based program evaluation, ideally with controlled randomized trials, but those often fail to be fully randomized when the practical challenges of data collection kick in. And current movements toward comparative effectiveness mean increased interest in answering questions regarding effects outside the narrow study population.⁶⁰ Here too statisticians have started to combine multiple datasets or integrate information from different sources to mitigate the shortcomings of the original study, and high hopes are expressed that organic big data can help here even more.

The act of combining several data sources, for example survey and administrative data, is not without its own challenges, and in several areas research is needed to ensure that this approach is one that can lead to success. Direct record linkage and statistical matching are two of the most common procedures for linking different data sources. Both methods involve trade-offs and rely on untested assumptions that threaten their practical use.

The quality of linked databases can be adversely affected by (a) inconsistencies between information collected from respondents in one dataset (e.g. the survey) and information contained in the other (e.g. an administrative database or social network site information) and (b) errors in the record linkage process itself.⁶¹ Such errors can be due to erroneous linkage, imprecise matching variables, or the lack of respondent consent to link their survey responses to administrative data. The impact of these issues on the quality of the linked datasets is often unknown. Evaluation of the procedures and criteria used in combining data sources and assessment of their impact on data quality are clearly needed if combined data are to become a key source of statistical information in the future.

Several studies have examined inconsistencies between observations made with survey and administrative data. A common assumption in those studies is that the administrative records are error free and suitable as a 'gold standard' against which survey responses can be compared, though this assumption often holds only for variables

necessary to administer a certain program. Similar studies are needed for the various big data sources as well.

Prior to linkage, informed consent is usually needed to ensure that respondents are aware of the risks and benefits involved in releasing information for research purposes. Research has shown that consent rates to linkage requests vary widely from study to study, by study population and country, and across the social, economic, and health fields, with percentages ranging anywhere from the mid-20s to the high-80s. But as in the discussion of coverage, here too it is not only the rate that matters but the difference between consenters and non-consenters. Studies have found gender, age, education, and wealth to be strongly related to the likelihood of consent; and health status, income item nonresponse, and prior-wave cooperation in longitudinal studies also correlate with linkage consent. Direct bias assessments with respect to a target variable, on the other hand, are rare.⁶²

We know by now that consent rates do not only vary by personal characteristic and study topic but also as a function of the consent request itself. So far no studies have experimented with the stated risk of data linkage, but research on informed consent for study participation have shown that stronger assurances of data confidentiality lead to higher response rates, with the exception that stronger assurances can backfire and lead to lower participation rates and more expressions of suspicion when the survey topic is innocuous and the confidentiality risk is minimal.⁶³ Recent experiments on the wording of data linkage requests have shown that the framing of the request as well as the consent placement influences consent rates.⁶⁴ The linkage indicator itself also matters, and greater hesitation arises when Social Security numbers are used. For large-scale linking, other identifiers that do not require self-reporting might be better suited. Geocodes of addresses might be the most fruitful approach,⁶⁵ though here too one can find differential error depending on the type of address used.⁶⁶ Geography also allows augmentation with other geographic information, which is a variable available in many big data sources. If higher levels of aggregation are used, requirements for exact placement are less stringent.

In contrast to direct record linkage, statistical matching, or the act of matching 'similar' records (in a statistical sense) based on a set of variables common to both datasets, does not require consent from survey respondents and can be performed on the entire sample. Several matching procedures have been proposed. However, many of these procedures depend on the assumption of conditional independence, asserting that any systematic differences between two or more data sources are explained by the matching variables. Only when this assumption is met will the true underlying distribution of all variables in both datasets be reflected in the matched dataset; otherwise, statistical relationships estimated from the matched dataset will be misleading.

One problem that cannot be solved by either technique is if people are missing for privacy reasons from all data sources that are intended for linkage. For example,

nonrespondents in surveys might also not be active on Facebook or decline the use of credit cards. To our knowledge there has so far been no systematic research on this issue.

Another real challenge to data integration is the linkage of data with different privacy restrictions. Several chapters in this volume, notably Chapter 9 by Greenwood et al. and Chapter 13 by Karr and Reiter, provide suggestions. Infrastructure needs to be in place that allows linkage and the addition of identifiable IDs to place people in locations, deduplicate, and so forth.⁶⁷

As the former census director Robert Groves summarized on the *Director's Blog*:⁶⁷ “The challenge to the Census Bureau is to discover how to combine designed data with organic data, to produce resources with the most efficient information-to-data ratio. [...] Combining data sources to produce new information not contained in any single source is the future. I suspect that the biggest payoff will lie in new combinations of designed data and organic data, not in one type alone.”

Discussion

Some of the challenges of big data may, in some sense, be addressed by adapting the approaches of ‘small data’ to a larger scale. Designed big data experiments are already commonplace, particularly so-called A/B testing in industry applications,⁶⁸ and likely will see much wider application. We will almost certainly encounter methodological challenges as we continue to move into big data and so the corresponding research will be essential. Another development that may serve to simplify many big data problems is the continuous refinement and sharpening of scientific questions and hypotheses. While one of the key advantages of big data is that it allows one to explore and discover previously unobserved patterns, such free-form exploration can lead to numerous problems with respect to inference and measurement, some of which we have discussed in this chapter. Eventually, science will move beyond the fascination with terabytes and petabytes, and the ability to collect data, and will focus more on the ability to analyze data and address specific questions. As investigators become more comfortable and familiar with the scope of big data, they will develop a much better understanding of what kinds of hypotheses can be addressed by such data and, more important, what kinds cannot.

Some research questions can only be addressed through the combination of data products. Thus research on linkage and matching needs to be expanded, covering the issues raised above and preparing for the analysis of linkage (consent) bias. When done successfully, linkage can be a rich resource, not just for substantive analysis when surveys, experiments, administrative data, and sensor data or other organic big data are respectively enriched, but also to evaluate the quality of the linked sources, including research on the missing elements in both.

As the march of big data progresses, continuous effort will be needed to ensure standards of transparency in the collection of data, so that there may be appropriate accountability and so that valid statistical inferences can be made. Principles developed for traditional data collection efforts, such as the use of metadata and paradata to describe and monitor data-generating processes, will need to be translated to modern high-volume data collection methods such as sensor networks, real-time data streams, and imaging modalities. So far there is no context-independent framework for evaluating data-generating processes, and more thought in this area would be useful.

Transparency in data collection goes hand in hand with transparency on the back end. The issues related to the reproducibility of any scientific investigation are potentially magnified in a big data analysis. Naturally, any linkage, data preparation and processing, analysis, and reporting must be fully transparent so that the handling of these tasks may be open to critique. However, because of the increased complexity of documenting the actions applied to big data, we will tend to rely on computer code descriptions rather than more traditional written forms. Ultimately, ensuring reproducibility through the proper documentation of any statistical analysis will be critical because the uniqueness and size of big data studies will make them less likely to be replicated in other investigations. (See Chapter 5 in this volume, by Stodden.)

Because many datasets are too large to travel from one place to the next, or privacy concerns may prevent them from being examined at an insecure location, data processing at the originator's site or through trusted third parties⁶⁹ will be an essential element in working with these combined data products. Examples of such procedures are already in place (e.g. the National Center for Health Statistics's Research Data Centers), though more work is needed to create legal agreements that allow for data access and also solve issues of liability, international border crossing, and mismatches in confidentiality standards. However, one potential upside of 'taking computation to the data' is that such remote data processing will require people to document and share the code for their analyses. Hence, there may be a built-in reproducibility safeguard if datasets must be analyzed remotely. It is these changes in data processing that will, if implemented in the right way, ultimately lead to greater transparency and with that, hopefully, to greater trust by individuals in the use of big data (at least in the context of statistical analyses), and to better data quality.

Acknowledgements We thank Fred Conrad, Stephanie Eckman, Lars Lyberg, and Eleanor Singer for critical review, Martin Feldkircher and Richard Valliant for comments and suggestions, and Felicitas Mittereder for research assistance.

¹ Robert Groves's talk at the World Bank event on December 19, 2012, "What Happens when Big Data Meets Official Statistics," <http://live.worldbank.org/what-happens-when-big-data-meets-official-statistics-live-webcast> (accessed January 20, 2014).

² C. O'Neil and R. Schutt, *Doing Data Science* (Sebastopol, CA: O'Reilly Media, 2014).

³ D. Agrawal, P. Bernstein, E. Bertino, S. Davidson, U. Dayal, M. Franklin, J. Gehrke, L. Haas, A. Halevy, J. Han, H. V. Jagadish, A. Labrinidis, S. Madden, Y. Papakonstantinou, J. Patel, R. Ramakrishnan, K. Ross, C. Shahabi, D. Suciu, S. Vaithyanathan, and J. Widom, "Challenges and Opportunities with Big Data 2011-1," Cyber Center Technical Reports, Paper 1, Purdue University, 2011,

<http://docs.lib.purdue.edu/cctech/1>.

⁴ R. M. Groves and L. Lyberg, "Total Survey Error," *Public Opinion Quarterly* 74, no. 5 (2010): 849–879.

⁵ For an overview, see R. Valliant, J. A. Dever, and F. Kreuter, *Practical Tools for Sampling and Weighting* (New York: Springer, 2013).

⁶ R. Rosenbaum and D. B. Rubin, "The Central Role of the Propensity Score in Observational Studies for Causal Effects," *Biometrika* 70, no. 1 (April 1983): 41–55; C. Frangakis and D. Rubin, "Principal Stratification in Causal Inference," *Biometrics* 58 (2002): 21–29.

⁷ See <http://www.nielsen.com/us/en/nielsen-solutions/nielsen-measurement/nielsen-tv-measurement.html>.

⁸ See <http://www.cdc.gov/nchs/nhanes.htm>.

⁹ See http://www.bls.gov/cps/cps_htgm.pdf (accessed January 20, 2014).

¹⁰ See http://www.bls.gov/cps/cps_over.htm#coverage (accessed January 20, 2014).

¹¹ See http://www.aapor.org/Transparency_Initiative.htm#.Ut2pHrQo7IU (accessed January 20, 2014).

¹² See <http://www.esomar.org/news-and-multimedia/news.php?idnews=104>.

¹³ For studies that show this effect, see E. Singer, "Confidentiality, Risk Perception, and Survey Participation," *Chance* 17, no. 3 (2004): 30–34; E. Singer, N. Mathiowetz, and M. P. Couper, "The Role of Privacy and Confidentiality as Factors in Response to the 1990 Census," *Public Opinion Quarterly* 57 (1993): 465–482.

¹⁴ For a meta-analysis on nonresponse bias, see R. Groves and E. Peytcheva, “The Impact of Nonresponse Rates on Nonresponse Bias: A Meta-Analysis. *Public Opinion Quarterly* 72, no. 2 (2008): 167–189.

¹⁵ R. M. Groves, “Three Eras of Survey Research,” *Public Opinion Quarterly* 75, no. 5 (2011): 861–871.

¹⁶ See <http://magazine.amstat.org/blog/2013/01/01/sci-policy-jan2013/> (accessed January 20, 2014).

¹⁷ For a recent example, see the Equality of Opportunity project (Ray Chetty, Nathaniel Hendren, Patrick Kline, Emmanuel Saez, and colleagues), which uses the Earned Income Tax Credit to study the impact of tax expenditure on intergenerational mobility, <http://www.equality-of-opportunity.org/>.

¹⁸ J. Schmieder, T. von Wachter, and S. Bender, “The Effects of Extended Unemployment Insurance over the Business Cycle: Evidence from Regression Discontinuity Estimates over 20 Years,” *Quarterly Journal of Economics* 127, no. 2 (2012): 701–752.

¹⁹ D. Card, J. Heining, and P. Kline, “Workplace Heterogeneity and the Rise of West German Wage Inequality,” *Quarterly Journal of Economics* 128, no. 3 (2013): 967–1015.

²⁰ For a discussion and examples of feature selection, see D. Antenucci, M. J. Cafarella, M. C. Levenstein, C. Ré, and M. D. Shapiro, “Ringtail: Feature Selection for Easier Nowcasting,” in *16th International Workshop on the Web and Databases (WebDB 2013)*, New York, <http://web.eecs.umich.edu/~michjc/papers/ringtail.pdf>.

²¹ R. Tourangeau and T. Yan, “Sensitive Questions in Surveys,” *Psychological Bulletin* 133, no. 5 (2007): 859–883.

²² V. R. Brown and E. D. Vaughn, “The Writing on the (Facebook) Wall: The Use of Social Networking Sites in Hiring Decisions,” *Journal of Business and Psychology* 26, no. 2 (2011): 219–225.

²³ K. Karl, J. Peluchette, and C. Schlaegel, “Who’s Posting Facebook Faux Pas? A Cross-Cultural Examination of Personality Differences,” *International Journal of Selection and Assessment* 18, no. 2 (2010): 174–186.

²⁴ F. Kreuter, S. Presser, and R. Tourangeau, “Social Desirability Bias in CATI, IVR, and Web Surveys: The Effects of Mode and Question Sensitivity,” *Public Opinion Quarterly* 72, no. 5 (2008): 847–865.

²⁵ A point made by M. P. Couper, “Is the Sky Falling? New Technology, Changing Media, and the Future of Surveys,” *Survey Research Methods* 7, no. 3 (2013): 145–156; K. Prewitt, “The 2012 Morris Hansen Lecture: Thank you Morris, et al., for Westat, et al.,” *Journal of Official Statistics* 29, no. 2 (2013): 223–231. This is different from the biomedical sciences where often a few cases have countless variables.

²⁶ See Bpp.mit.edu.

²⁷ M. Horrigan, “Big Data: A Perspective from the BLS,” *AMSTATNews*, January 1, 2013.

²⁸ See <http://econprediction.eecs.umich.edu/>.

²⁹ D. Antenucci, M. Cafarella, M. Levenstein, and M. Shapiro, “Creating Measures of Labor Market Flows Using Social Media,” presentation to the National Bureau of Economic Research, Cambridge, MA, July 16, 2012.

³⁰ O’Neil and Schutt, *Doing Data Science*.

³¹ T. Yan and K. Olson, “Analyzing Paradata to Investigate Measurement Error,” in *Improving Surveys with Paradata: Making Use of Process Information*, ed. F. Kreuter (Hoboken, NJ: Wiley, 2013).

³² V. Mayer-Schönberger and K. Cukier, *Big Data: A Revolution That Will Transform How We Live, Work and Think* (London: John Murray, 2013).

³³ M. Paul and M. Dredze, “You Are What You Tweet: Analyzing Twitter for Public Health,” in *5th International AAAI Conference on Weblogs and Social Media (July 2011)*, <http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/2880>.

³⁴ T. File, “Computer and Internet Use in the United States,” Current Population Survey Report P20-568 (Washington, DC: U.S. Census Bureau, 2013).

³⁵ See <http://www.statisticbrain.com/credit-card-ownership-statistics/>; <http://www.creditcards.com/credit-card-news/credit-card-industry-facts-personal-debt-statistics-1276.php> (accessed January 20, 2014).

³⁶ J. T. Lessler and W. D. Kalsbeek, *Nonsampling Error in Surveys* (Hoboken, NJ: Wiley, 1992).

³⁷ For a review, see S. Eckman, “Did the Inclusion of Non-Internet Households in the LISS Panel Reduce Coverage Bias?” Manuscript, Institute for Employment Research, Germany, 2014.

³⁸ M. Bosnjak, I. Haas, M. Galesic, L. Kaczmirek, W. Bandilla, and M. P. Couper, “Sample Composition Discrepancies in Different Stages of a Probability-Based Online Panel,” *Field Methods* 25, no. 4 (2013): 339–360.

³⁹ J. A. Dever, A. Rafferty, and R. Valliant, “Internet Surveys: Can Statistical Adjustment Eliminate Coverage Bias?” *Survey Research Methods* 2, no. 2 (2008): 47–62.

⁴⁰ M. P. Couper, A. Kapteyn, M. Schonlau, and J. Winter, “Noncoverage and Nonresponse in an Internet Survey,” *Social Science Research* 36, no. 1 (2007): 131–148; M. Schonlau, A. Van Soest, A. Kapteyn, and M. Couper, “Selection Bias in Web Surveys and the Use of Propensity Scores,” *Sociological Methods and Research* 37, no. 3 (2009): 291–318.

⁴¹ M. Link, “Emerging Technologies: New Opportunities, Old Challenges,” presented at FedCASIC Workshop, Washington, DC, March 19, 2013.

⁴² Paul and Dredze, “You Are What You Tweet.”

⁴³ See R. Kirkpatrick, “Big Data and Real-Time Analytics for Agile Global Development,” 2013, http://unstats.un.org/unsd/statcom/statcom_2013/seminars/Big_Data/BigData_UNGlobalPulse_Kirkpatrick.pdf (accessed January 20, 2014).

⁴⁴ J. Staddon, D. Huffaker, L. Brown, and A. Sedley, “Are Privacy Concerns a Turn-off? Engagement and Privacy in Social Networks,” in *Proc. 8th Symposium on Usable Privacy and Security (SOUPS '12)*, article 12, <https://static.googleusercontent.com/media/research.google.com/en/us/pubs/archive/38142.pdf>.

⁴⁵ T. Halevi, J. Lewis, and N. Memon, “Phishing, Personality Traits and Facebook,” Preprint arXiv:1301.7643 [CS.HC] (2013).

⁴⁶ E. Singer, “Toward a Benefit-Cost Theory of Survey Participation: Evidence, Further Tests, and Implications,” *Journal of Official Statistics* 27, no. 2 (2011): 379–392.

⁴⁷ See http://www.globalresearch.ca/nsa-spying-and-search-engine-tracking-technologies/5365435?utm_source=rss&utm_medium=rss&utm_campaign=nsa-spying-and-search-engine-tracking-technologies (accessed January 20, 2014).

⁴⁸ P. A. Zandbergen, “Accuracy of iPhone Locations: A Comparison of Assisted GPS, WiFi and Cellular Positioning,” *Transactions in GIS* 13, no. s1 (2009): 5–25.

⁴⁹ Paul and Dredze, “You Are What You Tweet.”

⁵⁰ Couper, “Is the Sky Falling?”

⁵¹ Link, “Emerging Technologies.”

⁵² M. Kosinski, D. Stillwell, and T. Graepel, “Private Traits and Attributes are Predictable from Digital Records of Human Behavior,” *Proceedings of the National Academy of Sciences* 110, no. 15 (2013): 5802–5805.

⁵³ Ibid.

⁵⁴ Couper, “Is the Sky Falling?”

⁵⁵ See <http://www.fastcoexist.com/3015860/people-are-changing-their-internet-habits-now-that-they-know-the-nsa-is-watching> (accessed January 20, 2014).

⁵⁶ E.g. post-stratification, general regression estimation, and raking.

⁵⁷ Valliant, Dever, and Kreuter, *Practical Tools for Sampling and Weighting*.

⁵⁸ R. Valliant and J. Dever, “Estimating Propensity Adjustments for Volunteer Web Surveys,” *Sociological Methods and Research* 40 (2011): 105–137; J. Dever, A. Rafferty, and R. Valliant, “Internet Surveys: Can Statistical Adjustments Eliminate Coverage Bias?” *Survey Research Methods* 2 (2008): 47–60. For a very accessible discussion of these approaches to non-probability samples, see Couper, “Is the Sky Falling,” and AAPOR, “Report of the AAPOR Task Force on Non-Probability Sampling,” *Journal of Survey Statistics and Methodology* 1 (2013): 90–143.

⁵⁹ See the volume edited by Douglas S. Massey and Roger Tourangeau, *The Nonresponse Challenge to Surveys and Statistics*, ANNALS of the American Academy of Political and Social Science Series 645 (Thousand Oaks, CA: Sage, 2013).

⁶⁰ E. A. Stuart, S. R. Cole, C. P. Bradshaw, and P. J. Leaf, “The Use of Propensity Scores to Assess the Generalizability of Results from Randomized Trials,” *Journal of the Royal Statistical Society, Series A* 174, no. 2 (2011): 369–386; S. R. Cole and E. A. Stuart, “Generalizing Evidence from Randomized Clinical Trials to Target Populations: The ACTG-320 Trial,” *American Journal of Epidemiology* 172 (2010): 107–115.

⁶¹ T. Smith, “The Report of the International Workshop on Using Multi-Level Data from Sample Frames, Auxiliary Databases, Paradata and Related Sources to Detect and Adjust for Nonresponse Bias in Surveys,” *International Journal of Public Opinion Research* 23 (2011): 389–402.

⁶² J. Sakshaug and F. Kreuter, “Assessing the Magnitude of Non-Consent Biases in Linked Survey and Administrative Data,” *Survey Research Methods* 6, no. 2 (2012): 113–122.

⁶³ E. Singer, H. J. Hippler, and N. Schwarz, “Confidentiality Assurances in Surveys: Reassurance or Threat?” *International Journal of Public Opinion Research* 4, no. 3 (1992): 256–268; N. Bates, J. Dalhamer, and E. Singer, “Privacy Concerns, Too Busy, or Just Not Interested: Using Doorstep Concerns to Predict Survey Nonresponse,” *Journal of Official Statistics* 24, no. 4 (2008): 591–612; M. P. Couper, E. Singer, F. G. Conrad, and R. M. Groves, “Experimental Studies of Disclosure Risk, Disclosure Harm, Topic Sensitivity, and Survey Participation,” *Journal of Official Statistics* 26, no. 2 (2010): 287–300.

⁶⁴ J. Sakshaug, V. Tutz, and F. Kreuter, “Placement, Wording, and Interviewers: Identifying Correlates of Consent to Link Survey and Administrative Data,” *Survey Research Methods* 7, no. 2 (2013): 133–144; F. Kreuter, J. Sakshaug, and R. Tourangeau, “Using Gain-Loss Framing to Obtain Respondent Consent to Link Survey and Administrative Data” (under review).

⁶⁵ For an overview and new developments, see R. Schnell, “Combining Surveys with Non-Questionnaire Data: Overview and Introduction,” in *Improving Surveys Methods: Lessons from Recent Research*, ed. U. Engel, B. Jann, P. Lynn, A. Scherpenzeel, and P. Sturgis (New York: Psychology Press, 2014); R. Schnell, “Getting Big Data but Preventing Big Brother: Entwicklung neuer technischer Lösungen für die datenschutzgerechte Zusammenführung personenbezogener Daten,” UNIKATE 45: Fusionen – Universität Duisburg-Essen.

⁶⁶ S. Eckman and N. English, “Creating Housing Unit Frames from Address Databases Geocoding Precision and Net Coverage Rates,” *Field Methods* 24, no. 4 (2012): 399–408.

⁶⁷ R. Groves, “Designed Data” and “Organic Data,” *Director’s Blog*, <http://directorsblog.blogs.census.gov/2011/05/31designed-data-and-organic-data/> (accessed January 20, 2014).

⁶⁸ R. Kohavi, R. Longbotham, D. Sommerfield, and R. M. Henne, “Controlled Experiments on the Web: Survey and Practical Guide,” *Data Mining and Knowledge Discovery* 18 (2009): 140–181.

⁶⁹ See http://unstats.un.org/unsd/statcom/statcom_2013/seminars/Big_Data/BigData_OECD_Wyckoff.pdf.

Chapter 13

Using Statistics to Protect Privacy

Alan F. Karr and Jerome P. Reiter

Introduction

Those who generate data – for example, official statistics agencies, survey organizations, and principal investigators, henceforth all called *agencies* – have a long history of providing access to their data to researchers, policy analysts, decision makers, and the general public. At the same time, these agencies are obligated ethically and often legally to protect the confidentiality of data subjects' identities and sensitive attributes. Simply stripping names, exact addresses, and other direct identifiers typically does not suffice to protect confidentiality. When the released data include variables that are readily available in external files, such as demographic characteristics or employment histories, ill-intentioned users – henceforth called *intruders* – may be able to link records in the released data to records in external files, thereby compromising the agency's promise of confidentiality to those who provided the data.

In response to this threat, agencies have developed an impressive variety of strategies for reducing the risks of unintended disclosures, ranging from restricting data access to altering data before release. Strategies that fall into the latter category are known as statistical disclosure limitation (SDL) techniques. Most SDL techniques have been developed for data derived from probability surveys or censuses. Even in complete form, these data would not typically be thought of as big data, with respect to scale (numbers of cases and attributes), complexity of attribute types, or structure: most datasets are released, if not actually structured, as flat files.

In this chapter, we explore interactions between data dissemination and big data. We suggest lessons that stewards of big data can learn from statistical agencies' experiences. Conversely, we discuss how big data and growing computing power could impact agencies' future dissemination practices. We conclude with a discussion of research needed and possible visions of the future of big data dissemination.

Experiences from Agencies

When disseminating a dataset to the public, agencies generally take three steps. First, after removing direct identifiers like names and addresses, the agency evaluates the disclosure risks inherent in releasing the data 'as is'. Almost always the agency determines that these

risks are too large, so that some form of restricted access or SDL is needed. We focus on SDL techniques here, because of the importance to researchers and others of direct access to the data. Second, the agency applies an SDL technique to the data. Third, the agency evaluates the disclosure risks and assesses the analytical quality of the candidate data release(s). In these evaluations, the agency seeks to determine whether the risks are sufficiently low, and the usefulness is adequately high, to justify releasing a particular set of altered data (Reiter 2012). Often, these steps are iterated multiple times; for example, a series of SDL techniques is applied to the data and subsequently evaluated for risk and utility. The agency stops when it determines that the risks are acceptable and the utility is adequate (Cox et al. 2011).

To set the stage for our discussion of SDL frameworks and big data releases, we begin with a short overview of common SDL techniques, risk assessment, and utility assessment. We are not comprehensive here; additional information can be found in, for example, Federal Committee on Statistical Methodology (1994), Willenborg and de Waal (2001), National Research Council (2005, 2007), Karr et al. (2010), Reiter (2012), and Hundepool et al. (2012).

Risk Assessment for Original Data

Most agencies are concerned with two types of disclosures, namely (i) identification disclosures, which occur when an intruder correctly identifies individual records in the released data, and (ii) attribute disclosures, which occur when an intruder learns the values of sensitive variables for individual records in the data (Reiter 2012). Often agencies fold assessment of attribute risk into assessment of identification risk. For concreteness, in this chapter, we focus on data regarding individuals. In the world of official statistics, many datasets contain information on establishments such as hospitals, manufacturers, and schools. Many of the problems we discuss here are significantly more challenging for establishment data (Kinney et al. 2011).

To assess identification disclosure risks, agencies make assumptions, either explicitly or implicitly, regarding what intruders know about the data subjects. Typical assumptions include whether the intruder knows that certain individuals participated in the survey, which quasi-identifying variables the intruder knows, and the amount of measurement error, or other error, in the intruder's data. For example, a common approach to risk assessment is to perform re-identification studies: the agency matches records in the original file with records from external databases that intruders could use to attempt identifications, matching on variables common to both files such as demographics, employment histories, or education. In such studies, the information on the external files operationally defines the agency's assumptions about intruder knowledge.

Agencies are particularly concerned about data subjects that are unique in the population with respect to characteristics deemed to be available to intruders, which often

are called *keys* in the SDL literature. An intruder who accurately matches the keys of a record that is unique in the population (on those keys) to an external file is guaranteed to be correct. Typically agencies only know that a record is unique on the keys in the sample. They have to estimate the probability that a data subject is unique in the population given that the subject is unique in the sample. See Skinner and Shlomo (2008) and Manrique-Vallier and Reiter (2012) for reviews of such methods. We also note that intruders who know that a particular record was in the sample can identify that record easily if it is unique in the sample on the keys.

Almost surely, the agency does not know very precisely what information intruders possess about the data subjects. Hence, and prudently, they examine risks under several scenarios, for example, different sets of quasi-identifiers known by intruders, and whether or not intruders know who participated in the study.

Statistical Disclosure Limitation Techniques

Most public use datasets released by national statistical agencies have undergone SDL treatment by one or more of the methods below.

Aggregation Aggregation turns atypical records – which generally are most at risk – into typical records. For example, there may be only one person with a particular combination of keys in a county, but many people with those characteristics in a state. Releasing data for this person with county indicators would pose a high disclosure risk, whereas releasing the data at the state level might not. Unfortunately, such aggregation makes analysis at finer levels difficult and often impossible, and it creates problems of ecological inferences. Another example is to report exact values only below specified thresholds, for example, reporting all ages above 90 as ‘90 or older’. Such top coding (or bottom coding) eliminates detailed inferences about the distribution beyond the thresholds. Chopping off tails also negatively impacts estimation of whole-data quantities (Kennickell and Lane 2006).

Suppression Agencies can delete at-risk values, or even entire variables, from the released data (Cox 1980). Suppression of at-risk values creates data that are not missing at random, which are difficult to analyze properly.

Data Swapping Agencies can swap data values between selected pairs of records – for example, switch counties for two households with the same number of people – to discourage users from matching, because matches may be based on ‘incorrect’ data (Dalenius and Reiss 1982). Swapping at high levels destroys relationships involving both swapped and unswapped variables. Even at low levels of swapping, certain analyses can be compromised (Drechsler and Reiter 2010; Winkler 2007).

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Adding Random Noise Agencies can add randomly sampled amounts to the observed numerical values, for example, adding a random deviate from a normal distribution with mean equal to zero (Fuller 1993). This reduces the potential to match accurately on the perturbed data and changes sensitive attributes. Generally, the amount of protection increases with the variance of the noise distribution; however, adding noise with large variance distorts marginal distributions and attenuates regression coefficients (Yancey et al. 2002).

Synthetic Data Agencies can replace original data values at high risk of disclosure with values simulated from probability distributions specified to reproduce as many of the relationships in the original data as possible (Reiter and Raghunathan 2007). Partially synthetic data comprise the original individuals with some subset of collected values replaced with simulated values. Fully synthetic data comprise entirely simulated records; the originally sampled individuals are not on the file. In both types, the agency generates and releases multiple versions of the data to enable users to account appropriately for uncertainty when making inferences. Synthetic data can provide valid inferences for analyses that are in accord with the synthesis models, but they may produce inaccurate inferences for other analyses. Despite being synthesized, synthetic data are not risk free, especially with respect to attribute disclosure.

Disclosure Risk and Data Utility Assessment after SDL

Disclosure Risk Assessment Many agencies perform re-identification experiments on SDL-protected data. In addition to matching records in the file being considered for release to external files, many agencies match the altered file against the confidential file. Agencies also specify conditional probability models that explicitly account for assumptions about what intruders might know about the data subjects and any information released about the disclosure control methods. For illustrative computations of model-based identification probabilities, see Duncan and Lambert (1986, 1989), Fienberg et al. (1997), Reiter (2005), and Shlomo and Skinner (2010).

It is worth noting that the concept of harm, such as a criminal act or loss of benefits, from a disclosure can be separated from the risk of disclosure (Skinner 2012). SDL techniques are designed to reduce risks, not harm. Agencies may decide to take on more risk if the potential for harm is low, or less risk if the potential for harm is high. We note that agencies could be concerned about the harm that arises from *perceived* identification or attribute disclosures – the intruder believes she has made an identification or learned an attribute, but is not correct – although in general agencies do not take this into account when designing SDL strategies. See Lambert (1993) and Skinner (2012) for discussion.

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Data Utility Assessment Data utility is usually assessed by comparing differences in results of specific analyses between the original and the released data. For example, agencies look at the similarity between a set of quantities estimated with the original data and with the data proposed for release, such as first and second moments, marginal distributions, and regression coefficients representative of anticipated analyses. Similarity across a wide range of analyses suggests that the released data have high utility (Karr et al. 2006). Of course, such utility measures only reveal selected features of the quality of the candidate releases; other features could be badly distorted.

The SDL literature also describes utility measures based on global functions of the data, such as differences in distributions (Woo et al. 2009). Our sense is that these methods are not widely used by agencies.

Current SDL and Big Data

Can typical SDL techniques be employed to protect big data? To be blunt, we believe the answer is no, except in special cases. We reach this opinion via informal, but we think plausible, assessments of the potential risk–utility trade-offs associated with applying these methods.

3.1 Disclosure Risks in Original Files

Confidential big data carry greater disclosure risks than the typical survey sample. Often confidential big data come from administrative or privately collected sources so that, by definition, someone other than the agency charged with sharing the data knows the identities of data subjects. This is in contrast to small-scale probability samples, which agencies believe inherently have a degree of protection from the fact that they are random subsets of the population, and because membership in a survey is rarely known to intruders. Confidential big data typically include many variables that, because the data arguably are known by others, have to be considered as keys, so that essentially everyone on the file is unique in the population. Further, as the quality of administrative databases gets better with time (particularly as profit incentives strengthen their alignment with information collection), agencies cannot rely on measurement error in external files as a buffer for data protection.

Effectiveness of SDL

Because of the risks inherent in big data, SDL methods that make small changes to data are not likely to be sufficiently protective; there simply will be too much identifying information remaining on the file. This renders ineffective the usual implementations of data swapping, such as swapping entire records across geographies. On the other hand, massive swapping within individual variables, or even within many small sets of variables, would essentially destroy joint relationships in the data. Suppression is not a viable

solution: so much would be needed to ensure adequate protection that the released data would be nearly worthless. Aggregation is likely to be problematic for similar reasons. When many variables are available to intruders, even after typical applications of aggregation, many data subjects will remain unique on the file. Very coarse aggregation/recoding is likely to be needed, which also leads to limited data utility. One potential solution is a fully synthetic, or barely partially synthetic, data release. With appropriate models, it is theoretically possible to preserve many distributional features of the original data. However, in practice it is challenging to find good-fitting models for joint distributions of large-scale data; to our knowledge there have been only a handful of efforts to synthesize large-scale databases with complex variable types (Abowd et al. 2006; Machanavajjhala et al. 2008; Kinney et al. 2011). Nonetheless, we believe that methods for generating massive synthetic databases merit further research.

Demands on Big Data

Through both necessity and desire, analyst demands on big data will be broader than what has been dealt with to date. We know some things about utility assessed in terms of ‘standard’ statistical analyses such as linear regressions (Karr et al. 2006), but almost nothing about utility associated with machine learning techniques such as neural networks or support vector machines, or data mining techniques such as association rules. Nor is it clear even what the right abstractions are. For instance, for surveys, SDL can to some extent be thought of as one additional source of error within a total survey error framework (Groves 2004), allowing use of utility measures that relate to uncertainties in inferences. For big data that are the universe (e.g. of purchases at Walmart), we do not yet know even how to think about utility, let alone measure it. To illustrate, consider partitioning analyses in which the data are split recursively into classes on the basis of a response and one or more predictor variables, producing a tree whose terminal nodes represent sets of similar data points. Measuring the nearness of two trees can be challenging, making it difficult to say how much SDL has altered a partitioning analysis.

Moreover, many demands on big data will be inherently privacy threatening. Most of today’s statistical analyses require only that the post-SDL data sufficiently resemble the original data in some low-dimensional, aggregate sense. For instance, if means and covariances are close, so will be the results of linear regressions. On the other hand, data-mining analyses such as searching for extremely rare phenomena, like Higgs bosons or potential terrorists, require ‘sufficiently resemble’ at the individual record level. Current SDL techniques are, virtually universally, based on giving up record-level accuracy, which reduces disclosure risk, in return for preserving aggregate accuracy, which is the current, but not necessarily the future, basis of utility.

Vision for the Future

We now present a vision for the future, including (i) discussions of what disclosure risk might mean and how it might be assessed with big data and big computation; (ii) how methods based on remote access and secure computation might be useful; and (iii) a vision for a big data dissemination engine involving interplay between unrestricted access, verification of results, and trusted access.

Changing Views of Privacy

It is possible, if not likely, that concomitantly with the move to big data, there also will be changes in the legal, political, and social milieu within which data release lies. Of the authors of this chapter, one (AK) is a baby boomer, and the other (JR) is a genX-er. Perhaps as a result of our research on data confidentiality, our views on privacy do not differ dramatically. But, almost daily, we observe others whose views do seem quite different from ours. These include cellphone users who discuss intimate details of their lives within earshot (and ‘lipshot’, since many of them seem aware that there are skilled lip readers everywhere) of others, social media users who seem not to realize how much privacy-compromising information a photograph can contain, and others. Whether these behaviors represent true changes in thinking about privacy, or will change as the individuals mature or societal attitudes evolve, remains to be seen. If the former, less protection of data may be required, although “who doesn’t care about privacy” may be a potent form of response bias in both surveys and administrative data. If the latter, less may change.

Also unclear is the denouement of the current trend of reluctance-to-refusal by individuals to provide data to government agencies. Some of the decline in response rates is the result of privacy concerns, some is the result of everyone’s increasingly complicated lives, some represents a belief that the government already has the information, and some is political opposition to *any* government data collection. It is hard not to think that response rates will continue to decline, but might they stabilize at a level at which statistical fixes still work?

What seemingly must change no matter what is the nature of the compact between data collectors and data subjects. Currently there is a major disequilibrium: official statistics agencies collect and protect much information about individuals and organizations that is readily accessible elsewhere, albeit sometimes there is a cost. Data subjects are clueless as to whether their information is protected adequately. Incentives to data subjects are seen as a means of payment for burden; subjects could (but are not now) also be compensated for (actual or risked) loss of privacy (Reiter 2011). A supremely intriguing thought experiment is to ask “What would happen if data subjects were promised no privacy at all, and simply paid enough to get them to agree to participate?” Would data quality be destroyed? Would the cost be affordable? We do not know.

The connection between privacy and big data is likewise evolving. Answering a few questions on a survey does not generate big data, nor does it cause most people to think anew about privacy. Collecting entire electronic medical records, DNA sequences, or videotapes of two years of driving (as in the Naturalistic Driving Study of the Virginia Tech Transportation Institute) may generate big data, and may change attitudes about privacy. Big data may also change what information is considered sensitive. Forty years ago, most people would have considered salary to be the most sensitive information about them. Today, a significant fraction of salaries are directly available, or accurately inferable, from public information. Instead, medical records may be more sensitive for many people. Partly this is because (in the same way that salaries were once seen as protectable) they are perceived still to be protectable; in addition, the risk associated with knowledge of medical records may be greater (e.g. loss of insurance or a job), as well as more nebulous.

SDL of the Future: A Framework

A significant change that big data and big computing will produce is the capability to enumerate all possible versions of the original dataset that could plausibly have generated the released data. To understand what this means, we sketch here a framework for this ‘SDL of the future’.

Let O be the original dataset and M be the released dataset after SDL is applied to O . Let \mathcal{O} denote the set of all possible input datasets that could have been redacted to generate M . In general, the extent to which an analyst or intruder can specify \mathcal{O} is a function of M , agency-released information about the SDL applied to O , and external knowledge. We denote this collective knowledge by (the σ -algebra) \mathcal{K} and for the moment restrict it to consist only of M and agency-released information. External knowledge is addressed in the next section.

To illustrate, suppose that O is a categorical dataset structured as a multiway contingency table containing integer cell counts. Suppose that M is generated from O by means of suppressing low-count cells deemed to be risky, but contains correct marginal totals. In this case, additional cells must almost always be suppressed in order to prevent reconstruction of the risky cells from the marginals. Figure 1 contains an illustration: in the table O , on the left, the four cells with counts less than 5 are suppressed because they are risky, and the cells with counts 5 and 6 are suppressed to protect them. In M , on the right, there is no distinction between the ‘primary’ and ‘secondary’ suppressions. Minimally, \mathcal{K} consists of M and the knowledge that cell suppression was performed; \mathcal{K} might or might not contain the value of the suppression threshold or information distinguishing primary from secondary suppressions. In the minimal case, \mathcal{O} consists of six tables: O and the tables obtained by putting 0, 2, 3, 4, and 5 as the upper left-hand entry and solving for the other entries. We denote these by O_0, \dots, O_5 , respectively. If the suppression threshold is known and zero is not considered risky, the first of these is ruled

out because applying the rules to it does not yield M . Every one of the other four is ruled out if \mathcal{K} distinguishes primary from secondary suppressions. Already one key implication for agencies is clear: the framework can distinguish what must be protected from what might be protected.

Equally important, the framework can distinguish analysts from intruders. The sardonic but apt comment that “One person’s risk is another person’s utility” demonstrates how subtle the issues are. Within our framework, both analysts and intruders wish to calculate the posterior distribution $P\{O = (\cdot)|\mathcal{K}\}$, but *use this conditional distribution in fundamentally different ways*.

Specifically, analysts wish to perform statistical analyses of the masked data M , as surrogates for analyses of O , and wish to understand how faithful the results of the former are to the results of the latter. (See also the section on microdata release, below.) Conditional on O , the results of an analysis are a deterministic (in general, vector-valued) function $f(O)$. To illustrate, for categorical data, $f(O)$ may consist of the entire set of fitted values of the associated contingency table under a well-chosen log-linear model. In symbols, given $P\{O = (\cdot)|\mathcal{K}\}$, analysts integrate to estimate $f(O)$:

$$\widehat{f(O)} = \int_O f(o) dP\{O = o|\mathcal{K}\}. \quad (1)$$

It is important to keep in mind that O depends on \mathcal{K} , even though the notation suppresses the dependence.

To illustrate with the example in Figure 1, if \mathcal{K} is only the knowledge that cell suppression was performed, then $O = \{O, O_0, O_2, O_3, O_4, O_5\}$ and $P\{O = (\cdot)|\mathcal{K}\}$ is the uniform distribution on this set. By contrast, if \mathcal{K} contains in addition the suppression rules, then $O = \{O, O_2, O_3, O_4, O_5\}$ and $P\{O = (\cdot)|\mathcal{K}\}$ is the uniform distribution on *this* set. Finally, if \mathcal{K} distinguishes primary from secondary suppressions, then $O = \{O\}$.

If the analysis of interest were a χ^2 test of independence, then, in the second case, the average of the five χ^2 statistics is 34.97, and independence would be rejected. Indeed, independence is rejected for all of O, O_2, O_3, O_4 , and O_5 so the analyst can be certain, even without knowing O , that independence fails.

The point is that big computing makes this approach feasible in realistic settings.

By contrast, intruders are interested in global or local maxima in $P\{O = o|\mathcal{K}\}$, which correspond to high posterior likelihood estimates of the original data O . In the extreme, *intruders maximize*, calculating

$$O^* = \arg \max_{o \in O} P\{O = o|\mathcal{K}\}. \quad (2)$$

We do not prescribe what intruders would do using O^* but assume only that any malicious acts would be done using O itself, for instance, re-identifying records by means of linkage to an external database containing identifiers.

This distinction allows the agency to reason in principled manners about risk and utility, especially in terms of how they relate to \mathcal{K} . *High utility* means that the integration in (1) can be performed or approximated relatively easily. *Low risk* means that the maximization in (2) is difficult to perform or approximate.

A central question is then: How large is the set \mathcal{O} of possible values of O given \mathcal{K} ? Of course, high utility and low risk remain competing objectives: when \mathcal{O} is very large, then the maximization in (2) is hard, but so may be the integration in (1). Because of the integration in (1), it may be more natural to view $|\mathcal{O}|$ as a measure of disclosure risk than as an inverse measure of data utility.

Incorporating External Information

The framework in the preceding section meshes perfectly with a Bayesian approach to external knowledge possessed by analysts or intruders. Once \mathcal{O} is known, such information exists independently of the knowledge embodied in \mathcal{K} , for instance, (1), so that it becomes completely natural to view as the product of a prior distribution on O and a likelihood function. See McClure and Reiter (2012a) for implementation of a related approach. In the example in the preceding section, the prior would simply weight the elements of \mathcal{O} on the basis of external knowledge.

More important from a computational perspective is that the integration in (1) can be performed by sampling from the posterior distribution $P\{O = (\cdot)|\mathcal{K}\}$, which is exactly what (Markov chain) Monte Carlo methods do!

Operational Implications

Most of today's (2013) big data are physical measurements that seem to need no SDL. There are, of course, very large transaction databases held by e-commerce websites, as well as databases containing information about telephone or e-mail communications. The extent to which any of the latter will be shared in any form is not clear. What is clear is that, in the short run at least, local storage and computing power will be supplemented or even supplanted by 'cloud computing', in which, transparently to the user, data and cycles reside in multiple physical machines.

Some implications of cloud computing are troubling to official statistics agencies. They may lose control over who has physical possession of their data, over who can view the data, and over how access to the data is controlled. The number of vulnerabilities increases in the cloud model, as does the possibility of secondary disclosure. In today's model, someone seeking illicitly to access Census Bureau data must penetrate Census Bureau servers, all of which are physically and electronically controlled by the Bureau. What happens if Census Bureau data might 'accidentally' be seen by someone attempting to access credit card records? Can the Census Bureau legitimately promise confidentiality

of records when ‘transparency’ means lack of knowledge rather than openness? Similar, and perhaps more challenging, issues arise for licensing of datasets.

These issues notwithstanding, we expect that the data access model of the future will be to take the analysis to the data rather than the data to the analyst or the analyst to the data. There are multiple reasons for this. Truly big data are too big to take to the users. Dataset size, coupled with the current impetus for availability of research datasets, seems to demand archives that can deal with complex issues of data format, metadata, paradata, provenance, and versioning. In our view, archives will also provide computational power. They will resemble today’s remote access servers (Karr et al. 2010), but with vastly increased computational power and flexibility.

Construction of such archives will require addressing issues we currently choose (mostly) to bypass by limiting server capability. If the data do require protection, perhaps the most pressing challenge is query interaction: both risk and utility increase in ways we do not currently understand when multiple queries are posed to the server. Answering one query may permanently preclude answering others (Dobra et al. 2002, 2003). Many current remote access servers in effect dodge this issue by severely limiting the space of allowable queries, for instance, by forbidding high-leverage variable transformations or limiting the degree of interactions. Others include manual review of both analyses and results, a strategy that is hopelessly non-scalable. Linkage to other datasets is rarely permitted, nor are exploratory tools such as visualizations. In virtually all of these instances, everything from sound abstractions to computational tools is lacking.

Because cloud data are distributed data, operational systems will require techniques for handling distributed data. A set of techniques from computer science known generically as secure multiparty computation (SMPC) has been shown to allow analyses based on sufficient statistics that are additive across component databases (Karr et al. 2005, 2007; Karr 2010; Karr and Lin 2010). These analyses include creation of contingency tables, linear and logistic regression (as well as extensions such as generalized linear models), and even iterative procedures such as numerical maximum likelihood estimation using Newton-Raphson methods. For almost all other analyses, the details remain to be worked out.

Is There a Future for Microdata Releases?

In view of the discussion in the preceding section, it is natural to ask whether there is a future for released microdata – individual records, as opposed to only summary statistics. We believe that there is, but that new tools will be required to attain it.

To begin, there *is* and will remain a case for releasing microdata. Microdata are essential to the education and training of early career researchers. Historically, there has been no substitute for working directly with data, and we do not believe that this will change. (Indeed, the risk that ‘big data’ means ‘disconnected from the data’ is both real and

disconcerting.) Perhaps more important, even skilled, mature researchers rarely know in advance which are the right questions to ask, and exploratory analyses dealing with the data themselves remain the best, if not only, path to the ‘right’ questions.

Currently available techniques for query-based analysis of distributed data using SMPC are notably poor in this respect. To illustrate, consider the example in Figure 2. There are three distributed datasets containing two variables, lying in the ranges shown. An analyst familiar with any one of the three databases would believe that the relationship between the two variables is linear, but, of course, it is quadratic instead. Existing query system models might thwart knowing the right question to ask. But even a small sample with intensive SDL from the integrated dataset would have made the right question apparent.

The question then: If highly redacted microdata are released publicly, for example, using novel methods of generating fully synthetic data, how can an analyst know whether he or she is on the right track to the right questions, which can then be posed to an archive/server? *Verification servers* are one technology that offers a solution (Reiter et al. 2009; McClure and Reiter 2012b). Briefly, a verification server is a web-accessible system based on a confidential database O with an associated public microdata release M – derived from O – that

- receives from the analyst a description of a statistical analysis Q performed on M ;
- performs the analysis on both M and O ;
- calculates one or more measures of the fidelity of $Q(M)$ to $Q(O)$;
- returns to the analyst the values of the fidelity measure(s).

The concept is illustrated pictorially in Figure 3. When the fidelity is high, the analyst may pose the query to a server, and receive a more detailed set of results.

Verification servers also could help reduce costs of accessing servers that host confidential data. Currently, and we expect also in the future, users who want access to confidential data via virtual or physical data enclaves are vetted by the data stewards. This involves cost that often is passed to the user, for example in the form of fees to access data. With the output from a verification server, users can decide if analysis results based on the redacted data are of satisfactory quality for their particular purposes. If so, they may choose to forgo the dollar and time costs of gaining access. Even users who are not satisfied with the quality of the results can benefit from starting with the redacted data. Storage and processing of big data is costly to data stewards, who likely will pass some costs to users. Analysts who have an informed analysis plan can improve their efficiency when using the server, thereby saving dollars and time.

Although attractive conceptually, verification servers remain an untested technology with both known and to-be-discovered risks. The former include risks shared with remote

access servers – unlimited and/or arbitrary queries, interaction among multiple queries, high-complexity variable transformations, subsetting of the data, and intruders with extreme computational resources. Too many, or too high precision, fidelity measures are among the latter. We do know that the latter *are* problems: if they are unaddressed, many SDL methods, including data swapping and top coding, can be reversed (Reiter et al. 2009). At the extreme, returning to the SDL framework we sketched above, with sufficiently many queries, sufficiently precise fidelity measures, and enough computational power, O can be recovered *exactly* from M .

Archive/server-based models also seem (at least currently) to be poor at handling record linkage, except in simple cases where the linkage amounts to a database join. Knowing which variables to link with, and understanding how uncertainties are affected by linkage, require – at least in exploratory stages – actual microdata.

One item of interest in this setting is that as a means of SDL, sampling is typically seen as ineffectual, at least by itself. If the goal is to produce an analysis-capable dataset M , most records must be retained. If no other SDL is performed and this information is known, then an intruder seeking to carry out the maximization in (2) need only worry about the possibility that the maximizer is not in M . Typically, this would be deemed insufficient protection. On other hand, if the goal is to produce an M that allows analysts to ask the right questions, small samples, especially if accompanied by weights, may be entirely adequate.

How Do Official Statistics Agencies Fit In?

Despite some of the challenges alluded to earlier (e.g. cloud computing), official statistics agencies are playing, and we expect will continue to play, significant roles in advancing methodology and practice for accessing big data. Many official statistics agencies that currently collect large-scale databases are experimenting with methods for providing access to these data. For example, as reported in presentations at the 2013 Joint Statistical Meetings, the Centers for Medicare and Medicaid Services (CMS) has contracted with the National Opinion Research Center (NORC) to develop an unrestricted-access, synthetic public use file for Medicare claims data. This file is intended to have limited analytic utility. It exists to help researchers develop methods and code to run on the actual data. After vetting, these researchers may be approved to access the restricted data in a data enclave setting. CMS and NORC had one team develop the synthetic datasets, while another team evaluated the disclosure risks, a separation we recommend as a general dissemination practice.

The Census Bureau has forged similar partnerships with researchers in academia to develop public use products for Longitudinal Employer-Household Dynamics (LEHD) data; see lehd.ces.census.gov for details.

At the same time, agencies are, properly and of necessity, conservative and slow to change. In particular, they must deal with extremely diverse sets of data users and other stakeholders. To illustrate, agencies have been slow to adopt multiple imputation as a means of dealing with item nonresponse, because not all users, even in the research community, are able to analyze such data. More ‘exotic’ technologies, such as synthetic data, other Bayesian methods, and differential privacy (Dwork 2006; also Chapter 14 in this volume), will replace existing methods, if at all, only at an evolutionary pace. One promising trend, however, is increasing agency attention to the fact that most people pay heed only to the decisions based on agency data, not to the data themselves (Karr 2012, 2013), which seems likely to yield important new insights about data utility.

Concluding Remarks

In spite of many steps toward wider data availability, legal, ethical, scale, and intellectual property restrictions are part of the foreseeable future (Karr 2014). “Make everything available to everyone” will not be ubiquitous, and SDL techniques are not likely to offer broadly the kind of one-off databases released by statistical agencies today. Statistical agencies already balance what to release to whom against other considerations, and this mode of thinking can, we believe, be crucial to big data.

For many big datasets, confidentiality risks of disseminating data may be so high that it is nearly impossible to share unrestricted-use microdata without massive data alterations, which call into question the usefulness of the released big data. We believe that methods for nonparametric estimation of distributions for large-scale data – a focus of significant research effort in the machine learning and statistical communities – offer potential to be converted to data synthesizers (Drechsler and Reiter 2011). Nonetheless, unrestricted-access, big datasets probably need to take on less ambitious roles than current agency practice permits; for example, they may serve as code testbeds or permit only a limited number of (valid) analyses. Verification servers, which promise to provide automated feedback on the quality of inferences from redacted data, could enhance the usefulness of such datasets, allowing users to determine when they can trust results and when they need to accept the costs of applying for access to the confidential data. Highly redacted datasets also should help users of remote query systems to identify sensible queries.

To conclude, we believe a way forward for big data access is an integrated system including (i) unrestricted access to highly redacted data, most likely some version of synthetic data, followed with (ii) means for approved researchers to access the confidential data via remote access solutions, glued together by (iii) verification servers that allow users to assess the quality of their inferences with the redacted data so as to be more efficient in their use (if necessary) of the remote data access. We look forward to seeing how this vision develops.

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Acknowledgement This work was partially supported by National Science Foundation grants CNS-1012141 and SES-1131897.

References

- Abowd, J., M. Stinson, and G. Benedetto. 2006. Final report to the Social Security Administration on the SIPP/SSA/IRS Public Use File Project. Technical report, U.S. Census Bureau Longitudinal Employer-Household Dynamics Program. Available at http://www.census.gov/sipp/synth_data.html.
- Cox, L. H. 1980. Suppression methodology and statistical disclosure control. *Journal of the American Statistical Association* 75:377–385.
- Cox, L. H., A. F. Karr, and S. K. Kinney. 2011. Risk-utility paradigms for statistical disclosure limitation: How to think, but not how to act (with discussion). *International Statistical Review* 79(2):160–199.
- Dalenius, T., and S. P. Reiss. 1982. Data-swapping: A technique for disclosure control. *Journal of Statistical Planning and Inference* 6:73–85.
- Dobra, A., S. E. Fienberg, A. F. Karr, and A. P. Sanil. 2002. Software systems for tabular data releases. *International Journal of Uncertainty, Fuzziness and Knowledge Based Systems* 10(5):529–544.
- Dobra, A., A. F. Karr, and A. P. Sanil. 2003. Preserving confidentiality of high-dimensional tabular data: Statistical and computational issues. *Statistics and Computing* 13(4):363–370.
- Drechsler, J., and J. P. Reiter. 2010. Sampling with synthesis: A new approach for releasing public use census microdata. *Journal of the American Statistical Association* 105:1347–1357.
- Drechsler, J., and J. P. Reiter. 2011. An empirical evaluation of easily implemented, nonparametric methods for generating synthetic datasets. *Computational Statistics and Data Analysis* 55:3232–3243.
- Duncan, G. T., and D. Lambert. 1986. Disclosure-limited data dissemination. *Journal of the American Statistical Association* 81:10–28.
- Duncan, G. T., and D. Lambert. 1989. The risk of disclosure for microdata. *Journal of Business and Economic Statistics* 7:207–217.
- Dwork, C. 2006. Differential privacy. In *Automata, Languages and Programming*, ed. M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, LNCS 4052, 1–12. Berlin: Springer.

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

Federal Committee on Statistical Methodology. 1994. *Report on Statistical Disclosure Limitation Methodology*. Statistical Policy Working Paper 22. Washington, DC: U.S. Office of Management and Budget.

Fienberg, S. E., U. E. Makov, and A. P. Sanil. 1997. A Bayesian approach to data disclosure: Optimal intruder behavior for continuous data. *Journal of Official Statistics* 13:75–89.

Fuller, W. A. 1993. Masking procedures for microdata disclosure limitation. *Journal of Official Statistics* 9:383–406.

Groves, R. M. 2004. *Survey Errors and Survey Costs*. New York: Wiley.

Hundepool, A., J. Domingo-Ferrer, L. Franconi, S. Giessing, E. Schulte-Nordholt, K. Spicer, and P.-P. de Wolf. 2012. *Statistical Disclosure Control*. New York: Wiley.

Karr, A. F. 2010. Secure statistical analysis of distributed databases, emphasizing what we don't know. *Journal of Privacy and Confidentiality* 1(2):197–211.

Karr, A. F. 2012. Discussion on statistical use of administrative data: Old and new challenges. *Statistica Neerlandica* 66(1):80–84.

Karr, A. F. 2013. Discussion of five papers on Systems and Architectures for High-Quality Statistics Production. *Journal of Official Statistics* 29(1):157–163.

Karr, A. F. 2014. Why data availability is such a hard problem. *Statistical Journal of the International Association for Official Statistics*, to appear.

Karr, A. F., W. J. Fulp, X. Lin, J. P. Reiter, F. Vera, and S. S. Young. 2007. Secure, privacy-preserving analysis of distributed databases. *Technometrics* 49(3):335–345.

Karr, A. F., S. K. Kinney, and J. F. Gonzalez, Jr. 2010. Data confidentiality – the next five years: Summary and guide to papers. *Journal of Privacy and Confidentiality* 1(2):125–134.

Karr, A. F., C. N. Kohnen, A. Oganian, J. P. Reiter, and A. P. Sanil. 2006. A framework for evaluating the utility of data altered to protect confidentiality. *The American Statistician* 60:224–232.

Karr, A. F., and X. Lin. 2010. Privacy-preserving maximum likelihood estimation for distributed data. *Journal of Privacy and Confidentiality* 1(2):213–222.

Karr, A. F., X. Lin, J. P. Reiter, and A. P. Sanil. 2005. Secure regression on distributed databases. *Journal of Computational and Graphical Statistics* 14(2):263–279.

Kennickell, A., and J. Lane. 2006. Measuring the impact of data protection techniques on data utility: Evidence from the Survey of Consumer Finances. In *Privacy in Statistical*

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

- Databases 2006*, ed. J. Domingo-Ferrer and L. Franconi, LNCS 4302, 291–303. New York: Springer.
- Kinney, S. K., J. P. Reiter, A. P. Reznek, J. Miranda, R. S. Jarmin, and J. M. Abowd. 2011. Towards unrestricted public use business microdata: The synthetic Longitudinal Business Database. *International Statistical Review* 79:363–384.
- Lambert, D. 1993. Measures of disclosure risk and harm. *Journal of Official Statistics* 9:313–331.
- Machanavajjhala, A., D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber. 2008. Privacy: Theory meets practice on the map. In *Proc. IEEE 24th International Conference on Data Engineering*, 277–286.
- Manrique-Vallier, D., and J. P. Reiter. 2012. Estimating identification disclosure risk using mixed membership models. *Journal of the American Statistical Association* 107:1385–1394.
- McClure, D., and J. P. Reiter. 2012a. Differential privacy and statistical disclosure risk measures: An illustration with binary synthetic data. *Transactions on Data Privacy* 5:535–552.
- McClure, D., and J. P. Reiter. 2012b. Towards providing automated feedback on the quality of inferences from synthetic datasets. *Journal of Privacy and Confidentiality*, 4(1): article 8.
- National Research Council. 2005. *Expanding Access to Research Data: Reconciling Risks and Opportunities*. Panel on Data Access for Research Purposes, Committee on National Statistics, Division of Behavioral and Social Sciences and Education. Washington, DC: The National Academies Press.
- National Research Council. 2007. *Putting People on the Map: Protecting Confidentiality with Linked Social-Spatial Data*. Panel on Confidentiality Issues Arising from the Integration of Remotely Sensed and Self-Identifying Data, Committee on the Human Dimensions of Global Change, Division of Behavioral and Social Sciences and Education. Washington, DC: The National Academies Press.
- Reiter, J. P. 2005. Estimating identification risks in microdata. *Journal of the American Statistical Association* 100:1103–1113.
- Reiter, J. P. 2011. Commentary on article by Gates. *Journal of Privacy and Confidentiality* 3: article 8.
- Reiter, J. P. 2012. Statistical approaches to protecting confidentiality for microdata and their effects on the quality of statistical inferences. *Public Opinion Quarterly* 76:163–181.

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

- Reiter, J. P., A. Oganian, and A. F. Karr. 2009. Verification servers: Enabling analysts to assess the quality of inferences from public use data. *Computational Statistics and Data Analysis* 53:1475–1482.
- Reiter, J. P., and T. E. Raghunathan. 2007. The multiple adaptations of multiple imputation. *Journal of the American Statistical Association* 102:1462–1471.
- Shlomo, N., and C.J. Skinner. 2010. Assessing the protection provided by misclassification-based disclosure limitation methods for survey microdata. *Annals of Applied Statistics* 4:1291–1310.
- Skinner, C. 2012. Statistical disclosure risk: Separating potential and harm. *International Statistical Review* 80:349–368.
- Skinner, C. J., and N. Shlomo. 2008. Assessing identification risk in survey microdata using log-linear models. *Journal of the American Statistical Association* 103:989–1001.
- Willenborg, L., and T. de Waal. 2001. *Elements of Statistical Disclosure Control*. New York: Springer.
- Winkler, W. E. 2007. Examples of easy-to-implement, widely used methods of masking for which analytic properties are not justified. U.S. Census Bureau Research Report Series, No. 2007-21. Washington, DC: U.S. Census Bureau.
- Woo, M. J., J. P. Reiter, A. Oganian, and A. F. Karr. 2009. Global measures of data utility for microdata masked for disclosure limitation. *Journal of Privacy and Confidentiality* 1:111–124.
- Yancey, W. E., W. E. Winkler, and R. H. Creecy. 2002. Disclosure risk assessment in perturbative microdata protection. In *Inference Control in Statistical Databases*, ed. J. Domingo-Ferrer, 135–152. Berlin: Springer.

Chapter 14

Differential Privacy: A Cryptographic Approach to Private Data Analysis

Cynthia Dwork

Introduction

Propose. Break. Propose again. So pre-modern cryptography cycled. An encryption scheme was proposed; a cryptanalyst broke it; a modification, or even a completely new scheme, was proposed. Nothing ensured that the new scheme would in any sense be better than the old. Among the astonishing breakthroughs of modern cryptography is the methodology of rigorously defining the goal of a cryptographic primitive – what it means to break the primitive – and providing a clear delineation of the power – information or computational ability – of the adversary to be resisted (Goldwasser and Micali 1984; Goldwasser et al. 1988). Then, for any proposed method, one proves that no adversary of the specified class can break the primitive. If the class of adversaries captures all feasible adversaries, the scheme can be considered to achieve the stated goal.

This does not mean the scheme is invulnerable, as the goal may have been too weak to capture the full demands placed on the primitive. For example, when the cryptosystem needs to be secure against a passive eavesdropper the requirements are weaker than when the cryptosystem needs to be secure against an active adversary that can determine whether or not a ciphertext is well formed (such an attack was successfully launched against PKCS#1; Bleichenbacher 1998). In this case the goal may be reformulated to be strictly more stringent than the original goal, and a new system proposed (and proved). This strengthening of the goal converts the propose–break–propose again cycle into a path of progress.

‘Big data’ mandates a mathematically rigorous theory of privacy, a theory amenable to measuring – and minimizing – cumulative privacy loss, as data are analyzed and re-analyzed, shared, and linked. This chapter discusses *differential privacy*, a definition of privacy tailored to statistical analysis of data and equipped with a measure of privacy loss. We will motivate and present the definition, give some examples of its use, and discuss the scientific and social challenges to adoption. We will argue that, whatever the measure on which the community eventually settles, data usage should be accompanied by publication of the amount of privacy lost, that is, its privacy ‘price’.

Toward Articulating a Privacy Goal

Following the paradigm of modern cryptography, we need to articulate what it is we are trying to prevent – what does it mean to ‘break’ privacy? What is a *privacy adversary* trying to achieve? Here, *adversary* is a term of art; we do not necessarily ascribe malicious intent to the government compliance monitor analyzing loan information data, to the citizen putting two and two together looking at census results and the neighbors’ blog posts, or to the research scientist poring over the published results of multiple studies of a small population of patients with a rare disease. We do, however, wish to understand and control what such parties can learn. Let us look at some examples of things that can go wrong when confidentiality is a stated goal.

Linkage Attacks In a linkage attack ‘anonymized’ records containing sensitive (say, medical encounter data) and ‘insensitive’ information (say, date of birth, sex, and ZIP code) are matched up, or ‘linked’, with records in a different dataset (say, voter registration records) on the basis of the insensitive fields. In fact exactly this happened, resulting in the identification of the medical records of the governor of Massachusetts (Sweeney 1997). In another famous example, the records of a Netflix user were identified among anonymized training data for a competition on movie recommendation systems, in this case by linkage with the Internet Movie Database (IMDb; Narayanan and Shmatikov 2008). Linkage attacks are powerful because a relatively small collection of seemingly innocuous facts often suffices to uniquely identify an individual. For example, among the observations in Narayanan and Shmatikov (2008): “with 8 movie ratings (of which we allow 2 to be completely wrong) and dates that may have a 3-day error, 96% of Netflix subscribers whose records have been released can be uniquely identified in the dataset.” In other words, the removal of all ‘personal information’ ultimately did not ‘anonymize’ the records in the Netflix training dataset.

Succinctly put, “‘De-identified’ data isn’t,” and the culprit is *auxiliary information* – that is, information from a source (voter registration records or IMDb) other than the database itself (HMO medical encounter data or Netflix Prize training dataset). This is not to say that many records, indeed most records, did not get identified *in these particular linkage attacks*. Rather, it demonstrates the fragility of the ‘anonymization’ protection. There are many ways of knowing that a family member, colleague, or public figure has watched a few movies on a few days. Were the ‘anonymized’ viewing habits of such a person made accessible, it could be very easy to learn things about her viewing habits that she would prefer, and is arguably entitled, to conceal.

The Statistics Masquerade Privacy problems do not disappear if we give up on ‘anonymizing’ individual records and instead release only statistics. For example, the *differencing attack* exploits the relationships between certain pairs of statistics, such as:

1. the number of members of the U.S. House of Representatives with the sickle cell trait and
2. the number of members of the House of Representatives, other than the Speaker of the House, with the sickle cell trait.

When taken together, these two statistics, each of which covers a large set of individuals, reveal the sickle cell status of the Speaker. Such dangerous pairs of queries are not always so easy to spot; indeed, if the query language is sufficiently rich the question of whether two queries pose such a threat is *undecidable*, meaning that there provably cannot be any algorithmic procedure for determining whether the pair of queries is problematic.

A more general adversarial strategy may be called the ‘Big Bang’ attack.¹ Given a large dataset, the attacker focuses on a relatively small subset, say, members of his extended family, of some size k . For concreteness, let us set $k = 128$. The attacker’s goal is to learn a single private bit – not necessarily the same bit – about each member of the extended family. For example, the attacker may wish to know if Aunt Wilma, who has two children, has had more than two pregnancies, and to know if Uncle William has a history of depression, and so on. Clearly, by asking k ‘counting queries’, each describing exactly one member of the extended family and the property in question, e.g. “How many people with the following identifying characteristics [description of Aunt Wilma and only Aunt Wilma] have had at least three pregnancies?” the attacker can learn the desired bits. But suppose the attacker does not receive perfectly accurate answers. Can introducing small inaccuracies into the query responses protect the family’s privacy? Intuitively this approach seems perfect: it renders useless any query about an individual, while not significantly distorting ‘statistical’ queries whose answers are expected to be fairly large.

The degree to which small distortions can protect against arbitrary counting query sequences depends on the size of ‘small’ compared to the number of queries. For example, there is a sequence of 128 counting queries with the following property. If the errors introduced are always of magnitude at most 1, then the adversary can reconstruct at least 124 of the 128 private bits. If the errors have magnitude bounded by 3, the number accurately reconstructed is still at least 92. With these same bounds on the magnitudes of the errors, taking $k = 256$, the adversary can correctly reconstruct at least 252 and 220 bits, respectively.

The general form of the bound is: if the magnitudes of the errors are all bounded by E , then at least $k - 4E^2$ bits can be correctly reconstructed (Dwork and Yekhanin 2008).² This turns out to be essentially the ‘right’ answer for arbitrary counting queries.

The Big Bang attack is concrete. It gives a simple and computationally very efficient method by which information released by a disclosure control method that yields accurate answers to a relatively small number of apparently statistical queries can be abused to

compromise privacy. The basic result is also very robust; with slight changes in bounds other attacks with similar outcomes can be launched using *random* linear queries, even if more than one-fifth of the responses are completely arbitrary (Dwork et al. 2007) and even under more general types of queries (Kasiviswanathan et al. 2012). The attacks are generally efficient and so, at least for non-Internet-scale datasets, can be launched against the entire database (i.e. $k = n$).

The Kindness of Strangers Now that the era of ‘big data’ is upon us, personal information – our searching, traveling, purchasing, and entertainment histories – flows from one individual to another via statistical learning systems. The set of search hits that receive clicks from one user affects which hits are returned to the next user; our presence on the road affects congestion, which in turn affects route suggestions; recommendation systems propose products based on observed paired purchases; Last.fm recommends music based on preferences of ‘similar’ users. Can these flows be used to compromise privacy?

Astonishingly, despite any potential adversary’s tremendous uncertainty regarding the dataset, such attacks are possible. The currently known examples require a small amount of auxiliary information. For example, a blog post about a recent purchase, taken together with the vendor’s (e.g. Amazon’s) continually changing public lists of ‘similar items’, can reveal purchases by the blogger that are not disclosed in the blog (Calandrino et al. 2011).

Smoking Causes Cancer Defining a *query* to be a function mapping datasets to some output range, we can view everything discussed so far – the production of microdata, statistics, predictors, classifiers, and so on – as queries. A user’s interaction with a dataset can be viewed as receiving responses to queries, and a natural attempt to articulate a privacy goal tries to relate what is known about a member of the dataset before, versus after, obtaining the response to a query or sequence of queries. Ideally, nothing would be learned about an individual from such an interaction.

This turns out to be unachievable if the responses are *useful*, in that they teach us things we did not know (Dwork 2006; Dwork and Naor 2010): We would like to learn facts such as “smoking causes cancer,” but in doing so our views and beliefs about individuals whom we know to smoke will change; for example, we will revise our predictions about their health. On the other hand, statistical analysis is meaningless without this type of *generalizability* – the whole point of a statistical database is to learn useful facts like “smoking causes cancer,” not just for the participants in the study but for human beings in general. Our definition of privacy must take into account this desired utility.

Framing Our Goal: In/Out vs. Before/After If the database teaches that smoking causes cancer, the bad (pays higher insurance premiums) and good (joins smoking cessation program) consequences for an individual smoker will be incurred *independent of*

whether or not the particular smoker is in the database. This suggests a new privacy goal: to ensure that, by participating in a dataset, one will be no worse off than one would be had one declined to participate. This is the heart of differential privacy.

Informed by our examples of attacks, we want this ‘In/Out’ privacy guarantee to hold regardless of the sources of auxiliary information – detailed information about family members and co-workers, blogs, other datasets, product recommendations, etc. – to which an attacker may have access.

An Ideal Scenario

Most of the literature on differential privacy assumes an ideal scenario in which the data are all held by a trusted and trustworthy *curator*, who carries out computations on the entire dataset and releases the results to the data analyst. Not to put too fine a point on it, the *data* (and the processing time and the power consumption, etc.) remain secret, the *responses* are published.

In reality, the data may not all reside in the same place – for example, the analyst may wish to study the combined medical records of multiple hospitals which do not choose to share their data with one another – or the data may reside, encrypted, in a semi-trusted cloud, where the cloud is trusted to keep data intact and to run programs, but it is desired that the cloud operator not have access to unencrypted data. For these situations, cryptography comes to the rescue. For example, the first may be addressed through *secure multiparty computation* (Prabhakaran and Sahai 2013) and the second through *fully homomorphic encryption* (Gentry 2009; Brakerski and Vaikuntanathan 2011). The role of cryptography in these cases is to abstract away the details and ensure that the system *looks just like*, or emulates, the ideal scenario.

Privacy-preserving data analysis is difficult even in the ideal scenario, but of course in any real implementation of differential privacy, whether in differentially private generation of synthetic data that are released to the public or in differentially private query/response systems, questions of physical security of the data, protection against timing and power consumption attacks (Kocher 1996; Kocher et al. 1999), errors in floating point implementations (Mironov 2012) do not go away, and must be addressed with additional technology.

Adversaries

Who are the ‘adversaries’ and what motivates them? To what kinds of information do they have access? Do they collude, intentionally or accidentally? Here it seems we are limited only by our imagination. We list a few examples.

- An abusive and controlling partner has copious auxiliary information about the victim, including dates and details of abuse, rendering useless anonymization of medical

records. Privacy of medical and police records may be a question of life or death in such a situation.

- Snake oil salesmen who prey on the desperate are financially motivated to find very sick individuals. Purchasing, through an online advertising system, the ability to track individuals based on the issuing of certain search queries could be very lucrative, and very easy.
- Blackmailers are motivated to find the unfaithful, for example, by analysis of telephony and mobility records.
- Learning the reading preferences of an employee or a prospective employee, via recommendation systems, can enable discrimination, or can inhibit intellectual exploration.
- A thief, observing patterns of power consumption through improperly aggregated smart grid data, learns good times to break into homes.
- Medical insurance companies wish to charge higher rates for customers with less healthy, or more risky, eating, exercise, and sexual habits, which may be revealed by purchase, search, and advertising click histories.
- A member of a middle-class community might find her relationships with her neighbors significantly altered were they able to deduce from an interactive census database that, despite her modest living style, she has a seven-figure income.
- ‘Anonymized’ social networks, published to enable social science research, may be vulnerable to ‘structural steganography’, revealing private social connections (Backstrom et al. 2007).
- Allele frequency statistics from a medical study, when combined with a DNA sample obtained on a date, can reveal membership in a case group (Homer et al. 2008).

1 Differential Privacy

On an intuitive level, the goal of differential privacy is to obscure the presence or absence of any individual, or small group of individuals, while at the same time preserving statistical utility. A little thought (and perhaps a lot of experience) shows that, absent constraints on auxiliary information, any such method must employ *randomization*. The introduction of randomness for preserving privacy is not new. For example, *randomized response* (Warner 1965) is a well-known technique from the social sciences used to survey respondents about embarrassing or illegal behavior. One version of randomized response goes as follows. Fix a specific yes/no question. The subject is told to flip a coin. If the outcome is heads, the subject answers the yes/no question honestly. If the outcome is tails, the subject flips a second coin and answers yes or no depending on the outcome of the second coin. Thus, in randomized response the subject randomizes his or her answers before handing them over to the researcher.³ Nonetheless, the researcher can recover

statistics such as the (approximate) fraction of subjects who engage in the behavior in question.

In our setting, the raw data have been collected by a trusted curator, who can therefore compute exact answers to these sorts of statistical queries. The twin concerns are privacy loss to potentially untrustworthy data users and privacy loss due to bad interactions among statistics (or other data products) published by virtuous, well-intentioned, users. We saw already in the case of the differencing and Big Bang attacks that exact answers can lead to loss of privacy, so, as in randomized response, our algorithms will inject carefully designed random noise. Algorithms that flip coins are said to be *randomized*.

Given a randomized data protection algorithm, a database, and a query, we get a probability distribution on query responses, where the probabilities come from the coin flips of the randomized algorithm. Similarly, given a randomized data protection algorithm, a database, and an adversary issuing queries and receiving responses, once we fix the randomness of the adversary, we get a probability distribution on transcripts, where the probabilities come from the coin flips of the data protection algorithm. Differential privacy says that the distribution on the outcome of any analysis is essentially unchanged independent of whether any individual opts into or opts out of the dataset. ‘Essentially’ is formalized by a parameter, usually called *epsilon* (ϵ), measuring privacy loss.

The property of being differentially private depends *only* on the data protection algorithm – something that the data curator, or ‘good guy’, controls. Thus, if an algorithm is differentially private then it remains differentially private no matter what an adversarial data analyst knows – including to which other datasets he or she has access. So differentially private algorithms automatically protect against linkage attacks. Differential privacy even guarantees that, if the analyst knows that the dataset is either D or $D' = D \cup \{p\}$, the outcome of the analysis will give at most an ϵ advantage in determining which of D, D' is the true dataset.

1.1 Formal Definition of Differential Privacy

A database is modeled as a collection of *rows*, with each row containing the data of a different individual. Differential privacy will ensure that the ability of an adversary to inflict harm (or good, for that matter) – of any sort, on any set of people – should be essentially the same, independent of whether any individual opts into, or opts out of, the dataset. This is achieved indirectly, simultaneously addressing all possible forms of harm and good, by focusing on the probability of any given output of a privacy mechanism and how this probability can change with the addition or deletion of any one person. Thus, we concentrate on pairs of databases (D, D') differing only in one row, meaning one is a subset of the other and the larger database contains just one additional row. (Sometimes it is easier to think about pairs of databases D, D' of the same size, say, n , in which case they

agree on $n - 1$ rows but one person in D has been replaced, in D' , by someone else.) Databases differing in at most one row are said to be *adjacent*.

Definition 1 (Dwork 2006; Dwork et al. 2006a, 2006b) A randomized mechanism M gives (ε, δ) -*differential privacy* if for all pairs of adjacent datasets D and D' and all $S \subseteq \text{Range}(M)$,

$$\Pr[M(D) \in S] \leq e^\varepsilon \times \Pr[M(D') \in S] + \delta,$$

where the probability space in each case is over the coin flips of M .⁴

S should always be very small, preferably less than the inverse of any polynomial in the size of the dataset.

For most of this chapter, we will take $\delta = 0$. This is sometimes referred to in the literature as ‘pure differential privacy’. Consider any possible set S of outputs that the mechanism might produce. Then the probability that the mechanism produces an output in S is essentially the same – specifically, to within an e^ε factor – on any pair of adjacent databases. This means that, from the output produced by M , it is hard to tell whether the database is D which, say, contains the data of the Speaker of the House, or the adjacent database D' , which does not contain the Speaker’s data. The intuition for privacy is: if you cannot even tell whether or not the database contains the Speaker’s data, then you cannot learn anything about the Speaker’s data (other than what you can learn from the data of the rest of the House). As this example shows, differential privacy defeats the differencing attack, even if the adversary knows everyone’s data except the Speaker’s!

1.2 Properties of Differential Privacy

Why is this a strong definition? We describe some properties implied by the definition itself; that is, *any* differentially private algorithm will enjoy the properties listed here.

Addresses Arbitrary Risks Any data access mechanism satisfying differential privacy should satisfy all concerns a participant might have about the leakage of her personal information, regardless of any auxiliary information – other databases, newspapers, websites, and so on – known to an adversary: even if the participant removed her data from the dataset, no outputs (and thus consequences of outputs) would become significantly more or less likely. For example, if the database were to be consulted by an insurance provider before deciding whether to insure a given individual, then the presence or absence of *any* individual’s data in the database will not significantly affect her chance of receiving coverage. Protection against arbitrary risks is, of course, a much stronger promise than the often-stated goal in anonymization of protection against re-identification. And so it should be! Without re-identifying anything, an adversary studying anonymized medical encounter

data could still learn that a neighbor, observed to have been taken to the emergency room (the ambulance was seen), has one of only, say, three possible complaints.⁵

Quantification of Privacy Loss Differential privacy is not binary; rather, privacy loss is *quantified*. For adjacent databases D, D' and any $y \in \text{Range}(M)$, the privacy loss incurred by observing y when the dataset is D is

$$L_{(M(D))||M(D'))}^{(y)} \ln \left[\frac{\Pr[M(D) = y]}{\Pr[M(D') = y]} \right].$$

In particular, $(\varepsilon, 0)$ -differential privacy ensures that this *privacy loss* is bounded by ε , and in general, $(\varepsilon, 0)$ -differential privacy ensures that this holds with probability at least $1 - \delta$. This quantification permits comparison of algorithms: given two algorithms with the same degree of accuracy (quality of responses), which one incurs smaller privacy loss? Or, given two algorithms with the same bound on privacy loss, which one permits the more accurate responses?

Automatic and Oblivious Composition Returning to the example of the sickle cell status of the Speaker of the House, we see that the method of ensuring privacy by only presenting counts of large groups *fails to compose*: each of the two counts in itself may not compromise privacy, but as a general method the approach cannot even tolerate an unfortunate set of two queries. In contrast, differential privacy immediately offers some composition guarantees. For example, given two differentially private computations, on the same or on different, possibly overlapping, databases, where one is $(\varepsilon_1, 0)$ -differentially private and the other is $(\varepsilon_2, 0)$ -differentially private, the cumulative privacy loss incurred by participating in (or opting out of) both databases is at worst $\varepsilon_1 + \varepsilon_2$. This is true even if the responses are generated obliviously of one another. This also teaches us one way to cope with high demand; for example, to ensure a cumulative loss bounded by ε^* over k computations, it is enough to ensure that each computation is $(\varepsilon^*/k, 0)$ -differentially private. Composition bounds are what allow us to reason about cumulative privacy loss of complex differentially private algorithms built from simple differentially private primitives (see e.g. Blum et al. 2005; Dwork et al. 2006b). This ‘programmability’ enables the construction of differentially private programming platforms (McSherry 2009; Roy et al. 2010).

Group Privacy Every $(\varepsilon, 0)$ -differentially private algorithm is *automatically* $(k\varepsilon, 0)$ -differentially private for groups of k individuals, for all k . This protects small groups, such as families. It will not necessarily offer protection for large groups, and indeed it should not! If two databases differ significantly, their statistics are expected to be different, and this should be observable if the databases are to be useful.

Closure under Post-Processing Differential privacy is immune to post-processing. A data analyst, without additional knowledge about the private database, cannot compute a function of the output of a differentially private algorithm M and make it less differentially private. That is, a data analyst cannot increase privacy loss, either under the formal definition or even in any intuitive sense, simply by sitting in a corner and thinking about the output of the algorithm, *no matter what auxiliary information is available*.

1.3 Achieving Differential Privacy

The differential privacy literature contains many astonishingly beautiful and powerful algorithmic techniques, some of which have given impressive results even on small datasets. For the most part, we will confine ourselves in this chapter to some simple techniques that, nonetheless, have nontrivial applications; the power of these techniques is illustrated in Section 2.

Differentially private algorithms hide the presence or absence of a single row. Consider a real-valued function f . The (worst-case, or global) *sensitivity* of f is the maximum absolute value by which the addition or deletion of a single database element can change the value of f :

$$\Delta f = \max_{D,D'} |f(D) - f(D')|,$$

where the maximum is taken over all pairs of adjacent databases. For vector-valued functions mapping databases to points in \mathbb{R}^k we extend this to the L_1 -norm:

$$\Delta f = \max_{D,D'} \|f(D) - f(D')\|_1 = \sum_{i=1}^k |f(D)_i - f(D')_i|.$$

Speaking intuitively, Δf is the worst-case difference that a differentially private algorithm for the function f will have to ‘hide’ in order to protect the presence or absence of an individual.

Definition 2 (The Laplace Distribution) The *Laplace distribution* (centered at 0) with scale b is the distribution with probability density function:

$$\text{Lap}(x|b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right).$$

The variance of this distribution is $\sigma^2 = 2b^2$. We will sometimes write $\text{Lap}(b)$ to denote the Laplace distribution with scale b , and will sometimes abuse notation and write $\text{Lap}(b)$ simply to denote a random variable $X \sim \text{Lap}(b)$.

We will now define the *Laplace mechanism*. As its name suggests, the Laplace mechanism will simply compute f , and perturb each coordinate with noise drawn from the Laplace distribution. The scale of the noise will be calibrated to the sensitivity of f (divided by ε).

Definition 3 (The Laplace Mechanism) Let f be a function mapping databases to \mathbb{R}^k .

The *Laplace mechanism* is defined as

$$M(D, f(\cdot), \varepsilon) = f(D) + (Y_1, \dots, Y_k),$$

where Y_i are i.i.d. random variables drawn from $\text{Lap}(\Delta f / \varepsilon)$.

Theorem 4 (Dwork et al. 2006b) *The Laplace mechanism preserves $(\varepsilon, 0)$ -differential privacy.*

Example 5: Counting Queries Queries of the form “How many people in the database are over six feet tall?” have sensitivity $\Delta f = 1$, since the presence or absence of any individual in D can affect the true answer by at most 1. Thus, the Laplace mechanism will return the true count perturbed by a random draw from $\text{Lap}(1/\varepsilon)$.

One way to handle $k > 1$ counting queries is via composition: by running each individual query with parameter ε/k we ensure that the cumulative privacy loss due to k queries is bounded by $k \cdot \varepsilon/k = \varepsilon$.

A second approach permits us to take advantage of the special properties of the particular set of counts we wish to compute, which may have lower sensitivity than the worst-case $\Delta f = k$, leading to better accuracy for the same privacy loss. An extreme case is illustrated in the next example.

Example 6: Histograms In a histogram query, the universe of possible database rows is partitioned into a fixed set of bins, say k , so that every database row belongs in exactly one bin. The true answer to the histogram query H when the database is D is, for each of the k bins in H , the number of rows in D that are in the given bin. For example, the bins may be income ranges $[0, 25K], [25K, 50K], \dots, [\geq 1,000,000]$ for the year 2011, so the query is asking about the distribution on incomes for the sample of the population that database D comprises. The sensitivity of a histogram query is 1, since the addition or deletion of one individual can change the count of at most one bin, and that change will have magnitude at most 1. Thus $\|H(D) - H(D')\|_1 \leq 1$ for all adjacent D, D' . Theorem 4 says that $(\varepsilon, 0)$ -differential privacy can be achieved by adding independently generated draws from $\text{Lap}(1/\varepsilon)$ to each output of $H(D)$. Compare this to the accuracy we would have obtained naively, by viewing the histogram query as k independent counting queries (one per bin) and applying the composition result mentioned in Section 1.2, which would have suggested adding noise drawn from $\text{Lap}(k/\varepsilon)$ to the count for each bin – a factor of k worse than what we get by thinking carefully about sensitivity.

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
 • in the book Cambridge University Press.

The Laplace mechanism provides *one* method for ensuring $(\varepsilon, 0)$ -differential privacy for any value of $\varepsilon > 0$. It does not necessarily give the best method for every setting. For example, the method gives poor responses if we seek answers to a superlinear (in the size of the database) number of queries, but other algorithms⁶ give meaningful responses even for a number of queries that grows *exponentially* in the size of the database! In Section 2 we briefly mention empirical results for computation of marginals using one of these algorithms.

Differentially private algorithms will typically be composed of several steps, and the Laplace mechanism is frequently employed at one or more of these individual steps. It is therefore an important *primitive*, or building block. We will see an example of extensive use of this primitive in constructing differentially private probability distributions in Section 2.

The Exponential Mechanism Differential privacy can be ensured for discrete output ranges by the *exponential mechanism* (McSherry and Talwar 2007). This mechanism uses a computation-specific *quality function* mapping (dataset, output) pairs to a real number. It assigns to each output a probability that grows exponentially in the quality, and then selects an output according to the resulting probability distribution. The exponential mechanism is another very important primitive. We will see an example of its use in Section 2 in selecting a ‘best’ (or perhaps just sufficiently good) distribution from a family of distributions.

The Gaussian Mechanism The Gaussian distribution is more tightly concentrated than the Laplace distribution, and may also be closer in form to other sources of variation in the data. The addition of Gaussian noise yields (ε, δ) -differential privacy with $\delta > 0$. Roughly speaking, (ε, δ) -differential privacy ensures that, given a pair D, D' of adjacent databases, with probability $1 - \delta$ the privacy loss on D with respect to D' will be bounded by ε . Recall that typically we have in mind cryptographically small values of δ .

Redefining sensitivity to be the maximum L_2 (i.e. Euclidean) distance $\|f(D) - f(D, D')\|_2$ on pairs of adjacent databases D, D' (rather than the L_1 difference $\|f(D) - f(D')\|_1$ we have discussed until this point), we obtain the following theorem for the *Gaussian mechanism*.

Theorem 7 *Let f be a function mapping databases to \mathbb{R}^k , and let Δ denote the L_2 -sensitivity of f . The ‘Gaussian mechanism’ that adds i.i.d. noise drawn from $\mathcal{N}(0, 2\Delta^2 \ln(2/\delta)/\varepsilon^2)$ to each of the k coefficients of f is (ε, δ) -differentially private.*

Like the Laplace mechanism, the Gaussian mechanism is an important primitive, especially in geometric algorithms for ensuring differential privacy (Hardt and Talwar 2010; Nikolov et al. 2013).

Beyond supporting the addition of Gaussian noise, this relaxation to $\delta > 0$ is also useful in differentially private programming. For example, suppose we have two methods for differentially private release of a given statistic, say, the median income. The first method, A , always maintains $(\varepsilon, 0)$ -differential privacy, but has poor accuracy on some inputs; the second method, B , has excellent accuracy, but its privacy loss exceeds ε on pathological inputs of a certain type, and only on these pathological inputs. We can use a differentially private test to determine whether it is safe to use algorithm B on the given dataset; but even if designed correctly there will be some very small probability, say, γ , that the test will erroneously indicate it is safe to use method B , potentially yielding probability γ of a large privacy loss. The best we can do in this case is to achieve (ε, γ) -differential privacy. We can make γ suitably small by designing the test to have an extremely small probability of error.

Remark 8 (Technical Remark) A more sophisticated analysis than that mentioned in Section 1.2 shows that the composition of k mechanisms, each of which is (ε, δ') -differentially private, $\varepsilon \leq 1$, satisfies $(\sqrt{2k \ln(1/\delta)}\varepsilon + k\varepsilon(e^\varepsilon - 1), k\delta' + \delta)$ -differential privacy for all $\delta > 0$ (Dwork et al. 2010). This translates into much better accuracy when $\varepsilon \leq 1/k$.

1.4 An Aside

The version of randomized response described at the very start of this section is $(\varepsilon, 0)$ -differentially private for $\varepsilon = \ln 3$. It is instructive to compare randomized response to the Laplace mechanism. For a single query such as “How many people in the data set ingested a controlled substance in the past week?” randomized response will yield an error on the order of \sqrt{n} , while the Laplace mechanism will yield an error on the order of $1/\varepsilon$, which is a constant independent of n .

What about multiple queries? Suppose we have a database with a single ‘sensitive’ binary attribute, and that attribute is recorded using randomized response. In this case the population can be sliced and diced at will, and privacy of this single attribute will be maintained. In contrast, the Laplace and Gaussian mechanisms appear to cease to give meaningful responses after just under n^2 queries.⁷ In this special case of a single sensitive attribute, randomized response is preferable once we require answers to a linear number of queries.⁸

2 Empirical Results

Since its inception, differential privacy has been the subject of intensive algorithmic research. There is also work on formal methods (e.g. Barthe et al. 2012) and a few programming platforms (e.g. McSherry 2009; Roy et al. 2010) that permit online interaction with the data.

OnTheMap (Machanavajjhala et al. 2008; Abowd et al. 2009), a privacy-preserving U.S. Census Bureau web-based mapping and reporting application that shows where people work and where workers live, and provides companion reports on age, earnings, industry distributions, and local workforce indicators, satisfies *probabilistic differential privacy*. While interactive, responding to queries issued by users of the site, the system gives exact answers computed from a privately generated *synthetic dataset* that was constructed offline (that is, before the website went live) from U.S. census data. To our knowledge, this is the only online system permitting anonymous members of the general public to issue queries while ensuring some form of differential privacy.⁹

We can think of a synthetic dataset as a collection of records with the same structure as real records, so that, for example, off-the-shelf software running on the original dataset could also run on the synthetic dataset. Given a (public) set \mathcal{Q} of queries and a (private) database D , the goal is to produce a synthetic dataset y with the property that for all $q \in \mathcal{Q}$, $q(y)$ yields a good approximation to $q(D)$.

A synthetic dataset does not preserve privacy simply by virtue of being synthetic. The process for generating the synthetic dataset matters. Moreover, it follows from the Big Bang attack that it is impossible to simultaneously preserve any reasonable notion of privacy and to release a synthetic dataset that answers ‘too many’ queries with ‘too much’ accuracy. Finally, there are also considerations of *computational complexity*, that is, the computational difficulty of creating a synthetic dataset with the desired properties. Two factors come into play here: the size of the set \mathcal{Q} of queries for which the curator promises correct answers, and the size of \mathcal{U} , the *universe* of possible data items. For example, if we wish to describe humans by their DNA sequences, the size of the universe is exponential in the length of the DNA sequence; if instead we describe the humans in our datasets by 6 binary attributes, the size of the universe is only $2^6 = 64$. Although theoretical results suggest formidable computational barriers to building synthetic datasets for certain large \mathcal{Q} or large \mathcal{U} cases (Dwork et al. 2009; Ullman and Vadhan 2011), the literature also contains some counterpoints with very encouraging experimental validation. We give two examples.

2.1 The MWEM Algorithm

The Multiplicative Weights with Exponential Mechanism (MWEM) algorithm (Hardt et al. 2012) optimizes an offline variant (Gupta et al. 2011) of the Private Multiplicative Weights update technique (Hardt and Rothblum 2010). A description of the techniques involved in these works is, unfortunately, beyond the scope of this book. The MWEM

algorithm was evaluated on *tables of marginals*. These are tables that answer counting queries of a special form. The universe \mathcal{U} of possible database elements are d -bit strings, representing, for each individual, the values of d binary attributes. A k -way marginal is specified by a set S of k of these d attributes together with an assignment to these attributes. Assuming binary attributes, there are $\binom{d}{k} 2^k$ k -way marginals. MWEM was evaluated on the sets of all 3-way marginals for three datasets discussed by Fienberg et al. (2011), for several values of $\epsilon \in [0,1]$. That is, \mathcal{Q} is the set of all $\binom{d}{3} 2^3$ 3-way marginals. The smallest dataset consisted of only 70(!) 6-attribute records. Of the $2^6 = 64$ possible settings of these bits, 22 appeared in the dataset (so the contingency table had 22 non-zero entries).

A byproduct of the MWEM algorithm is a synthetic database created solely from the privacy-preserving responses to the queries in \mathcal{Q} . In each case, the synthetic dataset was evaluated by computing the relative entropy, or Kullback-Leibler (KL) divergence, with respect to the real dataset and comparing this measurement with a report in the literature (Fienberg et al. 2011) that, roughly speaking, captures the best that can be done non-privately.¹⁰

Remarkably, even on the smallest dataset the relative entropy closely approaches the ideal when ϵ reaches about 0.7. For the other two datasets (665 records, 8 attributes, 91 non-zero cells; 1841 records, 6 attributes, 63 non-zero cells), the differentially private algorithms beat the non-private bounds once $\epsilon \approx 0.7$ and $\epsilon \approx 0.5$, respectively.

2.2 DP-WHERE

In this section we describe DP-WHERE (Mir et al. 2013), a differentially private version of the WHERE (Work and Home Extracted REgions) approach to modeling human mobility based on cellphone call detail records (Isaacman et al. 2012). For each individual, simultaneously, DP-WHERE protects *all* call detail records in the dataset; this is known in the literature as *user-level privacy* (here ‘user’ refers to a telephone user, not the data analyst).¹¹ The output of the system is a collection of *synthetic* call detail records. Example uses of synthetic call detail records include estimating daily ranges (the maximum distance a person travels in one day), modeling epidemic routing, and the modeling of hypothetical cities, in which the analyst creates a parameterized model of a city and user behavior patterns that cannot be observed in the real world, yielding the power to experiment with the effects of modifications to reality such as telecommuting (Isaacman et al. 2012).

2.3 A Sketch of DP-WHERE

Each call detail record corresponds to a single voice call or text message. Users making more than 120 calls in any hour are filtered out,¹² and it is assumed that the number of remaining users, denoted n , is known. Each of the n users is identified by an integer in

$\{1, \dots, n\}$. The calls were made in a metropolitan area divided by a grid into smaller geographic areas. Each call detail record is augmented with inferred home and work locations obtained by a combination of clustering and regression (Isaacman et al. 2011).¹³ Thus, in DP-WHERE each element in the dataset contains an id (number between 1 and n), date, time, latitude, longitude, and the inferred home and work locations.

The approach is to create several probability distributions, all in a differentially private manner. The synthetic call detail records are generated by appropriate sampling from these distributions.

Description of the Distributions

First, we list the distributions, briefly commenting on some of the differential privacy techniques used in their construction.

Home and Work For each of Home and Work, DP-WHERE computes a probability distribution on a square grid covering the metropolitan area in question, with a simple histogram query (Example 6). For example, for the Home distribution, the histogram reports, for each grid cell, the approximate (that is, noisy) number of users in the dataset whose home location is in this grid cell. In order to be able to transform this to a probability distribution – for example, to remove negative counts – post-processing techniques are applied that require no additional access to the true data (Hay et al. 2010).

Commute Distance DP-WHERE uses a coarser ‘commute grid’ for this computation. For each cell in the commute grid, the system computes an empirical distribution on commute distances for people whose home location falls in this cell.

We briefly describe the construction of one of these cumulative distribution functions, say, for the i th grid cell. The algorithm creates a *data-dependent* histogram of commute distances for the residents in this cell. Each histogram bin is a range of distances, and the (true, non-noisy) count in bin j is the number of users living in the i th grid cell whose true commute distance is in the range associated with the j th bin. There is some subtlety in determining the ‘right’ set of bins for this histogram. This is done by assuming that the commute distances for the residents of grid cell i are modeled by an exponential distribution of the form $\eta(x) = \lambda e^{-\lambda x}$ (each grid cell has its own distribution on commute distances, i.e. its own λ). The approach is to approximate the median using the exponential mechanism, set λ to be $\ln 2$ divided by this approximate median,¹⁴ and then define the histogram bins according to the deciles of the exponential distribution with parameter λ .

Calls per Day per User In this step, for a fixed, discrete, set of potential means $M = \{\mu_1, \mu_2, \dots, \mu_m\}$ and standard deviations $S = \{\sigma_1, \sigma_2, \dots, \sigma_s\}$, the algorithm computes

This is a preliminary version of the book *Privacy, Big Data, and the Public Good: Frameworks for Engagement*, ed. Julia Lane, Victoria Stodden, Stefan Bender, and Helen Nissenbaum (Cambridge University Press, 2014), www.cambridge.org/9781107637689.
© in the book Cambridge University Press.

a probability distribution on $M \times S$ from a histogram query (the cells of the histogram correspond to (mean, deviation) pairs (μ_i, σ_j)).

2-Means Clustering DP-WHERE runs a privacy-preserving 2-means clustering algorithm (McSherry 2009) to classify users based on a 24-dimensional probability vector describing their daily calling patterns. For each user i , using only the call records for this user, DP-WHERE first constructs a *non-private* probability vector P_i , describing for each hour $j \in \{1, 2, \dots, 24\}$ the fraction of i 's calls made in the j th hour of the day. These 24-dimensional P_i are clustered into two clusters, using a differentially private algorithm.

Hourly Calls per Location The last set of probability distributions generated by DP-WHERE yield, for each hour of the day, a probability distribution over grid cells, describing where the population as a whole is likely to be during the given hour. That is, the Hourly Location distribution for hour $j \in \{1, 2, \dots, 24\}$ yields a probability distribution on locations (grid cells) for the population as a whole during the j th hour of every day covered by the dataset. Ideally, this would be done by counting, for each grid cell and hour, the number of calls made from that grid cell during that hour of the day, summed over the different days covered by the dataset.

These counts are highly sensitive: for each hour, the total sensitivity of this computation is 120 times the number of days!¹⁵ Thus, even though, for a fixed $j \in \{1, 2, \dots, 24\}$, DP-WHERE builds something like a histogram, with one cell for each cell of the geographic grid, the L_1 -sensitivity of this data structure is 120 times the number of days. Applying the Laplace mechanism would add noise of this magnitude to *each* of the grid cells, which makes for too much distortion overall. This difficulty is addressed using a *grouping and smoothing* technique (Papadopoulos and Kellaris 2013), in which geographically close grid cells are ‘merged’, essentially coarsening the geographic grid, to give a data structure with fewer cells, yielding lower overall distortion.

This completes the overview of the (differentially privately generated) distributions used in generating synthetic call detail records. The differential privacy techniques used are the Laplace mechanism for histogram queries, post-processing to transform counts to distributions, the exponential mechanism, differentially private k -means clustering, and grouping and smoothing.

Generation of Synthetic Call Detail Records

Once the distributions have been generated in a differentially private fashion, privacy under post-processing tells us that sampling these distributions presents no additional risk to privacy. Thus, although DP-WHERE generates synthetic users and synthesizes

movements for these users, the system can also publish the distributions and the data analysts can sample from them at will.

A synthetic user is generated by sampling a home location h from the Home distribution, sampling a commute distance d from the Commute Distance distribution for the commute grid cell corresponding to h , and weighting the cells at distance d from h according to their distribution under the Work distribution and sampling from the resulting distribution on cells at distance d to obtain a work location w . Having determined the home and work locations of the synthetic user, the final steps are to sample (μ, σ) according to the Calls per Day per User distribution and finally to sample one of the two calling pattern clusters obtained in the 2-means clustering.

Having generated the synthetic users, DP-WHERE ‘moves’ them between their home and work locations. Fix $i \in \{1, 2, \dots, n\}$. The procedure described next generates a day in the life of synthetic user i .

1. Generate a number N of calls to be made during the day by sampling from a normal distribution with mean μ and variance σ^2 .
2. Allocate the total number N of calls to be made in this day to the 24 different hours of the day, according to the calling pattern (cluster) to which synthetic user i was assigned. Assign the exact time within the hour by interpolating between the beginning and end of the hour.
3. For each call made by user i during hour j , choose the location – select between user i ’s home (h) and work (w) location – by sampling according to the (differentially privately generated) Hourly Calls per Location densities for these two locations during hour j .

Experimental Validation of DP-WHERE

Experiments were carried out using call detail records from actual cellphone use over 91 consecutive days. The dataset contains over one billion records involving over 250,000 unique phones chosen at random from phones billed to ZIP codes within 50 miles of the center of New York City.

As in WHERE, accuracy of DP-WHERE is measured by a ‘normalized’ Earth Mover’s Distance. The results vary according to the coarseness of the commute grid (in both WHERE and DP-WHERE) and the choice of the *total, cumulative* privacy loss (in DP-WHERE). For a commute grid cell size of 0.01° , (non-private) WHERE yields an average hourly error of 3.2150; when $\epsilon = 0.33$ (respectively, 0.23 and 0.13) this quantity is 3.5136 (respectively, 3.4066 and 5.3391). A coarser grid cell size of 0.05° yields 3.0871 for WHERE and, respectively for these same values of ϵ , 4.5687, 5.1691, and 5.2754 for DP-WHERE.

Experiments showed that, in all cases, DP-WHERE as described above¹⁶ performed better than (non-private) WHERE based on public data, such as U.S. census data (and not call detail records; Isaacman et al. 2012). Thus, if the choice is between unfettered access to public data and differentially private access to the call detail records, these experiments show that differential privacy, even with $\epsilon = 0.13$, has the better utility.

Experiments were also carried out to measure the daily range, or the maximum distance between any two points visited by an individual in a day. The boxplots for daily range in DP-WHERE ($\epsilon = 0.23$), WHERE, and the real call detail records are qualitatively similar, with differences of 0.5–1.3 miles across the middle two quartiles (the smallest interquartile range of the three sets is 5.2 miles).

3 Challenges for Differential Privacy

The greatest *scientific* challenge for differential privacy is that, for a given computational task and a given value of ϵ , finding a low-error, differentially private algorithm can be hard. An analogy may be made to numerical analysis. Suppose, in the non-private world, we wish to compute a matrix decomposition. A naïve algorithm for the decomposition may be numerically unstable, so we first consult a textbook on numerical algorithms and write our program based on the stable algorithm in the text. It is easy now – but quite possibly the algorithm in the text was a PhD thesis when it was developed in the 1970s.

A different sort of challenge is posed by ‘non-algorithmic’ thinking in data analysis. From data cleaning through detailed investigation, many researchers who work with data do not, indeed cannot, provide an algorithmic description of their interactions with the data. With no algorithm for the non-private case, there is essentially no hope of finding a differentially private alternative. This is less of an issue in machine learning and the VLDB (Very Large Data Bases) communities, where the sheer volume of data rules out non-algorithmic approaches.

Differential privacy requires a new way of interacting with data, in which the analyst accesses data only through a privacy mechanism, and in which accuracy and privacy are improved by minimizing the viewing of intermediate results. But query minimization is a completely foreign concept to data analysts. A good analogy might be to running an industrial scale database without the benefit of query planning, leading to (literally) prohibitive computational costs.

By far the hardest to grapple with are the *social* challenges of a changing world, in which highly detailed research datasets are expected to be shared and reused, linked and analyzed, for knowledge that may or may not benefit the subjects, and all manner of information exploited for commercial gain, seemingly without limit. That this is fundamentally incompatible with privacy is proved by a host of lower bounds and attacks.¹⁷ What are we to make of this state of affairs? To paraphrase Latanya Sweeney (2012), computer science got us into this mess, can computer science get us out of it?

One thing seems certain: complexity of this type requires a mathematically rigorous theory of privacy and its loss. Other fields – economics, ethics, policy – cannot be brought to bear without a ‘currency’, or measure of privacy, with which to work. In this connected world, we cannot discuss trade-offs between privacy and statistical utility without a measure that captures cumulative harm over multiple releases.

Publish the Loss, and Pay a Fine for Infinity What should be the value of ϵ ? How should ϵ be meted out? How should ϵ depend on the nature of the data, the nature of the queries, the identity or affiliation of the data analyst? The anticipated social value of the investigation? The commercial utility? These questions will take time to sort out.

On a more philosophical level, consider an analogy to time: there are only so many hours in your lifetime, and once they are consumed you die. (This is sometimes worse than someone learning private information about you.) Yet, somehow, we as a society have found ways to arrive at values for an individual's time, and a fundamental part of that is the ability to quantitatively measure it.
(Dwork et al. 2011)

Differential privacy provides a measure that captures cumulative privacy loss over multiple releases. Whatever the measure of privacy loss on which the community ultimately settles, we should take a page from environmental law and require data usage to be accompanied by publication of privacy loss.¹⁸ In differential privacy, simply ensuring that the loss is finite helps to protect against many common avenues of attack. So let this be our starting point: publication of privacy losses – and a fine for infinite loss. If the data analyst cannot function without seeing raw data then, once the analyst has determined (in a non-private fashion) which statistics (or other data products) are to be released, the chosen statistics should be published using differential privacy (or whatever the community settles on), together with the privacy losses incurred in those calculations.¹⁹ The attention to privacy loss will raise awareness and lead to innovation and competition, deploying the talents and resources of a larger set of researchers and other marketers and consumers of data in the search for private algorithms.

Notes

¹ These are also known variously as *blatant non-privacy*, *reconstruction*, or *Dinur-Nissim* attacks, the last in homage to the computer scientists who first demonstrated attacks of this kind (Dinur and Nissim 2003).

² The attack involves computation of a Fourier transform and does not require knowledge of E .

³ Randomized response is also known as *the local model*, and it does not require a trusted curator; see Evfimievski et al. (2003), Blum et al. (2005), Dwork et al. (2006a), Kasiviswanathan et al. (2011), Hsu et al. (2012), and Duchi et al. (2013).

⁴ If $\varepsilon \ll 1$ then $e^\varepsilon \approx (1 + \varepsilon)$; for example, $e^{1/10} \approx 1.1052$.

⁵ Two of which, say, a broken limb and heart attack, might be ruled out when the neighbor is seen the next day, leaving only ‘panic attack’.

⁶ See Blum et al. (2008), Roth and Roughgarden (2010), Hardt and Rothblum (2010), Dwork et al. (2010), and Hardt et al. (2012).

⁷ Appearances can be deceptive: correlations between responses generated with independent noise can be exploited to extract surprisingly accurate approximate answers *on average*, even for a superpolynomial number of queries (Nikolov et al. 2013)!

⁸ See the literature on local learning mentioned in n.4, and in particular the careful treatment in Kasiviswanathan et al. (2011), for further understanding of the power and subtleties of randomized response.

⁹ The entry point is <http://lehdmap3.did.census.gov/>.

¹⁰ Any set of marginals determines the maximum likelihood estimator \hat{p} , which is the unique probability distribution in the log-linear model encoded by the given set of marginals that makes the observed dataset D the ‘most likely’ sample to have been observed. The bounds on KL divergence for the non-private case come from the KL divergence between \hat{p} and D . While considered important, small KL divergence is not necessarily viewed as a sufficient criterion for model selection.

¹¹ This is in contrast to *event-level* privacy, which would only hide the presence or absence of a single (or small number of) call record.

¹² These are assumed to be auto-dialers.

¹³ Since the clustering uses only information specific to the given user, together with global information about the locations of cell towers, the determination of these fields will not affect the privacy guarantee.

¹⁴ The median of the distribution $\eta(x) = \lambda e^{-\lambda x}$ is $\lambda^{-1}\ln 2$.

¹⁵ Recall that only users making more than 120 calls in a single hour have been filtered out.

¹⁶ That is, starting from real call detail records, generating the distributions in a differentially private fashion, creating synthetic users, and ‘moving’ these users.

¹⁷ It is also fundamentally incompatible with statistical power, where issues of false discovery arise.

¹⁸ See Hirsch (2006) for an investigation of basing privacy regulation on environmental law.

¹⁹ Note that this is still infinite loss – and should still incur a fine – because the *choice* of what to publish was made in a non-private fashion.

References

- Abowd, J., J. Gehrke, and L. Vilhuber. 2009. Parameter exploration for synthetic data with privacy guarantees for OnTheMap. In *Proc. Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality (Bilbao, Spain, 2-4 December)*. Available at <http://www.unece.org/fileadmin/DAM/stats/documents/ece/ces/ge.46/2009/wp.12.e.pdf> (accessed January 9, 2014).
- Backstrom, Lars, Cynthia Dwork, and Jon Kleinberg. 2007. Wherefore art thou r3579x? Anonymized social networks, hidden patterns, and structural steganography. In *Proc. 16th International Conference on World Wide Web*, 181–190.
- Barthe, Gilles, Boris Köpf, Federico Olmedo, and Santiago Zanella Beguelin. 2012. Probabilistic relational reasoning for differential privacy. In *Proc. POPL 2012*.
- Bleichenbacher, D. 1998. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS# 1. In *CRYPTO '98*, LNCS 1462, 1–12.
- Blum, A., C. Dwork, F. McSherry, and K. Nissim. 2005. Practical privacy: The SuLQ framework. In *Proc. 24th ACM Symposium on Principles of Database Systems (PODS)*, 128–138.
- Blum, A., K. Ligett, and A. Roth. 2008. A learning theory approach to non-interactive database privacy. In *Proc. 40th ACM SIGACT Symposium on Theory of Computing (STOC)*, 609–618.
- Brakerski, Z., and V. Vaikuntanathan. 2011. Efficient fully homomorphic encryption from (standard) LWE. In *Proc. 52nd Annual IEEE Symposium on Foundations of Computing (FOCS)*, 97–106.

- Calandrino, J., A. Kilzer, A. Narayanan, E. Felten, and V. Shmatikov. 2011. You might also like: Privacy risks of collaborative filtering. In *Proc. IEEE Symposium on Security and Privacy (SP)*, 231–246.
- Dinur, I., and K. Nissim. 2003. Revealing information while preserving privacy. In *Proc. 22nd ACM Symposium on Principles of Database Systems (PODS)*, 202–210.
- Duchi, John, Michael Jordan, and Martin Wainwright. 2013. Local privacy and statistical minimax rates. In *Proc. 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*.
- Dwork, C. 2006. Differential privacy. In *Proc. 33rd International Colloquium on Automata, Languages and Programming (ICALP)*, 2:1–12.
- Dwork, C., K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. 2006a. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology: Proc. EUROCRYPT*, 486–503.
- Dwork, C., F. McSherry, K. Nissim, and A. Smith. 2006b. Calibrating noise to sensitivity in private data analysis. In *Proc. 3rd Theory of Cryptography Conference (TCC)*, 265–284.
- Dwork, C., F. McSherry, and K. Talwar. 2007. The price of privacy and the limits of LP decoding. In *Proc. 39th ACM Symposium on Theory of Computing (STOC)*, 85–94.
- Dwork, C., and M. Naor. 2010. On the difficulties of disclosure prevention in statistical databases or the case for differential privacy. *Journal of Privacy and Confidentiality* 2. Available at <http://repository.cmu.edu/jpc/vol2/iss1/8>.
- Dwork, C., M. Naor, O. Reingold, G. Rothblum, and S. Vadhan. 2009. When and how can privacy-preserving data release be done efficiently? In *Proc. 41st ACM Symposium on Theory of Computing (STOC)*, 381–390.
- Dwork, C., and S. Yekhanin. 2008. New efficient attacks on statistical disclosure control mechanisms. In *Proc. CRYPTO*, 468–480.
- Dwork, Cynthia, Frank McSherry, Kobbi Nissim, and Adam Smith. 2011. Differential privacy – a primer for the perplexed. In *Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality*. Available at http://www.unece.org/fileadmin/DAM/stats/documents/ece/ces/ge.46/2011/26_Dwork-Smith.pdf.
- Dwork, Cynthia, Guy N. Rothblum, and Salil P. Vadhan. 2010. Boosting and differential privacy. In *Proc. 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 51–60.

- Evfimievski, Alexandre, Johannes Gehrke, and Ramakrishnan Srikant. 2003. Limiting privacy breaches in privacy preserving data mining. In *Proc. 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*.
- Fienberg, Stephen, Alessandro Rinaldo, and Xiaolin Yang. 2011. Differential privacy and the risk-utility tradeoff for multi-dimensional contingency tables. In *Privacy in Statistical Databases*, LNCS 6344, 187–199.
- Gentry, C. 2009. A fully homomorphic encryption scheme. PhD thesis, Stanford University.
- Goldwasser, S., and S. Micali. 1984. Probabilistic encryption. *Journal of Computer and Systems Sciences* 28:270–299.
- Goldwasser, Shafi, Silvio Micali, and Ron Rivest. 1988. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing* 17:281–308.
- Gupta, Anupam, Moritz Hardt, Aaron Roth, and Jonathan Ullman. 2011. Privately releasing conjunctions and the statistical query barrier. In *Proc. 43rd Annual ACM Symposium on Theory of Computing (STOC)*, 803–812.
- Hardt, M., K. Ligett, and F. McSherry. 2012. A simple and practical algorithm for differentially private data release. *Advances in Neural Information Processing Systems* 25:2348–2356.
- Hardt, M., and K. Talwar. 2010. On the geometry of differential privacy. In *Proc. 42nd ACM Symposium on Theory of Computing (STOC)*, 705–714.
- Hardt, Moritz, and Guy Rothblum. 2010. A multiplicative weights mechanism for privacy-preserving data analysis. In *Proc. 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 61–70.
- Hay, Michael, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. 2010. Boosting the accuracy of differentially private histograms through consistency. *Proc. VLDB Endowment* 3(1-2):1021–1032.
- Hirsch, D. 2006. Protecting the inner environment: What privacy regulation can learn from environmental law. *Georgia Law Review* 41.
- Homer, N., S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J.V. Pearson, D.A. Stephan, S.F. Nelson, and D.W. Craig. 2008. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS Genetics* 4(8):e1000167.

- Hsu, Justin, Sanjeev Khanna, and Aaron Roth. 2012. Distributed private heavy hitters. In *Proc. 39th International Colloquium Conference on Automata, Languages, and Programming (ICALP)(Track 1)*, 461–472.
- Isaacman, Sibren, Richard Becker, Ramón Cáceres, Stephen Kobourov, Margaret Martonosi, James Rowland, and Alexander Varshavsky. 2011. Identifying important places in people’s lives from cellular network data. In *Pervasive Computing*, LNCS 6696, 133–151.
- Isaacman, Sibren, Richard Becker, Ramón Cáceres, Margaret Martonosi, James Rowland, Alexander Varshavsky, and Walter Willinger. Human mobility modeling at metropolitan scales. 2012. In *Proc. 10th International Conference on Mobile Systems, Applications, and Services*, 239–252.
- Kasiviswanathan, Shiva, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing* 40:793–826.
- Kasiviswanathan, Shiva, Mark Rudelson, and Adam Smith. 2012. The power of linear reconstruction attacks. arXiv:1210.2381.
- Kocher, P., J. Jaffe, and B. Jun. 1999. Differential power analysis. In *Advances in Cryptology: Proc. CRYPTO’99*, 388–397.
- Kocher, Paul. 1996. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Advances in Cryptology: Proc. CRYPTO’96*, 104–113.
- Machanavajjhala, Ashwin, Daniel Kifer, John Abowd, Johannes Gehrke, and Lars Vilhuber. 2008. Privacy: Theory meets practice on the map. In *Proc. International Conference on Data Engineering (ICDE)*, 277–286.
- McSherry, F. 2009. Privacy integrated queries (codebase). Available on Microsoft Research downloads website. See also *Proc. SIGMOD 2009*, 19–30.
- McSherry, F., and K. Talwar. 2007. Mechanism design via differential privacy. In *Proc. 48th Annual Symposium on Foundations of Computer Science (FOCS)*, 94–103.
- Mir, Darakhshan, Sibren Isaacman, Ramón Cáceres, Margaret Martonosi, and Rebecca N. Wright. 2013. DP-WHERE: Differentially private modeling of human mobility. In *Proc. IEEE Conference on Big Data*, 580–588.
- Mironov, Ilya. 2012. On significance of the least significant bits for differential privacy. In *Proc. ACM Conference on Computer and Communications Security (CCS)*, 650–661.
- Narayanan, Arvind, and Vitaly Shmatikov. 2008. Robust de-anonymization of large sparse datasets. In *Proc. IEEE Symposium on Security and Privacy (SP)*, 111–125.

- Nikolov, Aleksandar, Kunal Talwar, and Li Zhang. 2013. The geometry of differential privacy: The sparse and approximate cases. In *Proc. 45th ACM Symposium on Theory of Computing (STOC)*, 351–360.
- Papadopoulos, S., and G. Kellaris. 2013. Practical differential privacy via grouping and smoothing. In *Proc. 39th International Conference on Very Large Data Bases*, 301–312.
- Prabhakaran, Manoj, and Amit Sahai. 2013. *Secure Multi-party Computation*. Washington, DC: IOS Press.
- Roth, Aaron, and Tim Roughgarden. 2010. Interactive privacy via the median mechanism. In *Proc. 42nd ACM Symposium on Theory of Computing (STOC)*, 765–774.
- Roy, Indrajit, Srinath Setty, Ann Kilzer, Vitaly Shmatikov, and Emmett Witchel. 2010. Airavat: Security and privacy for MapReduce. In *Proc. 7th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 10:297–312.
- Sweeney, L. 1997. Weaving technology and policy together to maintain confidentiality. *Journal of Law, Medicine and Ethics* 25:98–110.
- Sweeney, Latanya. 2012. Keynote Lecture, Second Annual iDASH All-Hands Symposium, UCSD, La Jolla, CA.
- Ullman, Jonathan, and Salil P. Vadhan. 2011. PCPs and the hardness of generating private synthetic data. In *Proc. 8th Theory of Cryptography Conference (TCC)*, 400–416.
- Warner, S. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association* 60:63–69.

Decision Making Under Uncertainty and Reinforcement Learning

Christos Dimitrakakis Ronald Ortner

August 12, 2018

Contents

1	Introduction	9
1.1	Uncertainty and Probability	10
1.2	The exploration-exploitation trade-off	11
1.3	Decision theory and reinforcement learning	12
1.4	Acknowledgements.	14
2	Subjective probability and utility	15
2.1	Subjective probability	16
2.1.1	Relative likelihood	16
2.1.2	Subjective probability assumptions	17
2.1.3	Assigning unique probabilities*	18
2.1.4	Conditional likelihoods	19
2.1.5	Probability elicitation	20
2.2	Updating beliefs: The theorem of Bayes	21
2.3	Utility theory	22
2.3.1	Rewards and preferences	22
2.3.2	Preferences among distributions	23
2.3.3	Utility	24
2.3.4	Measuring utility*	26
2.3.5	Convex and concave utility functions	27
2.4	Exercises	29
3	Decision problems	31
3.1	Introduction	32
3.2	Rewards that depend on the outcome of an experiment	32
3.2.1	Formalisation of the problem setting	33
3.2.2	Decision diagrams	35
3.2.3	Statistical estimation*	36
3.3	Bayes decisions	37
3.3.1	Convexity of the Bayes-optimal utility*	38
3.4	Statistical and strategic decision making	41
3.4.1	Alternative notions of optimality	42
3.4.2	Solving minimax problems*	43
3.4.3	Two-player games	45
3.5	Decision problems with observations	47
3.5.1	Decision problems in classification.	51
3.5.2	Calculating posteriors	53
3.6	Summary.	54

3.7	Exercises	55
3.7.1	Problems with no observations.	55
3.7.2	Problems with observations.	55
4	Estimation	57
4.1	Introduction	58
4.2	Sufficient statistics	58
4.2.1	Sufficient statistics	59
4.2.2	Exponential families	61
4.3	Conjugate priors	61
4.3.1	Bernoulli-Beta conjugate pair	62
4.3.2	Conjugates for the normal distribution	66
4.3.3	Normal with unknown precision and unknown mean	69
4.3.4	Conjugates for multivariate distributions	70
4.4	Credible intervals	74
4.5	Concentration inequalities	76
4.5.1	Chernoff-Hoeffding bounds	78
4.6	Approximate Bayesian approaches	79
4.6.1	Monte-Carlo inference	80
4.6.2	Approximate Bayesian Computation	80
4.6.3	Analytic approximations of the posterior.	81
4.6.4	Maximum Likelihood and Empirical Bayes methods.	82
5	Sequential sampling	85
5.1	Gains from sequential sampling	86
5.1.1	An example: sampling with costs	87
5.2	Optimal sequential sampling procedures	90
5.2.1	Multi-stage problems	93
5.2.2	Backwards induction for bounded procedures	93
5.2.3	Unbounded sequential decision procedures	94
5.2.4	The sequential probability ratio test	95
5.2.5	Wald's theorem	98
5.3	Martingales	99
5.4	Markov processes	100
5.5	Exercises.	102
6	Experiment design and Markov decision processes	105
6.1	Introduction	106
6.2	Bandit problems	107
6.2.1	An example: Bernoulli bandits	108
6.2.2	Decision-theoretic bandit process	109
6.3	Markov decision processes and reinforcement learning	111
6.3.1	Value functions	114
6.4	Finite horizon, undiscounted problems	115
6.4.1	Policy evaluation	115
6.4.2	Monte-Carlo policy evaluation	116
6.4.3	Backwards induction policy evaluation	117
6.4.4	Backwards induction policy optimisation	118
6.5	Infinite-horizon	119
6.5.1	Examples	119

CONTENTS	5
-----------------	----------

6.5.2	Markov chain theory for discounted problems	122
6.5.3	Optimality equations	124
6.5.4	MDP Algorithms	126
6.6	Summary	134
6.7	Further reading	134
6.8	Exercises	136
6.8.1	Medical diagnosis	136
6.8.2	Markov Decision Process theory	136
6.8.3	Automatic algorithm selection	136
6.8.4	Scheduling	137
6.8.5	General questions	139
7	Simulation-based algorithms	141
7.1	Introduction	142
7.1.1	The Robbins-Monro approximation	142
7.1.2	The theory of the approximation	144
7.2	Dynamic problems	147
7.2.1	Monte-Carlo policy evaluation and iteration	148
7.2.2	Monte Carlo updates	149
7.2.3	Approximate policy iteration	150
7.2.4	Temporal difference methods	150
7.2.5	Stochastic value iteration methods	152
7.3	Discussion	156
7.4	Exercises	159
8	Approximate representations	161
8.1	Introduction	162
8.1.1	Fitting a value function.	163
8.1.2	Fitting a policy.	163
8.1.3	Features	165
8.2	Approximate policy iteration (API)	167
8.2.1	Error bounds for approximate value functions	167
8.2.2	Estimation building blocks	169
8.2.3	The value estimation step	171
8.2.4	Policy estimation	172
8.2.5	Rollout-based policy iteration methods	173
8.2.6	Least Squares Methods	174
8.3	Approximate Value Iteration	175
8.3.1	Approximate backwards induction	176
8.3.2	State aggregation	176
8.3.3	Representative state approximation	177
8.3.4	Bellman error methods	178
8.4	Policy gradient	179
8.4.1	Specific instantiations.	182
8.4.2	Practical considerations.	183
8.5	Further reading	183
8.6	Exercises	185

9 Bayesian reinforcement learning	187
9.1 Introduction	188
9.2 Acting in unknown MDPs	188
9.2.1 Updating the belief	191
9.2.2 Finding Bayes-optimal policies	192
9.2.3 The maximum MDP heuristic	193
9.2.4 Bounds on the expected utility	194
9.2.5 Tighter lower bounds	195
9.2.6 Further sampling methods	197
9.2.7 The Belief-augmented MDP	198
9.2.8 The belief-augmented MDP tree structure	199
9.2.9 Stochastic branch and bound	199
9.2.10 Further reading	200
9.3 Partially observable Markov decision processes	200
9.3.1 Solving known POMDPs	201
9.3.2 Solving unknown POMDPs	202
9.4 Bayesian methods in continuous spaces	203
9.4.1 Transition models	203
9.4.2 Approximate dynamic programming	204
9.5 Exercises	206
10 Distribution-free reinforcement learning	209
10.1 Introduction	210
10.2 Finite Stochastic Bandit problems	210
10.2.1 The UCB1 algorithm	211
10.2.2 Non iid Rewards	213
10.3 Structured bandit problems	214
10.4 Reinforcement learning problems	215
10.4.1 Optimality Criteria	215
10.4.2 Introduction	217
10.4.3 An upper-confidence bound algorithm	217
10.4.4 Bibliographical remarks	224
11 Conclusion	225
A Symbols	227
B Probability concepts	231
B.1 Fundamental definitions	232
B.1.1 Experiments and sample spaces	232
B.2 Events, measure and probability	233
B.2.1 Events and probability	234
B.2.2 Measure theory primer	234
B.2.3 Measure and probability	235
B.3 Conditioning and independence	237
B.3.1 Mutually exclusive events	238
B.3.2 Independent events	240
B.3.3 Conditional probability	240
B.3.4 Bayes' theorem	241
B.4 Random variables	241

B.4.1	(Cumulative) Distribution functions	242
B.4.2	Discrete and continuous random variables	243
B.4.3	Random vectors	243
B.4.4	Measure-theoretic notation	244
B.4.5	Marginal distributions and independence	245
B.4.6	Moments	245
B.5	Divergences	246
B.6	Empirical distributions	247
B.7	Further reading	247
B.8	Exercises	248
C	Useful results	251
C.1	Functional Analysis	252
C.1.1	Series	252
C.1.2	Special functions	253
D	Index	255

Chapter 1

Introduction

The purpose of this book is to collect the fundamental results for decision making under uncertainty in one place, much as the book by Puterman [1994] on Markov decision processes did for Markov decision process theory. In particular, the aim is to give a unified account of algorithms and theory for sequential decision making problems, including reinforcement learning. Starting from elementary statistical decision theory, we progress to the reinforcement learning problem and various solution methods. The end of the book focuses on the current state-of-the-art in models and approximation algorithms.

The problem of decision making under uncertainty can be broken down into two parts. First, how do we learn about the world? This involves both the problem of *modeling our initial uncertainty* about the world, and that of drawing *conclusions* from *evidence* and our initial belief. Secondly, given what we currently know about the world, how should we *decide* what to do, taking into account future events and observations that may change our conclusions?

Typically, this will involve creating long-term plans covering possible future eventualities. That is, when planning under uncertainty, we also need to take into account what possible future knowledge could be generated when implementing our plans. Intuitively, executing plans which involve trying out new things should give more information, but it is hard to tell whether this information will be beneficial. The choice between doing something which is already known to produce good results and experiment with something new is known as the exploration-exploitation dilemma, and it is at the root of the interaction between learning and planning.

1.1 Uncertainty and Probability

A lot of this book is grounded in the essential methods of probability, in particular using it to represent uncertainty. While probability is a simple mathematical construction, philosophically it has had at least three different meanings. In the classical sense, a probability distribution is a description for a truly random event. In the subjectivist sense, probability is merely an expression of our uncertainty, which is not necessarily due to randomness. Finally, in the algorithmic sense, probability is linked with how “simple” a program that can generate a particular output is.

In all cases, we are dealing with a set Ω of possible outcomes: the result of a random experiment, the underlying state of the world and the program output respectively. In all cases, we use probability to model our uncertainty over Ω .

Classical Probability

A *random experiment* is performed, with a given set Ω of possible outcomes. An example is the 2-slit experiment in physics, where a particle is generated which can go through either one of two slits. According to our current understanding of quantum mechanics, it is impossible to predict which slit the particle will go through. Herein, the set Ω consists of two possible events corresponding to the particle passing through one or the other slit.

In the 2-slit experiment, the probabilities of either event can be actually accurately calculated through quantum theory. However, which slit the particle

will go through is fundamentally unpredictable. Such quantum experiments are the only ones that are currently thought of as truly random (though some people disagree about that too). Any other procedure, such as tossing a coin or casting a die, is inherently deterministic and only *appears* random due to our difficulty in predicting the outcome. That is, modelling a coin toss as a random process is usually the best approximation we can make in practice, given our uncertainty about the complex dynamics involved. This gives rise to the concept of subjective probability as a general technique to model uncertainty.

Subjective Probability

Here Ω can conceptually not only describe the outcomes of some experiment, but also a set of possible *worlds* or realities. This set can be quite large and include anything imaginable. For example, it may include worlds where dragons are real. However, in practice one only cares about certain aspects of the world, such as whether in this world, you will win the lottery if you buy a ticket. We can interpret the probability of a world in Ω as our degree of belief that it corresponds to reality.

In such a setting there is an actual, true world $\omega^* \in \Omega$, which is simply unknown. This could have been set by Nature to an arbitrary value deterministically. The probability only reflects our lack of knowledge, rather than any inherent randomness about the selection of ω^* .

No matter which view we espouse, we must always take into account our uncertainty when making decisions. When the problem we are dealing with is sequential, we are taking actions, obtaining new observations, and then taking further actions. As we gather more information, we learn more about the world. However, the things we learn about depends on what actions we take. For example, if we always take the same route to work, then we learn how much time this route takes on different days and times of the week. However, we don't obtain information about the time other routes take. So, we potentially miss out on better choices than the one we follow usually. This phenomenon gives rise to the so-called exploration-exploitation trade-off.

1.2 The exploration-exploitation trade-off

Consider the problem of selecting a restaurant to go to during a vacation. The best restaurant you have found so far was *Les Epinards*. The food there is usually to your taste and satisfactory. However, a well-known recommendations website suggests that *King's Arm* is really good! It is tempting to try it out. But there is a risk involved. It may turn out to be much worse than *Les Epinards*, in which case you will regret going there. On the other hand, it could also be much better. What should you do?

It all depends on how much information you have about either restaurant, and how many more days you'll stay in town. If this is your last day, then it's probably a better idea to go to *Les Epinards*, unless you are expecting *King's Arm* to be significantly better. However, if you are going to stay there longer, trying out *King's Arm* is a good bet. If you are lucky, you will be getting much better food for the remaining time, while otherwise you will have missed only one good meal out of many, making the potential risk quite small.

Thus, one must decide whether to *exploit* knowledge about the world, to gain a *known* reward, or to *explore* the world to *learn* something new. This will potentially give you less reward immediately, but the knowledge itself can usually be put to use in the future.

This exploration-exploitation trade-off only arises when data collection is *interactive*. If we are simply given a set of data and asked to decide upon a course or action, but our decision does not affect the data we shall collect in the future, then things are much simpler. However, a lot of real-world human decision making as well as modern applications in data science involve such trade-offs. Decision theory offers precise mathematical models and algorithms for such problems.

1.3 Decision theory and reinforcement learning

Decision theory deals with the formalization and solution of decision problems. Given a number of alternatives, what would be the rational choice in a particular situation depending on one's goals and desires? In order to answer this question we need to develop a good concept of *rational behavior*. This will serve two purposes. Firstly, this can serve as an *explanation* for what animals and humans (should) do. Secondly, it should be *useful* for developing models and algorithms for automated decision making in complex tasks.

A particularly interesting problem in this setting is *reinforcement learning*. This problem arises when the environment is unknown, and the learner has to make decisions solely through interaction, which only gives limited *feedback*. Thus, the learning agent does not have access to detailed instructions on which task to perform, nor on how to do it. Instead, it performs *actions*, which affect the environment and obtains some observations (i.e. sensory input) and feedback, usually in form of *rewards* which correspond to the agent's desires. The learning problem is then formulated as the problem of learning how to act to maximize total reward. In biological systems, reward is intrinsically hardwired to signals associated with basic needs. In artificial systems, we can choose the reward signals so as to reinforce behaviour that achieves the designer's goals.

Reinforcement learning is a fundamental problem in artificial intelligence, since frequently we can tell robots, computers, or cars only what we would like them to achieve, but we do not know the best way to achieve it. We would like to simply give them a description of our goals and then let them explore the environment on their own to find a good solution. Since the world is (at least partially) unknown, the learner always has to deal with the exploration-exploitation trade-off.

Similarly, animals and humans also learn through imitation, exploration, and shaping their behavior according to reward signals to finally achieve their goals. In fact, it has been known since the 1990s that there is some connection between some reinforcement learning algorithms and mechanisms in the basal ganglia.[Yin and Knowlton, 2006, Barto, 1995, Schultz et al., 1997]

Decision theory is closely related to other fields, such as logic, statistics, game theory and optimization. Those fields have slightly different underlying objectives, even though they may share the same formalisms. In the field of *optimization*, we are not only interested in optimal planning in complex environments but also in how to make *robust* plans given some uncertainty about

the environment. *Artificial intelligence* research is concerned with modelling the environments and developing algorithms that are able to learn by interaction with the environment or from demonstration by teachers. *Economics* and *game theory* deal with the problem of modeling the behavior of rational agents and with designing mechanisms (such as markets) that will give incentives to agents to behave in a certain way.

Beyond pure research, there are also many applications connected to decision theory. Commercial applications arise e.g. in advertising where one wishes to model the preferences and decision making of individuals. Decision problems also arise in *security*. There are many decision problems, especially in cryptographic and biometric authentication, but also in detecting and responding to intrusions in networked computer systems. Finally, in the natural sciences, especially in *biology and medicine*, decision theory offers a way to automatically design and run experiments and to optimally construct clinical trials.

Outline

1. Subjective probability and utility: The notion of subjective probability; eliciting priors; the concept of utility; expected utility.
 2. Decision problems: maximising expected utility; maximin utility; regret.
 3. Estimation: Estimation as conditioning; families of distributions that are closed under conditioning; conjugate priors; concentration inequalities; PAC and high-probability bounds; Markov Chain Monte Carlo; ABC estimation.
 4. Sequential sampling and optimal stopping: Sequential sampling problems; the cost of sampling; optimal stopping; martingales.
 5. Reinforcement learning I - Markov decision processes Belief and information state; bandit problems; Markov decision processes; backwards induction; value iteration; policy iteration; temporal differences; linear programming
 6. Reinforcement learning II – Stochastic and approximation algorithms: Sarsa; Q -learning; stochastic value iteration; $\text{TD}(\lambda)$
 7. Reinforcement learning III – Function approximation features and the curse of dimensionality; approximate value iteration; approximate policy iteration; policy gradient
 8. Reinforcement learning IV – Bayesian reinforcement learning: bounds on the utility; Thompson sampling; stochastic branch and bound; sparse sampling; partially observable MDPs.
 9. Reinforcement learning V – Distribution-free reinforcement learning: stochastic and metric bandits; UCRL; (*) bounds for Thompson sampling.
- B Probability refresher: measure theory; axiomatic definition of probability; conditional probability; Bayes' theorem; random variables; expectation
- C Miscellaneous mathematical results.

1.4 Acknowledgements.

Many thanks go to all the students of the *Decision making under uncertainty* and *Advanced topics in reinforcement learning and decision making* class over the years, for bearing with early drafts of this book. A big “thank you” goes to Nikolaos Tziortziotis, whose code is used in some of the examples in the book. Finally, thanks to Aristide Tossou and Hannes Eriksson for proof-reading various chapters. Finally, a lot of the coded examples in the book were run using the *parallel* package by Tange [2011].

Chapter 2

Subjective probability and utility

2.1 Subjective probability

In order to make decisions, we need to be able to make predictions about the possible outcomes of each decision. Usually, we have *uncertainty* about what those outcomes are. This can be due to *stochasticity*, which is frequently used to model games of chance and inherently unpredictable physical phenomena. It can also be due to *partial information*, a characteristic of many natural problems. For example, it might be hard to guess at any one moment how much change you have in your wallet, whether you will be able to catch the next bus, or to remember where you left your keys.

In either case, this uncertainty can be expressed as a *subjective belief*. This does not have to correspond to reality. For example, some people believe, quite inaccurately, that if a coin comes up tails for a long time, it is quite likely to come up heads very soon. Or, you might quite happily believe your keys are in your pocket, only to realise that you left them at home as soon you arrive at the office.

In this book, we assume the view that subjective beliefs can be modelled as *probabilities*. This allows us to treat uncertainty due to stochasticity and due to partial information in a unified framework. In doing so, we shall treat each part of the problem as specifying a space of possible outcomes. What we wish to do is to find a *consistent way* of defining probabilities in the space of outcomes.

2.1.1 Relative likelihood

Let us start with the simple example of guessing whether a tossed coin will come up head, or tails. In this case the sample space Ω would correspond to every possible way the coin can land. Since we are only interested in predicting which face will be up, let $A \subset \Omega$ be all those cases where the coin comes up heads, and $B \subset \Omega$ be the set of tosses where it comes up tails. Here $A \cap B = \emptyset$, but there may be some other events such as the coin becoming lost, so it does not necessarily hold that $A \cup B = \Omega$. Nevertheless, we only care about whether A is more likely to occur than B . As said, this likelihood may be based only on subjective beliefs. We can express that via the concept of relative likelihood:

(The relative likelihood of two events A and B)

- If A is *more likely* than B , then we write $A \succ B$, or equivalently $B \prec A$.
- If A is *as likely* as B , then we write $A \sim B$.

We also use \gtrsim and \lesssim for *at least as likely as* and for *no more likely than*.

Let us now speak more generally about the case where we have defined an appropriate σ -field \mathcal{F} on Ω . Then each element $A_i \in \mathcal{F}$ will be a subset of Ω . We now wish to define relative likelihood relations for the elements $A_i \in \mathcal{F}$.¹

¹More formally, we can define three classes: $C_{\succ}, C_{\prec}, C_{\sim} \subset \mathcal{F}^2$ such that a pair $(A_i, A_j) \in C_R$ if and only if it satisfies the relation $A_i R A_j$, where $R \in \{\succ, \prec, \sim\}$. These three classes form a partition of \mathcal{F}^2 under the subjective probability assumptions we will introduce in the

As we would like to use the language of probability to talk about likelihoods, we need to define a probability measure that agrees with our given relations. A probability measure $P : \mathcal{F} \rightarrow [0, 1]$ is said to *agree* with a relation $A \precsim B$, if it has the property that $P(A) \leq P(B)$ if and only if $A \precsim B$, for all $A, B \in \mathcal{F}$. In general, there are many possible measures that can agree with a given relation, cf. Example 1 below. However, it could also be that a given relational structure is incompatible with any possible probability measure. We also consider the question under which assumptions a likelihood relation corresponds to a unique probability measure.

2.1.2 Subjective probability assumptions

We would like our beliefs to satisfy some intuitive properties about what statements we can make concerning the relative likelihood of events. As we will see, these assumptions are also necessary to guarantee the existence of a corresponding probability measure. First of all, it must always be possible to say whether one event is more likely than the other, i.e. our beliefs must be complete. Consequently, we are not allowed to claim ignorance.

Assumption 2.1.1 (SP1). *For any pair of events $A, B \in \mathcal{F}$, one has either $A \succ B$, $A \prec B$, or $A \sim B$.*

Another important assumption is a principle of consistency: Informally, if we believe that every possible event A_i that leads to A is less likely than a unique corresponding event B_i that leads to an outcome B , then we should always conclude that A is less likely than B .

Assumption 2.1.2 (SP2). *Let $A = A_1 \cup A_2$, $B = B_1 \cup B_2$ with $A_1 \cap A_2 = B_1 \cap B_2 = \emptyset$. If $A_i \precsim B_i$ for $i = 1, 2$ then $A \precsim B$.*

We also require the simple technical assumption that any event $A \in \mathcal{F}$ is at least as likely as the empty event \emptyset , which never happens.

Assumption 2.1.3 (SP3). *For all A it holds that $\emptyset \precsim A$. Further, $\emptyset \prec \Omega$.*

As it turns out, these assumptions are sufficient for proving the following theorems [DeGroot, 1970]. The first theorem tells us that our belief must be consistent with respect to transitivity.

Theorem 2.1.1 (Transitivity). *Under Assumptions 2.1.1, 2.1.2, and 2.1.3, for all events A, B, C : If $A \precsim B$ and $B \precsim C$, then $A \precsim C$.*

The second theorem says that if two events have a certain relation, then their negations have the converse relation.

Theorem 2.1.2 (Complement). *For any A, B : $A \precsim B$ iff $A^c \succ B^c$.*

Finally, note that if $A \subset B$, then it must be the case that whenever A happens, B must happen and hence B must be at least as likely as A . This is demonstrated in the following theorem.

Theorem 2.1.3 (Fundamental property of relative likelihoods). *If $A \subset B$ then $A \precsim B$. Furthermore, $\emptyset \precsim A \precsim \Omega$ for any event A .*

next section.

Since we are dealing with σ -fields, we need to introduce properties for infinite sequences of events. While these are not necessary if the field \mathcal{F} is finite, it is good to include them for generality.

Assumption 2.1.4 (SP4). *If $A_1 \supset A_2 \supset \dots$ is a decreasing sequence of events in \mathcal{F} and $B \in \mathcal{F}$ is such that $A_i \succsim B$ for all i , then $\bigcap_{i=1}^{\infty} A_i \succsim B$.*

As a consequence, we obtain the following dual theorem:

Theorem 2.1.4. *If $A_1 \subset A_2 \subset \dots$ is an increasing sequence of events in \mathcal{F} and $B \in \mathcal{F}$ is such that $A_i \precsim B$ for all i , then $\bigcup_{i=1}^{\infty} A_i \precsim B$.*

We are now able to state a theorem for the unions of infinite sequences of disjoint events.

Theorem 2.1.5. *If $(A_i)_{i=1}^{\infty}$ and $(B_i)_{i=1}^{\infty}$ are infinite sequences of disjoint events in \mathcal{F} such that $A_i \precsim B_i$ for all i , then $\bigcup_{i=1}^{\infty} A_i \precsim \bigcup_{i=1}^{\infty} B_i$.*

The following theorem shows that if likelihood is induced by a probability measure P (that is, $A \succ B$ iff $P(A) > P(B)$, and $A \approx B$ if $P(A) = P(B)$, so that P agrees with \succsim), it always satisfies the stipulated assumptions.

Theorem 2.1.6. *Let P be a probability measure over Ω . Then*

- (i) $P(A) > P(B)$, $P(A) < P(B)$ or $P(A) = P(B)$ for all A, B .
- (ii) Consider (possibly infinite) partitions $\{A_i\}_i$, $\{B_i\}_i$ of A, B , respectively. If $P(A_i) \leq P(B_i)$ for all i , then $P(A) \leq P(B)$.
- (iii) For any A , $P(\emptyset) \leq P(A)$ and $P(\emptyset) < P(\Omega)$.

Proof. Part (i) is trivial, as $P : \mathcal{F} \rightarrow [0, 1]$. Part (ii) follows from $P(A) = P(\bigcup_i A_i) = \sum_i P(A_i) \leq \sum_i P(B_i) = P(B)$. Part (iii) follows from $P(\emptyset) = 0$, $P(A) \geq 0$, and $P(\Omega) = 1$. \square

2.1.3 Assigning unique probabilities*

In many cases, and particularly when \mathcal{F} is a finite field, there is a large number of probability distributions agreeing with our relative likelihoods. Choosing one specific probability over another does not seem easy. The following example underscores this ambiguity.

EXAMPLE 1. Consider $\mathcal{F} = \{\emptyset, A, A^c, \Omega\}$ and say $A \succ A^c$. Consequently, $P(A) > 1/2$. But this is insufficient for assigning a specific value to $P(A)$.

In some cases we would like to assign unique probabilities to events in order to facilitate computations.

This can be achieved by augmenting our set of events with random draws from a uniform distribution, defined below for intervals in $[0, 1]$. Intuitively, we may only be able to tell whether some event A is more likely than some other event B . However, we can create a new, uniformly distributed random variable x on $[0, 1]$ and ask ourselves, for each value $\alpha \in [0, 1]$ whether A more or less likely than the event $x > \alpha$. Since we need to compare both A and B with all such events, the distribution we'll obtain is unique. Essentially, we relate the

likelihoods of our two discrete events with the uniform distribution, in order to assign specific probabilities to them. Without further ado, here is the definition of the uniform distribution.

Definition 2.1.1 (Uniform distribution). Let $\lambda(A)$ denote the length of any interval $A \subseteq [0, 1]$. Then $x : \Omega \rightarrow [0, 1]$ has a uniform distribution on $[0, 1]$ if, for any subintervals A, B of $[0, 1]$,

$$(x \in A) \precsim (x \in B) \text{ iff } \lambda(A) \leq \lambda(B),$$

where $(x \in A)$ denotes the event that $x(\omega) \in A$. Then $(x \in A) \precsim (x \in B)$ means that ω is such that $x \in A$ is not more likely than $x \in B$.

This means that *any* larger interval is more likely than *any* smaller interval. Now we shall connect the uniform distribution to the original sample space Ω by assuming that there is some function with uniform distribution.

Assumption 2.1.5 (SP5). *It is possible to construct a random variable $x : \Omega \rightarrow [0, 1]$ with a uniform distribution in $[0, 1]$.*

Constructing the probability distribution

We can now use the uniform distribution to create a unique probability measure that agrees with our likelihood relation. First, we have to map each event in Ω to an equivalent event in $[0, 1]$.

Theorem 2.1.7 (Equivalent event). *For any event $A \in \mathcal{F}$, there exists some $\alpha \in [0, 1]$ such that $A \approx (x \in [0, \alpha])$.*

This means that we can now define the probability of an event A by matching it to a specific equivalent event on $[0, 1]$.

Definition 2.1.2 (The probability of A). Given any event A , define $P(A)$ to be the α with $A \approx (x \in [0, \alpha])$.

Hence

$$A \approx (x \in [0, P(A)]).$$

The above is sufficient to show the following theorem.

Theorem 2.1.8 (Relative likelihood and probability). *If assumptions SP1-SP5 are satisfied, then the probability measure P defined above is unique. Furthermore, for any two events A, B , $A \precsim B$ iff $P(A) \leq P(B)$.*

2.1.4 Conditional likelihoods

So far we have only considered the problem of forming opinions about which events are more likely *a priori*. However, we also need to have a way to incorporate evidence which may adjust our opinions. For example, while we ordinarily may think that $A \precsim B$, we may have additional information D , given which we think the opposite is true. We can formalise this through the notion of conditional likelihoods.

EXAMPLE 2. Say that A is the event that it rains in Gothenburg, Sweden tomorrow. We know that Gothenburg is quite rainy due to its oceanic climate, so we set $A \precsim A^c$. Now, let us try and incorporate some additional information. Let D denote the fact that good weather is forecast. I personally believe that $(A | D) \precsim (A^c | D)$, i.e. that good weather is more probable than rain, given the evidence of the weather forecast.

Conditional likelihoods

Define $(A | D) \lesssim (B | D)$ to mean that B is at least as likely as A when it is known that D has occurred.

Assumption 2.1.6 (CP). *For any events A, B, D ,*

$$(A | D) \lesssim (B | D) \quad \text{iff} \quad A \cap D \lesssim B \cap D.$$

Theorem 2.1.9. *If a likelihood relation \lesssim satisfies assumptions SP1 to SP5, as well as CP, then there exists a probability measure P such that: For any A, B, D such that $P(D) > 0$,*

$$(A | D) \lesssim (B | D) \quad \text{iff} \quad P(A | D) \leq P(B | D).$$

It turns out that there are very few ways that a conditional probability definition can satisfy all of our assumptions. One natural definition, indeed employed pretty much everywhere in probability theory, is the following.

Definition 2.1.3 (Conditional probability).

$$P(A | D) \triangleq \frac{P(A \cap D)}{P(D)}. \quad (2.1.1)$$

This definition effectively answers the question of how much evidence for A we have, now that we have observed D . This is expressed as the ratio between the combined event $A \cap D$, also known as the joint probability of A and D , and the marginal probability of D itself. The intuition behind the definition becomes clearer once we rewrite it as $P(A \cap D) = P(A | D)P(D)$. Then conditional probability is effectively used as a way to factorise joint probabilities.

2.1.5 Probability elicitation

Probability elicitation is the problem of quantifying the subjective probabilities that a particular individual uses. One of the simplest, and most direct, methods, is to simply ask. However, because we cannot simply ask somebody to completely specify a probability distribution, we can ask for this distribution iteratively.

EXAMPLE 3 (Temperature prediction). Let τ be the temperature tomorrow at noon in Gothenburg. What are your estimates?

Eliciting the prior / forming the subjective probability measure P

- Select temperature x_0 s.t. $(\tau \leq x_0) \approx (\tau > x_0)$.
- Select temperature x_1 s.t. $(\tau \leq x_1 | \tau \leq x_0) \approx (\tau > x_1 | \tau \leq x_0)$.

By repeating this procedure recursively we will slowly build the complete distribution, quantile by quantile.

Note that, necessarily, $P(\tau \leq x_0) = P(\tau > x_0) = p_0$. Since $P(\tau \leq x_0) + P(\tau > x_0) = P(\tau \leq x_0 \cup \tau > x_0) = P(\tau \in \mathbb{R}) = 1$, it follows that $p_0 = 1/2$. Similarly, $P(\tau \leq x_1 | \tau \leq x_0) = P(\tau > x_1 | \tau \leq x_0) = 1/4$.

EXERCISE 1. Propose another way to arrive at a prior probability distribution. For examples, define a procedure for eliciting a single probability distribution from a group of people without any interaction between the participants.

2.2 Updating beliefs: The theorem of Bayes

Although we always start with a particular belief, this belief must be adjusted when we receive new evidence. In probabilistic inference, the updated beliefs are simply the probability of future events conditioned on observed events. This idea is captured neatly by Bayes' theorem, which links the prior probability $P(A_i)$ of events to their posterior probability $P(A_i | B)$ given some event B and the probability $P(B | A_i)$ of observing the evidence B given that the events A_i are true.

Theorem 2.2.1 (Bayes' theorem). *Let A_1, A_2, \dots be a (possibly infinite) sequence of disjoint events such that $\bigcup_{i=1}^n A_i = \Omega$ and $P(A_i) > 0$ for all i . Let B be another event with $P(B) > 0$. Then*

$$P(A_i | B) = \frac{P(B | A_i)P(A_i)}{\sum_{j=1}^n P(B | A_j)P(A_j)}. \quad (2.2.1)$$

Proof. By definition, $P(A_i | B) = P(A_i \cap B)/P(B)$, and $P(A_i \cap B) = P(B | A_i)P(A_i)$, so:

$$P(A_i | B) = \frac{P(B | A_i)P(A_i)}{P(B)}, \quad (2.2.2)$$

As $\bigcup_{j=1}^n A_j = \Omega$, we have $B = \bigcup_{j=1}^n (B \cap A_j)$. Since A_i are disjoint, so are $B \cap A_i$. As P is a probability, the union property and an application of (2.2.2) give

$$P(B) = P\left(\bigcup_{j=1}^n (B \cap A_j)\right) = \sum_{j=1}^n P(B \cap A_j) = \sum_{j=1}^n P(B | A_j)P(A_j).$$

□

A simple exercise in updating beliefs

EXAMPLE 4 (The weather forecast). Form a subjective probability for the probability that it rains.

A_1 : Rain.

A_2 : No rain.

First, choose $P(A_1)$ and set $P(A_2) = 1 - P(A_1)$. Now assume that there is a weather forecasting station that predicts *no rain* for tomorrow. However, you know the following fact about the station: on the days when it rains, half of the time the station had predicted it *was not* going to rain. On days when it doesn't rain, the station had said *no rain* 9 times out of 10.

Solution. Let B denote the event that the station predicts no rain. According to our information, $P(B | A_1) = 1/2$, i.e. whenever there is rain (A_1), the previous day's prediction said no rain (B). On the other hand, $P(B | A_2) = 0.9$. Combining these with Bayes rule, we obtain.

$$\begin{aligned} P(A_1 | B) &= \frac{P(B | A_1)P(A_1)}{P(B | A_1)P(A_1) + P(B | A_2)[1 - P(A_1)]} \\ &= \frac{1/2P(A_1)}{0.9 - 0.4P(A_1)}. \end{aligned}$$

□

2.3 Utility theory

While probability can be used to describe how likely an event is, utility can be used to describe how desirable it is. More concretely, our subjective probabilities are numerical representations of our beliefs and information. They can be taken to represent our “internal model” of the world. By analogy, our utilities are numerical representations of our tastes and preferences. Even if the consequences of our actions are not directly known to us, we assume that we act to maximise our utility, in some sense.

2.3.1 Rewards and preferences

Rewards

Consider that we have to choose a *reward* r from a set \mathcal{R} of possible rewards. While the elements of \mathcal{R} may be arbitrary, we shall in general find that we prefer some rewards to others. In fact, some elements of \mathcal{R} may not even be desirable. As an example, \mathcal{R} might be a set of tickets to different musical events, or a set of financial commodities.

Preferences

EXAMPLE 5 (Musical event tickets). We have a set of tickets \mathcal{R} , and we must choose the ticket $r \in \mathcal{R}$ we prefer best.

- Case 1: \mathcal{R} are tickets to different music events at the same time, at equally good halls with equally good seats and the same price. Here preferences simply coincide with the preferences for a certain type of music or an artist.
- Case 2: \mathcal{R} are tickets to different events at different times, at different quality halls with different quality seats and different prices. Here, preferences may depend on all the factors.

EXAMPLE 6 (Route selection). We have a set of alternate routes and must pick one.

- \mathcal{R} contains two routes, one short and one long, of the same quality.
- \mathcal{R} contains two routes, one short and one long, but the long route is more scenic.

Preferences among rewards

We will treat preferences in a similar manner as we have treated probabilities. That is, we will define a linear ordering among possible rewards.

Let $a, b \in \mathcal{R}$ be two rewards. When we prefer a to b , we write $a \succ^* b$. Conversely, when we like a less than b we write $a \prec^* b$. If we like a as much as b , we write $a \approx^* b$. We also use \gtrsim^* and \lesssim^* for *I like at least as much as* and for *I don't like any more than*, respectively. We make the following assumptions about the preference relations.

Assumption 2.3.1. (i) For any $a, b \in \mathcal{R}$, one of the following holds: $a \succ^* b$, $a \prec^* b$, $a \approx^* b$.

(ii) If $a, b, c \in \mathcal{R}$ are such that $a \lesssim^* b$ and $b \lesssim^* c$, then $a \lesssim^* c$.

The first assumption means that we must always be able to decide between any two rewards. It may seem that it does not always hold in practice, since humans are frequently indecisive. However, without the second assumption, it is still possible to create preference relations that are cyclic.

EXAMPLE 7 (Counter-example for transitive preferences). Consider vector rewards in $\mathcal{R} = \mathbb{R}^2$, with $r_i = (a_i, b_i)$, and some $\epsilon, \epsilon' > 0$. Our preference relation is:

- $r_i \succ^* r_j$ if $b_i \geq b_j + \epsilon'$.
- $r_i \succ^* r_j$ if $a_i \geq a_j$ and $|b_i - b_j| < \epsilon$.

This may correspond for example to an employer deciding to hire one of two employees, i, j , depending on their experience (a) or their school grades (b). Since grades are not very reliable, if two people have grades, then we prefer the one with the most experience. However, that may lead to a cycle. Consider a sequence of candidates $i = 1, \dots, n$, such that each candidate satisfies $b_i = b_{i+1} + \delta$, with $\delta < \epsilon$ and $a_i > a_{i+1}$. Then clearly, we must always prefer r_i to r_{i+1} . However, if $\delta n > \epsilon$, we will prefer r_n to r_1 .

2.3.2 Preferences among distributions

When we cannot select rewards directly

In most problems, we cannot choose the rewards directly. Rather, we must make some decision, and then obtain a reward depending on this decision. Since we may be uncertain about the outcome of a decision, we can specify our uncertainty regarding the rewards obtained by a decision in terms of a probability distribution.

EXAMPLE 8 (Route selection). Assume that you have to pick between two routes P_1, P_2 . Your preferences are such that shorter time routes are preferred over longer ones. For simplicity, let $\mathcal{R} = \{10, 15, 25, 30\}$ be the possible times we might take to reach your destination. Route P_1 takes 10 minutes when the road is clear, but 30 minutes when the traffic is heavy. The probability of heavy traffic on P_1 is 0.5. On the other hand, route P_2 takes 15 minutes when the road is clear, but 25 minutes when the traffic is heavy. The probability of heavy traffic on P_2 is 0.2.

Preferences among probability distributions

As seen in the previous example, we frequently have to define preferences between probability distributions, rather than over rewards. To represent our

preferences, we can use the same notation as before. Let P_1, P_2 be two distributions on (R, \mathcal{F}_R) . If we prefer P_1 to P_2 , we write $P_1 \succ^* P_2$. If we like P_1 less than P_2 , write $P_1 \prec^* P_2$. If we like P_1 as much as P_2 , we write $P_1 \sim^* P_2$. Finally, we also use \gtrsim^* and \lesssim^* do denote strict preference relations.

What would be a good principle for choosing between the two routes in Example 8? Clearly route P_1 gives both the lowest best-case time and the highest worst-case time. It thus appears as though both an extremely cautious person (who assumes the worst-case) and an extreme optimist (who assumes the best case) would say $P_2 \succ^* P_1$. However, the average time taken in P_1 is only 17 minutes versus 20 minutes for P_1 . Thus, somebody that only took the average time into account would prefer P_1 . In the following sections, we will develop one of the most fundamental methodologies for choices under uncertainty, based on the idea of utilities.

2.3.3 Utility

The concept of utility allows us to create a unifying framework, such that given a particular set of rewards and probability distributions on them, we can define preferences among distributions automatically. The first step is to define utility as a way to define a preference relation among rewards.

Definition 2.3.1 (Utility). A utility function $U : \mathcal{R} \rightarrow \mathbb{R}$ is said to agree with the preference relation \gtrsim^* , if for all rewards $a, b \in \mathcal{R}$

$$a \gtrsim^* b \quad \text{iff} \quad U(a) \geq U(b). \quad (2.3.1)$$

The above definition is very similar to how we defined relative likelihood in terms of probability. For a given utility function, its expectation for a distribution over rewards is defined as follows:

Definition 2.3.2 (Expected utility). Given a utility function U , the expected utility of a distribution P on \mathcal{R} is:

$$\mathbb{E}_P(U) = \int_{\mathcal{R}} U(r) dP(r) \quad (2.3.2)$$

We make the assumption that the utility function is such that the expected utility remains consistent with the preference relations between all probability distributions we are choosing between.

Assumption 2.3.2 (The expected utility hypothesis). *Given a preference relation \gtrsim^* over \mathcal{R} and a corresponding utility function U , the utility of any probability measure P on \mathcal{R} is equal to the expected utility of the reward under P . Consequently,*

$$P \gtrsim^* Q \quad \text{iff} \quad \mathbb{E}_P(U) \geq \mathbb{E}_Q(U). \quad (2.3.3)$$

EXAMPLE 9. Consider the following decision problem. You have the option of entering a lottery, for 1 currency unit (CU), that gives you a prize of 10 CU. The probability of winning is 0.01. This can be formalised by making it a choice between two probability distributions: P , where you do not enter the lottery and Q , which represents entering the lottery. Now we can calculate the expected utility for each choice. This is simply $\mathbb{E}(U | P) = \sum_r U(r)P(r)$ and $\mathbb{E}(U | Q) = \sum_r U(r)Q(r)$ respectively. Hence the utility of entering the lottery is -0.9 , while it is 0 for not entering.

r	U(r)	P	Q
did not enter	0	1	0
paid 1 CU and lost	-1	0	0.99
paid 1 CU and won 10	9	0	0.01

Table 2.1: A simple gambling problem

Monetary rewards

Frequently, rewards come in the form of money. In general, it is assumed that people prefer to have more money than less money. However, the utility of additional money is not constant, i.e. 1,000 Euros are probably worth more to somebody with only 100 Euros in the bank than to somebody with 100,000 Euros. Hence, the utility of monetary rewards is generally assumed to be increasing, but not necessarily linear. Indeed, we would expect the utility of money to be concave. Nevertheless, we would in any case expect the behaviour of individuals to follow the tenets of expected utility theory. You should be able to verify this for following example for any increasing utility function U .

EXAMPLE 10. Choose between the following two gambles:

1. The reward is 500,000 with certainty.
2. The reward is 2,500,000 with probability 0.10. It is 500,000 with probability 0.89, and 0 with probability 0.01.

EXAMPLE 11. Choose between the following two gambles:

1. The reward is 500,000 with probability 0.11, or 0 with probability 0.89.
2. The reward is: 2,500,000 with probability 0.1, or 0 with probability 0.9.

EXERCISE 2. Show that if gamble 1 is preferred in the first example, gamble 1 must also be preferred in the second example, irrespective of the form of our utility function, under the expected utility hypothesis.

In practice, you may find that your preferences are not aligned with what this exercise suggests. This implies that either your decisions do not conform to the expected utility hypothesis, or that you are not internalising the given probabilities. We will explore this further in following example.

The St. Petersburg Paradox

The following simple example illustrates the fact that, internally, most humans do not seem to behave in ways that are not compatible with a linear utility for money. Ask yourself, or other classmates, how much money they would be willing to bet, in order to play the following game.

EXAMPLE 12 (The St. Petersburg Paradox (Bernoulli, 1713)). In this game, we first pay k currency units, and then the *bank* tosses a fair coin repeatedly, until the coin comes up heads. Then the game ends and we obtain 2^n units, where n is the number of times the coin was thrown. So $n \in \{1, 2, \dots, \infty\}$. The coin is *fair*, meaning that the probability of heads is always $1/2$.

How many units k are you willing to pay, to play this game once?

As you can see below, the expected amount of money is infinite. First of all, the probability to stop at round n is 2^{-n} . Thus, the expected monetary gain of the game is

$$\sum_{n=1}^{\infty} 2^n 2^{-n} = \infty.$$

Were your utility function linear you'd be willing to pay any amount k to play, as the expected utility for playing the game is

$$\sum_{n=1}^{\infty} U(2^n - k) 2^{-n} = \infty$$

for any finite k .

It would be safe to assume that very few readers would be prepared to pay any amount to play this game. One way to explain this is that the utility function used by the player is not necessarily linear. For example, if we also assume that the player has an initial capital C from which k has to be paid, we can consider a logarithmic utility function so that

$$\mathbb{E} U = \sum_{n=1}^{\infty} \ln(C + 2^n - k) 2^{-n},$$

where C is our initial capital. In that case, for $C = 10,000$, the maximum bet is 14. For $C = 100$, the maximum bet is 6, while for $C = 10$, it is just 4.

There is another reason why one may not pay an arbitrary amount to play this game. The player may not fully internalise the fact (or rather, the promise) that the coin is unbiased. Other explanations include whether you really believe that I can pay off an unbounded amount of money, or whether the sum only reaches up to some finite N . In the linear expected utility scenario, for a coin with probability p of coming heads, and sums only up to N , we have

$$\sum_{n=1}^N 2^n p^{n-1} (1-p) = 2(1-p) \frac{1 - (2p)^N}{1 - 2p}.$$

For large N , it turns out that if $p = 0.45$, so slightly biased off heads, you should only expect about 10 dollars. But even if you believe the coin is fair, there is another possibility: if you think the *bank* only has a reserve of 1024 dollars, then again you should only bet up to 10 dollars. These are possible subjective beliefs that an individual might have that would influence their behaviour when dealing with a formally specified decision problem.

2.3.4 Measuring utility*

Since we cannot even rely on linear utility for money, we need to ask ourselves how we can measure the utility of different rewards. There are a number of ways, including trying to infer it from the actions of people. The simplest approach is to simply ask them to make even money bets. No matter what approach we use, however, we need to make some assumptions about the utility structure. This includes whether or not we should accept that the expected utility hypothesis holds for the observed human behaviour.

Experimental measurement of utility

EXAMPLE 13. We shall try and measure the utility of all monetary rewards in some interval $[a, b]$.

Let $\langle a, b \rangle$ denote a lottery ticket that yields a or b CU with equal probability. Consider the following sequence:

1. Find x_1 such that receiving x_1 CU with certainty is equivalent to receiving $\langle a, b \rangle$.
2. Find x_2 such that receiving x_2 CU with certainty is equivalent to receiving $\langle a, x_1 \rangle$.
3. Find x_3 such that receiving x_3 CU with certainty is equivalent to receiving $\langle x_1, b \rangle$.
4. Find x_4 such that receiving x_4 CU with certainty is equivalent to receiving $\langle x_2, x_3 \rangle$.

The above example algorithm allows us to measure the utility of money under the assumption that the expected utility hypothesis holds. However, if $x_1 \neq x_4$, then the preferences do not appear to meet the requirements of the expected utility hypothesis, which implies that $U(x_1) = U(x_4) = \frac{1}{2}(U(a) + U(b))$.

2.3.5 Convex and concave utility functions

As previously mentioned, utility functions of monetary rewards are not necessarily linear. In general, we'd say that a concave utility function implies risk aversion and a convex one risk taking. Intuitively, a risk averse person prefers a fixed amount of money to a random amount of money with the same expected value. A risk taker prefers to gamble. Let's start with the definition of a convex function.

Definition 2.3.3. A function $g : \Omega \rightarrow \mathbb{R}$, is convex on $A \subset \Omega$ if, for any points $x, y \in A$, and any $\alpha \in [0, 1]$:

$$\alpha g(x) + (1 - \alpha)g(y) \geq g(\alpha x + (1 - \alpha)y)$$

An important property of convex functions is that they are bounded from above by linear segments connecting their points. This property is formally given below.

Theorem 2.3.1 (Jensen's inequality). *If g is convex on Ω and $x \in \Omega$ and P is a measure with $P(\Omega) = 1$ and $\mathbb{E}(x)$ and $\mathbb{E}[g(x)]$ exist, then:*

$$\mathbb{E}[g(x)] \geq g[\mathbb{E}(x)]. \quad (2.3.4)$$

EXAMPLE 14. If the utility function is convex, then we would prefer obtaining a random reward x rather than a fixed reward $y = \mathbb{E}(x)$. Thus, a convex utility function implies risk-taking. This is illustrated by Figure 2.1 which shows a linear function, x , a convex function, $e^x - 1$, and a concave function, $\ln(x + 1)$.

Definition 2.3.4. A function g is concave on Ω if, for any points $x, y \in \Omega$, and any $\alpha \in [0, 1]$:

$$\alpha g(x) + (1 - \alpha)g(y) \leq g[\alpha x + (1 - \alpha)y]$$

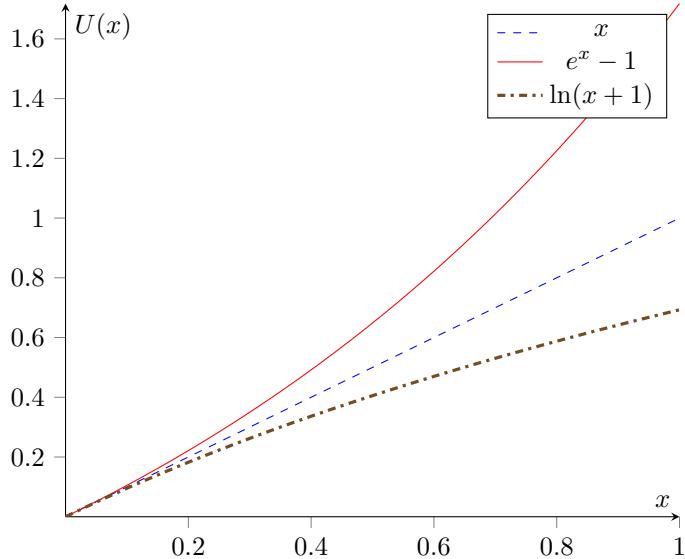


Figure 2.1: Linear, convex and concave functions

For concave functions, the inverse of Jensen's inequality holds (i.e. in the other direction). If the utility function is concave, then we choose a gamble giving a fixed reward $\mathbb{E}[x]$ rather than one giving a random reward x . Consequently, a concave utility function implies risk aversion. The act of buying insurance can be related to concavity of our utility function. Consider the following example, where we assume individuals are risk-averse, but insurance companies are risk-neutral.

EXAMPLE 15 (Insurance). Let x be the insurance cost, h our insurance cover, ϵ the probability of needing the cover, and U an increasing utility function (for monetary values). Then we are going to buy insurance if the utility of losing x with certainty is greater than the utility of losing $-h$ with probability ϵ .

$$U(-x) > \epsilon U(-h) + (1 - \epsilon)U(0). \quad (2.3.5)$$

The company has a linear utility, and fixes the premium x high enough for

$$x > \epsilon h. \quad (2.3.6)$$

Consequently, we see from (2.3.6) that $U(-\epsilon h) \geq U(-x)$, as U is an increasing function. From (2.3.5) we obtain $U(-\epsilon h) > \epsilon U(-h) + (1 - \epsilon)U(0)$. Now the $-\epsilon h$ term is the utility of our expected monetary loss, while the right hand side is our expected utility. Consequently if the inequality holds, our utility function is (at least locally) concave.

2.4 Exercises

EXERCISE 3. If \mathcal{R} is our set of rewards, our utility function is $U : \mathcal{R} \rightarrow \mathbb{R}$ and $a \succ^* b$ iff $U(a) > U(b)$, then our preferences are transitive. Give an example of a utility function, not necessarily mapping to \mathbb{R} , and a binary relation $>$ such that transitivity can be violated. Back your example with a thought experiment.

EXERCISE 4. Assuming that U is increasing and absolutely continuous, consider the following experiment:

1. You specify an amount a , then observe random value Y .
2. If $Y \geq a$, you receive Y currency units.
3. If $Y < a$, you receive a random amount X with known distribution (independent of Y).

Show that we should choose a s.t. $U(a) = \mathbb{E}[U(X)]$.

EXERCISE 5. The usefulness of probability and utility.

- Would it be useful to separate randomness from uncertainty? What would be desirable properties of an alternative concept to probability?
- Give an example of how the expected utility assumption might be violated.

EXERCISE 6. Consider two urns, each containing red and blue balls. The first urn contains an equal number of red and blue balls. The second urn contains a *randomly* chosen proportion X of red balls, i.e. the probability of drawing a red ball from urn 2 is X .

1. Suppose that you were to select an urn, and then choose a random ball from that urn. If the ball is red, you win 1 CU, otherwise nothing. Show that: if your utility function is increasing with monetary gain, you should prefer urn 1 iff $\mathbb{E}(X) < \frac{1}{2}$.
2. Suppose that you were to select an urn, and then choose n random balls from that urn and that urn only. Each time you draw a red ball, you gain 1 CU. After you draw a ball, you put it back in the urn. Assume that the utility U is strictly concave and suppose that $\mathbb{E}(X) = \frac{1}{2}$. Show that you should always select balls from urn 1.

Hint: Show that for urn 2, $\mathbb{E}(U | x)$ is concave for $0 \leq x \leq 1$ (this can be done by showing $\frac{d^2}{dx^2} \mathbb{E}(U | x) < 0$). In fact,

$$\frac{d^2}{dx^2} \mathbb{E}(U | x) = n(n-1) \sum_{k=0}^{n-2} [U(k) - 2U(k+1) + U(k+2)] \binom{n-2}{k} x^k (1-x)^{n-2-k}.$$

Then apply Jensen's inequality.

EXERCISE 7. Probability measures as a way to define likelihood relations.

Show that a probability measure P on (Ω, \mathcal{F}) satisfies the following: For any events $A, B \in \mathcal{F}$, one of the following holds: $P(A) > P(B)$, $P(B) > P(A)$ or $P(A) = P(B)$. If A_i, B_i are partitions of A, B such that for all $P(A_i) \leq P(B_i)$ for all i , then $P(A) \leq P(B)$. For any event A , $P(\emptyset) \leq P(A)$ and $P(\emptyset) < P(\Omega)$.

EXERCISE 8. The definition of conditional probability

Recall that $P(A | B) \triangleq \frac{P(A \cap B)}{P(B)}$ is only a definition. Give a plausible alternative that agrees with the basic properties of a probability measure. It helps if you see the conditional probability as a new probability measure $M_B(A) \triangleq P(A | B)$. The

properties are: (a) Null probability: $P(\emptyset | B) = 0$ (b) Total probability: $P(\Omega | B) = 1$ (c) Union of disjoint subsets: $P(A_1 \cup A_2 | B) = P(A_1 | B) + P(A_2 | B)$ (d) Conditional Probability: $P(A | D) \leq P(B | D)$ if and only if $P(A \cap D) \leq P(B \cap D)$.

“

EXERCISE 9 (30!). Alternatives to the expected utility hypothesis The expected utility hypothesis states that we prefer decision P over Q if and only if our expected utility under the distribution P is larger than that under Q , i.e. $\mathbb{E}_P(U) \geq \mathbb{E}_Q(U)$. Under what conditions do you think this is a reasonable hypothesis? Can you come up with a different rule for making decisions under uncertainty? Would it still satisfy the total order and transitivity properties of preference relations? In other words, could you still unambiguously say whether you prefer P to Q ? If you had three choices, P, Q, W and you preferred P to Q and Q to W , would you always prefer P to W ?

EXERCISE 10. Rational Arthur-Merlin games. You are Arthur, and you wish to pay Merlin to do a very difficult computation for you. More specifically, you perform a query $q \in Q$ and obtain an answer $r \in R$, from Merlin. However, there exists a unique correct answer $r^* = f(q)$. After he gives you the answer, you give Merlin a random amount of money m , depending on r, q such that $\mathbb{E}(m | r, q) = \sum_m m P(m | r, q)$ is maximised by the correct answer, i.e.

$$\mathbb{E}(m | r^*, q) > \mathbb{E}(m | r, q)$$

for any $r \neq r^*$. Assume that Merlin knows P and the function f .

Is this sufficient to incentivise Merlin to respond with the correct answer? If not, what other assumptions or knowledge do we require?

EXERCISE 11. Assume that you need to travel over the weekend. You wish to decide whether to take the train or take the car. Assume that the train and car trip cost exactly the same amount of money. The train trip takes 2 hours. If it does not rain, then the car trip takes 1.5 hour. However, if it rains the road becomes both more slippery and more crowded and so the average trip time is 2.5 hours. Assume that your utility function is equal to the negative amount of time spent travelling: $U(t) = -t$.

1. Let it be Friday. What is the expected utility of taking the car on Sunday? What is the expected utility of taking the train on Sunday? What is the Bayes-optimal decision, assuming you will travel on Sunday?
2. Let it be a rainy Saturday, i.e. that A holds. What is your posterior probability over the two weather stations, given that it has rained, i.e. $P(H_i | A)$? What is the new marginal probability of rain on Sunday, i.e. $P(B | A)$? What is now the expected utility of taking the car versus taking the train on Sunday? What is the Bayes-optimal decision?

EXERCISE 12. It is possible for the utility function to be nonlinear.

1. One example is $U(t) = 1/t$, which is a *convex* utility function. How would you interpret the utility in that case? Without performing the calculations, can you tell in advance whether your optimal decision can change? Verify your answer by calculating the expected utility of the two possible choices.
2. How would you model a problem where the objective involves arriving in time for a particular appointment?

Chapter 3

Decision problems

3.1 Introduction

In this chapter we describe how to formalise statistical decision problems. These involve making decisions whose utility depends on an unknown *state of the world*. In this setting, it is common to assume that the state of the world is a fundamental property that is not influenced by our decisions. However, we can calculate a probability distribution for the state of the world, using a prior belief and some data, and the data that we do obtain may depend on our decisions.

A classical application of this framework is parameter estimation. Therein, we stipulate the existence of a parameterised *law of nature*, and we wish to choose a best-guess set of parameters for the law through measurements and some prior information. An example would be determining the gravitational attraction constant from observations of planetary movements. These measurements are always obtained through experiments, the automatic design of which will be covered in later chapters.

The decisions we make will necessarily depend on both our prior belief and the data we obtain. In the last section of this chapter will examine how sensitive our decisions are to the prior, and how we can choose it so that our decisions are robust.

3.2 Rewards that depend on the outcome of an experiment

Consider the problem of choosing one of two different types of tickets in a raffle. Each type of ticket gives you the chance to win a different prize. The first is a bicycle and the second is a tea set. After n_i tickets are bought for the i -th prize, a number p_i is drawn uniformly from $\{1, \dots, n_i\}$ and the holder of that ticket wins that particular prize. Thus, the raffle guarantees that somebody will win either price. If most people opt for the bicycle, your chance of actually winning it by buying a single ticket is much smaller. However, if you prefer winning a bicycle to winning the tea set, it is not clear what choice you should make in the raffle. The above is the quintessential example for problems where the reward that we obtain depends not only on our decisions, but also in the outcome of an *experiment*.

This problem can be viewed more generally for scenarios where the reward you receive depends not only on your own choice, but also on some other, unknown fact in the world. This may be something completely uncontrollable, and hence you only can make an informed guess.

More formally, given a set of possible actions \mathcal{A} , we must make a decision $a \in \mathcal{A}$ before knowing the outcome ω of an experiment with outcomes in Ω . After the experiment is performed, we obtain a *reward* $r \in \mathcal{R}$ which depends on both the outcome ω of the experiment and our decision. As discussed in the previous chapter, our preferences for some rewards over others are determined by a *utility* function $U : \mathcal{R} \rightarrow \mathbb{R}$, such that we prefer r to r' if and only if $U(r) \geq U(r')$. Now, however, we cannot choose rewards directly. Another example, which will be used throughout this section, is the following.

EXAMPLE 16 (Taking the umbrella). We must decide whether to take an umbrella to work. Our reward depends on whether we get wet and the amount of objects that we

carry. We would rather not get wet and not carry too many things, which can be made more precise by choosing an appropriate utility function. For example, we might put a value of -1 for carrying the umbrella and a value of -10 for getting wet. In this example, the only events of interest are whether it rains or not.

3.2.1 Formalisation of the problem setting

The elements we need to formulate the problem setting are a random variable, a decision variable, a reward function mapping the random and decision variable to a reward, and a utility function that says how much we prefer each reward.

Assumption 3.2.1 (Outcomes). *There exists a probability measure P on $(\Omega, \mathcal{F}_\Omega)$ such that the probability of the random outcome ω being in $A \in \mathcal{F}_\Omega$ is*

$$\mathbb{P}(\omega \in A) = P(A). \quad (3.2.1)$$

The probability measure P is completely independent of any decision that we make.

Assumption 3.2.2 (Utilities). *Given a set of rewards \mathcal{R} , our preferences satisfy Assumptions 2.1.1, 2.1.2, 2.1.3, i.e. preferences are transitive, all rewards are comparable, and there exists a utility function U , measurable with respect to $\mathcal{F}_\mathcal{R}$ such that $U(r) \geq U(r')$ iff $r \succ^* r'$.*

Since the random outcome ω does not depend on our decision a , we must find a way to connect the two. This can be formalised via a reward function, so that the reward that we obtain (whether we get wet or not) depends on both our decision (to take the umbrella) and the random outcome (whether it rains).

Definition 3.2.1 (Reward function). A reward function $\rho : \Omega \times \mathcal{A} \rightarrow \mathcal{R}$ defines the reward we obtain if we select $a \in \mathcal{A}$ and the experimental outcome is $\omega \in \Omega$:

$$r = \rho(\omega, a). \quad (3.2.2)$$

When we discussed the problem of choosing between distributions in Section 2.3.2, we had directly defined probability distributions on the set of rewards. We can now formulate our problem in that setting. First, we define a set of distributions $\{P_a \mid a \in \mathcal{A}\}$ on the reward space $(\mathcal{R}, \mathcal{F}_\mathcal{R})$, such that the decision a amounts to choosing a particular distribution P_a on the rewards.

EXAMPLE 17 (Rock/Paper/Scissors). Consider a simple game of rock-paper-scissors, where your opponent plays a move at the same time as you, so that you cannot influence his move. The opponents moves are thus $\Omega = \{\omega_R, \omega_P, \omega_S\}$.

You have studied your opponent for some time and you *believe* that he is most likely to play rock $P(\omega_R) = 3/6$, somewhat likely to play paper $P(\omega_P) = 2/6$, and less likely to play scissors: $P(\omega_S) = 1/6$. Your decision set is your own moves: $\mathcal{A} = \{a_R, a_P, a_S\}$, for rock, paper, scissors, respectively. The reward set is $\mathcal{R} = \{\text{Win}, \text{Draw}, \text{Lose}\}$.

What is the probability of each reward, for each decision you make? Taking the example of a_R , we see that you win if the opponent plays scissors with probability $1/6$, you lose if the opponent plays paper ($2/6$), and you draw if he plays rock ($3/6$). Consequently, we can convert the outcome probabilities to reward probabilities for every decision:

$$\begin{aligned} P_{a_R}(\text{Win}) &= 1/6, & P_{a_R}(\text{Draw}) &= 3/6, & P_{a_R}(\text{Lose}) &= 2/6 \\ P_{a_P}(\text{Win}) &= 3/6, & P_{a_P}(\text{Draw}) &= 2/6, & P_{a_P}(\text{Lose}) &= 1/6 \\ P_{a_S}(\text{Win}) &= 2/6, & P_{a_S}(\text{Draw}) &= 1/6, & P_{a_S}(\text{Lose}) &= 3/6. \end{aligned}$$

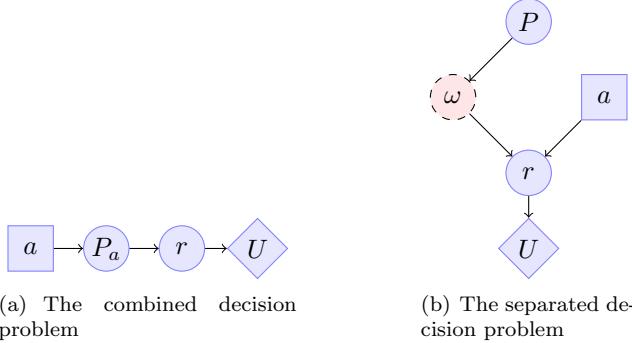


Figure 3.1: Decision diagrams for the combined and separated formulation of the decision problem. Squares denote decision variables, diamonds denote utilities. All other variables are denoted by circles. Arrows denote the flow of dependency.

Of course, what you play depends on our own utility function. If we prefer winning over drawing or losing, we could for example have the utility function $U(\text{Win}) = 1$, $U(\text{Draw}) = 0$, $U(\text{Lose}) = -1$. Then, since $\mathbb{E}_a U = \sum_{\omega \in \Omega} U(\omega, a) P_a(\omega)$, we have

$$\begin{aligned} E_{a_R} U &= -1/6 \\ E_{a_P} U &= 2/6 \\ E_{a_S} U &= -1/6, \end{aligned}$$

so that based on your belief, choosing paper is best.

The above example illustrates that every decision that we make creates a corresponding probability distribution on rewards. While the outcome of the experiment is independent of the decision, the distribution of rewards is effectively chosen by our decision.

Expected utility

The expected utility of any decision $a \in \mathcal{A}$ under P is:

$$\mathbb{E}_{P_a}(U) = \int_{\mathcal{R}} U(r) dP_a(r) = \int_{\Omega} U[\rho(\omega, a)] dP(\omega) \quad (3.2.3)$$

From now on, we shall use the simple notation

$$U(P, a) \triangleq \mathbb{E}_{P_a} U \quad (3.2.4)$$

to denote the expected utility of a under distribution P .

Instead of viewing the decision as effectively choosing a distribution over rewards (Fig. 3.1(a)) we can separate the random part of the process from the deterministic part (Fig. 3.1(b)) by considering a measure P on some space of outcomes Ω , such that the reward depends on both a and the outcome $\omega \in \Omega$ through the reward function $\rho(\omega, a)$. The optimal decision is of course always the $a \in \mathcal{A}$ maximising $\mathbb{E}(U | P_a)$. However, this structure allows us to clearly distinguish the controllable from the random part of the rewards.

The probability measure induced by decisions

For every $a \in \mathcal{A}$, the function $\rho : \Omega \times \mathcal{A} \rightarrow \mathcal{R}$ induces a probability distribution P_a on \mathcal{R} . In fact, for any $B \in \mathcal{F}_{\mathcal{R}}$:

$$P_a(B) \triangleq \mathbb{P}(\rho(\omega, a) \in B) = P(\{\omega \mid \rho(\omega, a) \in B\}). \quad (3.2.5)$$

The above equation requires that the following technical assumption is satisfied. As usual, we employ the expected utility hypothesis (Assumption 2.3.2). Thus, we should choose the decision that results in the highest expected utility.

Assumption 3.2.3. *The sets $\{\omega \mid \rho(\omega, a) \in B\}$ must belong to \mathcal{F}_{Ω} . That is, ρ must be \mathcal{F}_{Ω} -measurable for any a .*

The dependency structure of this problem in either formulation can be visualised in the *decision diagram* shown in Figure 3.1.

EXAMPLE 18 (Continuation of Example 16). You are going to work, and it might rain. The forecast said that the probability of rain (ω_1) was 20%. What do you do?

- a_1 : Take the umbrella.
- a_2 : Risk it!

The reward of a given outcome and decision combination, as well as the expected utility is given in Table 3.1.

$\rho(\omega, a)$	a_1	a_2
ω_1	dry, carrying umbrella	wet
ω_2	dry, carrying umbrella	dry
$U[\rho(\omega, a)]$	a_1	a_2
ω_1	-1	-10
ω_2	-1	1
$\mathbb{E}_P(U \mid a)$	-1	-1

Table 3.1: Rewards, utilities, expected utility for 20% probability of rain.

3.2.2 Decision diagrams

Decision diagrams are also known as *decision networks* or *influence diagrams*. Like the examples shown in Figure 3.1, they are used to show dependencies between different variables. In general, these include the following types of nodes:

- Choice nodes, denoted by squares. These are nodes whose values the decision maker can directly choose. Sometimes there is more than one decision maker involved.
- Value nodes, denoted by diamonds. These are the nodes that the decision maker is interested in influencing. The utility of the decision maker is always a function of the value nodes.

- Circle nodes are used to denote all other types of variables. These include deterministic, stochastic, known or unknown variables.

The nodes are connected via directed edges. These denote the dependencies between nodes. For example, in Figure 3.1(b), the reward is a function of both ω and a , i.e. $r = \rho(\omega, a)$, while ω depends only on the probability distribution P . Typically, there must be a path from a choice node to a value node, otherwise nothing the decision maker can do will influence its utility. Nodes belonging to or observed by different players will usually be denoted by different lines or colors. In Figure 3.1(b), ω , which is not observed, is shown in a lighter color.

3.2.3 Statistical estimation*

Statistical decision problems arise particularly often in *parameter estimation*, such as estimating the covariance matrix of a Gaussian random variable. In this setting, the unknown outcome of the experiment ω is called a *parameter*, while the set of outcomes Ω is called the *parameter space*. Classical statistical estimation involves selecting a single parameter value on the basis of observations. This requires us to specify a preference for different types of estimation errors, and is distinct from the standard Bayesian approach to estimation, which simply calculates a full distribution over all possible parameters.

A simple example is estimating the distribution of votes in an election from a small sample. Depending on whether we are interested in predicting the vote share of individual parties or the most likely winner of the election, we can use a distribution over vote shares (possibly estimated through standard Bayesian methodology) to decide on a share or the winner.

EXAMPLE 19 (Voting). Assume you wish to estimate the number of votes for different candidates in an election. The *unknown parameters* of the problem mainly include: the percentage of likely voters in the population, the probability that a likely voter is going to vote for each candidate. One simple way to estimate this is by polling.

Consider a nation with k political parties. Let $\omega = (\omega_1, \dots, \omega_k) \in [0, 1]^k$ be the voting proportions for each party. We wish to make a guess $a \in [0, 1]^k$. How should we guess, given a distribution $P(\omega)$? How should we select U and ρ ? This depends on what our goal is, when we make the guess.

If we wish to give a reasonable estimate about the votes of all the k parties, we can use the squared error: First, set the error vector $r = (\omega_1 - a_1, \dots, \omega_k - a_k) \in [0, 1]^k$. Then we set $U(r) \triangleq -\|r\|^2$, where $\|r\|^2 = \sum_i |\omega_i - a_i|^2$.

If on the other hand, we just want to predict the winner of the election, then the actual percentages of all individual parties are not important. In that case, we can set $r = 1$ if $\arg \max_i \omega_i = \arg \max_i a_i$ and 0 otherwise, and $U(r) = r$.

Losses and risks

In such problems, it is common to specify a loss instead of a utility. This is usually the negative utility:

Definition 3.2.2 (Loss).

$$\ell(\omega, a) = -U[\rho(\omega, a)]. \quad (3.2.6)$$

Given the above, instead of the expected utility, we consider the expected loss, or risk.

Definition 3.2.3 (Risk).

$$\kappa(P, a) = \int_{\Omega} \ell(\omega, a) dP(\omega). \quad (3.2.7)$$

Of course, the optimal decision is a minimising κ .

3.3 Bayes decisions

The decision which maximises the expected utility under a particular distribution P , is called the *Bayes-optimal* decision, or simply the *Bayes decision*. The probability distribution P is supposed to reflect all our uncertainty about the problem.

Definition 3.3.1 (Bayes-optimal utility). Consider an outcome (or parameter) space Ω , decision space \mathcal{A} , and a utility function $U : \Omega \times \mathcal{A} \rightarrow \mathbb{R}$. For any probability distribution P on Ω , the Bayes-optimal utility $U^*(P)$ is defined as the smallest upper bound on $U(P, a)$ for all decisions $a \in \mathcal{A}$. That is,

$$U^*(P) = \sup_{a \in \mathcal{A}} U(P, a). \quad (3.3.1)$$

The maximisation over decision is usually not easy. However, there exist a few cases where it is relatively simple. The first of those is when the utility function is the negative squared error.

EXAMPLE 20 (Quadratic loss). Consider $\Omega = \mathbb{R}^k$ and $\mathcal{A} = \mathbb{R}^k$. The utility function that, for any point $\omega \in \mathbb{R}$, is defined as

$$U(\omega, a) = -\|\omega - a\|^2 \quad (3.3.2)$$

is called quadratic loss.

Quadratic loss is a very important special case of utility functions, as it is easy to calculate the optimal solution. This is illustrated by the following theorem.

Theorem 3.3.1. *Let P be a measure on Ω and $U : \Omega \times \mathcal{A} \rightarrow \mathbb{R}$ be the quadratic loss defined in Example 20. Then the decision*

$$a = \mathbb{E}_P(\omega), \quad (3.3.3)$$

maximises the expected utility $U(P, a)$, under the technical assumption that $\partial/\partial a|\omega - a|^2$ is measurable with respect to $\mathcal{F}_{\mathbb{R}}$.

Proof. We first write out the expected utility of a decision a .

$$U(P, a) = - \int_{\Omega} \|\omega - a\|^2 dP(\omega).$$

We now take derivatives – due to the measurability assumption, we can swap the order of differentiation and integration:

$$\begin{aligned} \frac{\partial}{\partial a} \int_{\Omega} \|\omega - a\|^2 dP(\omega) &= \int_{\Omega} \frac{\partial}{\partial a} \|\omega - a\|^2 dP(\omega) \\ &= 2 \int_{\Omega} (a - \omega) dP(\omega) \\ &= 2 \int_{\Omega} a dP(\omega) - 2 \int_{\Omega} \omega dP(\omega) \\ &= 2a - 2\mathbb{E}(\omega). \end{aligned}$$

Setting the derivative equal to 0 and noting that the utility is concave, we see that the expected utility is maximised for $a = \mathbb{E}_P(\omega)$. \square

Another simple example is the absolute error, where $U(\omega, a) = |\omega - a|$. The solution in this case differs significantly from the squared error. As can be seen from Figure 3.2(a), for absolute loss, the optimal decision is to choose the a that is closest to the most likely ω . Figure 3.2(b) illustrates the finding of Theorem 3.3.1.

3.3.1 Convexity of the Bayes-optimal utility*

Although finding the optimal decision for an arbitrary utility U and distribution P may be difficult, fortunately the Bayes-optimal utility has some nice properties which enable it to be approximated rather well. In particular, for any decision, the expected utility is linear with respect to our belief P . Consequently, the Bayes-optimal utility is convex with respect to P . This firstly implies that there is a unique “worst” distribution P , against which we cannot do very well. Secondly, we can approximate the Bayes-utility very well for all possible distributions by generalising from a small number of distributions. In order to define linearity and convexity, we first introduce the concept of a mixture of distributions.

Consider two probability measures P, Q on $(\Omega, \mathcal{F}_{\Omega})$. These define two alternative distributions for ω . For any P, Q and $\alpha \in [0, 1]$, we define the *mixture of distributions*

$$Z_{\alpha} \triangleq \alpha P + (1 - \alpha)Q \quad (3.3.4)$$

to mean the probability measure such that $Z_{\alpha}(A) = \alpha P(A) + (1 - \alpha)Q(A)$ for any $A \in \mathcal{F}_{\Omega}$. For any fixed choice a , the expected utility varies linearly with α :

Remark 3.3.1 (Linearity of the expected utility). If Z_{α} is as defined in (3.3.4), then, for any $a \in \mathcal{A}$:

$$U(Z_{\alpha}, a) = \alpha U(P, a) + (1 - \alpha)U(Q, a). \quad (3.3.5)$$

Proof.

$$\begin{aligned} U(Z_{\alpha}, a) &= \int_{\Omega} U(\omega, a) dZ_{\alpha}(\omega) \\ &= \alpha \int_{\Omega} U(\omega, a) dP(\omega) + (1 - \alpha) \int_{\Omega} U(\omega, a) dQ(\omega) \\ &= \alpha U(P, a) + (1 - \alpha)U(Q, a). \end{aligned}$$

\square

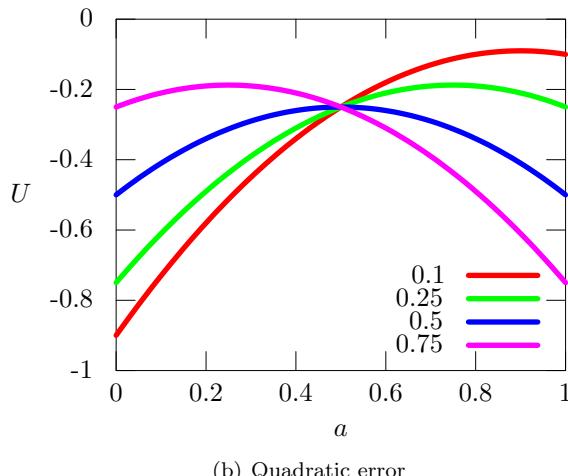
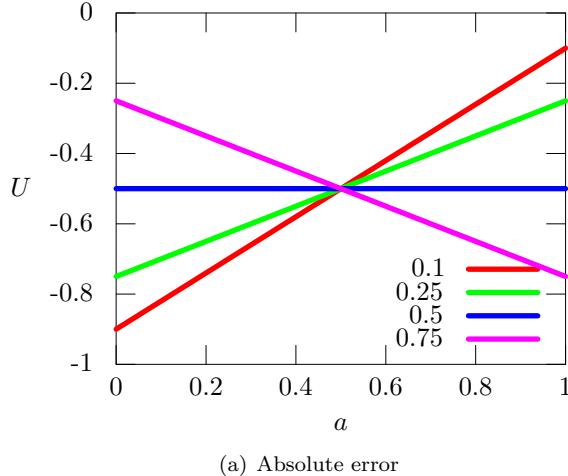


Figure 3.2: Expected utility curves for different values of ω , as the decision a varies in $[0, 1]$.

However, if we consider Bayes-optimal decisions, this is no longer true, because the optimal decision depends on the distribution. In fact, the utility of Bayes-optimal decisions is convex, as the following theorem shows.

Theorem 3.3.2. *For probability measures P, Q on Ω and any $\alpha \in [0, 1]$,*

$$U^*[Z_\alpha] \leq \alpha U^*(P) + (1 - \alpha)U^*(Q), \quad (3.3.6)$$

where $Z_\alpha = \alpha P + (1 - \alpha)Q$.

Proof. From the definition of the expected utility (3.3.5), for any decision $a \in \mathcal{A}$,

$$U(Z_\alpha, a) = \alpha U(P, a) + (1 - \alpha)U(Q, a).$$

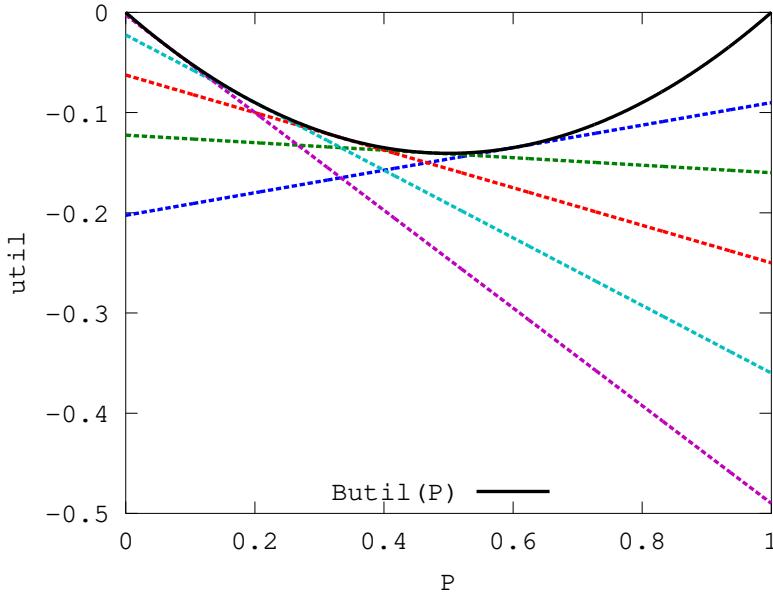


Figure 3.3: A strictly convex Bayes utility.

Hence, by definition (3.3.1) of the Bayes-utility:

$$\begin{aligned} U^*(Z_\alpha) &= \sup_{a \in \mathcal{A}} U(Z_\alpha, a) \\ &= \sup_{a \in \mathcal{A}} [\alpha U(P, a) + (1 - \alpha)U(Q, a)]. \end{aligned}$$

As $\sup_x [f(x) + g(x)] \leq \sup_x f(x) + \sup_x g(x)$, we obtain:

$$\begin{aligned} U^*[Z_\alpha] &\leq \alpha \sup_{a \in \mathcal{A}} U(P, a) + (1 - \alpha) \sup_{a \in \mathcal{A}} U(Q, a) \\ &= \alpha U^*(P) + (1 - \alpha)U^*(Q). \end{aligned}$$

□

As we have proven, the expected utility is linear with respect to Z_α . Thus, for any fixed action a we obtain one of the lines in Fig. 3.3. Due to the theorem just proved, the Bayes-optimal utility is convex. Furthermore, the minimising decision for any Z_α is tangent to the Bayes-optimal utility at the point $(Z_\alpha, U^*(Z_\alpha))$. If we take a decision that is optimal with respect to some Z , but the distribution is in fact $Q \neq Z$, then we are not far from the optimal, Q, Z are close and U^* is smooth. Consequently, we can trivially lower bound the Bayes utility by examining any arbitrary finite set of decisions $\hat{\mathcal{A}}$:

$$U^*(P) \geq \max_{a \in \hat{\mathcal{A}}} U(P, a),$$

for any probability distribution P on Ω . In addition, we can upper-bound the Bayes utility as follows. Take any two distributions P_1, P_2 over Ω . Then, the following upper bound

$$U^*(\alpha P_1 + (1 - \alpha)P_2) \leq \alpha U^*(P_1) + (1 - \alpha)U^*(P_2)$$

holds due to convexity. The two bounds suggest an algorithm for successive approximation of the Bayes-optimal utility, by looking for the largest gap between the lower and the upper bounds.

3.4 Statistical and strategic decision making

We do not need to be restricted to simply choosing one of a finite number of decisions. For example, we could choose a distribution over decisions. In addition, we may wish to consider other criteria than maximising expected utility / minimising risk.

Strategies Instead of choosing a specific decision, we could instead choose to randomise our decision somehow. In other words, instead of our choices being specific decisions, we can choose among distributions over decisions. For example, instead of choosing to eat lasagna or beef, we choose to throw a coin and eat lasagna if the coin comes heads and beef otherwise.

Definition 3.4.1 (Strategies). A strategy σ is a probability measure on \mathcal{A} such that $\sigma(A)$ is the probability that we select a decision $a \in A \subseteq \mathcal{A}$.

Interestingly, *for the type of problems that we have considered so far*, even if we expand our choices to the set of all possible probability measures on \mathcal{A} , there always is one decision (rather than a strategy) which is optimal. In the following we remove the reward function ρ from the decision problem, summarising everything with the utility function U for simplicity.

Theorem 3.4.1. Consider any statistical decision problem with probability measure P on outcomes Ω and with utility function $U : \Omega \times \mathcal{A} \rightarrow \mathbb{R}$. Further let $a^* \in \mathcal{A}$ such that $U(P, a^*) \geq U(P, a)$ for all $a \in \mathcal{A}$. Then for any probability measure σ on \mathcal{A} ,

$$U(P, a^*) \geq U(P, \sigma).$$

Proof.

$$\begin{aligned} U(P, \sigma) &= \int_{\mathcal{A}} U(P, a) d\sigma(a) \\ &\leq \int_{\mathcal{A}} U(P, a^*) d\sigma(a) \\ &= U(P, a^*) \int_{\mathcal{A}} d\sigma(a) = U(P, a^*) \end{aligned}$$

□

This theorem should not be applied naively. It only states that if we know P then the expected utility of the best fixed/deterministic decision $a^* \in \mathcal{A}$ cannot be increased by randomising between decisions.

For example, it does not make sense to apply this theorem to cases where P itself is unknown. This can happen in two cases. The first is when P is chosen by somebody else, analogously to how we choose σ , and its value remains hidden to us. The second is when P is only known partially.

$U(\omega, a)$	a_1	a_2
ω_1	-1	0
ω_2	10	1
$\mathbb{E}(U P, a)$	4.5	0.5
$\min_{\omega} U(\omega, a)$	-1	0

Table 3.2: Utility function, expected utility and maximin utility.

3.4.1 Alternative notions of optimality

There are some situations where maximising expected utility with respect to the distribution on outcomes is unnatural. Two simple examples—where, for simplicity, we consider utility functions $U : \Omega \times \mathcal{A} \rightarrow \mathbb{R}$ on outcomes and decisions directly—are the following.

Maximin/Minimax policies. These policies are useful when we have no information about ω . In that case, we may want to take a worst-case approach and select a^* that maximises the utility in the worst-case ω .

$$U_* = \max_a \min_{\omega} U(\omega, a) = \min_{\omega} U(\omega, a^*) \quad (\text{maximin})$$

The maximin value of the problem can essentially be seen as how much utility we would be able to obtain, if we were to make a decision a first, and nature were to select an adversarial decision ω later. On the other hand, the minimax value is:

$$U^* = \min_{\omega} \max_a U(\omega, a) = \max_a U(\omega^*, a), \quad (\text{minimax})$$

where $\omega^* \triangleq \arg \min_{\omega} \max_a U(\omega, a)$ is the worst-case choice nature could make, if we were to select our own decision a after its own choice was revealed to us.

To illustrate this, consider Table 3.2. Here, we see that a_1 maximises expected utility. However, under a worst-case assumption this is not the case, i.e. the maximin solution is a_2 . Note that by definition

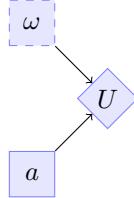
$$U^* \geq U(\omega^*, a^*) \geq U_*. \quad (3.4.1)$$

Maximin/minimax problems are a special case of problems in game theory, in particular two-player zero-sum games. The minimax problem can be seen as a game where the maximising player plays first, and the minimising player second. If $U^* = U_*$, then the game is said to have a value, which implies that if both players are playing optimal, then it doesn't matter which player moves first. More details about these types of problems will be given in Section 3.4.2.

Regret. Instead of calculating the expected utility for each possible decision, we could instead calculate how much utility we would have obtained if we had made the best decision in hindsight. Consider, for example the problem in Table 3.2. There the optimal action is either a_1 or a_2 , depending on whether we accept the probability P over Ω , or adopt a worst-case approach. However, after we make a specific decision, we can always look at the best decision we could have made given the actual outcome ω , as shown in Table 3.3.

$L(\omega, a)$	a_1	a_2
ω_1	1	0
ω_2	0	9
$\mathbb{E}(L P, a)$	0.5	4.5
$\max_{\omega} L(\omega, a)$	1	9

Table 3.3: Regret, in expectation and minimax.

Figure 3.4: Simultaneous two-player stochastic game. The first player (nature) chooses ω , and the second player (the decision maker) chooses a . Then the second player obtains utility $U(\omega, a)$.

Definition 3.4.2 (Regret). The regret of σ is how much we lose compared to the best decision in hindsight, that is,

$$L(\omega, a) \triangleq \max_{a'} U(\omega, a') - U(\omega, a). \quad (3.4.2)$$

The notion of regret is given in Table 3.3, which reuses Example 16. Here, the decision maker has a choice between two actions, while nature has a choice between two outcomes. We can see that the choice minimising regret either in expectation or in the minimax sense is a_1 . This is in contrast to what occurs when we are considering utility. Given the regret of each action-outcome pair, we can now find the decision minimising expected regret $\mathbb{E}(L | P, a)$ and minimising maximum regret $\max_{\omega} L(\omega, a)$, analogously to expected utility and minimax utility. Interestingly, as Table 3.3 shows, in this setting we always prefer action a_2 to a_1 , showing that the concept of regret results in quantitatively different decisions.

3.4.2 Solving minimax problems*

We now view minimax problems as two player games, where one player chooses a and the other player chooses ω . The decision diagram for this problem is given in Figure 3.4.2, where the dashed line indicates that, from the point of view of the decision maker, nature's choice is unobserved before she makes her own decision. A simultaneous two-player game is a game where both players act without knowing each other's decision. From the point of view of the player that chooses a , this is equivalent to assuming that ω is hidden, as shown in Figure 3.4.2. There are other variations of such games, however. For example, their moves may still be revealed after they have played. This is important in the case where the game is played *repeatedly*. However, what is usually revealed is not the belief ξ , which is something assumed to be internal to player one, but ω , the actual decision made by the first player. In other cases, we might have that U itself is not known, and we only observe $U(\omega, a)$ for the choices made.

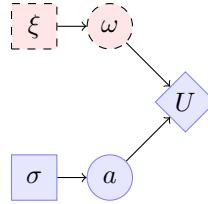


Figure 3.5: Simultaneous two-player stochastic game. The first player (nature) chooses ξ , and the second player (the decision maker) chooses σ . Then $\omega \sim \xi$ and $a \sim \sigma$ and the second player obtains utility $U(\omega, a)$.

Definition 3.4.3 (Strategy). A strategy $\sigma \in \Delta \mathcal{A}$, is a probability distribution over simple decisions $a \in \mathcal{A}$.

In this setting, by allowing the decision maker to select arbitrary strategies σ , we permit her to select arbitrary probability distributions over simple decisions, rather than fixing one decision.

Minimax utility, regret and loss If the decision maker knows the outcome, then the additional flexibility does not help. As we showed for the general case of a distribution over Ω , a simple decision is as good as any randomised strategy:

Remark 3.4.1. For each ω , there is some a such that:

$$U(\omega, a) \in \max_{\sigma \in \Delta \mathcal{A}} U(\omega, \sigma). \quad (3.4.3)$$

What follows are some rather trivial remarks connecting regret with utility in various cases.

Remark 3.4.2.

$$L(\omega, \sigma) = \sum_a \sigma(a) L(\omega, a) \geq 0, \quad (3.4.4)$$

with equality iff σ is ω -optimal.

Proof.

$$\begin{aligned} L(\omega, \sigma) &= \max_{\sigma'} U(\omega, \sigma') - U(\omega, \sigma) = \max_{\sigma'} U(\omega, \sigma') - \sum_a \sigma(a) U(\omega, a) \\ &= \sum_a \sigma(a) \left(\max_{\sigma'} U(\omega, \sigma') - U(\omega, a) \right) \geq 0. \end{aligned}$$

The equality on optimality is obvious. \square

Remark 3.4.3.

$$L(\omega, \sigma) = \max_a U(\omega, a) - U(\omega, \sigma). \quad (3.4.5)$$

Proof. As (3.4.3) shows, for any fixed ω , the best decision is always deterministic,

$$\sum_a \sigma(a) L(\omega, a) = \sum_a \sigma(a) [\max_{a' \in \mathcal{A}} U(\omega, a') - U(\omega, a)] = \max_{a' \in \mathcal{A}} U(\omega, a') - \sum_a \sigma(a) U(\omega, a).$$

\square

U	ω_1	ω_2
a_1	1	-1
a_2	0	0

Table 3.4: Even-bet utility

Remark 3.4.4. $L(\omega, \sigma) = -U(\omega, \sigma)$ iff $\max_a U(\omega, a) = 0$.

Proof. If $\max_{\sigma'} U(\omega, \sigma') - U(\omega, \sigma) = -U(\omega, \sigma)$ then $\max_{\sigma'} U(\omega, \sigma') = \max_a U(\omega, a) = 0$. The converse follows trivially. \square

EXAMPLE 21. (An even-money bet) For this problem, the maximum regret of a policy

L	ω_1	ω_2
a_1	0	1
a_2	1	0

Table 3.5: Even-bet regret

σ can be written as

$$\max_{\omega} L(\omega, \sigma) = \max_{\omega} \sum_a \sigma(a) L(\omega, a) = \max_{\omega} \sigma(a) \mathbb{I}\{\{\} a = a_i \wedge \omega \neq \omega_i\} \cdot 1 \geq 1/2, \quad (3.4.6)$$

since $L(a, \omega) = 0$ when $a = a_i$ and $\omega \neq \omega_i$. In fact, equality is obtained iff $\sigma(a) = 1/2$, giving minimax regret $L^* = 1/2$.

3.4.3 Two-player games

Here we go into some more detail in the connections between minimax theory and the theory of two-player games. In particular, we extend the actions of nature to $\Delta(\Omega)$, the distributions over Ω and our actions to distributions $\Delta(\mathcal{A})$, the distributions over \mathcal{A} .

For two distributions σ, ξ on \mathcal{A} and Ω , define our expected utility to be:

$$U(\xi, \sigma) \triangleq \sum_{\omega \in \Omega} \sum_{a \in \mathcal{A}} U(\omega, a) \xi(\omega) \sigma(a). \quad (3.4.7)$$

Then we define the maximin policy σ^* for which:

$$\min_{\xi} U(\xi, \sigma^*) = U_* \triangleq \max_{\sigma} \min_{\xi} U(\xi, \sigma), \quad (3.4.8)$$

The minimax prior ξ^* satisfies

$$\max_{\sigma} U(\xi^*, \sigma) = U^* \triangleq \min_{\xi} \max_{\sigma} U(\xi, \sigma), \quad (3.4.9)$$

where the solution exists as long as \mathcal{A}, Ω are finite, which we will assume in the following.

Expected regret

We can now define the expected regret for a given pair of distributions ξ, σ as

$$\begin{aligned} L(\xi, \sigma) &= \max_{\sigma'} \sum_{\omega} \xi(\omega) \{U(\omega, \sigma') - U(\omega, \sigma)\} \\ &= \max_{\sigma'} U(\xi, \sigma') - U(\xi, \sigma). \end{aligned} \quad (3.4.10)$$

Not all minimax and maximin policies result in the same value. The following theorem gives a condition under which the game does have a value.

Theorem 3.4.2. *If there exist (perhaps singular) distributions ξ^*, σ^* and $C \in \mathbb{R}$ such that*

$$U(\xi^*, \sigma) \leq C \leq U(\xi, \sigma^*) \quad \forall \xi, \sigma$$

then

$$U^* = U_* = U(\xi^*, \sigma^*) = C.$$

Proof. Since $C \leq U(\xi, \sigma^*)$ for all ξ we have

$$C \leq \min_{\xi} U(\xi, \sigma^*) \leq \max_{\sigma} \min_{\xi} U(\xi, \sigma) = U_*.$$

Similarly

$$C \geq \max_{\sigma} U(\xi^*, \sigma) \geq \min_{\xi} \max_{\sigma} U(\xi, \sigma) = U^*.$$

But then due to (3.4.1)

$$C \geq U^* \geq U_* \geq C.$$

□

One question is whether a solution exists, and if so we can find it. In fact, the type of games we have been looking at so far are called bilinear games. For these, a solution always exists and there are efficient methods for finding it.

Definition 3.4.4. A bilinear game is a tuple $(U, \Xi, \Sigma, \Omega, \mathcal{A})$ with $U : \Xi \times \Sigma \rightarrow \mathbb{R}$ such that all $\xi \in \Xi$ are arbitrary distributions on Ω and all $\sigma \in \Sigma$ are arbitrary distributions on \mathcal{A} :

$$U(\xi, \sigma) \triangleq \mathbb{E}(U | \xi, \sigma) = \sum_{\omega, a} U(\omega, a) \sigma(a) \xi(\omega).$$

Theorem 3.4.3. *For a bilinear game, $U^* = U_*$. In addition, the following three conditions are equivalent:*

1. σ^* is maximin, ξ^* is minimax and $U^* = C$.
2. $U(\xi, \sigma^*) \geq C \geq U(\xi^*, \sigma)$ for all ξ, σ .
3. $U(\omega, \sigma^*) \geq C \geq U(\xi^*, a)$ for all ω, a .

Linear programming formulation

While general games may be hard, bilinear games are easy, in the sense that minimax solutions can be found with well-known algorithms. One example is linear programming. The problem

$$\max_{\sigma} \min_{\xi} U(\xi, \sigma),$$

where ξ, σ are distributions over finite domains, can be converted to finding σ corresponding to the greatest lower bound $v_\sigma \in \mathbb{R}$ on the utility. Using matrix notation, set \mathbf{U} to be the matrix such that $\mathbf{U}_{\omega,a} = U(\omega, a)$, $\pi(a) = \sigma(a)$ and $\xi(\omega) = \xi(\omega)$. Then the problem can be written as:

$$\max \left\{ v_\sigma \mid (\mathbf{U}\pi)_j \geq v_\sigma \forall j, \sum_i \sigma_i = 1, \sigma_i \geq 0 \forall i \right\}.$$

Equivalently, we can find ξ with the least upper bound:

$$\min \left\{ v_\xi \mid (\xi^\top \mathbf{U})_i \leq v_\xi \forall i, \sum_j \xi_j = 1, \xi_j \geq 0 \forall j \right\},$$

where everything has been written in matrix form. In fact, one can show that $v_\xi = v_\sigma$, thus obtaining Theorem 3.4.3.

To understand the connection of two-person games with Bayesian decision theory, take a look at Figure 3.3, seeing the risk as negative expected utility, or as the opponent's gain. Each of the decision lines represents nature's gain as she chooses different prior distributions, while we keep our policy σ fixed. The bottom horizontal line that would be tangent to the Bayes-optimal utility curve would be minimax: if nature were to change priors, since the line is horizontal, it would not increase its gain. On the other hand, if we were to choose another tangent line, we would only increase nature's gain (and decrease our utility).

3.5 Decision problems with observations

So far we have only examined problems where the outcomes were drawn from some fixed distribution. This distribution constituted our subjective belief about what the unknown parameter is. Now, we examine the case where we can obtain some observations that depend on the unknown ω before we make our decision. These observations should give us more information about ω , before making a decision. Intuitively, we should be able to make decisions by simply considering the posterior distribution.

In this setting, we once more need to take some decision $a \in \mathcal{A}$ so as to maximise expected utility. As before, we have a prior distribution ξ on some parameter $\omega \in \Omega$, representing what we know about ω . Consequently, the expected utility of any fixed decision a is going to be $\mathbb{E}_\xi(U | a)$.

However, now we may obtain more information about ω before making a final decision. In particular, each ω corresponds to a *model* of the world P_ω , which is a probability distribution over the observation space \mathcal{S} , such that $P_\omega(X)$ is

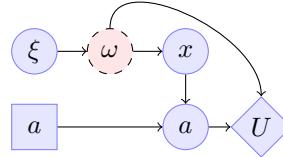


Figure 3.6: Statistical decision problem with observations

the probability that the observation is in $X \subset \mathcal{S}$. The set of parameters Ω thus defines a family of models:

$$\mathcal{P} \triangleq \{P_\omega \mid \omega \in \Omega\}. \quad (3.5.1)$$

Now, consider the case where we take an observation x from the true model P_{ω^*} before making a decision. We can represent the dependency of our decision on the observation by making our decision a function of x :

Definition 3.5.1 (policy). A policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ maps from any observation to a decision.¹

The expected utility of a policy π is:

$$U(\xi, \pi) \triangleq \mathbb{E}_\xi \{U[\omega, \pi(x)]\} = \int_{\Omega} \left(\int_{\mathcal{S}} U[\omega, \pi(x)] dP_\omega(x) \right) d\xi(\omega). \quad (3.5.2)$$

This is the standard Bayesian framework for decision making. It may be slightly more intuitive in some cases to use the notation $\psi(x \mid \omega)$, in order to emphasize that this is a conditional distribution. However, there is no technical difference between the two notations.

When the set of policies includes all constant policies, then there is a policy π^* at least as good as the best fixed decision a^* . More formally:

Remark 3.5.1. Let Π denote a set of policies $\pi : \mathcal{S} \rightarrow \mathcal{A}$. If, $\forall a \in \mathcal{A} \exists \pi \in \Pi$ such that $\pi(x) = a \forall x \in \mathcal{S}$, then $\max_{\pi \in \Pi} \mathbb{E}_\xi(U \mid \pi) \geq \max_{a \in \mathcal{A}} \mathbb{E}_\xi(U \mid a)$.

Proof. The proof follows by setting Π_0 to be the set of constant policies. The result follows since $\Pi_0 \subset \Pi$. \square

We conclude this section with a simple example, about deciding whether or not to go to a restaurant given expert opinions.

EXAMPLE 22. Consider the problem of deciding whether or not to go to a particular restaurant. Let $\Omega = [0, 1]$ with $\omega = 0$ meaning the food is in general horrible and $\omega = 1$ meaning the restaurant is great. Let x_1, \dots, x_n be n expert opinions in $\mathcal{S} = \{0, 1\}$ about the restaurant. Under our model, the probability of observing $x_i = 1$ when the quality of the restaurant is ω is given by $P_\omega(1) = \omega$ and conversely $P_\omega(0) = 1 - \omega$. The probability of observing a particular² sequence x of length n is

$$P_\omega(x) = \omega^s (1 - \omega)^{n-s}$$

with $s = \sum_{i=1}^n x_i$.

¹For that reason, policies are also sometimes called *decision functions* or *decision rules* in the literature.

²We obtain a different probability of observations under the binomial model, but the resulting posterior, and hence the policy, is the same.

Maximising utility when making observations

Statistical procedures based on the assumption that a distribution can be assigned to any parameter in a statistical decision problem, which we are considering here, are called *Bayesian statistical methods*. The scope of these methods has been the subject of much discussion in the statistical literature. See e.g. Savage [1972].

In the following, we shall look at different expressions for the expected utility. We shall overload the utility operator U for various cases: when the parameter is fixed, when the parameter is random, when the decision is fixed, and when the decision depends on the observation x and thus is random as well.

Expected utility of a fixed decision a with $\omega \sim \xi$

We first consider the expected utility of taking a fixed decision $a \in \mathcal{A}$, when $\mathbb{P}(\omega \in B) = \xi(B)$. This is the case we have dealt with so far.

$$U(\xi, a) \triangleq \mathbb{E}_\xi(U | a) = \int_{\Omega} U(\omega, a) d\xi(\omega). \quad (3.5.3)$$

Expected utility of a policy π with fixed $\omega \in \Omega$

Now assume that ω is fixed, but instead of selecting a decision directly, we select a decision that depends on the random observation x , which is distributed according to P_ω on \mathcal{S} . We do this by defining a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$.

$$U(\omega, \pi) = \int_{\mathcal{S}} U(\omega, \pi(x)) dP_\omega(x). \quad (3.5.4)$$

Expected utility of a policy π with $\omega \sim \xi$

Now we generalise to the case where ω is distributed with measure ξ . Note that the expectation of the previous expression (3.5.4) is by definition written as:

$$U(\xi, \pi) = \int_{\Omega} U(\omega, \pi) d\xi(\omega), \quad U^*(\xi) \triangleq \sup_{\pi} U(\xi, \pi) = U(\xi, \pi^*). \quad (3.5.5)$$

Bayes decision rules

We wish to construct the Bayes decision rule, that is, the policy with maximal ξ -expected utility. However, doing so by examining all possible policies is hard, because (usually) there are many more policies than decisions. It is however, easy to find the Bayes decision for each possible observation. This is because it is usually possible to rewrite the expected utility of a policy in terms of the posterior distribution. While this is trivial to do when the outcome and observation spaces are finite, it can be extended to the general case as shown in the following theorem.

Theorem 3.5.1. *If U is non-negative or bounded, then we can reverse the integration order of*

$$U(\xi, \pi) = \mathbb{E}\{U[\omega, \pi(x)]\} = \int_{\Omega} \int_{\mathcal{S}} U[\omega, \pi(x)] dP_{\omega}(x) d\xi(\omega),$$

which is the normal form, to obtain the utility in extensive form, shown below:

$$U(\xi, \pi) = \int_{\mathcal{S}} \int_{\Omega} U[\omega, \pi(x)] d\xi(\omega | x) dP_{\xi}(x), \quad (3.5.6)$$

where $P_{\xi}(x) = \int_{\Omega} P_{\omega}(x) d\xi(\omega)$.

Proof. To prove this when U is non-negative, we shall use Tonelli's theorem. First we need to construct an appropriate product measure. Let $p(x | \omega) \triangleq \frac{dP_{\omega}(x)}{d\nu(x)}$ be the Radon-Nikodym derivative of P_{ω} with respect to some dominating measure ν on \mathcal{S} . Similarly, let $p(\omega) \triangleq \frac{d\xi(\omega)}{d\mu(x)}$ be the corresponding derivative for ξ . Now, the utility can be written as:

$$\begin{aligned} U(\xi, \pi) &= \int_{\Omega} \int_{\mathcal{S}} U[\omega, \pi(x)] p(x | \omega) p(\omega) d\nu(x) d\mu(\omega) \\ &= \int_{\Omega} \int_{\mathcal{S}} h(\omega, x) d\nu(x) d\mu(\omega). \end{aligned}$$

Clearly, if U is non-negative, then so is $h(\omega, x) \triangleq U[\omega, \pi(x)] p(x | \omega) p(\omega)$. Then, Tonelli's theorem can be applied and:

$$\begin{aligned} U(\xi, \pi) &= \int_{\mathcal{S}} \int_{\Omega} h(\omega, x) d\mu(\omega) d\mu(x) \\ &= \int_{\mathcal{S}} \int_{\Omega} U[\omega, \pi(x)] p(x | \omega) p(\omega) d\mu(\omega) d\nu(x) \\ &= \int_{\mathcal{S}} \int_{\Omega} U[\omega, \pi(x)] p(\omega | x) d\mu(\omega) p(x) d\nu(x) \\ &= \int_{\mathcal{S}} \left[\int_{\Omega} U[\omega, \pi(x)] p(\omega | x) d\mu(\omega) \right] \frac{dP_{\xi}(x)}{d\nu(x)} d\nu(x) = \int_{\mathcal{S}} \int_{\Omega} U[\omega, \pi(x)] d\xi(\omega | x) dP_{\xi}(x), \end{aligned}$$

□

We can construct an optimal policy π^* as follows. For any specific observed $x \in \mathcal{S}$, we set $\pi^*(x)$ to:

$$\pi^*(x) \triangleq \arg \max_{a \in \mathcal{A}} \mathbb{E}_{\xi}(U | x, a) = \arg \max_{a \in \mathcal{A}} \int_{\Omega} U(\omega, a) d\xi(\omega | x).$$

So now we can plug π^* in the extensive form to obtain:

$$\int_{\mathcal{S}} \int_{\Omega} U[\omega, \pi^*(x)] d\xi(\omega | x) dP_{\xi}(x) = \int_{\mathcal{S}} \left\{ \max_a \int_{\Omega} U[\omega, a] d\xi(\omega | x) \right\} dP_{\xi}(x).$$

Consequently, there is no need to completely specify the policy before we have seen x . In particular, this would create problems when \mathcal{S} is large.

Definition 3.5.2 (Prior distribution). The distribution ξ is called the *prior distribution* of ω .

Definition 3.5.3 (Marginal distribution). The distribution P_ξ is called the (prior) *marginal distribution* of x .

Definition 3.5.4 (Posterior distribution). The conditional distribution $\xi(\cdot | x)$ is called the *posterior distribution* of ω .

Bayes decision rule.

The *optimal decision* given x , is the optimal decision with respect to the *posterior* $\xi(\omega | x)$. Thus, we do not need to pre-compute the complete Bayes-optimal decision rule.

3.5.1 Decision problems in classification.

Classification is the problem of deciding which class $y \in \mathcal{Y}$ some particular observation $x_t \in \mathcal{X}$ belongs to. From a decision-theoretic viewpoint, the problem can be seen at three different levels. In the first, we are given a classification model in terms of a probability distribution, and we simply wish to classify optimally given the model. In the second, we are given a family of models, a prior distribution on the family, and a training data set, and we wish to classify optimally according to our belief. In the last form of the problem, we are given a set of *policies* $\pi : \mathcal{X} \rightarrow \mathcal{Y}$ and we must choose the one with highest expected performance. The two last classes of the problem are equivalent when the set of policies contains all Bayes decision rules for a specific model family.

Deciding the class given a probabilistic model

In the simple form of the problem, we are already given a classifier P that can calculate probabilities $P(y_t | x_t)$, and we simply must decide upon some class $a_t \in \mathcal{Y}$, so as to maximise a specific utility function. One standard utility function is the prediction accuracy

$$U_t \triangleq \mathbb{I}\{y_t = a_t\}.$$

The probability $P(y_t | x_t)$ is the posterior probability of the class given the observation x_t . If we wish to maximise expected utility, we can simply choose

$$a_t \in \arg \max_{a \in \mathcal{Y}} P(y_t = a | x_t).$$

This defines a particular, simple policy. In fact, for two-class problems with $\mathcal{Y} = \{0, 1\}$, such a rule can be often visualised as a *decision boundary* in \mathcal{X} , on whose one side we decide for class 0 and on whose other side for class 1.

Deciding the class given a model family

In the general form of the problem, we are given a *training* data set $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$, a set of *classification models* $\{P_\omega | \omega \in \Omega\}$, and a prior

distribution ξ on Ω . For each model, we can easily calculate $P_\omega(y_1, \dots, y_n | x_1, \dots, x_n)$. Consequently, we can calculate the posterior distribution

$$\xi(\omega | S) = \frac{P_\omega(y_1, \dots, y_n | x_1, \dots, x_n)\xi(\omega)}{\sum_{\omega' \in \Omega} P_{\omega'}(y_1, \dots, y_n | x_1, \dots, x_n)\xi(\omega')}$$

and the posterior marginal label probability

$$P_{\xi|S}(y_t | x_t) \triangleq P_\xi(y_t | x_t, S) = \sum_{\omega \in \Omega} P_\omega(y_t | x_t)\xi(\omega | S).$$

We can then construct the following simple policy:

$$a_t \in \arg \max_{a \in \mathcal{Y}} \sum_{\omega \in \Omega} P_\omega(y_t | x_t)\xi(\omega | S),$$

Bayes rule

The Bayes-optimal policy under parametrisation constraints*

In some cases, we are restricted to functionally simple policies, which do not contain any Bayes rules as defined above. For example, we might be limited to linear functions of x . Let $\pi : \mathcal{X} \rightarrow \mathcal{A}$ be such a rule and let Π be the set of allowed policies. Given a family of models and a set of training data, we wish to calculate the policy that maximises our expected utility. For a given ω , we can indeed calculate:

$$U(\omega, \pi) = \sum_{x,y} U(y, \pi(x))P_\omega(y | x)P_\omega(x),$$

where we assume an i.i.d. model, i.e. $x_t | \omega \sim P_\omega(x)$ independently of previous observations. Note that to select the optimal rule $\pi \in \Pi$ we also need to know $P_\omega(x)$. For the case where ω is unknown and we have a posterior $\xi(\omega | S)$, the Bayesian framework is easily extensible:

$$U(\xi(\cdot | S), \pi) = \sum_{\omega} \xi(\omega | S) \sum_{x,y} U(y, \pi(x))P_\omega(y | x)P_\omega(x).$$

This maximisation is not generally trivial. However, if our policy Π is parametrised, we can employ optimisation algorithms such as gradient ascent to find a maximum. In particular, it is true that if we sample $\omega \sim \xi(\cdot | S)$, then

$$\nabla_{\pi} U(\xi(\cdot | S), \pi) = \sum_{x,y} \nabla_{\pi} U(y, \pi(x))P_\omega(y | x)P_\omega(x).$$

Fairness in classification problems*

Any policy, when applied to large-scale, real world problems, has certain externalities. This implies that considering only the decision maker's utility is not sufficient. One such issue is fairness.

This concerns desirable properties of policies applied to a population of individuals. For example, college admissions should be decided on variables that inform about merit, but fairness may also require taking into account the fact

that certain communities are inherently disadvantaged. At the same time, a person should not feel that another in a similar situation obtained an unfair advantage. All this must be taken into account while still caring about optimizing the decision maker's utility function. As another example, consider mortgage decisions: while lenders should take into account the creditworthiness of individuals in order to make a profit, society must ensure that they do not unduly discriminate against socially vulnerable groups.

Recent work in fairness for statistical decision making in the classification setting has considered two main notions of fairness. The first uses (conditional) *independence* constraints between a sensitive variable (such as ethnicity) and other variables, such as decisions made. The second type ensures that decisions are *meritocratic*, so that better individuals are favoured, but also smoothness,³ in order to avoid elitism. While a thorough discussion of fairness is beyond the scope of this book, it is useful to note that some of these concepts are impossible to strictly achieve simultaneously, but may be approximately satisfied by careful design of the policy. The recent work by Dwork et al. [2012], Chouldechova [2016], Corbett-Davies et al. [2017], Kleinberg et al. [2016], Kilbertus et al. [2017], Dimitrakakis et al. [2017] goes much more deeply on this topic.

3.5.2 Calculating posteriors

Posterior distributions for multiple observations

We now consider how we can re-write the posterior distribution over Ω incrementally. Assume that we have a prior ξ on Ω . We then observe $x^n \triangleq x_1, \dots, x_n$. For the observation probability, we write:

Observation probability given history x^{n-1} and parameter ω

$$P_\omega(x_n | x^{n-1}) = \frac{P_\omega(x^n)}{P_\omega(x^{n-1})}$$

Now we can write the posterior as follows:

Posterior recursion

$$\xi(\omega | x^n) = \frac{P_\omega(x^n)\xi(\omega)}{P_\xi(x^n)} = \frac{P_\omega(x_n | x^{n-1})\xi(\omega | x^{n-1})}{P_\xi(x_n | x^{n-1})}. \quad (3.5.7)$$

Here $P_\xi(\cdot | \cdot) = \int_\Omega P_\omega(\cdot | \cdot) d\xi(\omega)$ is a marginal distribution.

Posterior distributions for multiple independent observations

Now we consider the case where, given the parameter ω , the next observation does not depend on the history: If $P_\omega(x_n | x^{n-1}) = P_\omega(x_n)$ then $P_\omega(x^n) = \prod_{k=1}^n P_\omega(x_k)$. Then:

³More precisely Lipschitz conditions on the policy

Posterior recursion with conditional independence

$$\xi_n(\omega) \triangleq \xi_0(\omega | x^n) = \frac{P_\omega(x^n)\xi_0(\omega)}{P_{\xi_0}(x_n)} \quad (3.5.8)$$

$$= \xi_{n-1}(\omega | x_n) = \frac{P_\omega(x_n)\xi_{n-1}(\omega)}{P_{\xi_{n-1}}(x_n)}, \quad (3.5.9)$$

where ξ_t is the belief at time t . Here $P_{\xi_n}(\cdot | \cdot) = \int_{\Omega} P_\omega(\cdot | \cdot) d\xi_n(\omega)$ is the marginal distribution with respect to the n -th posterior.

Conditional independence allows us to write the posterior update as an identical recursion at each time t . We shall take advantage of that when we look at *conjugate prior* distributions in Chapter 4. For such models, the recursion involves a particularly simple parameter update.

3.6 Summary.

In this chapter, we introduced a general framework for making decisions $a \in \mathcal{A}$ whose optimality depends on an unknown outcome or parameter ω . We saw that, when our knowledge about $\omega \in \Omega$ is in terms of a probability distribution ξ on Ω , then the utility of the Bayes-optimal decision is convex with respect to ξ .

In some cases, observations $x \in \mathcal{X}$ may affect our belief, leading to a posterior $\xi(\cdot | x)$. This requires us to introduce the notion of a policy $\pi : \mathcal{X} \rightarrow \mathcal{A}$ mapping observations to decisions. While it is possible to construct a complete policy by computing $U(\xi, \pi)$ for all *policies* (normal form) and maximising, it is frequently simpler to just wait until we observe x and compute $U[\xi(\cdot | x), a]$ for all *decisions* (extensive form).

In minimax settings, we can consider a fixed but unknown parameter ω or a fixed but unknown prior ξ . This links statistical decision theory to game theory.

3.7 Exercises

The first part of this exercise set considers problems where we are simply given some distribution over Ω . In the second part, the distribution is a posterior distribution that depends on observations x .

3.7.1 Problems with no observations.

For the first part of exercises, we consider a set of worlds Ω and a decision set \mathcal{A} , as well as the following utility function $U : \Omega \times \mathcal{A} \rightarrow \mathbb{R}$:

$$U(\omega, a) = \text{sinc}(\omega - a) \quad (3.7.1)$$

where $\text{sinc}(x) = \sin(x)/x$. If ω is known and $\mathcal{A} = \Omega = \mathbb{R}$ then obviously the optimal decision is $a = \omega$, as $\text{sinc}(x) \leq \text{sinc}(0) = 1$. However, we consider the following case:

$$\Omega = \mathcal{A} = \{-2.5, \dots, -0.5, 0, 0.5, \dots, 2.5\}.$$

EXERCISE 13. Assume ω is drawn from ξ , with $\xi(\omega) = 1/11$ for all $\omega \in \Omega$, calculate and plot the expected utility $U(\xi, a) = \sum_{\omega} \xi(\omega) U(\omega, a)$ for each a . Report $\max_a U(\xi, a)$.

EXERCISE 14 (5). Assume $\omega \in \Omega$ is arbitrary (but deterministically selected). Calculate the utility $U(a) = \min_{\omega} U(\omega, a)$ for each a . Report $\max(U)$.

EXERCISE 15. Again assume $\omega \in \Omega$ is arbitrary (but deterministically selected). We now allow for stochastic policies π on \mathcal{A} . Then the expected utility is $U(\omega, \pi) = \sum_a U(\omega, a)\pi(a)$.

(a) Calculate and plot the expected utility when $\pi(a) = 1/11$ for all a , reporting values for all ω .

(b) Find

$$\max_{\pi} \min_{\xi} U(\xi, \pi).$$

Hint: Use the linear programming formulation, adding a constant to the utility matrix U so that all elements are non-negative.

EXERCISE 16. Consider the definition of rules that, for some $\epsilon > 0$, select a maximising

$$P \left(\left\{ \omega \mid U(\omega, a) > \sup_{d' \in \mathcal{A}} U(\omega, d') - \epsilon \right\} \right). \quad (3.7.2)$$

Prove that this is indeed a statistical decision problem, i.e. it corresponds to maximising the expectation of some utility function.

3.7.2 Problems with observations.

For this section, we consider a set of worlds Ω and a decision set \mathcal{A} , as well as the following utility function $U : \Omega \times \mathcal{A} \rightarrow \mathbb{R}$:

$$U(\omega, a) = -|\omega - a|^2. \quad (3.7.3)$$

In addition, we consider a family of distributions on a sample space $S = \{0, 1\}^n$,

$$\mathcal{F} \triangleq \{f_{\omega} \mid \omega \in \Omega\}, \quad (3.7.4)$$

such that f_ω is the binomial probability mass function with parameters ω (with the number of draws n being implied).

Consider the parameter set:

$$\Omega = \{0, 0.1, \dots, 0.9, 1\}. \quad (3.7.5)$$

Let ξ be the uniform distribution on Ω , such that $\xi(\omega) = 1/11$ for all $\omega \in \Omega$. Let the decision set be:

$$\mathcal{A} = [0, 1]. \quad (3.7.6)$$

EXERCISE 17. What is the decision a^* maximising $U(\xi, a) = \sum_\omega \xi(\omega)U(\omega, a)$ and what is $U(\xi, a^*)$?

EXERCISE 18. In the same setting, we now observe the sequence $x = (x_1, x_2, x_3) = (1, 0, 1)$.

1. Plot the posterior distribution $\xi(\omega | x)$ and compare it to the posterior we would obtain if our prior on ω was $\xi' = \text{Beta}(2, 2)$.
2. Find the decision a^* maximising the *a posteriori* expected utility

$$\mathbb{E}_\xi(U | a, x) = \sum_\omega U(\omega, a)\xi(\omega | x).$$

3. Consider $n = 2$, i.e. $S = \{0, 1\}^2$. Calculate the Bayes-optimal expected utility in extensive form:

$$\mathbb{E}_\xi(U | \pi^*) = \sum_S \phi(x) \sum_\omega U[\omega, \pi^*(x)]\xi(\omega | x) = \sum_S \phi(x) \max_a \sum_\omega U[\omega, a]\xi(\omega | x), \quad (3.7.7)$$

where $\phi(x) = \sum_\omega f_\omega(x)\xi(\omega)$ is the prior marginal distribution of x and $\delta^* : S \rightarrow \mathcal{A}$ is the Bayes-optimal decision rule.

Hint: You can simplify the computational complexity somewhat, since you only need to calculate the probability of $\sum_t x_t$. This is not necessary to solve the problem though.

EXERCISE 19. In the same setting, we consider nature to be adversarial. Once more, we observe $x = (1, 0, 1)$. Assume that nature can choose a prior among a set of priors $\Xi = \{\xi_1, \xi_2\}$. Let $\xi_1(\omega) = 1/11$ and $\xi_2(\omega) = \omega/5.5$.

1. Calculate and plot our value for deterministic decisions a :

$$\min_{\xi \in \Xi} \mathbb{E}_\xi(U | a, x).$$

2. Find the minimax prior ξ^*

$$\min_{\xi \in \Xi} \max_{a \in \mathcal{A}} \mathbb{E}_\xi(U | a)$$

Hint: Apart from the adversarial prior selection, this is very similar to the previous exercise.

Chapter 4

Estimation

4.1 Introduction

In the previous unit, we have seen how to make optimal decisions with respect to a given utility function and belief. However, one important question is how a new belief can be calculated from observations and a prior belief. More generally, we wish to examine how much information we can obtain about an unknown parameter from observations, and how to bound our errors. Hence, while most of this chapter will focus on the Bayesian framework for estimating parameters, we shall also look at tools for making conclusions about the value of parameters without making specific assumptions about the data distribution, i.e. without providing specific prior information.

In the Bayesian setting, we calculate posterior distributions of parameters given data. The basic problem can be stated as follows. Let $\mathcal{P} \triangleq \{P_\omega \mid \omega \in \Omega\}$ be a family of probability measures on $(\mathcal{S}, \mathcal{F}_S)$ and ξ be our prior probability measure on $(\Omega, \mathcal{F}_\Omega)$. Given some data $x \sim P_{\omega^*}$, with $\omega^* \in \Omega$, how can we estimate ω^* ? The Bayesian approach is to estimate the posterior distribution $\xi(\cdot \mid x)$, instead of guessing a single ω^* . In general, the posterior measure is a function $\xi(\cdot \mid x) : \mathcal{F}_\Omega \rightarrow [0, 1]$, with:

$$\xi(B \mid x) = \frac{\int_B P_\omega(x) d\xi(\omega)}{\int_\Omega P_\omega(x) d\xi(\omega)}. \quad (4.1.1)$$

The posterior distribution allows us to quantify our uncertainty about the unknown ω^* . This in turn enables us to take decisions that take uncertainty into account.

The first question we are concerned with in this chapter is how to calculate this posterior for any value of x in practice. If x is a complicated object, this may be computationally difficult. In fact, the posterior distribution can also be a complex function. However, there exist distribution families and priors such that this calculation is very easy, in the sense that the functional form of the posterior depends upon a small number of parameters. This happens when a summary of the data that contains all necessary information can be calculated easily. Formally, this is captured via the concept of a sufficient statistic.

4.2 Sufficient statistics

Sometimes we want to summarise the data we have observed. This can happen when the data is a long sequence of simple observations $x^t = (x_1, \dots, x_t)$. It may also be useful to do so when we have a single observation x , such as a high-resolution image. For some applications, it may be sufficient to only calculate a really simple function of the data, such as the sample mean, defined below:

Definition 4.2.1 (Sample mean). The sample mean $\bar{x}_t : \mathbb{R}^t \rightarrow \mathbb{R}$ of a sequence $x_k \in \mathbb{R}$ is defined as:

$$\bar{x}_t \triangleq \frac{1}{t} \sum_{k=1}^t x_k. \quad (4.2.1)$$

statistic

This summary, or any other function of the observations is called a *statistic*. In particular, we are interested in statistics that can replace all the complete original data in our calculations, without losing any information. Such statistics are called *sufficient*.

4.2.1 Sufficient statistics

We consider the standard probabilistic setting. Let \mathcal{S} be a sample space and Ω be a parameter space defining a family of measures on \mathcal{S} :

$$\mathcal{P} = \{P_\omega \mid \omega \in \Omega\}.$$

In addition, we must also define an appropriate prior distribution ξ on the parameter space Ω . Now let us proceed to the definition of a sufficient statistic in the Bayesian sense.¹

Definition 4.2.2. Let Ξ be a set of prior distributions on Ω , which indexes a family $\mathcal{P} = \{P_\omega \mid \omega \in \Omega\}$ of distributions on \mathcal{S} . A statistic $\phi : \mathcal{S} \rightarrow \mathcal{Z}$, where \mathcal{Z} is a vector space² is a *sufficient statistic* for $\langle \mathcal{P}, \Xi \rangle$ if:

$$\xi(\cdot \mid x) = \xi(\cdot \mid x') \quad (4.2.2)$$

for any prior $\xi \in \Xi$ and any $x, x' \in \mathcal{S}$ such that $\phi(x) = \phi(x')$.

This simply states that the statistic is sufficient if, whenever we obtain the same value of the statistic for two different datasets x, x' , then the resulting posterior distribution over the parameters is identical, no matter what the prior distribution. In other words, the value of the statistic is sufficient for computing the posterior. Interestingly, a sufficient statistic always implies the following factorisation for members of the family.

Theorem 4.2.1. *A statistic $\phi : \mathcal{S} \rightarrow \mathcal{Z}$ is sufficient $\langle \mathcal{P}, \text{Bels} \rangle$ iff there exist functions $u : \mathcal{S} \rightarrow (0, \infty)$, and $v : \mathcal{Z} \times \Omega \rightarrow [0, \infty)$ such that $\forall x \in \mathcal{S}, \omega \in \Omega$:*

$$P_\omega(x) = u(x)v[\phi(x), \omega], \quad u > 0, v \geq 0. \quad (4.2.3)$$

Proof. The proof will be for the general case. The case when Ω is finite is technically simpler and is left as an exercise. Assume the existence of u, v . Then for any $B \in \mathcal{F}_\Omega$:

$$\begin{aligned} \xi(B \mid x) &= \frac{\int_B u(x)v[\phi(x), \omega] d\xi(\omega)}{\int_\Omega u(x)v[\phi(x), \omega] d\xi(\omega)} \\ &= \frac{\int_B v[\phi(x), \omega] d\xi(\omega)}{\int_\Omega v[\phi(x), \omega] d\xi(\omega)}. \end{aligned}$$

If $\phi(x) = \phi(x')$, then the above is also equal to $\xi(B \mid x')$, so $\xi(\cdot \mid x) = \xi(\cdot \mid x')$. Thus, ϕ satisfies the definition of a sufficient statistic.

Conversely, let ϕ be a sufficient statistic. Let μ be a dominating measure on \mathcal{S} so that we can define the densities $p(\omega) \triangleq \frac{d\xi(\omega)}{d\mu(\omega)}$ and

$$p(\omega \mid x) \triangleq \frac{d\xi(\omega \mid x)}{d\mu(\omega)} = \frac{P_\omega(x)p(\omega)}{\int_\Omega P_\omega(x) d\xi(\omega)}.$$

¹There is an alternative definition, which replaces equality of posterior distributions with point-wise equality on the family members, i.e. $P_\omega(x) = P_\omega(x') \forall \omega$. This is a stronger definition, as it implies the Bayesian one we use here.

²Typically $\mathcal{Z} \subset \mathbb{R}^k$ for finite-dimensional statistics.

Consequently, we can write:

$$P_\omega(x) = \frac{p(\omega | x)}{p(\omega)} \int_{\Omega} P_\omega(x) d\xi(\omega).$$

Since ϕ is sufficient, there is by definition some function $g : \mathcal{Z} \times \Omega \rightarrow [0, \infty)$ such that $p(\omega | x) = g[\phi(x), \omega]$. Consequently, we can factorise P_ω as:

$$P_\omega(x) = v[\phi(x), \omega] u(x),$$

where $u(x) = \int_{\Omega} P_\omega(x) d\xi(\omega)$ and $v[\phi(x), \omega] = g[\phi(x), \omega] / \xi(\omega)$. \square

In the factorisation of Theorem 4.2.1, u is the only factor that depends directly on x . Interestingly, it *does not appear* in the posterior calculation at all. So, the posterior only depends on x through the statistic.

EXAMPLE 23. Suppose $x^t = (x_1, \dots, x_t)$ is a random sample from a Bernoulli distribution with parameter ω . Then the joint probability is

$$P_\omega(x^t) = \prod_{k=1}^t P_\omega(x_k) = \omega^{s_t} (1 - \omega)^{t - s_t}$$

with $s_t = \sum_{k=1}^t x_k$ being the number of times 1 has been observed until time t . Then the statistic $\phi(x^t) = s_t$ satisfies (4.2.3) with $u(x) = 1$, while $P_\omega(x^t)$ only depends on the data through the statistic $s_t = \phi(x^t)$.

Another, example is when we have a *finite* set of models. Then the sufficient statistic is always a finite-dimensional vector.

Lemma 4.2.1. *Let a family $\mathcal{P} = \{P_\theta | \theta \in \Theta\}$, where each model P_θ is a probability measure on \mathcal{X} and Θ contains n models. If $\mathbf{p} \in \Delta^n$ is a vector representing our prior distribution, i.e. $\xi(\theta) = p_\theta$, then a sufficient statistic is the finite-dimensional vector $q_\theta = p_\theta P_\theta(x)$.*

Proof. Simply note that the posterior distribution in this case is

$$\xi(\theta | x) = \frac{q_\theta}{\sum_{\theta'} q_{\theta'}}.$$

Thus, all the information we need to compute the posterior is \mathbf{q} . Alternatively, we could also use a vector \mathbf{w} with $w_\theta = \frac{q_\theta}{\sum_{\theta'} q_{\theta'}}$. \square

In other words, when dealing with a finite set of models, it's always possible to maintain a finite dimensional sufficient statistic. This could simply be the actual posterior distribution, since that is also a finite-dimensional vector.

More generally, however, the prior and posterior distributions are functions (i.e. they have an infinite number of points and so cannot be represented as finite vectors). There are nevertheless still cases where we can compute posterior distributions efficiently.

4.2.2 Exponential families

Many well-known distributions such as the Gaussian, Bernoulli and Dirichlet distribution are members of the exponential family of distributions. All those distributions are factorisable in the manner shown below, while at the same time they have fixed-dimension sufficient statistics.

Definition 4.2.3. A distribution family $\mathcal{P} = \{P_\omega \mid \omega \in \Omega\}$ with P_ω being a probability function (or density) on the sample space \mathcal{S} , is said to be an *exponential family* if for any $x \in \mathcal{S}, \omega \in \Omega$:

$$P_\omega(x) = a(\omega)b(x) \exp \left[\sum_{i=1}^k g_i(\omega)h_i(x) \right]. \quad (4.2.4)$$

Informally, it is interesting to know that among families of distributions satisfying certain smoothness conditions, only exponential families have a fixed-dimension sufficient statistic.

Because of this, exponential family distributions admit so-called parametric *conjugate* prior distribution families. These have the property that any posterior distribution calculated will remain within the conjugate family. Frequently, because of the simplicity of the statistic used, calculation of the conjugate posterior parameters is very simple.

4.3 Conjugate priors

In this section, we examine some well-known conjugate families. First, we give sufficient conditions for the existence of conjugate family of priors for a given distribution family and statistic. While this section can be used as a reference, the reader may wish to initially only look at the first few example families.

The following remark gives sufficient conditions for the existence of a finite-dimensional sufficient statistic.

Remark 4.3.1. If a family \mathcal{P} of distributions on \mathcal{S} has a sufficient statistic $\phi : \mathcal{S} \rightarrow \mathcal{Z}$ of *fixed* dimension for any $x \in \mathcal{S}$, then there exists a conjugate family of priors $\Xi = \{\xi_\alpha \mid \alpha \in A\}$, where A is a set of possible parameters for the prior distribution, such that:

1. $P_\omega(x)$ is proportional to some $\xi_\alpha \in \Xi$:

$$\forall x \in S, \exists \xi_\alpha \in \Xi, c > 0 : \int_B P_\omega(x) d\xi_\alpha(\omega) = c \xi_\alpha(B), \forall B \in \mathcal{F}_\Omega$$

2. The family is closed under multiplication:

$$\forall \xi_1, \xi_2 \in \Xi, \exists \xi_\alpha \in \Xi, c > 0$$

such that:

$$\xi_\alpha = c \xi_1 \xi_2.$$

While conjugate families exist for statistics with unbounded dimension, here we shall focus on finite-dimensional families. We will start with the simplest example, the Bernoulli-Beta pair.

4.3.1 Bernoulli-Beta conjugate pair

The Bernoulli-Beta conjugate pair of families is useful for problems where we wish to measure success rates of independent trials. First, we shall give details on the Bernoulli distribution. Then, we shall define the Beta distribution and describe its conjugate relation to the Bernoulli.

The Bernoulli distribution is a discrete distribution with outcomes taking values in $\{0, 1\}$. It is ideal for modelling the outcomes of independent random trials with fixed probability of success. The structure of the graphical model in

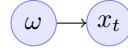


Figure 4.1: Bernoulli graphical model

Figure 4.1 shows the dependencies between the different variables of the model, which is explained below.

Definition 4.3.1 (Bernoulli distribution). The Bernoulli distribution is discrete with outcomes $\mathcal{S} = \{0, 1\}$, parameter $\omega \in [0, 1]$, and probability function:

$$P_\omega(x_t = u) = \begin{cases} \omega, & u = 1 \\ 1 - \omega, & u = 0 \end{cases} = \omega^u(1 - \omega)^{1-u}.$$

If x_t is distributed according to a Bernoulli distribution with parameter ω , we write $x_t \sim \text{Bern}(\omega)$.

The Bernoulli distribution can be extended to $\mathcal{S} = \{0, 1\}^n$ by modelling each outcome as independent. Then $P_\omega(x^n) = \prod_{t=1}^n P_\omega(x_t)$. This is the probability of observing the exact sequence x^t under the Bernoulli model. However, in many cases we are interested in the probability of observing the particular number of 1s and 0s and do not care about the actual order. In that case, we use what is called the binomial distribution.

We first need a way to *count* the cases where, out of n trials, we have k positive outcomes. This is given by the *binomial coefficient*, defined below:

$$\binom{n}{k} \triangleq \frac{\prod_{i=0}^{k-1}(n-i)}{k!}, \quad k, n \in \mathbb{N}, \quad (4.3.1)$$

and $\binom{0}{k} = 1$. It follows that

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \quad k, n \in \mathbb{N}, k \geq n. \quad (4.3.2)$$

Now we can define the binomial distribution in terms of the binomial coefficient. This is just a scaled product-Bernoulli distribution for multiple independent outcomes, but where we want to measure the probability of a particular number of 1s or 0s.

Definition 4.3.2 (Binomial Distribution). Let us denote the total number of 1's observed until time t by $s_t = \sum_{k=1}^t x_k$. Then the probability that, k out

of t trials will be positive can be written in terms of the probability function $f(k | t, \omega)$ of the binomial distribution:

$$\mathbb{P}(s_t = k | \omega) = f(k | t, \omega) \triangleq \binom{t}{k} \omega^k (1 - \omega)^{t-k}. \quad (4.3.3)$$

If s_t is drawn from a binomial distribution with parameters ω, t , we write $s_t \sim \text{Binom}(\omega, t)$.

The Bernoulli is a distribution on a sequence of outcomes, while the binomial is a distribution on the total number of positive outcomes. This is why the binomial distribution includes the binomial coefficient, which basically counts the number of possible sequences of length t that have k positive outcomes.

Let us return to the Bernoulli distribution. If the ω parameter is known, then all the observations are independent of each other. However, this is not the case when ω is unknown. For example, let $\Omega = \{\omega_1, \omega_2\}$. Then

$$\mathbb{P}(x^t) = \sum_{\omega \in \Omega} \mathbb{P}(x^t | \omega) \mathbb{P}(\omega) = \sum_{\omega} \prod_{k=1}^t \mathbb{P}(x_k | \omega) \mathbb{P}(\omega) \neq \prod_{k=1}^t \mathbb{P}(x_k).$$

In general, however $\Omega = [0, 1]$. In that case, is there a prior distribution that could succinctly describe our uncertainty about the parameter? Indeed, there is, and it is called the *Beta distribution*. This distribution is defined on the interval $[0, 1]$.

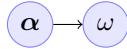


Figure 4.2: Beta graphical model

$[0, 1]$ has two parameters that determine the density of the observations. Because the Bernoulli distribution has a parameter in $[0, 1]$, the outcomes of the Beta can be used to specify a prior on the parameters of the Bernoulli distribution. Let us now call the distribution's outcomes ω and its parameter α . The dependencies between the parameters are described in the graphical model of Figure 4.2.

Definition 4.3.3 (Beta distribution). The Beta distribution has outcomes $\omega \in \Omega = [0, 1]$ and parameters $\alpha_0, \alpha_1 > 0$, $\alpha = (\alpha_1, \alpha_0)$. It is defined via its probability density function:

$$p(\omega | \alpha) = \frac{\Gamma(\alpha_0 + \alpha_1)}{\Gamma(\alpha_0)\Gamma(\alpha_1)} \omega^{\alpha_1-1} (1 - \omega)^{\alpha_0-1}, \quad (4.3.4)$$

where Γ is the *gamma function*. If ω is distributed according to a Beta distribution with parameters α_1, α_0 , we write: $\omega \sim \text{Beta}(\alpha_1, \alpha_0)$.

A Beta distribution with parameter α has expectation

$$\mathbb{E}(\omega | \alpha) = \alpha_1 / (\alpha_0 + \alpha_1) //$$

and variance

$$\mathbb{V}(\omega | \alpha) = \frac{\alpha_1 \alpha_0}{(\alpha_1 + \alpha_0)^2 (\alpha_1 + \alpha_0 + 1)}.$$

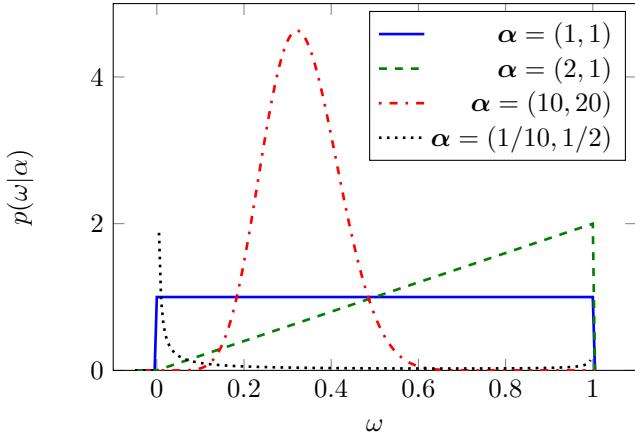


Figure 4.3: Four example Beta densities

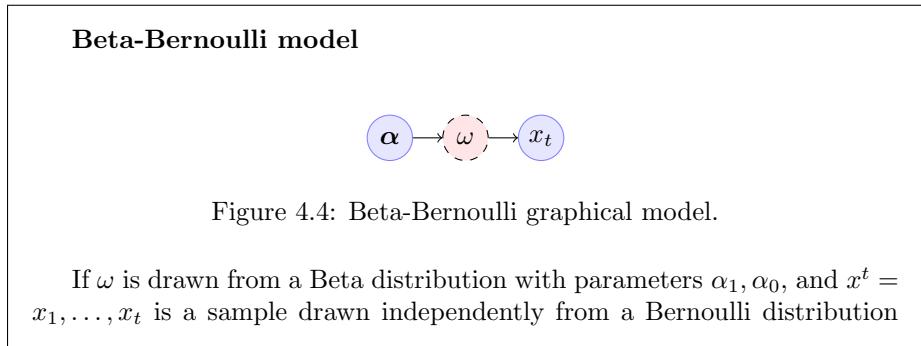
Figure 4.3 shows the density of a Beta distribution for four different parameter vectors. When $\alpha_0 = \alpha_1 = 1$, the distribution is equivalent to a uniform one. The Beta distribution is useful for expressing probabilities of random variables in bounded intervals. In particular, since probabilities of events take values in $[0, 1]$, the Beta distribution is an excellent choice for expressing uncertainty about a probability.

Beta prior for Bernoulli distributions

We can encode our uncertainty about an unknown parameter of the Bernoulli distribution using a Beta distribution. The main idea is to assume that the Bernoulli parameter $\omega \in [0, 1]$ is unknown but fixed. We define a Beta prior distribution for ω to represent our uncertainty. This can be summarised by a parameter vector $\boldsymbol{\alpha}$ and we write $\xi_0(B) \triangleq \int_B p(\omega | \boldsymbol{\alpha}) d\omega$ for our prior distribution ξ_0 . It is easy to see that in that case, the posterior probability is:

$$p(\omega | x^t, \boldsymbol{\alpha}) = \frac{\prod_{k=1}^t P_\omega(x_k) p(\omega | \boldsymbol{\alpha})}{\int_\Omega \prod_{k=1}^t P_\omega(x_k) p(\omega | \boldsymbol{\alpha}) d\omega} \propto \omega^{s_{t,1} + \alpha_1 - 1} (1 - \omega)^{s_{t,0} + \alpha_0 - 1},$$

where $s_{t,1} = \sum_{k=1}^t x_k$ and $s_{t,0} = t - s_{t,1}$ is the total number of 1s and 0s respectively. As you can see, this again has the form of a Beta distribution.



with parameter ω , i.e.:

$$\omega \sim \text{Beta}(\alpha_1, \alpha_0) \quad x^t | \omega \sim \text{Bern}^t(\omega), \quad (4.3.5)$$

then the posterior distribution of ω given the sample the posterior distribution is also Beta:

$$\omega | x^t \sim \text{Beta}(\alpha'_1, \alpha'_0), \quad \alpha'_1 = \alpha_1 + \sum_{k=1}^t x_k, \quad \alpha'_0 = \alpha_0 + t - \sum_{k=1}^t x_k \quad (4.3.6)$$

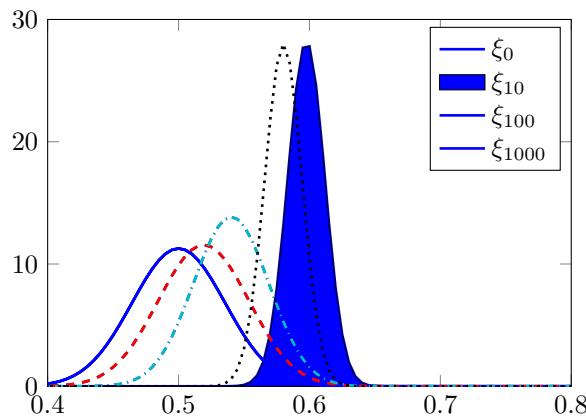


Figure 4.5: Changing beliefs as we observe tosses from a coin with probability $\omega = 0.6$ of heads.

EXAMPLE 24. The parameter $\omega \in [0, 1]$ of a randomly selected coin can be modelled as a Beta distribution peaking around 1/2. Usually one assumes that coins are fair. However, not all coins are exactly the same. Thus, it is possible that each coin deviates slightly from fairness. We can use a Beta distribution to model how likely (we think) different values ω of coin parameters are.

To demonstrate how belief changes, we perform the following simple experiment. Imagine a coin such that, when it is tossed, it has a probability 0.6 of coming heads every time it is tossed, independently of previous outcomes. Thus, the distribution of outcomes is a Bernoulli distribution with parameter $\omega = 0.6$.

We wish to form an accurate belief about how biased the coin is, under the assumption that the outcomes are Bernoulli with parameter ω . Our initial belief, ξ_0 , is modelled as a Beta distribution on the parameter space $\Omega = [0, 1]$, with parameters $\alpha_0 = \alpha_1 = 100$. This places a strong prior on the coin being close to fair. However, we still allow for the possibility that the coin is biased.

Figure 4.5 shows a sequence of beliefs at times 0, 10, 100, 1000 respectively, from a coin with bias $\omega = 0.6$. Due to the strength of our prior, after 10 observations, the situation has not changed much and the belief ξ_{10} is very close to the starting one. However, after 100 observations, our belief has now shifted towards 0.6, the true bias of the coin. After a total of 1000 observations, our belief is centered very close to 0.6, and is now much more concentrated, reflecting the fact that we are almost certain about the value of ω .

4.3.2 Conjugates for the normal distribution

The well-known normal distribution is also endowed with suitable conjugate priors. We first give the definition of the normal distribution, then consider the cases where we wish to estimate its mean, its variance, or both at the same time.

Definition 4.3.4 (Normal distribution). The normal distribution is a continuous distribution, with outcomes in \mathbb{R} . It has two parameters, the mean $\omega \in \mathbb{R}$, and the variance $\sigma^2 \in \mathbb{R}^+$, or alternatively the precision $r \in \mathbb{R}^+$, where $\sigma^2 = r^{-1}$. It has the following probability density function:

$$f(x_t | \omega, r) = \sqrt{\frac{r}{2\pi}} \exp\left(-\frac{r}{2}(x_t - \omega)^2\right). \quad (4.3.7)$$

When x is distributed according to a normal distribution with parameters ω, r^{-1} , we write $x \sim \mathcal{N}(\omega, r^{-1})$. For a sample of size t , we write $x^t \sim \mathcal{N}^t(\omega, r^{-1})$. Independent samples satisfy the following independence condition

$$f(x^t | \omega, r) = \prod_{k=1}^t f(x_k | \omega, r) = \left(\frac{r}{\sqrt{2\pi}}\right)^t \exp\left(-\frac{r}{2} \sum_{k=1}^t (x_k - \omega)^2\right) \quad (4.3.8)$$

The dependency graph in Figure 4.6 shows the dependencies between the parameters of a normal distribution and observations x_t . In this graph, only a

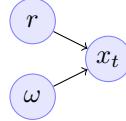


Figure 4.6: Normal graphical model

single sample x_t is shown, and it is implied that all x_t are independent of each other given r, ω .

*standard normal
 χ^2 distribution*

Transformations of normal samples. The normal distribution is interesting mainly because many actual distributions turn out to be approximately normal. Further interesting properties of the normal distribution concern transformations of normal samples. For example, if x^n is drawn from a normal distribution with mean ω and precision r , then $\sum_{k=1}^n x_k \sim \mathcal{N}(n\omega, nr^{-1})$. Finally, if they are drawn from the *standard normal* distribution, i.e. $x_t \sim \mathcal{N}(0, 1)$, then $\sum_{k=1}^n x_t^2$ has a χ^2 distribution with n degrees of freedom.

Normal distribution with known precision, unknown mean

The simplest normal estimation problem occurs when we only need to estimate the mean, but we assume that the variance (or equivalently the precision) is known. For Bayesian estimation, it is convenient to assume that the mean ω is drawn from *another* normal distribution with known mean, as this results in a conjugate pair. Hence, we end up with a posterior normal distribution for the mean as well.

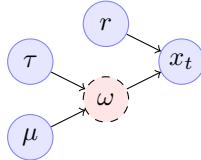
Normal-Normal conjugate pair


Figure 4.7: Normal with unknown mean, graphical model

If ω is drawn from a Normal distribution with parameters mean μ and precision τ , while x_1, \dots, x_n is a sample drawn independently from a Normal distribution with mean ω and precision r , i.e.

$$x^n \sim \mathcal{N}^n(\omega, r^{-1}), \quad \omega \sim \mathcal{N}(\mu, \tau^{-1}),$$

then the posterior distribution of ω given the sample is also normal:

$$\omega | x^n \sim \mathcal{N}(\mu', 1/\tau'), \quad \mu' = \frac{\tau\mu + nr\bar{x}_t}{\tau'}, \quad \tau' = \tau + nr,$$

and $\bar{x}_n \triangleq \frac{1}{n} \sum_{k=1}^n x_k$.

In this case, our new estimate for the mean is shifted towards the empirical mean \bar{x}_t , and our precision increases linearly with the number of samples. Now we examine the case where we know the mean, but not the precision, of the normal distribution. This requires introducing another distribution as a prior for the value of the precision.

Normal with unknown precision and known mean

To model normal distributions with known mean, but unknown precision (or equivalently, unknown variance), we use the Gamma distribution to represent our uncertainty about the precision. The Gamma distribution itself is a two-parameter distribution, whose graphical model is shown in Figure 4.8. Since

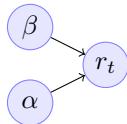


Figure 4.8: Gamma graphical model

the precision of the normal distribution is a positive parameter, the Gamma distribution only has support on $[0, \infty)$. Its two parameters determine the shape and scale of the distribution, as illustrated in Figure 4.9.

Definition 4.3.5 (Gamma distribution). A random variable $r \sim \text{Gamma}(\alpha, \beta)$ is a random variable with outcomes in $[0, \infty)$, and probability density function:

$$f(r | \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} r^{\alpha-1} e^{-\beta r},$$

where $\alpha, \beta > 0$ and $\Gamma(\alpha) = \int_0^\infty u^{\alpha-1} e^{-u} du$ is the Gamma function (see also Appendix C.1.2).

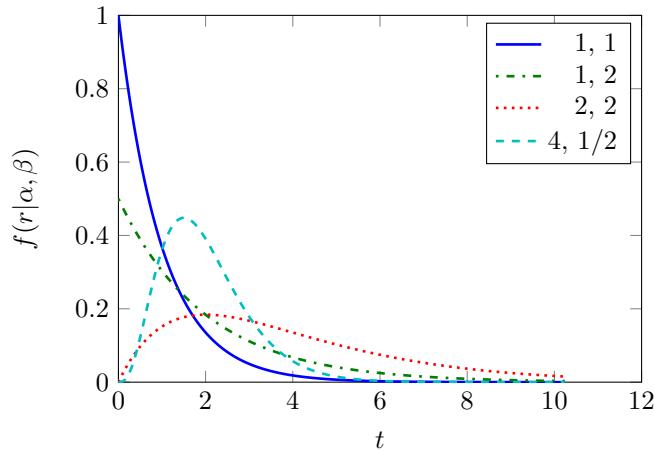


Figure 4.9: Example Gamma densities

exponential distribution

A few example Gamma densities are shown in Figure 4.9. Some of those choices are special, as the Gamma distribution is a generalisation of a number of other standard distributions. For $\alpha = 1$, $\beta > 0$ one obtains an *exponential distribution* with parameter β and probability density function

$$f(x | \beta) = \beta e^{-\beta x}, x > 0. \quad (4.3.9)$$

For $n \in \mathbb{N}$ and $\alpha = n/2$, $\beta = 1/2$ one obtains a χ^2 distribution with n degrees of freedom.

As already mentioned, the Gamma distribution is a natural choice for representing uncertainty about the accuracy of a normal distribution with known mean and unknown accuracy.

Normal-Gamma model

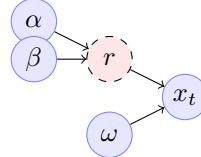


Figure 4.10: Normal-Gamma graphical model for normal distributions with unknown precision.

If r is drawn from a Gamma distribution with parameters α, β , while x^n is a sample drawn independently from a normal distribution with mean ω and precision r , i.e.

$$x^n | r \sim \mathcal{N}^n(\omega, 1/r), \quad r \sim \text{Gamma}(\alpha, \beta)$$

then the posterior distribution of r given the sample is also Gamma:

$$r | x^n \sim \text{Gamma}(\alpha', \beta'), \quad \alpha' = \alpha + \frac{n}{2}, \quad \beta' = \beta + \frac{1}{2} \sum_{i=1}^n (x_i - \omega)^2$$

4.3.3 Normal with unknown precision and unknown mean

The more general problem is estimating a normal distribution when both the mean and the precision are unknown. In that case, we can use the same prior distributions for the mean and precision as when just one of them was unknown. The important thing to note is that the precision is independent of the mean, while the mean has a normal distribution given the precision.

Normal with unknown mean and precision

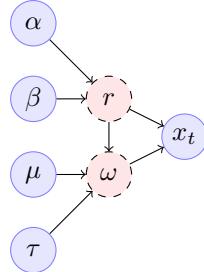


Figure 4.11: Graphical model for a normal distribution with unknown mean and precision, graphical model

Given a sample x^n from a normal distribution with unknown mean ω and precision r , whose prior joint distribution satisfies

$$\omega | r \sim \mathcal{N}(\mu, 1/(\tau)), \quad r \sim \text{Gamma}(\alpha, \beta), \quad (4.3.10)$$

then the posterior distribution is

$$\omega | r, x^n \sim \mathcal{N}(\mu', 1/(\tau'r)), \quad r | x^n \sim \text{Gamma}(\alpha', \beta'). \quad (4.3.11)$$

where

$$\mu' = \frac{\tau\mu + n\bar{x}}{\tau + n}, \quad \tau' = \tau + n, \quad (4.3.12)$$

$$\alpha' = \alpha + \frac{n}{2}, \quad \beta' = \beta + \frac{1}{2} \sum_{i=1}^n (x_i - \bar{x})_n^2 + \frac{\tau n(\bar{x} - \mu)^2}{2(\tau + n)}. \quad (4.3.13)$$

In fact, while $\omega | r$ has normal distribution, the marginal distribution of ω is not normal. In fact, it can be shown that it has a *student t-distribution*. However, we are frequently interested in the marginal distribution of a set of observations x^n . This is also has a generalised student *t*-distribution, which is described below.

The marginal distribution of x . For a normal distribution with mean ω , precision r , we have

$$f(x | \omega, r) \propto r^{1/2} \exp\left(-\frac{r}{2}(\omega - x)^2\right).$$

For a prior $\omega|r \sim \mathcal{N}(\mu, 1/\tau r)$ and $r \sim \text{Gamma}(\alpha, \beta)$, as before, we have the following joint distribution for the mean and precision:

$$\xi(\omega, r) \propto r^{1/2} e^{-(\tau r/2)(\omega - \mu)^2} r^{\alpha-1} e^{-\beta r},$$

as $\xi(\omega, r) = \xi(\omega | r)\xi(r)$. Now we can write the marginal density of new observations as

$$\begin{aligned} p_\xi(x) &= \int f(x | \omega, r) d\xi(\omega, r) \\ &\propto \int_0^\infty \int_{-\infty}^\infty r^{1/2} e^{-\frac{r}{2}(\omega-x)^2} e^{-(\tau r/2)(\omega-\mu)^2} r^{\alpha-1} e^{-\beta r} d\omega dr \\ &= \int_0^\infty r^{\alpha-1/2} e^{-\beta r} \int_{-\infty}^\infty e^{-\frac{r}{2}(\omega-x)^2 - (\tau r/2)(\omega-\mu)^2} d\omega dr \\ &= \int_0^\infty r^{\alpha-1/2} e^{-\beta r} \left(\int_{-\infty}^\infty e^{-\frac{r}{2}[(\omega-x)^2 + \tau(\omega-\mu)^2]} d\omega \right) dr \\ &= \int_0^\infty r^{\alpha-1/2} e^{-\beta r} e^{-\frac{\tau r}{2(\tau+1)}(\mu-x)^2} \sqrt{\frac{2\pi}{r(1+\tau)}} dr \end{aligned}$$

4.3.4 Conjugates for multivariate distributions

The binomial distribution, as well as the normal distribution can be extended to multiple dimensions. Fortunately, multivariate extensions exist for their corresponding conjugate priors as well.

Multinomial-Dirichlet conjugates

The multinomial distribution is the extension of the binomial distribution to an arbitrary number of outcomes. Consider an outcome set $S = \{1, \dots, K\}$.

This is a common model for independent random trials with a finite number of possible outcomes, such as repeated dice throws, multi-class classification problems, etc.

We now perform n trials, such that the outcome of each trial is independent of the rest. This is an extension of a sequence of n Bernoulli trials, but with a potentially larger set of possible outcomes in each trial.

The *multinomial* distribution gives the probability of obtaining outcome i exactly n_i times, given that we perform a total of n . The dependencies between the variables are given in Figure 4.12

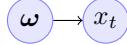


Figure 4.12: Multinomial graphical model.

Definition 4.3.6 (Multinomial distribution). This is a discrete distribution with K outcomes $x_k \in S = \{1, \dots, K\}$. There is a vector parameter $\omega \in \mathbb{R}^K$, with $\|\omega\|_1 = 1$ and $\omega_i \geq 0$, with ω_i representing the probability of obtaining the i -th outcome. In other words, it is defined on the simplex Δ^K .³ The outcomes are i.i.d., so that $\mathbb{P}(x_t = i | \omega) = \omega_i$ for all i, t . Let us denote the number of times the i -th outcome was observed until time t by $n_{t,i} \triangleq \sum_{k=1}^t \mathbb{I}\{x_k = i\}$. Then the probability of obtaining a particular vector of outcome counts $\mathbf{n}_t = (n_{t,i})_{i=1}^K$ at time t is:

$$\mathbb{P}(\mathbf{n}_t | \omega) = \frac{t!}{\prod_{i=1}^K n_{t,i}!} \prod_{i=1}^K \omega_i^{n_{t,i}}, \quad (4.3.14)$$

The Dirichlet distribution

The Dirichlet distribution is the multivariate extension of the Beta distribution. It has a vector parameter α that determines the density of the observations, as shown in Figure 4.13.

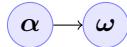


Figure 4.13: Dirichlet graphical model

Definition 4.3.7 (Dirichlet distribution). The Dirichlet distribution is a continuous distribution with outcomes $\omega \in \Omega = \Delta^K$, i.e. $\|\omega\|_1 = 1$ and $\omega_i \geq 0$ and parameter vector $\alpha \in \mathbb{R}_+^K$. If

$$\omega \sim \text{Dir}(\alpha),$$

then it is distributed according to the following probability density function:

$$f(\omega | \alpha) = \frac{\Gamma(\sum_{i=1}^K \alpha_i)}{\prod_{i=1}^K \Gamma(\alpha_i)} \prod \omega_i^{\alpha_i - 1}, \quad (4.3.15)$$

³ Due to these constraints, given $\omega_1, \dots, \omega_{K-1}$, the value of ω_K is fully determined from the remaining parameters.

The Dirichlet distribution is consequently a natural candidate for a prior on the multinomial distribution, as its support coincides with the parameter space of the latter. In fact, the Dirichlet distribution is conjugate to the multinomial distribution in the same way that the Beta distribution is conjugate to the Bernoulli/binomial distribution.

Multinomial distribution with unknown parameter.



Figure 4.14: Dirichlet-multinomial graphical model.

When the multinomial parameter ω is unknown, we can assume it is generated from a Dirichlet distribution as shown in Figure 4.14. In particular, if we observe $x^t = (x_1, \dots, x_t)$, and our prior is given by $\text{Dir}(\alpha)$, so that our initial belief is $\xi_0(\omega) \triangleq f(\omega | \alpha)$, the resulting posterior after t observations is:

$$\xi_t(\omega) \propto \prod_{i=1}^K \omega_i^{n_{t,i} + \alpha_i - 1} \quad (4.3.16)$$

where $n_{t,i} = \sum_{k=1}^t \mathbb{I}\{x_k = i\}$.

Multivariate normal conjugate families

The last conjugate pair we shall discuss is that for multivariate normal distributions. Similarly to the extension of the Bernoulli distribution to the multinomial, and the corresponding extension of the Beta to the Dirichlet, the normal priors can be extended to the multivariate case. The prior of the mean becomes a multivariate normal distribution, while that of the precision becomes a Wishart distribution.

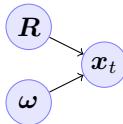


Figure 4.15: Multivariate normal graphical model

Definition 4.3.8 (Multivariate normal distribution). The multivariate normal distribution is a continuous distribution, with outcome space $S = \mathbb{R}^K$, and the following parameters: the mean $\omega \in \mathbb{R}^K$ and the precision⁴ $R \in \mathbb{R}^{K \times K}$, with $\mathbf{x}^\top R \mathbf{x} > 0$ for any $\mathbf{x} \neq 0$, as shown in Figure 4.15. Its probability density function, where $|R|$ denotes the *matrix determinant*, is:

⁴In other words, the inverse of the covariance.

$$f(\mathbf{x}_t | \boldsymbol{\omega}, \mathbf{R}) = (2\pi)^{-K/2} |\mathbf{R}|^{1/2} \exp\left(-\frac{1}{2}(\mathbf{x}_t - \boldsymbol{\omega}^\top \mathbf{R}(\mathbf{x}_t - \boldsymbol{\omega}))\right). \quad (4.3.17)$$

Samples from the multivariate normal distribution are i.i.d. so that $f(\mathbf{x}^t | \boldsymbol{\omega}, \mathbf{R}) = \prod_{k=1}^t f(\mathbf{x}_k | \boldsymbol{\omega}, \mathbf{R})$.

First, we remind ourselves of the definition of a matrix trace:

Definition 4.3.9. The trace of a $n \times n$ square matrix A is

$$\text{trace}(A) \triangleq \sum_{i=1}^n a_{ii}.$$

Definition 4.3.10 (Wishart distribution). The Wishart distribution is a *matrix distribution* on $\mathbb{R}^{K \times K}$ with $n > K - 1$ degrees of freedom and precision matrix $\mathbf{T} \in \mathbb{R}^{K \times K}$. Its probability density function, for any positive $\mathbf{V} \in \mathbb{R}^{K \times K}$, is given by:

$$f(\mathbf{V} | n, \mathbf{T}) \propto |\mathbf{T}|^{n/2} |\mathbf{V}|^{(n-K-1)/2} e^{-\frac{1}{2} \text{trace}(\mathbf{T}\mathbf{V})}. \quad (4.3.18)$$

Construction of the Wishart distribution. Let \mathbf{x}^n be drawn independently from a multivariate normal distribution with mean $\boldsymbol{\omega} \in \mathbb{R}^K$, and precision matrix $\mathbf{T} \in \mathbb{R}^{K \times K}$, that is $\mathbf{x}^n \sim \mathcal{N}^n(\boldsymbol{\omega}, \mathbf{T}^{-1})$. Let $\bar{\mathbf{x}}_n$ be the empirical mean, and define the covariance matrix $\mathbf{S} = \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}}_n)(\mathbf{x}_i - \bar{\mathbf{x}}_n)^\top$. Then \mathbf{S} has a Wishart distribution with $n - 1$ degrees of freedom and precision matrix \mathbf{T} and we write $\mathbf{S} \sim \text{Wish}(n - 1, \mathbf{T})$.

Normal-Wishart conjugate prior

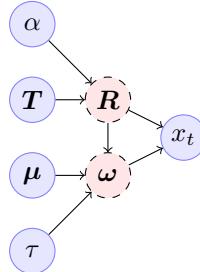


Figure 4.16: Normal-Wishart graphical model.

Theorem 4.3.1. Given a sample \mathbf{x}^n from a multivariate normal distribution in \mathbb{R}^K with unknown mean $\boldsymbol{\omega} \in \mathbb{R}^K$ and precision $\mathbf{R} \in \mathbb{R}^{K \times K}$ whose joint prior distribution satisfies:

$$\boldsymbol{\omega} | \mathbf{R} \sim \mathcal{N}(\boldsymbol{\mu}, (\tau \mathbf{R})^{-1}), \quad \mathbf{R} \sim \text{Wish}(\alpha, \mathbf{T}), \quad (4.3.19)$$

with $\tau > 0$, $\alpha > K - 1$, $\mathbf{T} > 0$, the posterior distribution is

$$\boldsymbol{\omega} | \mathbf{R} \sim \mathcal{N}\left(\frac{\tau \boldsymbol{\mu} + n \bar{\mathbf{x}}}{\tau + n}, [(\tau + n) \mathbf{R}]^{-1}\right), \quad (4.3.20)$$

$$\mathbf{R} \sim \text{Wish}\left(\alpha + n, \mathbf{T} + \mathbf{S} + \frac{\tau n}{\tau + n} (\boldsymbol{\mu} - \bar{\mathbf{x}})(\boldsymbol{\mu} - \bar{\mathbf{x}})^\top.\right), \quad (4.3.21)$$

where $\mathbf{S} = \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$.

4.4 Credible intervals

According to our current belief ξ , there is a certain subjective probability that the unknown parameter ω takes a certain value. However, we are not always interested in the precise probability distribution itself. Instead, we can use the complete distribution to describe an interval that we think contains the true value of the unknown parameter. In Bayesian parlance, this is called a credible interval.

Definition 4.4.1 (Credible interval). Given some probability measure ξ on Ω representing our belief and some interval $A \subset \Omega$,

$$\xi(A) = \int_A d\xi = \mathbb{P}(\omega \in A | \xi).$$

is our subjective belief that the unknown parameter ω is in A . If $\xi(A) = s$, then we say that A is an s -credible interval (or set), or an interval of size (or measure) s .

As an example, for prior distributions on \mathbb{R} , constructing an s -credible interval is usually done by finding $\omega_l, \omega_u \in \mathbb{R}$ such that

$$\xi([\omega_l, \omega_u]) = s.$$

Note that, *any* choice of A such that $\xi(A) = s$ is valid. However, typically the interval is chosen so as to exclude the tails (extremes) of the distribution and centered in the maximum. Figure 4.17 shows the 90% credible interval for

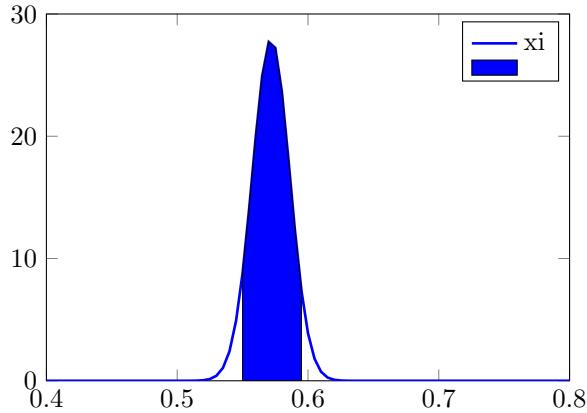


Figure 4.17: 90% credible interval after 1000 observations from a Bernoulli with $\omega = 0.6$.

the Bernoulli parameter. We see that the true parameter lies slightly outside it. (The measure of A under ξ is $\xi(A) = 0.9$.)

What is the probability that the true value of ω will be within a particular credible interval? This will depend on how well our prior ξ_0 matches the true distribution from which the parameter ω was drawn.

Reliability of credible intervals

Assume ϕ, ξ_0 are probability measures on the parameter set Ω , where our prior belief is ξ_0 and ϕ is the actual distribution of $\omega \in \Omega$. Each ω defines a measure P_ω on the observation set \mathcal{S} . We would like to construct a credible interval $A_t \subset \Omega$ (which is a random variable $A_t : S^t \rightarrow \mathcal{F}_\Omega$) such that it has measure $s = \xi_t(A_t)$ for all t . Finally, let $Q \triangleq \int_\omega P_\omega d\phi(\omega)$ be the marginal distribution on \mathcal{S} . Then the probability that the credible interval A_t will not include ω is

$$Q(\{x^t \in S^t \mid \omega \notin A_t\}).$$

The main question is how this failure probability relates to s, t and ξ_0 . So, let us design and conduct experiment for examining how often a typical credible interval includes the parameter we are interested in. In order to do so, we will have Nature draw the parameter from some arbitrary distribution ϕ , which may differ from our own assumed prior distribution ξ_0 .

Experimental testing of a credible interval

- 1: Given a probability family $\mathcal{P} = \{P_\omega \mid \omega \in \Omega\}$.
- 2: Nature chooses distribution ϕ over Ω .
- 3: We choose another distribution ξ_0 over Ω .
- 4: **for** $k = 1, \dots, n$ **do**
- 5: Draw $\omega_k \sim \phi$.
- 6: Draw $x^T \mid \omega_k \sim P_{\omega_k}$.
- 7: **for** $t = 1, \dots, T$ **do**
- 8: Calculate $\xi_t(\cdot) = \xi_0(\cdot \mid x^t)$ for all t .
- 9: Calculate A_t , for all t with $\xi_t(A_t) = 0.5$.
- 10: Check failure: $\epsilon_{t,k} = \mathbb{I}\{\omega_k \notin A_t\}$
- 11: **end for**
- 12: **end for**
- 13: Average over all k : $\epsilon_t = \frac{1}{n} \sum_{k=1}^n \epsilon_{t,k}$.

We performed this experiment for $n = 1000$ trials and for $T = 100$ observations per trial. Figure 4.18 illustrates what happens when $\phi = \xi_0$. We see that the credible interval is always centered around our initial mean guess and that it is quite tight. Figure 4.19 shows the failure rate the credible interval A_t around our estimated mean did not match the actual value of ω_k . Since the measure of our interval A_t is always $\xi_t(A_t) = 1/2$, we expect our error probability to be $1/2$, and this is borne out by the experimental results.

On the other hand, Figure 4.20 illustrates what happens when $\phi \neq \xi_0$. In fact in that case, $\phi(\omega) = \delta(\omega - 0.6)$, so that $\omega_k = 0.6$ for all trials k . We see that the credible interval is always centered around our initial mean guess and that it is always quite tight. Figure 4.21 shows the average number of failures. We see that initially, due to the fact that our prior is different from the distribution from

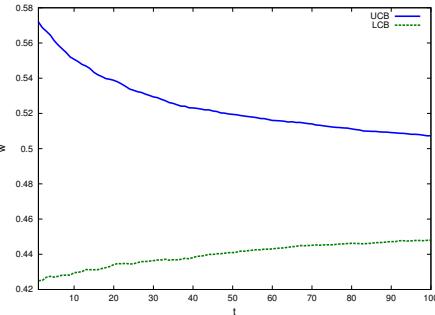


Figure 4.18: 50% credible intervals for a prior $\text{Beta}(10, 10)$, matching the distribution of ω .

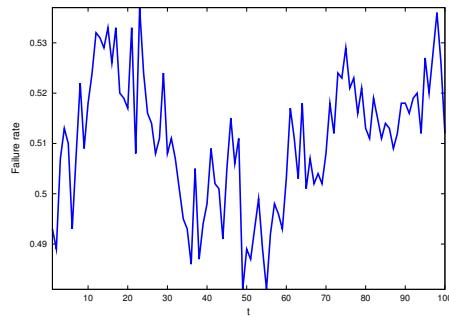


Figure 4.19: Failure rate of 50% CI for a prior $\text{Beta}(10, 10)$, matching the distribution of ω .

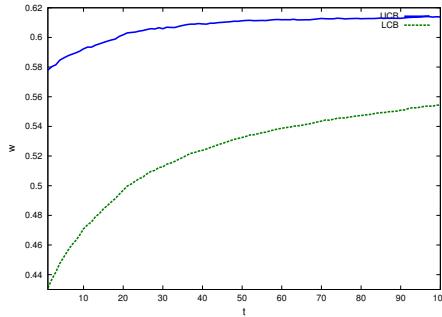
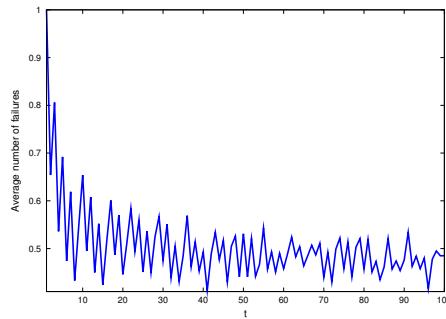
which the ω_k are selected, we make many more mistakes. However, eventually, our prior is swamped by the data and our error rate converges to 50%.

4.5 Concentration inequalities

While Bayesian ideas are useful, as they allow us to express our subjective beliefs about a particular unknown quantity, they nevertheless are difficult to employ when we have no good intuition about what prior to use. One way to overcome this difficulty is by looking at the Bayesian estimation problem as a minimax game between us and nature, as seen in the previous chapter. In this case, we assume that nature chooses the prior distribution in the worst possible way. However, even in that case, we must select a family of distributions and priors.

This section will examine what guarantees we can give about any calculation we make from observations, if we make only very minimal assumptions about the distribution generating these observations. The results are fundamental, in the sense that they rely on a very general phenomenon, called *concentration of measure*. As a consequence, they are much stronger than results such as the central limit theorem (which is not treated in this textbook). However, here we shall focus on their most common applications.

It is interesting to consider the case calculating a sample mean, as given in Definition 4.2.1. We have seen that, for the Beta-Bernoulli conjugate prior, it

Figure 4.20: 50% credible intervals for a prior $\text{Beta}(10, 10)$, when $\omega = 0.6$.Figure 4.21: Failure rate of 50% CI for a prior $\text{Beta}(10, 10)$, when $\omega = 0.6$.

is a simple enough matter to calculate a posterior distribution. From that, we can obtain a credible interval on the expected value of the unknown Bernoulli distribution. However, we would like to do the same for arbitrary distributions on $[0, 1]$, rather than just the Bernoulli. We shall now give an overview of a set of tools that can be used to do this.

Theorem 4.5.1 (Markov's inequality). *If $X \sim P$ with P a distribution on $[0, \infty)$, then:*

$$\mathbb{P}(X \geq t) \leq \frac{\mathbb{E} X}{t}, \quad (4.5.1)$$

where $\mathbb{P}(X \geq t) = P(\{x \mid x \geq t\})$.

Proof. The expectation of X is:

$$\begin{aligned} \mathbb{E} X &= \int_0^\infty x dP(x) \\ &= \int_0^u x dP(x) + \int_u^\infty x dP(x) \\ &\geq 0 + \int_u^\infty u dP(x) \\ &= u P(\{x \mid x \geq u\}) = u \mathbb{P}(X \geq u). \end{aligned}$$

□

Consequently, if \bar{x}_t is the empirical mean after t observations, for a random variable X with expectation $\mathbb{E} X = \mu$, we can use Markov's inequality to show that $\mathbb{P}(|\bar{x}_t - \mu| \geq \epsilon) \leq \mathbb{E} |\bar{x}_t - \mu|/\epsilon$. For $X \in [0, 1]$, we obtain the bound

$$\mathbb{P}(|\bar{x}_t - \mu| \geq \epsilon) \leq 1/\epsilon.$$

Unfortunately this bound does not improve for a larger number of observations t . However, we can get significantly better bounds through various transformations. For monotonic f ,

$$\mathbb{P}(X \geq t) = \mathbb{P}(f(X) \geq f(t)) \quad (4.5.2)$$

as $\{x \mid x \geq t\} = \{x \mid f(x) \geq f(t)\}$. Thus, we can apply Markov's inequality as a building block in other inequalities. The first of those is Chebyshev's inequality.

Theorem 4.5.2 (Chebyshev inequality). *Let X be a random variable with expectation $\mu = \mathbb{E} X$ and variance $\sigma^2 = \mathbb{V} X$. Then, for all $k > 0$:*

$$\mathbb{P}(|X - \mu| \geq k\sigma) \leq k^{-2}. \quad (4.5.3)$$

Proof.

$$\begin{aligned} \mathbb{P}(|X - \mu| \geq k\sigma) &= \mathbb{P}\left(\frac{|X - \mu|}{k\sigma} \geq 1\right) \stackrel{(4.5.2)}{=} \mathbb{P}\left(\frac{|X - \mu|^2}{k^2\sigma^2} \geq 1\right) \\ &\stackrel{(4.5.1)}{\leq} \mathbb{E}\left(\frac{(X - \mu)^2}{k^2\sigma^2}\right) = \frac{\mathbb{E}(X - \mu)^2}{k^2\sigma^2} = k^{-2}. \end{aligned}$$

□

We can now apply (4.5.3) to our sample mean estimator in order to obtain a t -dependent bound on the probability that the sample mean is more than ϵ -far away from the actual mean.

EXAMPLE 25 (Application to sample mean). It is easy to show that the sample mean has expectation μ and variance σ_x^2/t , where $\sigma_x^2 = \mathbb{V} x$. Consequently:

$$\mathbb{P}(|\bar{x}_t - \mu| \geq k\sigma_x/\sqrt{t}) \leq k^{-2}.$$

Setting $\epsilon = k\sigma_x/\sqrt{t}$ we get $k = \epsilon\sqrt{t}/\sigma_x$ and hence

$$\mathbb{P}(|\bar{x}_t - \mu| \geq \epsilon) \leq \frac{\sigma_x^2}{\epsilon^2 t}.$$

4.5.1 Chernoff-Hoeffding bounds

The previous inequality can be quite loose. In fact, one can prove tighter bounds for the estimation of an expected value. All these bounds rest upon a different application of the Markov inequality, due to Chernoff.

Main idea of Chernoff bounds.

Let $S_t = \sum_{k=1}^t X_k$, with $X_k \sim P$ independently, i.e. $X^t \sim P^t$. By definition, from Markov's inequality we obtain in turn, for any $\theta > 0$

$$\mathbb{P}(S_t \geq u) = \mathbb{P}(e^{\theta S_t} \geq e^{\theta u}) \leq e^{-\theta u} \mathbb{E} e^{\theta S_t} = e^{-\theta u} \prod_k \mathbb{E} e^{\theta X_k}, \quad \text{for } x \in [a, b]. \quad (4.5.4)$$

Theorem 4.5.3. *Hoeffding inequality (Hoeffding [1963], Theorem 2)* Let $x_k \sim P_k$ with $x_k \in [a_k, b_k]$ with $\mathbb{E} X_k = \mu_k$. Then

$$\mathbb{P}(\bar{x}_t - \mu \geq \epsilon) \leq \exp\left(-\frac{2t^2\epsilon^2}{\sum_{k=1}^t (b_k - a_k)^2}\right), \quad (4.5.5)$$

where $\bar{x}_t = \frac{1}{t} \sum_{k=1}^t x_k$ and $\mu = \frac{1}{t} \sum_{k=1}^t \mu_k$.

Proof. Use (4.5.4), setting $X_k = x_k - \mu_k$ so that $S_t = t(\bar{x}_t - \mu)$ and $u = t\epsilon$. Then:

$$\mathbb{P}(\bar{x}_t - \mu \geq \epsilon) = \mathbb{P}(S_t \geq u) \leq e^{-\theta u} \prod_{k=1}^t \mathbb{E} e^{\theta X_k} = e^{-\theta t\epsilon} \prod_{k=1}^t \mathbb{E} e^{\theta(x_k - \mu)}. \quad (4.5.6)$$

Applying Jensen's inequality directly to the expectation does not help. However, we can use convexity in another way. Let $f(x)$ be the linear upper bound on $e^{\theta x}$ on the interval $[a, b]$, i.e.

$$f(x) = \frac{b-x}{b-a}e^{\theta a} + \frac{x-a}{b-a}e^{\theta b} \geq e^{\theta x}.$$

Then obviously $\mathbb{E} e^{\theta x} \leq \mathbb{E} f(x)$ for $x \in [a, b]$. Applying this to the expectation term (4.5.6) above we get,

$$e^{\theta(x_k - \mu_k)} \leq \frac{e^{-\theta\mu_k}}{b_k - a_k} \{(b_k - \mu_k)e^{\theta a_k} + (\mu_k - a_k)e^{\theta b_k}\}.$$

Taking derivatives and computing the Taylor expansion, we get

$$\begin{aligned} \mathbb{E} e^{\theta(x_k - \mu_k)} &\leq e^{\frac{1}{8}\theta^2(b_k - a_k)^2} \\ \mathbb{P}(\bar{x}_t - \mu \geq \epsilon) &\leq e^{-\theta t\epsilon + \frac{1}{8}\theta^2 \sum_{k=1}^t (b_k - a_k)^2}. \end{aligned}$$

This is minimised for $\theta = 4t\epsilon / \sum_{k=1}^t (b_k - a_k)^2$ and we obtain the required result. \square

We can apply this inequality directly to the sample mean example, for $x_k \in [0, 1]$, to obtain

$$\mathbb{P}(|\bar{x}_t - \mu| \geq \epsilon) \leq 2e^{-2t\epsilon^2}.$$

4.6 Approximate Bayesian approaches

Unfortunately, being able to exactly calculate posterior distributions is only possible in special cases. In this section, we give a brief overview of some classic methods for approximate Bayesian inference. The first, Monte-Carlo methods, rely on stochastic approximations of the posterior distributions, where at least the likelihood function is computable. The second, approximate Bayesian computation, extends Monte Carlo methods to the case where the probability function is incomputable or not available at all. In the third which includes, variational Bayes methods, we replace distributions with an analytic approximation. Finally, in empirical Bayes methods, some parameters are replaced by an empirical estimate.

4.6.1 Monte-Carlo inference

Monte-Carlo inference has been a cornerstone of approximate Bayesian statistics ever since computing power was sufficient for such methods to become practical. Let us begin with a simple example, that of estimating expectations.

Estimating expectations.

Let $f : \mathcal{S} \rightarrow [0, 1]$ and P a measure on \mathcal{S} . Then

$$\mathbb{E}_P f = \int_{\mathcal{S}} f(x) dP(x). \quad (4.6.1)$$

Estimating expectations is relatively easy, as long as we can generate samples from P . Then, we can bound our error in estimating its expectation by using the Hoeffding bound.

Corollary 4.6.1. *Let $\hat{f}_n = \frac{1}{n} \sum_t f(x_t)$ with $x_t \sim P$ and $f : \mathcal{S} \rightarrow [0, 1]$. Then:*

$$P \left(\left\{ x^n \in S^n \mid |\hat{f}_n - \mathbb{E} f| \geq \epsilon \right\} \right) \leq 2e^{-2n\epsilon^2}. \quad (4.6.2)$$

This technique is simple and fast. However, we frequently cannot sample from P , but only from some alternative distribution Q . Then it is hard to bound our error.

Another interesting application of this technique is the calculation of posterior distributions.

EXAMPLE 26 (Calculation of posterior distributions). Assume a probability family $\mathcal{P} = \{P_\omega \mid \omega \in \Omega\}$ and a prior distribution ξ on Ω such that we can draw $\omega \sim \xi$. The posterior distribution can be written according to (4.1.1). The nominator can be written as

$$\int_B P_\omega(x) d\xi(\omega) = \int_\Omega \mathbb{I}\{\omega \in B\} P_\omega(x) d\xi(\omega) = \mathbb{E}_\xi [\mathbb{I}\{\omega \in B\} P_\omega(x)]. \quad (4.6.3)$$

Similarly, the denominator can be written as $\mathbb{E}_\xi[P_\omega(x)]$. If P_ω is bounded, then the errors can be bounded too.

An extension of this approach involves Markov chain Monte-Carlo (MCMC) methods. These are sequential sampling procedures, where data is sampled iteratively. At the k -th iteration, we obtain a sample $x^{(k)} \sim Q_k$, where Q_k depends on the previous sample drawn, $x^{(k-1)}$. Although under mild conditions $Q_k \rightarrow P$, there is no easy way to determine *a priori* when the procedure has converged. For more details see for example [Casella et al., 1999].

4.6.2 Approximate Bayesian Computation

The main problem we wish to solve in approximate Bayesian computation (ABC) is how to weigh the evidence we have for or against different models. The assumption is that we have a family of models $\{M_\omega \mid \omega \in \Omega\}$, from which we can generate data. However, there is no easy way to calculate the probability of any model having generated the data. On the other hand, like in the standard

Bayesian setting, we start with a prior ξ over Ω , and given some data $x \in \mathcal{W}$ we wish to calculate the posterior $\xi(\omega | x)$. ABC methods generally rely on what is called an *approximate statistic*, in order to weigh the relative likelihood of models for the data.

An approximate statistic $\phi : \mathcal{X} \rightarrow \mathcal{S}$ maps the data to some lower dimensional space. Then it is possible to compare different data points in terms of how similar their statistics are. For this, we also define some distance $D : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}_+$.

ABC methods are useful in two specific situations. The first is when the family of models that we consider has an intractable likelihood. This means that calculating $M_\omega(x)$ is prohibitively expensive. The second is in some applications which admit a class of *parametrised simulators*, which have no probabilistic description. Then, one reasonable approach is to find the best simulator in the class, and then apply it to the actual problem.

The simplest algorithm in this context is rejection sampling (Alg 1). Here, we repeatedly sample a model from the prior distribution, and then generate data \hat{x} from the model. If the sampled data is ϵ -close to the original data in terms of the statistic, we accept the sample as an approximate posterior sample.

For an overview of ABC methods see [Csilléry et al., 2010, Marin et al., 2011]. Early ABC methods were developed for applications, such as econometric modelling [e.g. Geweke, 1999], where detailed simulators were available, but no useful analytical probabilistic models. ABC methods have also been used for inference in dynamical systems [e.g Toni et al., 2009], the reinforcement learning problem [Dimitrakakis and Tziortziotis, 2013, 2014].

Algorithm 1 ABC Rejection Sampling from $\xi(\omega | x)$.

- 1: **input** prior ξ , data x , generative model family $\{M_\omega | \omega \in \Omega\}$, statistic ϕ , error bound ϵ .
 - 2: **repeat**
 - 3: $\hat{\omega} \sim \xi$
 - 4: $\hat{x} \sim M_{\hat{\omega}}$.
 - 5: **until** $D[\phi(x), \phi(\hat{x})] \leq \epsilon$
 - 6: Return $\hat{\omega}$.
-

4.6.3 Analytic approximations of the posterior.

Another type of approximation involves substituting complex distributions with members from a simpler family. For example, one could replace a multimodal posterior distribution $\xi(\omega | x)$ with a Gaussian. However, a more principled approximation would involve selecting a distribution that is the closest with respect to some divergence or distance, in this case the KL divergence. In particular, we would like to approximate the target distribution $\xi(\omega | x)$ with some other distribution $Q_\theta(\omega)$ in a family $\{Q_\theta | \theta \in \Theta\}$. While a number of distances such as the total variation or Wasserstein distance. However, the most popular algorithms employ the KL divergence

$$D(Q \parallel P) \triangleq \int_{\Omega} \ln \frac{dQ}{dP} dQ, \quad (4.6.4)$$

where one term is the target posterior distribution and the other the approximation. As the KL divergence is asymmetric, its use results in two distinct

approximation methods: variational Bayes and expectation propagation.

Variational approximation. In this formulation, we wish to minimise the KL divergence

$$D(Q_\theta \parallel \xi_{|x}) = \int_{\Omega} \ln \frac{dQ_\theta}{d\xi_{|x}} dQ_\theta, \quad (4.6.5)$$

where $\xi_{|x}$ is shorthand for the distribution $\xi(\omega | x)$. An efficient method for minimising this divergence is rewriting it as follows:

$$\begin{aligned} D(Q_\theta \parallel \xi_{|x}) &= - \int_{\Omega} \ln \frac{d\xi_{|x}}{dQ_\theta} dQ_\theta \\ &= - \int_{\Omega} \ln \frac{d\xi_x}{dQ_\theta} dQ_\theta + \ln \xi(x), \end{aligned}$$

where ξ_x is shorthand for the joint distribution $\xi(\omega, x)$ for a fixed value of x . As the latter term does not depend on θ , we can find the best element of the family by the following optimisation:

$$\max_{\theta \in \Theta} \int_{\Omega} \ln \frac{d\xi_x}{dQ_\theta} dQ_\theta, \quad (4.6.6)$$

where the term we are maximising can also be seen as a lower bound on the marginal log likelihood.

Expectation propagation. The other direction requires us to minimise the divergence

$$D(P \parallel Q) = \int_{\Omega} \ln \frac{dP}{dQ} dP.$$

An algorithm for achieving this in the case of data terms that are independent given the parameter is expectation propagation [Minka, 2001a]. There, the approximation has a factored form and is iteratively updated, with each term minimising the KL divergence while keeping the remaining terms fixed.

4.6.4 Maximum Likelihood and Empirical Bayes methods.

When a full posterior distribution is not necessary, some parameter may be estimated point-wise. One simple such approach is maximum likelihood. In the simplest case, we replace the posterior distribution $\xi(\theta | x)$ with a point estimate corresponding to the parameter value that maximises the likelihood:

$$\theta_{\text{ML}}^* \in \arg \max_{\theta} P_\theta(x). \quad (4.6.7)$$

Alternatively, the *maximum a posteriori* parameter maybe obtained:

$$\theta_{\text{MAP}}^* \in \arg \max_{\theta} \xi(\theta | x). \quad (4.6.8)$$

In the latter case, even though we cannot compute the full function $\xi(\theta | x)$, we can still maximise (perhaps locally) for θ .

More generally, there might be some parameters ϕ for which we *can* compute a posterior distribution. Then we can still use the same approaches, maximising one of:

$$P_\theta(x) = \int P_{\theta,\phi}(x) d\xi(\phi | x) \quad (4.6.9)$$

$$\xi(\theta | x) = \int_{\bar{\Phi}} \xi(\theta | \phi, x) d\xi(\phi | x) \quad (4.6.10)$$

Empirical Bayes methods, pioneered by Robbins [1955], some parameters are replaced by an empirical estimate, not necessary corresponding to the maximum likelihood. These methods are quite diverse Laird and Louis [1987], Lwin and Maritz [1989], Robbins [1964, 1955], Deely and Lindley [1981] and unfortunately beyond the scope of this book.

Chapter 5

Sequential sampling

5.1 Gains from sequential sampling

So far, we have mainly considered decision problems where the sample size was fixed. However, frequently the sample size can also be part of the decision. Since normally larger sample sizes give us more information, in this case the decision problem is only interesting when obtaining new samples has a cost. Consider the following example.

EXAMPLE 27. Consider that you have 100 produced items and you want to determine whether there are fewer than 10 faulty items among them. If testing has some cost, it pays off to think about whether it is possible to do without testing all 100 items. Indeed, this is possible by the following simple online testing scheme: You test one item after another until you either have discovered 10 faulty items or 91 good items. In either case you have the correct answer at considerably lower cost than when testing all items.

A sequential sample from some unknown distribution P is generated as follows. First, let us fix notation and assume that each new sample x_i we obtain belongs in some alphabet \mathcal{X} , so that at time t , we have observed $x_1, \dots, x_t \in \mathcal{X}^t$. It is also convenient to define the set of all sequences in the alphabet \mathcal{X} as $\mathcal{X}^* \triangleq \bigcup_{t=0}^{\infty} \mathcal{X}^t$. The distribution P defines a probability on \mathcal{X}^* so that x_{t+1} may depend on the previous samples x_1, \dots, x_t in an arbitrary manner. At any time t , we can either *stop sampling* or obtain one *more* observation x_{t+1} . A sample obtained in this way is called a *sequential sample*. More formally, we give the following definition:

stopping function

Definition 5.1.1 (Sequential sampling). A sequential sampling procedure on a probability space¹ $(\mathcal{X}^*, \mathfrak{B}(\mathcal{X}^*), P)$ involves a *stopping function* $\pi_s : \mathcal{X}^* \rightarrow \{0, 1\}$, such that we stop sampling at time t if and only if $\pi_s(x^t) = 1$, otherwise we obtain a new sample $x_{t+1} \mid x^t \sim P(\cdot \mid x^t)$.

Thus, the sample obtained depends both on P and the sampling procedure π_s . In our setting, we don't just want to sample sequentially, but also to take some action after sampling is complete. For that reason, we can generalise the above definition to sequential decision procedures.

Definition 5.1.2 (Sequential decision procedure). A sequential decision procedure $\pi = (\pi_s, \pi_d)$ is tuple composed of

1. a stopping rule $\pi_s : \mathcal{X}^* \rightarrow \{0, 1\}$ and
2. a decision rule $\pi_d : \mathcal{X}^* \rightarrow \mathcal{A}$.

The stopping rule π_s specifies whether, at any given time, we should stop and make a decision in \mathcal{A} or take one more sample. That is, stop if

$$\pi_s(x^t) = 1,$$

otherwise observe x_{t+1} . Once we have stopped (i.e. $\pi_s(x^t) = 1$), we choose the decision

$$\pi_d(x^t).$$

¹This is simply a sample space and associated algebra, together with a probability measure. See Appendix B for a complete definition.

Deterministic stopping rules If the stopping rule π_s is deterministic, then for any t , there exists some *stopping set* $B_t \subset \mathcal{X}^t$ such that

$$\pi_s(x^t) = \begin{cases} 1, & \text{if } x^t \in B_t \\ 0, & \text{if } x^t \notin B_t. \end{cases} \quad (5.1.1)$$

As with any Bayesian decision problem, it is sufficient to consider only deterministic decision rules.

We are interested in sequential sampling problems especially when there is a reason for us to stop sampling early enough, like the case when we incur a cost with each sample we take. A detailed example is given in the following section.

5.1.1 An example: sampling with costs

We once more consider problems where we have some observations x_1, x_2, \dots , with $x_t \in \mathcal{X}$, which are drawn from some distribution with parameter $\theta \in \Theta$, or more precisely from a family $\mathcal{P} = \{P_\theta \mid \theta \in \Theta\}$, such that each $(\mathcal{X}, \mathcal{B}(\mathcal{X}), P_\theta)$ is a probability space for all $\theta \in \Theta$. Since we take repeated observations, the probability of a sequence $x^n = x_1, \dots, x_n$ under an i.i.d. model θ is $P_\theta^n(x^n)$. We have a prior probability measure ξ on $\mathcal{B}(\Theta)$ for the unknown parameter, and we wish to take an action $a \in \mathcal{A}$ that maximises the expected utility according to a utility function $u : \Theta \times \mathcal{A} \rightarrow \mathbb{R}$.

In the classical case, we obtain a complete sample of fixed size n , $x^n = (x_1, \dots, x_n)$ and calculate a posterior measure $\xi(\cdot \mid x^n)$. We then take the decision maximising the expected utility according to our posterior. Now consider the case of sampling with costs, such that a sample of size n results in a cost of cn . For that reason we define a new utility function U which depends on the number of observations we have.

Samples with costs

$$U(\theta, a, x^n) = u(\theta, a) - cn, \quad (5.1.2)$$

$$\mathbb{E}_\xi(U \mid a, x^n) = \int_{\Theta} u(\theta, a) d\xi(\theta \mid x^n) - cn. \quad (5.1.3)$$

In the remainder of this section, we shall consider the following simple decision problem, where we need to make a decision the value of an unknown parameter. As we get more data, we have a better chance of discovering the right parameter. However, there is always a small chance of getting no information.

EXAMPLE 28. Consider the following decision problem, where the goal is to distinguish between two possible hypotheses θ_1, θ_2 , with corresponding decisions a_1, a_2 . We have three possible observations $\{1, 2, 3\}$, with 1, 2 being more likely under the first and second hypothesis, respectively. However, the third observation gives us no information about the hypothesis, as its probability is the same under θ_1 and θ_2 . In this problem γ is the probability that we obtain an uninformative sample.

- Parameters: $\Theta = \{\theta_1, \theta_2\}$.
- Decisions: $\mathcal{A} = \{a_1, a_2\}$.
- Observation distribution $f_i(k) = \mathbb{P}_{\theta_i}(x_t = k)$ for all t with

$$f_1(1) = 1 - \gamma, \quad f_1(2) = 0, \quad f_1(3) = \gamma, \quad (5.1.4)$$

$$f_2(1) = 0, \quad f_2(2) = 1 - \gamma, \quad f_2(3) = \gamma. \quad (5.1.5)$$

- Local utility: $u(\theta_i, a_j) = 0$, for $i = j$ and $b < 0$ otherwise.
- Prior: $P_\xi(\theta = \theta_1) = \xi = 1 - P_\xi(\theta = \theta_2)$.
- Observation cost per example: c .

At any step t , you have the option of continuing for one more step, or stopping and taking an action in \mathcal{A} . The question is what is the policy for sampling and selecting an action that maximises expected utility?

In this problem, it is immediately possible to distinguish θ_1 from θ_2 when you observe $x_t = 1$ or $x_t = 2$. However, the values $x_t = 3$ provide no information. Hence, the utility of stopping only depends on. So, the expected utility of stopping if you have only observed 3s after t steps is $\xi b - ct$. In fact, if your posterior parameter after t steps is ξ_t , then the expected utility of stopping is $b \min\{\xi_t, 1 - \xi_t\} - ct$. In general, you should expect ξ_t to approach 0 or 1 with high probability, and hence taking more samples is better. However, if we pay utility $-c$ for each additional sample, there is a point of diminishing returns, after which it will not be worthwhile to take any more samples.

value

We first investigate the setting where the number of observations is fixed. In particular, the *value* of the optimal procedure taking n observation is defined to be the expected utility that maximises the *a posteriori* utility given x^n , i.e.

$$V(n) = \sum_{x^n} P_\xi^n(x^n) \max_a \mathbb{E}_\xi(U | x^n, a),$$

where $P_\xi^n = \int_\Theta P_\theta^n d\xi(\theta)$ is the marginal distribution over n observations. For this specific example, it is easy to calculate the value of the procedure that takes n observations, by noting the following facts.

- The probability of observing $x_t = 3$ for all $t = 1, \dots, n$ is γ^n . Then we must rely on our prior ξ to make a decision.
- If we observe any other sequence, we know the value of θ .

Consequently, the total value $V(n)$ of the optimal procedure taking n observations is

$$V(n) = \xi b \gamma^n - cn. \quad (5.1.6)$$

Based on this, we now want to find the optimal number of samples n . Since V is a smooth function, an approximate maximiser can be found by viewing n as a continuous variable.² Taking derivatives, we get

$$n^* = \left[\log \frac{c}{\xi b \log \gamma} \right] \frac{1}{\log \gamma} \quad (5.1.7)$$

$$V(n^*) = \frac{c}{\log \gamma} \left[1 + \log \frac{c}{\xi b \log \gamma} \right] \quad (5.1.8)$$

²In the end, we can find the optimal maximiser by looking at the nearest two integers to the value found.

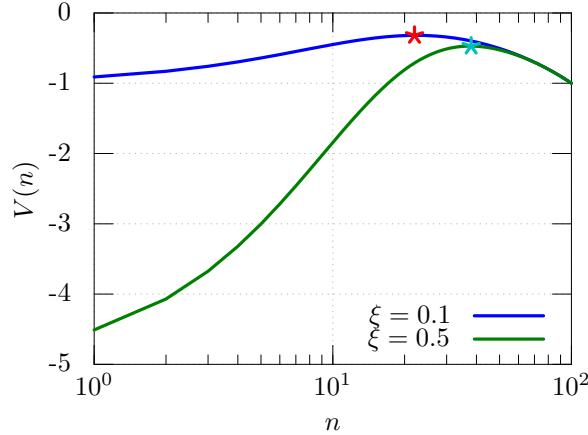


Figure 5.1: Illustration of P1, the procedure taking a fixed number of samples n . The value of taking exactly n observations under two different beliefs, for $\gamma = 0.9$, $b = -10$, $c = 10^{-2}$.

The results of applying this procedure are illustrated in Figure 5.1. Here we can see that, for two different choices of priors, the optimal number of samples is different. In both cases, there is a clear choice for how many samples to take, when we must fix the number of samples before seeing any data.

However, we may *not* be constrained to fix the number of samples *a priori*. As illustrated in Example 27, many times it is a good idea to adaptively decide when to stop taking samples. This is illustrated by the following *sequential* procedure. In this one, since we already know that there is an optimal *a priori* number of steps n^* , we can choose to look at all possible stopping times for that are smaller or equal to n^* .

P2. A sequential procedure stopping after at most n^* steps.

- If $t < n^*$, use the stopping rule $\pi_s(x^t) = 1$. iff $x_t = 3$.
- In other words, stop as soon as you observe a 3, or until you reach $t = n^*$.
- Our posterior after stopping is, just $\xi(\theta | x^n)$, where both x^n and the number of observations n are random.

Since the probability of $x_t = 3$ is always the same for both θ_1 and θ_2 , we have:

$$\mathbb{E}_\xi(n) = \mathbb{E}(n | \theta = \theta_1) = \mathbb{E}(n | \theta = \theta_2) < n^*$$

We can calculate the expected number of steps as follows:

$$\mathbb{E}_\xi(n \mid n \leq n^*) = \mathbb{E}_\xi(n \mid \theta = \theta_1) = \sum_{t=1}^{n^*} t \mathbb{P}_\xi(n = t \mid \theta = \theta_1) \quad (5.1.9)$$

$$= \sum_{t=1}^{n^*-1} t \gamma^{t-1} (1 - \gamma) + n^* \gamma^{n^*-1} = \frac{1 - \gamma^{n^*}}{1 - \gamma}, \quad (5.1.10)$$

from the *geometric series* (see equation C.1.4). Consequently, the value of this procedure is

$$\begin{aligned} \bar{V}(n^*) &= \mathbb{E}_\xi(U \mid n = n^*) \mathbb{P}_\xi(n = n^*) + \mathbb{E}_\xi(U \mid n < n^*) \mathbb{P}_\xi(n < n^*) \\ &= \xi b \gamma^{n^*} - c \mathbb{E}_\xi(n) \end{aligned}$$

and from the definition of n^* :

$$\bar{V}(n^*) = \frac{c}{\gamma - 1} + \frac{c}{\log \gamma} \left[1 + \frac{c}{\xi b(1 - \gamma)} \right]. \quad (5.1.11)$$

unbounded procedures

As you can see, there is a non-zero probability that $n = n^*$, at which time we will have not resolved the true value of θ . In that case, we are still not better off than at the very beginning of the procedure, when we had no observations. If our utility is linear with the number of steps, it thus makes sense that we should still continue. For that reason, we should consider *unbounded procedures*.

The unbounded procedure for our example is simply this to use the stopping rule $\pi_s(x^t) = 1$ iff $x_t \neq 3$. Since we only obtain information whenever $x_t \neq 3$, and that information is enough to fully decide θ , once we observe $x_t \neq 3$, we can make a decision that has value 0, as we can guess correctly. So, the value of the unbounded sequential procedure is just $V^* = -c \mathbb{E}_\xi(n)$.

$$\mathbb{E}_\xi(n) = \sum_{t=1}^{\infty} t \mathbb{P}_\xi(n = t) = \sum_{t=1}^{\infty} t \gamma^{t-1} (1 - \gamma) = \frac{1}{1 - \gamma}, \quad (5.1.12)$$

again using the formula for the geometric series.

In the given example, it is clear that bounded procedures are (in expectation) better than fixed-sampling procedures, as seen in Figure 5.2. In turn, the unbounded procedure is (in expectation) better than the bounded procedure. Of course, an unbounded procedure may end up costing much more than taking a decision without observing any data, as it disregards the amount spent to time t . This relates to the economic idea of *sunk costs*: since our utility is additive in terms of the cost, our optimal decision now should not be dependent on previously accrued costs.

5.2 Optimal sequential sampling procedures

We now turn our attention to the general case. While it is easy to define the optimal stopping rule and decision in this simple example, how can actually do the same thing for *arbitrary* problems? The following section characterises optimal sequential sampling procedures and gives an algorithm for constructing them.

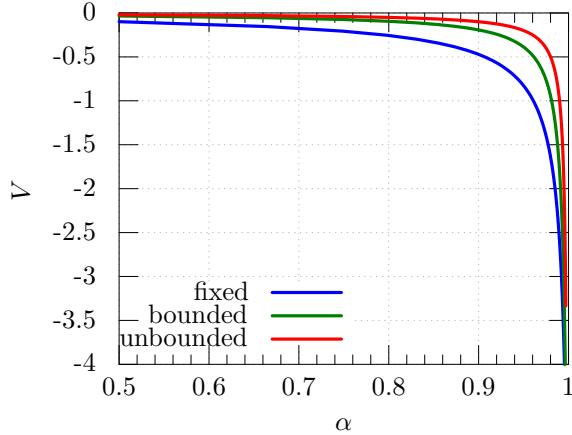


Figure 5.2: The value of three strategies for $\xi = 1/2$, $b = -10$, $c = 10^{-2}$ and varying γ . Higher values of γ imply a longer time before the true θ is known.

Once more, consider a distribution family $\mathcal{P} = \{P_\theta \mid \theta \in \Theta\}$ and a prior ξ over $\mathfrak{B}(\Theta)$. For a decision set \mathcal{A} , a utility function $U : \Theta \times \mathcal{A} \rightarrow \mathbb{R}$, and a sampling cost c , the utility of a sequential decision procedure is the local utility at the end of the procedure, minus the sampling cost. In expectation, this can be written as

$$U(\xi, \pi) = \mathbb{E}_\xi \{ u[\theta, \pi(x^n)] - nc. \} \quad (5.2.1)$$

Here the cost is inside the expectation, since the number of samples we take is random. Summing over all the possible stopping times n , and taking $B_n \subset \mathcal{X}^*$ as the set of observations for which we stop, we have:

$$U(\xi, \pi) = \sum_{n=1}^{\infty} \int_{B_n} \mathbb{E}_\xi [U(\theta, \pi(x^n)) \mid x^n] dP_\xi(x^n) - \sum_{n=1}^{\infty} P_\xi(B_n)nc \quad (5.2.2)$$

$$\sum_{n=1}^{\infty} \int_{B_n} \left\{ \int_{\Theta} U[\theta, \pi(x^n)] d\xi(\theta \mid x^n) \right\} dP_\xi(x^n) - \sum_{n=1}^{\infty} P_\xi(B_n)nc \quad (5.2.3)$$

where P_ξ is the marginal distribution under ξ . Although it may seem difficult to evaluate this, it can be done by a simple dynamic programming technique called *backwards induction*. We first give the algorithm for the case of bounded procedures (i.e. procedures that must stop after a particular time) and later for unbounded ones.

Definition 5.2.1 (Bounded sequential decision procedure). A sequential decision procedure δ is *bounded* if there is a positive integer T such that $\mathbb{P}_\xi(n \leq T) = 1$. Similarly, the procedure is T -bounded if it is bounded for a specific T .

We can analyse such a procedure by recursively analysing procedures of larger T , starting from the final point of the process and working our way backwards. Consider a π that is T -bounded. Then we know that we shall take at most T samples. If the process ends at stage T , we will have observed some

sequence x^T , which gives rise to a posterior $\xi(\theta | x^T)$. Since we *must* stop at T , we must choose a maximising expected utility at that stage:

$$\mathbb{E}_\xi[U | x^T, a] = \int_{\Theta} U(\theta, a) d\xi(\theta | x^T)$$

Since need not take another sample, the respective value (maximal expected utility) of that stage is:

$$V^0[\xi(\cdot | x^T)] \triangleq \max_{a \in \mathcal{A}} U(\xi(\cdot | x^T), a)$$

where we introduce the notation V^n to denote the expected utility, given that we are stopping after at most n steps.

More generally, we need to consider the effect on subsequent decisions. Consider the following simple two-stage problem as an example. Let $\mathcal{X} = \{0, 1\}$ and the prior ξ on the θ parameter of $Bern(\theta)$. We wish to either decide immediately on a parameter θ , or take one more observation, at cost c , before deciding. The problem we consider has two stages, as illustrated in Figure 5.3. In this exam-

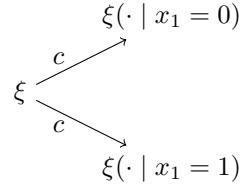


Figure 5.3: An example of a sequential decision problem with two stages. The initial belief is ξ and there are two possible subsequent beliefs, depending on whether we observe $x_t = 0$ or $x_t = 1$. At each stage we pay c .

ple, we begin with a prior ξ at the first stage. There are two possible outcomes for the **second stage**.

1. If we observe $x_1 = 0$ then our value is $V^0[\xi(\cdot | x_1 = 0)]$.
2. If we observe $x_1 = 1$ then our value is $V^0[\xi(\cdot | x_1 = 1)]$.

At the first stage, we can:

1. Stop with value $V^0(\xi)$.
2. Pay a sampling cost c for value: $V^0[\xi(\cdot | x_1)]$, with $P_\xi(x_1) = \int_{\Theta} P_\theta(x_1) d\xi(w)$.

So the expected value of continuing for one more step is

$$V^1(\xi) \triangleq \int_{\mathcal{X}} V^0[\xi(\cdot | x_1)] dP_\xi(x_1).$$

Thus, the overall value for this problem is:

$$\max \left\{ V^0(\xi), \sum_{x_1=0}^1 V^0[\xi(\cdot | x_1)] P_\xi(x_1) - c \right\}$$

The above is simply the maximum of the value of stopping immediately (V^0), and the value of continuing for at most one more step (V^1). This procedure can be applied recursively for multi-stage problems, as explained below.

5.2.1 Multi-stage problems

For simplicity, we use ξ_n to denote a posterior $\xi(\cdot \mid x^n)$, omitting the specific value of x^n . For any specific ξ_n , there is a range of given possible next beliefs ξ_{n+1} , depending on what the value of the next observation x_{n+1} is. This is illustrated in Figure 5.4. The value of the process can be calculated as follows, more

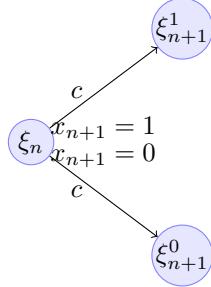


Figure 5.4: A partial view of the multi-stage process.

generally:

$$\begin{aligned} V^0(\xi_t) &= \sup_{a \in \mathcal{A}} \int_{\Theta} u(\theta, a) d\xi_t(\theta) && \text{(Immediate value)} \\ \xi_n(\cdot) &\triangleq \xi(\cdot \mid x^n) && \text{(posterior)} \\ \mathbb{E}_{\xi_n} V^0[\xi_{n+1}] &= \int_{\mathcal{X}} V^0[\xi_n(\cdot \mid x_n)] d\xi_n(x_n) && \text{(Next-step value)} \\ V^1(\xi_n) &= \max \{V^0(\xi_n), \mathbb{E}_{\xi_n} V^0(\xi_{n+1}) - c\} && \text{(Optimal value)} \end{aligned}$$

The immediate value is the expected value if we stop immediately at time t . The next-step value is the expected value of the next step, ignoring the cost. Finally, the optimal value at the n -th step is just the maximum between the value of stopping immediately plus the next-step value. We can generalise this procedure over all states $1, 2, \dots, T$, to obtain a general procedure.

5.2.2 Backwards induction for bounded procedures

The main idea expressed in the previous section is to start from the last stage of our decision problem, where the utility is known, and then move backwards. At each stage, we know the probability of reaching different points in the next stage, as well as their values. Consequently, we can compute the value of any point in the current stage as well. This notion is formalised below, via the algorithm of backwards induction.

Theorem 5.2.1 (Backwards induction). *The utility of a k -bounded optimal procedure with prior ξ_0 is $V^k(\xi_0)$ and is given by the recursion:*

$$V^{j+1}(\xi_n) = \max \{V^0(\xi_n), \mathbb{E}_{\xi_n} V^j(\xi_{n+1}) - c\}, \quad (5.2.4)$$

for every belief ξ_n in the set of beliefs that arise from the prior ξ_0 , with $j = k-n$.

The proof of this theorem follows by induction. However, we shall prove a more general version in Chapter 6. Equation 5.2.4 essentially gives a recursive calculation of the value of the k -bounded optimal procedure. To evaluate it, we first need to calculate all possible beliefs ξ_1, \dots, ξ_k . For each belief ξ_k , we calculate $V^0(\xi_k)$. We then move backwards, and calculate $V^0(\xi_{k-1})$ and $V^1(\xi_{k-1})$. Proceeding backwards, for $n = k-1, k-2, \dots, 1$, we calculate $V^{k+1}(\xi_n)$ for all beliefs ξ_n with $j = k-n$. The value of the procedure also determines the optimal sampling strategy, as shown by the following theorem.

Theorem 5.2.2. *The optimal k -bounded procedure stops at time t if the value of stopping right now is better than that of continuing, i.e. if*

$$V^0(\xi_t) \geq V^{k-t}(\xi_t)$$

and chooses a maximising $\mathbb{E}_{\xi_t} U(\theta, a)$, otherwise takes one more sample.

Finally, longer procedures (i.e. procedures that allow for stopping later) are always better than shortest ones. Whenever we wish to make a as shown by the following theorem.

Theorem 5.2.3. *For any probability measure ξ on Θ ,*

$$V^n(\xi) \leq V^{n+1}(\xi). \quad (5.2.5)$$

That is, the procedure that stops after at most n steps is never better than the procedure that stops after at most $n+1$ time steps. To obtain an intuition of why this is the case, consider the example of Section 5.1.1. In that example, if we have a sequence of 3s, then we obtain no information. Consequently, when we compare the value of a plan taking at most n samples with that of a plan taking at most $n+1$ samples, we see that the latter plan is better for the event where we obtain n 3s, but has the same value for all other events.

5.2.3 Unbounded sequential decision procedures

Given the monotonicity of the value of bounded procedures (5.2.5), one may well ask what is the value of unbounded procedures, i.e. procedures that may never stop sampling. The value of an unbounded procedure procedure is

$$U(\xi, \pi) = \int_{\mathcal{X}^*} \{V^0[\xi(\cdot | x^n)] - cn\} dP_\xi^\pi(x^n) = \mathbb{E}_\xi^\pi \{V^0[\xi(\cdot | x^n)] - cn\}, \quad (5.2.6)$$

where $P_\xi^\pi(x^n)$ is the probability that we observe samples x^n and stop under the marginal distribution defined by ξ and π , while n is the random number of samples taken by π . As before, this is random because the observations x are random; π itself can be deterministic.

Definition 5.2.2 (Regular procedure). Let $B_{>k}(\pi) \subset \mathcal{X}^*$ be the set of sequences such that π takes more than k samples. Then π is regular if $U(\xi, \pi) \geq V^0(\xi)$ and if, for all $x^n \in B_{>n}(\pi)$ and for all $n \in \mathbb{N}$,

$$U[\xi(\cdot | x^n), \pi] > V^0[\xi(\cdot | x^n)] - cn., \quad (5.2.7)$$

i.e. the expected utility given for any sample that starts with x^n where we don't stop, is greater than that of stopping at n .

In other words, if π specifies that at least one observation should be taken, then the value of π is greater than the value of choosing a decision without any observation. Furthermore, whenever π specifies that another observation should be taken, the expected value of continuing must be larger than the value of stopping. If the procedure is *not* regular, then there may be stages where the procedure specifies that sampling should be continued, though the value would be increased by stopping.

Theorem 5.2.4. *If π is not regular, then there exists a regular π' such that $U(\xi, \pi') \geq U(\xi, \pi)$.*

Proof. First, consider the case that π is not regular because $U(\xi, \pi) \leq V^0(\xi)$. Then π' can be the regular procedure which chooses $a \in \mathcal{A}$ without any observations.

Now consider the case that $V(\xi, \pi) > V^0(\xi)$ and that π specifies at least one sample should be taken. We can let π' be the procedure which stops as soon as the observed x^n does not satisfy (5.2.7).

If, for x^n , π stops, then both sides of (5.2.7) are equal. Consequently, π' stops no later than π for any x^n . Finally, let

$$B_k(\pi) = \{x \in \mathcal{X}^* \mid n = k\} \quad (5.2.8)$$

be the set of observations such that exactly k samples are taken by rule π and

$$B_{\leq k}(\pi) = \{x \in \mathcal{X}^* \mid n \leq k\} \quad (5.2.9)$$

be the set of observations such that at most k samples are taken by rule π . Then

$$\begin{aligned} V(\xi, \pi') &= \sum_{k=1}^{\infty} \int_{B_k(\pi')} \{V^0[\xi(\cdot \mid x^k) - ck]\} dP_{\xi}(x^k) \\ &\geq \sum_{k=1}^{\infty} \int_{B_k(\pi')} \mathbb{E}_{\xi}\{V[\xi, \pi \mid x^k]\} dP_{\xi}(x^k) \\ &= \sum_{k=1}^{\infty} \mathbb{E}_{\xi}\{V(\xi, \pi) \mid B_k(\pi')\} P_{\xi}(B_k(\pi')) = E_{\xi}V(\xi, \pi) = V(\xi, \pi). \end{aligned}$$

□

5.2.4 The sequential probability ratio test

Sometimes we wish to collect just enough data in order to be able to confirm or disprove a particular hypothesis. More specifically, we have a set of parameters Θ , and we need to pick the right one. However, rather than simply using an existing set of data, we are collecting data sequentially, and we need to decide when to stop and select a model. In this case, each one of our decisions a_i corresponds to choosing the model θ_i , and we have a utility function that favours our picking the correct model. However, data collection has some cost, which we must balance against the expected utility of picking a parameter.

As an illustration, consider a problem we must decide for one out of two possible parameters θ_1, θ_2 . At each step, we can either take another sample from the unknown $P_{\theta}(x_t)$, or decide for one or the other of the parameters.

EXAMPLE 29. A two-point sequential decision problem.

- Observations $x_t \in \mathcal{X}$
- Distribution family: $\mathcal{P} = \{P_\theta \mid \theta \in \Theta\}$
- Probability space $(\mathcal{X}^*, \mathfrak{B}(\mathcal{X}^*), P_\theta)$.
- Parameter set: $\Theta = \{\theta_1, \theta_2\}$.
- Action set: $\mathcal{A} = \{a_1, a_2\}$.
- Prior $\xi = \mathbb{P}(\theta = \theta_1)$.
- Sampling cost $c > 0$.

U(θ, d)		a_1	a_2
θ_1	0	λ_1	
	λ_2	0	

Table 5.1: The utility function, with $\lambda_1, \lambda_2 < 0$

As will be the case for all our sequential decision problems, we only need to consider our current belief ξ , and its possible evolution, when making a decision. To obtain some intuition about this procedure, we are going to analyse this problem by examining what the optimal decision is under all possible beliefs ξ .

Under some belief ξ , the immediate value (i.e. the value we obtain if we stop immediately), is simply:

$$V^0(\xi) = \max \{\lambda_1 \xi, \lambda_2(1 - \xi)\}. \quad (5.2.10)$$

The worst-case immediate value, i.e. the minimum, is attained when both terms are equal. Consequently, setting $\lambda_1 \xi = \lambda_2(1 - \xi)$, which gives $\xi = \lambda_2 / (\lambda_1 + \lambda_2)$. Intuitively, this is the worst-case belief, as the uncertainty it induces leaves us unable to choose between either hypothesis. Replacing in (5.2.10) gives a lower bound for the value for any belief.

$$V^0(\xi) \geq \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}.$$

Let π' denote the set of procedures π which take at least one observation and define:

$$V'(\xi) = \sup_{\pi \in \pi'} V(\xi, \pi). \quad (5.2.11)$$

Then the ξ -expected utility $V^*(\xi)$ must satisfy:

$$V^*(\xi) = \max \{V^0(\xi), V'(\xi)\}. \quad (5.2.12)$$

As we showed in Section 3.3.1, V' is a convex function of ξ . Now let

$$\Xi_0 \triangleq \{\xi \mid V^0(\xi) \leq V'(\xi)\}, \quad (5.2.13)$$

be the set of priors where it is optimal to terminate sampling. It follows that $\Xi \setminus \Xi_0$, the set of priors where we must not terminate sampling, is a convex set.

Figure 5.5 illustrates the above arguments, by plotting the immediate value against the optimal continuation after taking one more sample. For the worst-case belief, we must always continue sampling. When we are absolutely certain

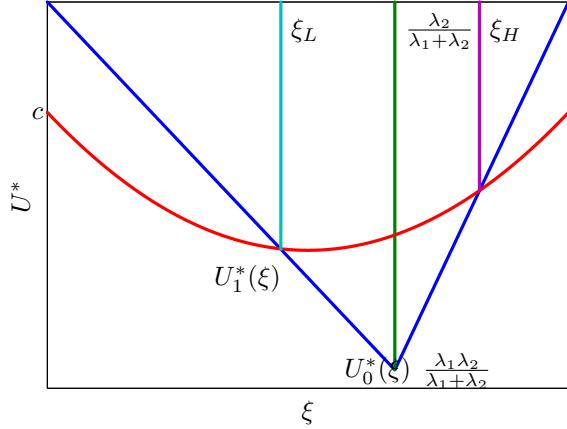


Figure 5.5: The value of the optimal continuation V' versus stopping V^0 .

about the model, then it's always better to stop immediately. There are two points where the curves intersect. Together, these define three subsets of beliefs: On the left, if $\xi < \xi_L$, we decide for one parameters. On the right, if $\xi > \xi_H$, we decide for the other parameter. Otherwise, we continue sampling. This is the main idea of the sequential probability ratio test, explained below.

The sequential probability ratio test (SPRT)

Figure 5.5 offers a graphical illustration of when it is better or take one more sample in this setting. In particular, if $\xi \in (\xi_L, \xi_T)$, then it is optimal to take at least one more sample. Otherwise, it is optimal to make an immediate decision with value $\rho_0(\xi)$.

This has a nice interpretation as a standard tool in statistics: the *sequential probability ratio test*. First note that our posterior at time t can be written as

$$\xi_t = \frac{\xi P_{\theta_1}(x^t)}{\xi P_{\theta_1}(x^t) + (1 - \xi)P_{\theta_2}(x^t)}.$$

Then, for any posterior, the optimal procedure is:

- If $\xi_L < \xi_t < \xi_T$, take one more sample.
- If $x_L \geq \xi_t$, stop and decide a_2 .
- If $x_T \leq \xi_t$, stop and decide a_1 .

We can now restate the optimal procedure can be restated in terms of a probability ratio, i.e. we should always take another observation as long as

$$A < \frac{P_{\theta_2}(x^t)}{P_{\theta_1}(x^t)} < B,$$

where

$$A \triangleq \frac{\xi(1 - \xi_T)}{(1 - \xi)\xi_T}, \quad B \triangleq \frac{\xi(1 - \xi_L)}{(1 - \xi)\xi_L}.$$

If the first inequality is violated, choose a_1 . If the second inequality is violated, choose a_2 . So, there is an equivalence between SPRT and optimal sampling procedures, when the optimal policy is to continue sampling whenever our belief is within a specific interval.

5.2.5 Wald's theorem

An important tool in the analysis of SPRT as well as other procedures that stop at random times is the following theorem by Wald.

Theorem 5.2.5 (Wald's theorem). *Let z_1, z_2, \dots be a sequence of i.i.d. random variables with measure G , such that $\mathbb{E} z_i = m$ for all i . Then for any sequential procedure with $\mathbb{E} n < \infty$:*

$$\mathbb{E} \sum_{i=1}^n z_i = m \mathbb{E} n. \quad (5.2.14)$$

Proof.

$$\begin{aligned} \mathbb{E} \sum_{i=1}^n z_i &= \sum_{k=1}^{\infty} \int_{B_k} \sum_{i=1}^k z_i dG^k(z^k) \\ &= \sum_{k=1}^{\infty} \sum_{i=1}^k \int_{B_k} z_i dG^k(z^k). \\ &= \sum_{i=1}^{\infty} \sum_{k=i}^{\infty} \int_{B_k} z_i dG^k(z^k) \\ &= \sum_{i=1}^{\infty} \int_{B_{\geq i}} z_i dG^i(z^i) \\ &= \sum_{i=1}^{\infty} \mathbb{E}(z_i) \mathbb{P}(n \geq i) = m \mathbb{E} n. \end{aligned}$$

□

We now consider application of this theorem to the SPRT. Let $z_i = \log \frac{P_{\theta_2}(x_i)}{P_{\theta_1}(x_i)}$. Consider the equivalent formulation of the SPRT which uses

$$a < \sum_{i=1}^n z_i < b$$

as the test. Using Wald's theorem and the previous properties and assuming $c \approx 0$, we obtain the following approximately optimal values for a, b :

$$a \approx \log c - \log \frac{I_1 \lambda_2 (1 - \xi)}{\xi} \quad b \approx \log \frac{1}{c} - \log \frac{I_2 \lambda_1 \xi}{1 - \xi}, \quad (5.2.15)$$

where $I_1 = -\mathbb{E}(z \mid \theta = \theta_1)$ and $I_2 = \mathbb{E}(z \mid \theta = \theta_2)$ is the *information*, better known as the *KL divergence*. If the cost c is very small, then the information terms vanish and we can approximate the values by $\log c$ and $\log \frac{1}{c}$.

5.3 Martingales

Martingales are a fundamentally important concept in the analysis of stochastic processes. The main idea of a martingale is that the expectation of a random variable at time $t + 1$ only depends on the value of another variable at time t .

An example of a martingale sequence is when x_t is the amount of money you have at a given time, and where at each time-step t you are making a gamble such that you lose or gain 1 currency unit with equal probability. Then, at any step t , it holds that $\mathbb{E}(x_{t+1} | x_t) = x_t$. This concept can be generalised by considering two random processes, x_t and y_t , which are dependent.

Definition 5.3.1. Let $x^n \in \mathcal{S}^n$ be a sequence of observations $x^n \in \mathcal{S}^n$ with distribution P_n , and $y_n : \mathcal{S}^n \rightarrow \mathbb{R}$ be a random variable. Then the sequence $\{y_n\}$ is a *martingale with respect to $\{x_n\}$* if for all n the expectation

$$\mathbb{E}(y_n) = \int_{\mathcal{S}^n} y_n(x^n) dP_n(x^n) \quad (5.3.1)$$

exists and

$$\mathbb{E}(y_{n+1} | x^n) = y_n \quad (5.3.2)$$

holds with probability 1. If $\{y_n\}$ is a martingale with respect to itself, i.e. $y_i(x) = x$, then we call it simply a *martingale*.

It is also useful to consider the following generalisations of martingale sequences to those that increase and decrease in expectation.

Definition 5.3.2. Similarly, sequence $\{y_n\}$ is a super-martingale if $\mathbb{E}(y_{n+1} | x^n) \leq y_n$ and a sub-martingale if $\mathbb{E}(y_{n+1} | x^n) \geq y_n$, w.p. 1.

At a first glance, it might appear that martingales are not very frequently encountered, apart from some niche applications. However, we can always construct a martingale from any sequence of random variables as follows.

Definition 5.3.3. Doob martingale Let a function $f : \mathcal{S}^m \rightarrow \mathbb{R}$ and some associated random variables $x^m \triangleq x_1, \dots, x_m$. Then, for some $n \leq m$, assuming the expectation $\mathbb{E}(f | x^n) = \int_{\mathcal{S}^{m-n}} f(x^m) d\mathbb{P}(x_{n+1}, \dots, x_m | x^n)$ exists, we can construct the random variable

$$y_n(x^n) = \mathbb{E}[f | x^n].$$

Then $\mathbb{E}(y_{n+1} | x^n) = y_n$, and so y_n is a martingale sequence with respect to x_n .

Another interesting type of martginale sequence are martingale *difference* sequences. They are particularly important as they are related to some useful concentration bounds.

Definition 5.3.4. A sequence $\{y_n\}$ is a *martingale difference sequence* with respect to $\{x_n\}$ if

$$\mathbb{E}(y_{n+1} | x^n) = 0 \quad \text{with probability 1.} \quad (5.3.3)$$

For bounded difference sequences, we can obtain the following well-known concentration bound.

Theorem 5.3.1. Let b_k be a random variable depending on x^{k-1} . and $\{y_k\}$ be a martingale difference sequence with respect to the $\{x_k\}$, such that $y_k \in [b_k, b_k + c_k]$ w.p. 1, Then, defining $s_k \triangleq \sum_{i=1}^k y_i$, it holds that:

$$\mathbb{P}(s_n \geq t) \leq \exp\left(\frac{-2t^2}{\sum_{i=1}^n c_i^2}\right). \quad (5.3.4)$$

This allows us to bound the probability that the difference sequence deviates from zero. Since it is only in very few problems whose default random variables are difference sequences, use of this theorem is most common by defining a new random variable sequence that is a difference sequence.

5.4 Markov processes

A more general type of sequence of random variables than Martingales are Markov processes. Informally speaking, a Markov process is a sequence of variables $\{x_n\}$ such that the next value x_{t+1} is predictable only from the current value x_t , with no previous values x_{t-k} adding any further information.

Definition 5.4.1 (Markov Process). Let $(\mathcal{S}, \mathfrak{B}(\mathcal{S}))$ be a measurable space. If $\{x_n\}$ is a sequence of random variables $x_n : \mathcal{S} \rightarrow \mathcal{X}$ such that

$$\mathbb{P}(x_t \in A | x_{t-1}, \dots, x_1) = \mathbb{P}(x_t \in A | x_{t-1}), \quad \forall A \in \mathfrak{B}(\mathcal{X}), \quad (5.4.1)$$

i.e. so that x_t is independent of x_{t-2}, \dots given x_{t-1} , then $\{x_n\}$ is a Markov process, and x_t is called the *state* of the Markov process at time t . If $\mathbb{P}(x_t \in A | x_{t-1} = x) = \tau(A | x)$ where $\tau : \mathfrak{B}(\mathcal{S}) \times \mathcal{S} \rightarrow [0, 1]$, *transition kernel*, then $\{x_n\}$ is a *stationary Markov process*

We have already encountered an example of Markov processes. This is the sequence of posterior parameters obtained in Bayesian inference, which is equivalent to the information state in the backwards induction tree.

Summary

- Sequential sampling is always better than a fixed sample size.
- Unbounded procedures are better than bounded procedures.
- Bounded procedures can be calculated using backwards induction.
- Unbounded procedures can be approximated as the limit of a sequence of bounded procedures.
- The sequential probability ratio test (SPRT) is a type of unbounded sequential decision procedure.
- The error probabilities of the SPRT $A < P_{\theta_2}(x^t)/P_{\theta_1}(x^t) < B$ are approximately $A, 1/B$. For sample cost $c \rightarrow 0$, the near-optimal values are $A = c, B = 1/c$.
- Martingales are a special type of sequence of random variables such that $\mathbb{E}(y_{n+1} | x^n) = y_n(x^n)$.

- Concentration inequalities can be derived for martingales.
- All the above problems can be modelled as Markov processes.

5.5 Exercises.

EXERCISE 20 (60). Consider a stationary Markov process with state space S and whose transition kernel is a matrix τ . At time t , we are at state $x_t = s$ and we can either, 1: Terminate and receive reward $b(s)$, or 2: Pay $c(s)$ and continue to a random state x_{t+1} from the distribution $\tau(z' | z)$.

Assuming b, c, τ are known, that $b, c > 0$, and design a backwards induction algorithm that optimises for the utility function

$$U(x_1, \dots, x_T) = b(x_T) - \sum_{t=1}^{T-1} c(x_t).$$

Finally, show that the expected utility of the optimal policy starting from any state must be bounded.

EXERCISE 21 (60). Consider the problem of classification with features $x \in \mathcal{X}$ and labels $y \in \mathcal{Y}$, where each label costs us c . Assume a Bayesian model with some parameter space Θ on which we have a prior distribution ξ_0 . Let ξ_t be the posterior distribution after t examples $(x_1, y_1), \dots, (x_t, y_t)$.

Let our expected utility be the expected accuracy (i.e. the marginal probability of correctly guessing the correct class over all possible models) of the Bayes-optimal classifier $\pi : \mathcal{X} \rightarrow \mathcal{Y}$ minus the cost paid:

$$\mathbb{E}_t(U) \triangleq \max_{\pi} \int_{\Theta} \int_{\mathcal{X}} P_{\theta}(\pi(x) | x) dP_{\theta}(x) d\xi_t(\theta) - ct$$

Show that the Bayes-optimal accuracy can be also rewritten as

$$\int_{\Theta} \int_{\mathcal{X}} \max_{y \in \mathcal{Y}} P_{\theta}(y | x) d\xi_t(\theta | x) d\mathbb{P}_t(x), -$$

where \mathbb{P}_t and \mathbb{E}_t denote marginal distributions under the belief ξ_t .

Write the expression for the expected gain in accuracy when obtaining one more sample and label.

Implement the above for a model family of your choice. Two simple options are the following. The first is a finite model family composed of two different classifiers $P_{\theta}(y | x)$. The second is the family of discrete classifier models with a Dirichlet product prior, i.e. where $\mathcal{X} = \{1, \dots, n\}$, and each different $x \in \mathcal{X}$ corresponds to a different multinomial distribution over \mathcal{Y} . In both cases, you can assume a common (and known) data distribution $P(x)$, in which case $\xi_t(\theta | x) = \xi_t(\theta)$.

Figure 5.6 shows the performance for a family of discrete classifier models with $|\mathcal{X}| = 4$. It shows the **expected** classifier performance (based on the posterior marginal), the **actual** performance on a small test set, as well as the cumulative **predicted** expected performance gain. As you can see, even though the expected performance gain is zero in some cases, cumulatively it reaches the actual performance of the classifier. You should be able to produce a similar figure for your own setup.

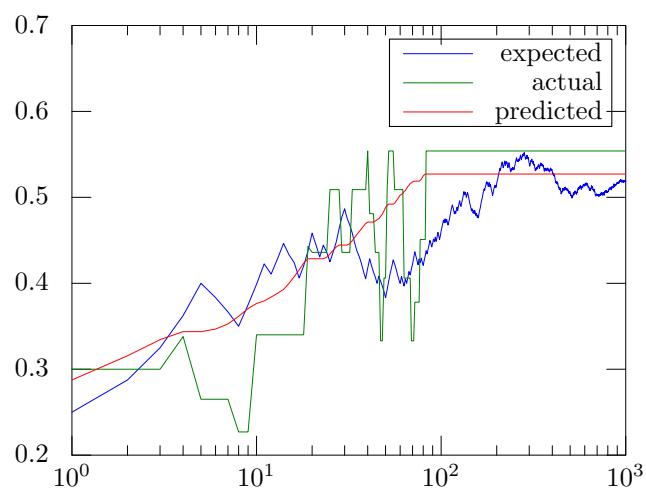


Figure 5.6: Illustrative results for an implementation of Exercise 21 on a discrete classifier model.

Chapter 6

Experiment design and Markov decision processes

6.1 Introduction

Markov decision process

This unit describes the very general formalism of Markov decision processes (MDPs) for formalising problems in sequential decision making. Thus a *Markov decision process* can be used to model stochastic path problems, stopping problems, reinforcement learning problems, experiment design problems, and control problems.

experimental design

We begin by taking a look at the problem of *experimental design*. One instance of this problem occurs when considering how to best allocate treatments with unknown efficacy to patients in an adaptive manner, so that the best treatment is found, or so as to maximise the number of patients that are treated successfully. The problem, originally considered by Chernoff [1959, 1966], informally can be stated as follows.

We have a number of treatments of unknown efficacy, i.e. some of them work better than the others. We observe patients one at a time. When a new patient arrives, we must choose which treatment to administer. Afterwards, we observe whether the patient improves or not. Given that the treatment effects are initially unknown, how can we maximise the number of cured patients? Alternatively, how can we discover the best treatment? The two different problems are formalised below.

Adaptive treatment allocation

EXAMPLE 30. Consider k treatments to be administered to T volunteers. To each volunteer only a single treatment can be assigned. At the t -th trial, we treat one volunteer with some treatment $a_t \in \{1, \dots, k\}$. We then obtain a reward $r_t = 1$ if the patient is treated and 0 otherwise. We wish to choose actions maximising the utility $U = \sum_t r_t$. This would correspond to maximising the number of patients that get treated over time.

Adaptive hypothesis testing

EXAMPLE 31. An alternative goal would be to do a *clinical trial*, in order to find the best possible treatment. For simplicity, consider the problem of trying to find out whether a particular treatment is better or not than a placebo. We are given a hypothesis set Ω , with each $\omega \in \Omega$ corresponding to different models for the effect of the treatment and the placebo. Since we don't know what is the right model, we place a prior ξ_0 on Ω . We can perform T experiments, after which we must make a decision whether or not the treatment is significantly better than the placebo. To model this, we define a decision set $\mathcal{D} = \{d_0, d_1\}$ and a utility function $U : \mathcal{D} \times \Omega \rightarrow \mathbb{R}$, which models the effect of each decision d given different versions of reality ω . One hypothesis $\omega \in \Omega$ is true. To distinguish them, we can choose from a set of k possible experiments to be performed over T trials. At the t -th trial, we choose experiment $a_t \in \{1, \dots, k\}$ and observe outcome $x_t \in \mathcal{X}$, with $x_t \sim P_\omega$ drawn from the true hypothesis. Our posterior is

$$\xi_t(\omega) \triangleq \xi_0(\omega | a_1, \dots, a_t, x_1, \dots, x_t).$$

The reward is $r_t = 0$ for $t < T$ and

$$r_T = \max_{d \in \mathcal{D}} \mathbb{E}_{\xi_T}(U | d).$$

Our utility in this can again be expressed as a sum over individual rewards, $U = \sum_{t=1}^T r_t$.

Both formalizations correspond to so-called *bandit problems* which we take a closer look at in the following section.

6.2 Bandit problems

The simplest bandit problem is the stochastic n -armed bandit. We are faced with n different one-armed bandit machines, such as those found in casinos. In this problem, at time t , you have to choose one *action* (i.e. a machine) $a_t \in \mathcal{A} = \{1, \dots, n\}$. In this setting, each time t you play a machine, you receive a reward r_t , with fixed expected value $\omega_i = \mathbb{E}(r_t | a_t = i)$. Unfortunately, you do not know ω_i , and consequently the best arm is also unknown. How do you then choose arms so as to maximise the total expected reward?

Definition 6.2.1 (The stochastic n -armed bandit problem.). This is the problem of selecting a sequence of actions $a_t \in \mathcal{A}$, with $\mathcal{A} = \{1, \dots, n\}$, so as to maximise expected utility, where the utility is

$$U = \sum_{t=0}^{T-1} \gamma^t r_t,$$

where $T \in (0, \infty]$ is the horizon and $\gamma \in (0, 1]$ is a *discount factor*. The reward r_t is stochastic, and only depends on the current action, with expectation $\mathbb{E}(r_t | a_t = i) = \omega_i$.

In order to select the actions, we must specify some *policy* or decision rule. This can only depend on the sequence of previously taken actions and observed rewards. Usually, the policy $\pi : \mathcal{A}^* \times \mathbb{R}^* \rightarrow \mathcal{A}$ is a deterministic mapping from the space of all sequences of actions and rewards to actions. That is, for every observation and action history $a_1, r_1, \dots, a_{t-1}, r_{t-1}$ it suggests a single action a_t . However, it could also be a stochastic policy, that specifies a mapping to action distributions. We use the following notation for stochastic history-dependent bandit policies,

$$\pi(a_t | a^{t-1}, r^{t-1}) \tag{6.2.1}$$

to mean the probability of actions a_t given the history until time t .

How can we solve bandit problems? One idea is to apply the Bayesian decision-theoretic framework we have developed earlier to maximise utility in expectation. More specifically, given the horizon $T \in (0, \infty]$ and the discount factor $\gamma \in (0, 1]$, we define our utility from time t to be:

$$U_t = \sum_{k=1}^{T-t} \gamma^k r_{t+k}. \tag{6.2.2}$$

To apply the decision theoretic framework, we need to define a suitable family of probability measures \mathcal{P} , indexed by parameter $\omega \in \Omega$ describing the reward distribution of each bandit, together with a prior distribution ξ on Ω . Since ω is unknown, we cannot maximise the expected utility with respect to it. However, we can always maximise expected utility with respect to our belief ξ . That is, we replace the ill-defined problem of maximising utility in an unknown model with that of maximising expected utility given a distribution over possible models. The problem can be written in a simple form:

$$\max_{\pi} \mathbb{E}_{\xi}^{\pi} U_t = \max_{\pi} \int_{\Omega} \mathbb{E}_{\omega}^{\pi} U_t d\xi \omega. \tag{6.2.3}$$

The difficulty lies not in formalising the problem, but in the fact that the set of learning policies is quite large, rendering the optimisation infeasible. The following figure summarises the statement of the bandit problem in the Bayesian setting.

Decision-theoretic statement of the bandit problem

- Let \mathcal{A} be the set of arms.
- Define a family of distributions $\mathcal{P} = \{P_{\omega,i} \mid \omega \in \Omega, i \in \mathcal{A}\}$ on \mathbb{R} .
- Assume the i.i.d model $r_t \mid \omega, a_t = i \sim P_{\omega,i}$.
- Define prior ξ on Ω .
- Select a policy $\pi : \mathcal{A}^* \times \mathbb{R}^* \rightarrow \mathcal{A}$ maximising

$$\mathbb{E}_\xi^\pi U = \mathbb{E}_\xi^\pi \sum_{t=0}^{T-1} \gamma^t r_t$$

There are two main difficulties with this approach. The first is specifying the family and the prior distribution: this is effectively part of the problem formulation and can severely influence the solution. The second is calculating the policy that maximises expected utility given a prior and family. The first problem can be resolved by either specifying a subjective prior distribution, or by selecting a prior distribution that has good worst-case guarantees. The second problem is hard to solve, because in general, such policies are history dependent and the set of all possible histories is exponential in the horizon T .

6.2.1 An example: Bernoulli bandits

As a simple illustration, consider the case when the reward for choosing one of the n actions is either 0 or 1, with some fixed, yet unknown probability depending on the chosen action. This can be modelled in the standard Bayesian framework using the Beta-Bernoulli conjugate prior. More specifically, we can formalise the problem as follows.

Consider n Bernoulli distributions with unknown parameters ω_i ($i = 1, \dots, n$) such that

$$r_t \mid a_t = i \sim \text{Bern}(\omega_i), \quad \mathbb{E}(r_t \mid a_t = i) = \omega_i. \quad (6.2.4)$$

Each Bernoulli distribution thus corresponds to the distribution of rewards obtained from each bandit that we can play. In order to apply the statistical decision theoretic framework, we have to quantify our uncertainty about the parameters ω in terms of a probability distribution.

We model our belief for each bandit's parameter ω_i as a Beta distribution $\text{Beta}(\alpha_i, \beta_i)$, with density $f(\omega \mid \alpha_i, \beta_i)$ so that

$$\xi(\omega_1, \dots, \omega_n) = \prod_{i=1}^n f(\omega_i \mid \alpha_i, \beta_i).$$

Recall that the posterior of a Beta prior is also a Beta. Let

$$N_{t,i} \triangleq \sum_{k=1}^t \mathbb{I}\{a_k = i\}$$

be the number of times we played arm i and

$$\hat{r}_{t,i} \triangleq \frac{1}{N_{t,i}} \sum_{k=1}^t r_t \mathbb{I}\{a_k = i\}$$

be the *empirical reward* of arm i at time t . We can let this equal 0 when $N_{t,i} = 0$. Then, the posterior distribution for the parameter of arm i is

$$\xi_t = \text{Beta}(\alpha_i + N_{t,i} \hat{r}_{t,i}, \beta_i + N_{t,i}(1 - \hat{r}_{t,i})).$$

Since $r_t \in \{0, 1\}$ the possible states of our belief given some prior are \mathbb{N}^{2n} .

In order for us to be able to evaluate a policy, we need to be able to predict the expected utility we obtain. This only depends on our current belief, and the state of our belief corresponds to the state of the bandit problem. This means that everything we know about the problem at time t can be summarised by ξ_t . For Bernoulli bandits, sufficient statistic for our belief is the number of times we played each bandit and the total reward from each bandit. Thus, our state at time t is entirely described by our priors α, β (the initial state) and the vectors

$$N_t = (N_{t,1}, \dots, N_{t,i}) \tag{6.2.5}$$

$$\hat{r}_t = (\hat{r}_{t,1}, \dots, \hat{r}_{t,i}). \tag{6.2.6}$$

At any time t , we can calculate the probability of observing $r_t = 1$ or $r_t = 0$ if we pull arm i as:

$$\xi_t(r_t = 1 | a_t = i) = \frac{\alpha_i + N_{t,i} \hat{r}_{t,i}}{\alpha_i + \beta_i + N_{t,i}}$$

So, not only we can predict the immediate reward based on our current belief, but we can also predict all next possible beliefs: the next state is well-defined and depends only on the current state. As we shall see later, this type of decision problem is more generally called a Markov decision process (Definition 6.3.1). For now, we shall more generally (and precisely) define the bandit process itself.

6.2.2 Decision-theoretic bandit process

The basic bandit process can be seen in Figure 6.2(a). We can now define the general decision-theoretic bandit process, not restricted to independent Bernoulli bandits.

Definition 6.2.2. Let \mathcal{A} be a set of actions, not necessarily finite. Let Ω be a set of possible parameter values, indexing a family of probability measures $\mathcal{P} = \{P_{\omega,a} | \omega \in \Omega, a \in \mathcal{A}\}$. There is some $\omega \in \Omega$ such that, whenever we take action $a_t = a$, we observe reward $r_t \in \mathcal{R} \subset \mathbb{R}$ with probability measure:

$$P_{\omega,a}(R) \triangleq \mathbb{P}_{\omega}(r_t \in R | a_t = a), \quad R \subseteq \mathbb{R}. \tag{6.2.7}$$

Let ξ_1 be a prior distribution on Ω and let the posterior distributions be defined as:

$$\xi_{t+1}(B) \propto \int_B P_{\omega, a_t}(r_t) d\xi_t(\omega). \quad (6.2.8)$$

The next belief is random, since it depends on the random quantity r_t . In fact, the probability of the next reward lying in R if $a_t = a$ is given by the following marginal distribution:

$$P_{\xi_t, a}(R) \triangleq \int_{\Omega} P_{\omega, a}(R) d\xi_t(\omega). \quad (6.2.9)$$

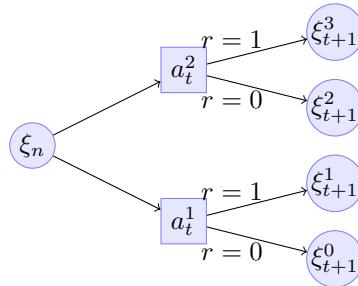


Figure 6.1: A partial view of the multi-stage process. Here, the probability that we obtain $r = 1$ if we take action $a_t = i$ is simply $P_{\xi_t, i}(\{1\})$.

Finally, as ξ_{t+1} deterministically depends on ξ_t, a_t, r_t , the probability of obtaining a particular next belief is the same as the probability of obtaining the corresponding rewards leading to the next belief. In more detail, we can write:

$$\mathbb{P}(\xi_{t+1} = \xi | \xi_t, a_t) = \int_{\mathcal{R}} \mathbb{I}\{\xi_t(\cdot | a_t, r_t = r) = \xi\} dP_{\xi_t, a}(r). \quad (6.2.10)$$

In practice, although multiple reward sequences may lead to the same beliefs, we frequently ignore that possibility for simplicity. Then the process becomes a tree. A solution to the problem of what action to select is given by a backwards induction algorithm similar to that given in Section 5.2.2.

$$U^*(\xi_t) = \max_{a_t} \mathbb{E}(r_t | \xi_t, a_t) + \sum_{\xi_{t+1}} \mathbb{P}(\xi_{t+1} | \xi_t, a_t) U^*(\xi_{t+1}). \quad (6.2.11)$$

backwards induction

The above equation is the *backwards induction* algorithm for bandits. If you look at this structure, you can see that next belief only depends on the current belief, action and reward, i.e. it satisfies the Markov property, as seen in Figure 6.1. Consequently, a decision-theoretic bandit process can be modelled more generally as a Markov decision process, explained in the following section. It turns out that backwards induction, as well as other efficient algorithms, can provide optimal solutions for Markov decision processes.

In reality, the reward depends only on the action and the unknown ω , as can be seen in Figure 6.2(b). This is the point of view of an external observer. However, from the point of view of the decision maker, the distribution of ω

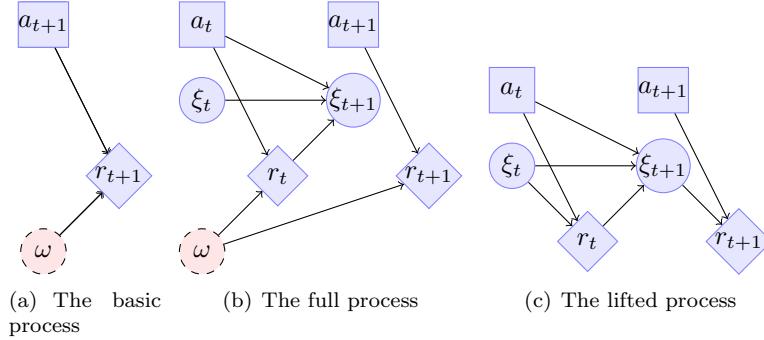


Figure 6.2: Three views of the bandit process. Figure 6.2(a) shows the basic bandit process, from the view of an external observer. The decision maker selects a_t , while the parameter ω of the process is hidden. It then obtains reward r_t . The process repeats for $t = 1, \dots, T$. The decision-theoretic bandit process is shown in Figures 6.2(b) and 6.2(c). While ω is not known, at each time step t we maintain a belief ξ_t on Ω . The reward distribution is then defined through our belief. In Figure 6.2(b), we can see that complete process, where the dependency on ω is clear. In Figure 6.2(c), we marginalise out ω and obtain a model where the transitions only depend on the current belief and action.

only depends on his current belief. Consequently, the distribution of rewards also only depends on the current belief, as we can marginalise over ω . This gives rise to the decision-theoretic bandit process shown in Figure 6.2(c). In the following section, we shall consider Markov decision processes more generally.

6.3 Markov decision processes and reinforcement learning

Bandit problems are one of the simplest instances of reinforcement learning problems. Informally, speaking, these are problems of learning how to act in an unknown environment, only through interaction with the environment and limited reinforcement signals. The learning agent interacts with the environment through actions and observations, and simultaneously obtains rewards. For example, we can consider a rat running through a maze designed by an experimenter, which from time to time finds a piece of cheese, the reward. The goal of the agent is usually to maximise some measure of the total reward. In summary, we can state the problem as follows.

The reinforcement learning problem.

The reinforcement learning problem is the problem of *learning* how to act in an *unknown* environment, only by [interaction](#) and [reinforcement](#).

Generally, we assume that the environment μ that we are acting in has an underlying state $s_t \in \mathcal{S}$, which changes with discrete time steps t . At each step, the agent obtains an observation $x_t \in \mathcal{X}$ and chooses actions $a_t \in \mathcal{A}$.

We usually assume that the environment is such that its next state s_{t+1} only depends on its current state s_t and the last action taken by the agent, a_t . In addition, the agent observes a reward signal r_t , and its goal is to maximise the total reward during its lifetime.

Doing so when the environment μ is unknown, is hard even in seemingly simple settings, like n -armed bandits, where the underlying state never changes. In many real-world applications, the problem is even harder, as the state is not directly observed. Instead, we may simply have some measurements x_t , which give only partial information about the true underlying state s_t .

Reinforcement learning problems typically fall into one of the following three groups: (1) Markov decision processes (MDPs), where the state s_t is observed directly, i.e. $x_t = s_t$; (2) Partially observable MDPs (POMDPs), where the state is hidden, i.e. x_t is only probabilistically dependent on the state; and (3) stochastic Markov games, where the next state also depends on the move of other agents. While all of these problem *descriptions* are different, in the Bayesian setting, they all can be reformulated as MDPs, by constructing an appropriate belief state, similarly to how we did it for the decision theoretic formulation of the bandit problem.

In this chapter, we shall restrict our attention to Markov decision processes. Hence, we shall not discuss the existence of other agents, or the case where we cannot observe the state directly.

*transition distribution
reward distribution*

Definition 6.3.1 (Markov Decision Process). A Markov decision process μ is a tuple $\mu = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$, where \mathcal{S} is the *state space* and \mathcal{A} is the *action space*. The *transition distribution* being $\mathcal{P} = \{P(\cdot | s, a) | s \in \mathcal{S}, a \in \mathcal{A}\}$ is a collection of probability measures on \mathcal{S} , indexed in $\mathcal{S} \times \mathcal{A}$ and the *reward distribution* $\mathcal{R} = \{\rho(\cdot | s, a) | s \in \mathcal{S}, a \in \mathcal{A}\}$ is a collection of probability measures on \mathbb{R} , such that:

$$P(S | s, a) = \mathbb{P}_\mu(s_{t+1} \in S | s_t = s, a_t = a) \quad (6.3.1)$$

$$\rho(R | s, a) = \mathbb{P}_\mu(r_t \in R | s_t = s, a_t = a). \quad (6.3.2)$$

For simplicity, we shall also use

$$r_\mu(s, a) = \mathbb{E}_\mu(r_{t+1} | s_t = s, a_t = a), \quad (6.3.3)$$

for the expected reward.

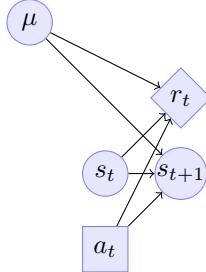
Of course, the transition and reward distributions are different for different environments μ . For that reason, we shall usually subscript the relevant probabilities and expectations with μ , unless the MDP is clear from the context.

Dependencies of rewards. Sometimes it is more convenient to have rewards that depend on the next state as well, i.e.

$$r_\mu(s, a, s') = \mathbb{E}_\mu(r_{t+1} | s_t = s, a_t = a, s_{t+1} = s'), \quad (6.3.6)$$

though this is complicates the notation considerably since now the reward is obtained on the next time step. However, we can always replace this with the expected reward for a given state-action pair:

$$r_\mu(s, a) = \mathbb{E}_\mu(r_{t+1} | s_t = s, a_t = a) = \sum_{s' \in \mathcal{S}} P_\mu(s' | s, a) r_\mu(s, a, s') \quad (6.3.7)$$



Markov property of the reward and state distribution

$$\mathbb{P}_\mu(s_{t+1} \in S | s_1, a_1, \dots, s_t, a_t) = \mathbb{P}_\mu(s_{t+1} \in S | s_t, a_t) \quad (6.3.4)$$

$$\mathbb{P}_\mu(r_t \in R | s_1, a_1, \dots, s_t, a_t) = \mathbb{P}_\mu(r_t \in R | s_t, a_t) \quad (6.3.5)$$

where $S \subset \mathcal{S}$ and $R \subset \mathcal{R}$ are reward and state subsets respectively.

Figure 6.3: The structure of a Markov decision process.

In fact, it is notationally more convenient to have rewards that only depend on the current state:

$$r_\mu(s) = \mathbb{E}_\mu(r_t | s_t = s). \quad (6.3.8)$$

For simplicity, we shall mainly consider the latter case.

The agent. The environment does not exist in isolation. The actions are taken by an agent, who is interested in obtaining high rewards. Instead of defining an algorithm for choosing actions directly, we define an algorithm for computing policies, which define distributions on actions.

The agent's policy π

$$\mathbb{P}^\pi(a_t | s_t, \dots, s_1, a_{t-1}, \dots, a_1) \quad (\text{history-dependent policy})$$

$$\mathbb{P}^\pi(a_t | s_t) \quad (\text{Markov policy})$$

In some sense, the agent is defined by its *policy* π , which is a conditional distribution on actions given the history. The *policy* π is otherwise known as a *policy decision function*. In general, the policy can be history-dependent. In certain cases, however, there are optimal policies that are Markov. This is for example the case with additive utility functions. In particular, the utility function maps from the sequence of all possible rewards to a real number $U : \mathcal{R}^* \rightarrow \mathbb{R}$, given below:

Definition 6.3.2 (Utility). Given a horizon T and a discount factor $\gamma \in (0, 1]$, the utility function $U : \mathcal{R}^* \rightarrow \mathbb{R}$ is defined as

$$U(r_0, r_1, \dots, r_T) = \sum_{k=0}^T \gamma^k r_k. \quad (6.3.9)$$

It is convenient to give a special name to the utility starting from time t , i.e. the sum of rewards from that time on:

$$U_t \triangleq \sum_{k=0}^{T-t} \gamma^k r_{t+k}. \quad (6.3.10)$$

At any time t , the agent wants to find a policy π *maximising the expected total future reward*

$$\mathbb{E}_\mu^\pi U_t = \mathbb{E}_\mu^\pi \sum_{k=0}^{T-t} \gamma^k r_{t+k}. \quad (\text{expected utility})$$

This is so far identical to the expected utility framework we had seen so far, with the only difference that now the reward space is a sequence of numerical rewards and that we are acting within a dynamical system with state space \mathcal{S} . In fact, it is a good idea to think about the *value* of different states of the system under certain policies, in the same way that one thinks about how good different positions are in chess.

6.3.1 Value functions

A value function represents the expected utility of a given state, or state-and-action pair for a specific policy. They are really useful as shorthand notation and as the basis of algorithm development. The most basic of those is the state value function.

State value function

$$V_{\mu,t}^\pi(s) \triangleq \mathbb{E}_\mu^\pi(U_t | s_t = s) \quad (6.3.11)$$

The state value function for a particular policy π can be interpreted as how much utility you should expect if you follow the policy starting from state s at time t , for the particular MDP μ .

State-action value function

$$Q_{\mu,t}^\pi(s, a) \triangleq \mathbb{E}_\mu^\pi(U_t | s_t = s, a_t = a) \quad (6.3.12)$$

The state-action value function for a particular policy π can be interpreted as how much utility you should expect if you play action a , at state s at time t , and then follow the policy π , for the particular MDP μ .

It is also useful to define the optimal policy and optimal value functions for a given MDP. In the following, a star indicates optimal quantities. The *optimal policy* π^*

$$\pi^*(\mu) : V_{t,\mu}^{\pi^*(\mu)}(s) \geq V_{t,\mu}^\pi(s) \quad \forall \pi, t, s \quad (6.3.13)$$

dominates all other policies π everywhere in \mathcal{S} .

The *optimal value function* V^*

$$V_{t,\mu}^*(s) \triangleq V_{t,\mu}^{\pi^*(\mu)}(s), \quad Q_{t,\mu}^*(s) \triangleq Q_{t,\mu}^{\pi^*(\mu)}(s, a). \quad (6.3.14)$$

is the value function of the optimal policy π^* .

Finding the optimal policy when μ is known

When the MDP μ is known, the expected utility of any policy can be calculated. Therefore, one could find the optimal policy by brute force, i.e. by calculating the utility of every possible policy. This might be a reasonable strategy if the number of policies is small. However, there are many better approaches. First, there are iterative/offline methods where an optimal policy is found for all states of the MDP. These either try to estimate the optimal value function directly, or try to iteratively improve a policy until it is optimal. The second type of methods tries to find an optimal policy online. That is, the optimal actions are estimated only for states which can be visited in the future starting from the current state. However, the same main ideas are used in all of these algorithms.

6.4 Finite horizon, undiscounted problems

The conceptually simplest type of problems are finite horizon problems where $T < \infty$ and $\gamma = 1$. The first thing we shall try to do is to evaluate a given policy for a given MDP. There are a number of algorithms that can achieve this.

6.4.1 Policy evaluation

Here we are interested in the problem of determining the value function of a policy π (for $\gamma = 1, T < \infty$). All the algorithms we shall consider can be recovered from the following recursion. Noting that $U_{t+1} = \sum_{k=1}^{T-t} r_{t+k}$ we have:

$$V_{\mu,t}^\pi(s) \triangleq \mathbb{E}_\mu^\pi(U_t \mid s_t = s) \quad (6.4.1)$$

$$= \sum_{k=0}^{T-t} \mathbb{E}_\mu^\pi(r_{t+k} \mid s_t = s) \quad (6.4.2)$$

$$= \mathbb{E}_\mu^\pi(r_t \mid s_t = s) + \mathbb{E}_\mu^\pi(U_{t+1} \mid s_t = s) \quad (6.4.3)$$

$$= \mathbb{E}_\mu^\pi(r_t \mid s_t = s) + \sum_{i \in \mathcal{S}} V_{\mu,t+1}^\pi(i) \mathbb{P}_\mu^\pi(s_{t+1} = i \mid s_t = s). \quad (6.4.4)$$

Note that the last term can be calculated easily through marginalisation.

$$\mathbb{P}_\mu^\pi(s_{t+1} = i \mid s_t = s) = \sum_{a \in \mathcal{A}} \mathbb{P}_\mu(s_{t+1} = i \mid s_t = s, a_t = a) \mathbb{P}^\pi(a_t = a \mid s_t = s).$$

This derivation directly gives a number of *policy evaluation algorithms*.

Direct policy evaluation Direct policy evaluation is based on (6.4.2), which can be implemented by Algorithm 2. One needs to *marginalise out* all possible state sequences to obtain the expected reward given the state at time $t + k$ giving the following:

$$\mathbb{E}_\mu^\pi(r_{t+k} \mid s_t = s) = \sum_{s_{t+1}, \dots, s_{t+k} \in \mathcal{S}^k} \mathbb{E}_\mu^\pi(r_{t+k} \mid s_{t+k}) \mathbb{P}_\mu^\pi(s_{t+1}, \dots, s_{t+k} \mid s_t).$$

By using the Markov property, we calculate the probability of reaching any state from any other state at different times, and then add up the expected reward we would get in that state under our policy. Then $\hat{V}_t(s) = V_{\mu,t}^\pi(s)$ by definition.

Unfortunately it is not a very good idea to use direct policy evaluation. The most efficient implementation involves calculating $P(s_t \mid s_0)$ recursively for every state. This would result in a total of $|\mathcal{S}|^3T$ operations. Monte-Carlo evaluations should be considerably cheaper, especially when the transition structure is sparse.

Algorithm 2 Direct policy evaluation

```

1: for  $s \in \mathcal{S}$  do
2:   for  $t = 0, \dots, T$  do
3:
4:   
$$\hat{V}_t(s) = \sum_{k=t}^T \sum_{j \in \mathcal{S}} \mathbb{P}_\mu^\pi(s_k = j \mid s_t = s) \mathbb{E}_\mu^\pi(r_k \mid s_k = j).$$

5: end for
end for

```

6.4.2 Monte-Carlo policy evaluation

Another conceptually simple algorithm is Monte-Carlo policy evaluation shown as Algorithm 3. The idea is that instead of summing over all possible states to be visited, we just draw states from the Markov chain defined jointly by the policy and the Markov decision process. Unlike direct policy evaluation the algorithm needs a parameter K , the number of trajectories to generate. Nevertheless, this is a very useful method, employed within a number of more complex algorithms.

Algorithm 3 Monte-Carlo policy evaluation

```

for  $s \in \mathcal{S}$  do
  for  $k = 0, \dots, K$  do
    Choose initial state  $s_1$ .
    for  $t = 1, \dots, T$  do
       $a_t \sim \pi(a_t | s_t)$                                 // Take action
      Observe reward  $r_t$  and next state  $s_{t+1}$ .
      Set  $r_{t,k} = r_t$ .
    end for
    Save total reward:
    
$$\hat{V}_k(s) = \sum_{t=1}^T r_{t,k}.$$

  end for
  Calculate estimate:
  
$$\hat{V}(s) = \frac{1}{K} \sum_{k=1}^K \hat{V}_k(s).$$

end for

```

Remark 6.4.1. The estimate \hat{V} of the Monte Carlo evaluation algorithm satisfies

$$\|V - \hat{V}\|_\infty \leq \sqrt{\frac{\ln(2|\mathcal{S}|/\delta)}{2K}} \quad \text{with probability } 1 - \delta$$

Proof. From Hoeffding's inequality (4.5.5) we have for any state s that

$$\mathbb{P}\left(|\hat{V}(s) - V(s)| \geq \sqrt{\frac{\ln(2|\mathcal{S}|/\delta)}{2K}}\right) \leq \delta/|\mathcal{S}|.$$

Consequently, using a union bound of the form $P(A_1 \cup A_2 \cup \dots \cup A_n) \leq \sum_i P(A_i)$ gives the required result. \square

The main advantage of Monte-Carlo policy evaluation is that it can be used in very general settings. It can be used not only in Markovian environments such as MDPs, but also in partially observable and multi-agent settings.

6.4.3 Backwards induction policy evaluation

Finally, the backwards induction algorithm shown as Algorithm 4 is similar to the backwards induction algorithm we saw for sequential sampling and bandit problems. However, here we are only evaluating a policy rather than finding the optimal one. This algorithm is slightly less generally applicable than the Monte-Carlo method because it makes Markovian assumptions. The Monte-Carlo algorithm, can be used for environments that with a non-Markovian variable s_t .

Algorithm 4 Backwards induction policy evaluation

For each state $s \in S$, for $t = 1, \dots, T - 1$:

$$\hat{V}_t(s) = r_\mu^\pi(s) + \sum_{j \in S} \mathbb{P}_\mu^\pi(s_{t+1} = j \mid s_t = s) \hat{V}_{t+1}(j), \quad (6.4.5)$$

with $\hat{V}_T(s) = r_\mu^\pi(s)$.

Theorem 6.4.1. *The backwards induction algorithm gives estimates $\hat{V}_t(s)$ satisfying*

$$\hat{V}_t(s) = V_{\mu,t}^\pi(s) \quad (6.4.6)$$

Proof. For $t = T - 1$, the result is obvious. We can prove the remainder by induction. Let (6.4.6) hold for all $t \geq n + 1$. Now we prove that it holds for n . Note that from the recursion (6.4.5) we have:

$$\begin{aligned} \hat{V}_t(s) &= r_\mu(s) + \sum_{j \in S} \mathbb{P}_{\mu,\pi}(s_{t+1} = j \mid s_t = s) \hat{V}_{t+1}(j) \\ &= r(s) + \sum_{j \in S} \mathbb{P}_{\mu,\pi}(s_{t+1} = j \mid s_t = s) V_{\mu,t+1}^\pi(j) \\ &= r(s) + \mathbb{E}_{\mu,\pi}(U_{t+1} \mid s_t = s) \\ &= \mathbb{E}_{\mu,\pi}(U_t \mid s_t = s) = V_{\mu,t}^\pi(s), \end{aligned}$$

where the second equality is by the induction hypothesis, the third and fourth equalities are by the definition of the utility, and the last by definition of $V_{\mu,t}^\pi$. \square

6.4.4 Backwards induction policy optimisation

Backwards induction as given in algorithm 5 is the first non-naive algorithm for finding an optimal policy for the sequential problems with T stages. It is basically identical to the backwards induction algorithm we saw in Chapter 5, which was for the very simple sequential sampling problem, as well as the backwards induction algorithm for the decision-theoretic bandit problem.

Algorithm 5 Finite-horizon backwards induction

Input μ , set \mathcal{S}_T of states reachable within T steps.

Initialise $V_T(s) := \max_a r(s, a)$, for all $s \in \mathcal{S}_T$.

for $n = T - 1, T - 2, \dots, 1$ **do**

for $s \in \mathcal{S}_n$ **do**

$$\pi_n(s) = \arg \max_a r(s, a) + \sum_{s' \in \mathcal{S}_{n+1}} P_\mu(s' \mid s, a) V_{n+1}(s')$$

$$V_n(s) = r(s, a) + \sum_{s' \in \mathcal{S}_{n+1}} P_\mu(s' \mid s, \pi_n(s)) V_{n+1}(s')$$

end for

end for

Return $\pi = (\pi_n)_{n=1}^T$.

Theorem 6.4.2. *For T -horizon problems, backwards induction is optimal, i.e.*

$$V_n(s) = V_{\mu,n}^*(s) \quad (6.4.7)$$

Proof. Note that the proof below also holds for $r(s, a) = r(s)$. First we show that $V_t \geq V_t^*$. For $n = T$ we evidently have $V_T(s) = \max_a r(s, a) = V_{\mu, T}^*(s)$. Now assume that for $n \geq t + 1$, (6.4.7) holds. Then it also holds for $n = t$, since for any policy π'

$$\begin{aligned} V_t(s) &= \max_a \left\{ r(s, a) + \sum_{j \in \mathcal{S}} P_\mu(j | s, a) V_{t+1}(j) \right\} \\ &\geq \max_a \left\{ r(s, a) + \sum_{j \in \mathcal{S}} P_\mu(j | s, a) V_{\mu, t+1}^*(j) \right\} \quad (\text{by induction assumption}) \\ &\geq \max_a \left\{ r(s, a) + \sum_{j \in \mathcal{S}} P_\mu(j | s, a) V_{\mu, t+1}^{\pi'}(j) \right\} \\ &\geq V_t^{\pi'}(s). \end{aligned}$$

This holds for any policy π' , including $\pi' = \pi$, the policy returned by backwards induction. Then:

$$V_{\mu, t}^*(s) \geq V_{\mu, t}^\pi(s) = V_t(s) \geq V_{\mu, t}^*(s).$$

□

Remark 6.4.2. A similar theorem can be proven for arbitrary \mathcal{S} . This requires using sup instead of max and proving the existence of a π' that is arbitrary-close in value to V^* . For details, see [Puterman, 1994].

6.5 Infinite-horizon

When problems have no fixed horizon, they usually can be modelled as infinite horizon problems, sometimes with help of a *terminating state*, whose visit terminates the problem, or discounted rewards, which indicate that we care less about rewards further in the future. When reward discounting is exponential, these problems can be seen as undiscounted problems with random and geometrically distributed horizon. For problems with no discounting and no termination states there are some complications in the definition of optimal policy. However, we defer discussion of such problems to Chapter 10.

6.5.1 Examples

We begin with some examples, which will help elucidate the concept of terminating states and infinite horizon. The first is shortest path problems, where the aim is to find the shortest path to a particular goal. Although the process terminates when the goal is reached, not all policies may be able to reach the goal, and so the process may never terminate.

Shortest-path problems

We shall consider two types of shortest path problems, deterministic and stochastic. Although conceptually very different, both problems have essentially the same complexity.

Consider an agent moving in a maze, aiming to get to some terminating goal state X . That is, when reaching this state, the agent cannot move anymore, and receives a reward of 0. In general, the agent can move deterministically in the four cardinal directions, and receives a negative reward at each time step. Consequently, the optimal policy is to move to X as quickly as possible.

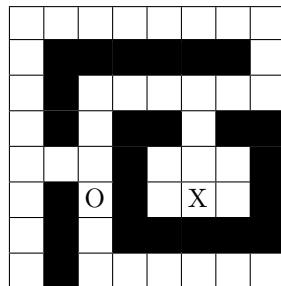
14	13	12	11	10	9	8	7
15		13					6
16	15	14		4	3	4	5
17					2		
18	19	20		2	1	2	
19		21		1	0	1	
20		22					
21		23	24	25	26	27	28

Properties

- $\gamma = 1, T \rightarrow \infty$.
- $r_t = -1$ unless $s_t = X$, in which case $r_t = 0$.
- $\mathbb{P}_\mu(s_{t+1} = X | s_t = X) = 1$.
- $\mathcal{A} = \{\text{North, South, East, West}\}$
- Transitions are deterministic and walls block.

Solving the shortest path problem can be done simply by looking at the distance of any point to X . Then the reward obtained by the optimal policy starting from any point, is simply the negative distance. The optimal policy simply moves to the state with the smallest distance to X .

Stochastic shortest path problem with a pit Now assume the shortest path problem with stochastic dynamics. That is, at each time-step there is a small probability ω that move to a random direction. In addition, there is a pit O , that is a terminating state with a reward of -100 .



Properties

- $\gamma = 1, T \rightarrow \infty$.
- $r_t = -1$, but $r_t = 0$ at X and -100 at O and episode ends.
- $\mathbb{P}_\mu(s_{t+1} = X | s_t = X) = 1$.
- $\mathcal{A} = \{\text{North, South, East, West}\}$
- Moves to a random direction with probability ω . Walls block.

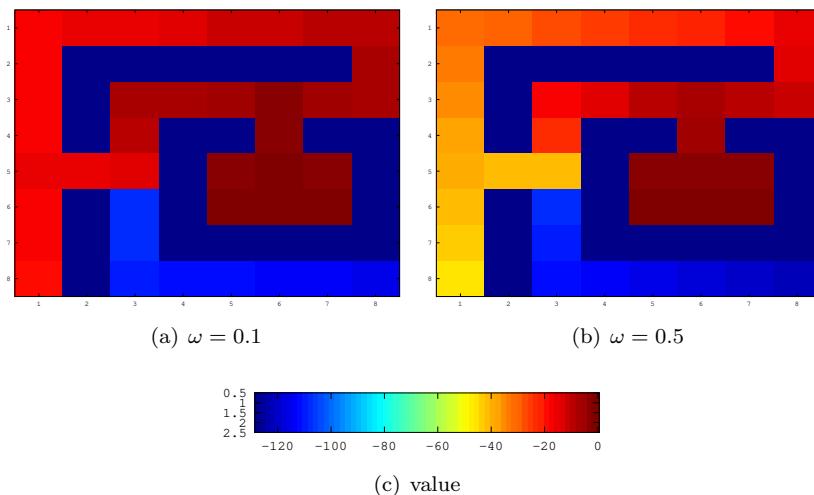


Figure 6.4: Pit maze solutions for two values of ω .

Randomness changes the solution significantly in this environment. When ω is relatively small, it is worthwhile (in expectation) for the agent to pass past the pit, even though there is a risk of falling in and getting a reward of -100 . In the example given, even starting from the third row, the agent prefers taking the short-cut. For high enough ω , the optimal policy avoids approaching the pit. Still, the agent prefers jumping in the pit, than being trapped at the bottom of the maze forever.

Continuing problems

Finally, many problems have no natural terminating state, but are continuing *ad infinitum*. Frequently, we model those problems using a utility that discounts future rewards exponentially. This way, we can guarantee that the utility is bounded. In addition, exponential discounting also has some economical sense. This is partially because of the effects of inflation, and partially because money now may be more useful than money in the future. Both these effects diminish the value of money over time. As an example, consider the following inventory management problem.

EXAMPLE 32 (Inventory management). There are K storage locations, and each location i can store n_i items. At each time-step there is a probability ϕ_i that a client tries to buy an item from location i , where $\sum_i \phi_i \leq 1$. If there is an item available, when this happens, you gain reward 1. There are two types of actions, one for ordering a certain number u units of stock, paying $c(u)$. Further one may move u units of stock from one location i to another location j , paying $\psi_{ij}(u)$.

An easy special case is when $K = 1$, and we assume that deliveries happen once every m timesteps, and each time-step a client arrives with probability ϕ . Then the state set $\mathcal{S} = \{0, 1, \dots, n\}$ corresponds to the number of items we have, the action set $\mathcal{A} = \{0, 1, \dots, n\}$ to the number of items we may order. The transition probabilities are given by $P(s'|s, a) = \binom{m}{d} \phi^d (1 - \phi)^{m-d}$, where $d = s + a - s'$, for $s + a \leq n$.

6.5.2 Markov chain theory for discounted problems

Here we consider MDPs with infinite horizon and discounted rewards. We shall consider undiscounted rewards only in Chapter 10. Our utility in this case is the discounted total reward:

$$U_t = \lim_{T \rightarrow \infty} \sum_{k=t}^T \gamma^k r_k, \quad \gamma \in (0, 1)$$

For simplicity, in the following we assume that rewards only depend on the current state instead of both state and action. It can easily be verified that results still hold in the latter case. More importantly, we also assume that the state and action spaces \mathcal{S}, \mathcal{A} are finite, and that the transition kernel of the MDP is time-invariant. This allows us to use the following simplified vector notation:

- $\mathbf{v}^\pi = (\mathbb{E}^\pi(U_t | s_t = s))_{s \in \mathcal{S}}$ is a vector in $\mathbb{R}^{|\mathcal{S}|}$ representing the value of policy π .
- Sometimes we will use $p(j|s, a)$ as a shorthand for $\mathbb{P}_\mu(s_{t+1} = j | s_t = s, a_t = a)$.
- $\mathbf{P}_{\mu, \pi}$ is a transition matrix in $\mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ for policy π , such that

$$\mathbf{P}_{\mu, \pi}(i, j) = \sum_a p(j | i, a) \mathbb{P}^\pi(a | i).$$

- \mathbf{r} is a reward vector in $\mathbb{R}^{|\mathcal{S}|}$.
- The space of value functions \mathcal{V} is a Banach space (i.e., a complete, normed vector space) equipped with the norm

$$\|\mathbf{v}\| = \sup \{|\mathbf{v}(s)| \mid s \in \mathcal{S}\}$$

For infinite-horizon discounted MDPs, stationary policies are sufficient. This can be proven by induction, using arguments similar to other proofs given here. For a detailed set of proofs, see Puterman [1994].

Definition 6.5.1. A policy π is stationary if $\pi(a_t | s_t) = \pi(a_n | s_n)$ for all n, t .

We now present a set of important results that link Markov decision processes to linear algebra.

Remark 6.5.1. We can use the Markov chain kernel \mathbf{P} to write the expected reward vector as

$$\mathbf{v}^\pi = \sum_{t=0}^{\infty} \gamma^t \mathbf{P}_{\mu, \pi}^t \mathbf{r} \tag{6.5.1}$$

Proof.

$$\begin{aligned} V^\pi(s) &= \mathbb{E} \left(\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s \right) \\ &= \sum_{t=0}^{\infty} \gamma^t \mathbb{E}(r_t \mid s_0 = s) \\ &= \sum_{t=0}^{\infty} \gamma^t \sum_{i \in \mathcal{S}} \mathbb{P}(s_t = i \mid s_0 = s) \mathbb{E}(r_t \mid s_t = i). \end{aligned}$$

Since for any distribution vector \mathbf{p} over \mathcal{S} , we have $\mathbb{E}_{\mathbf{p}} r_t = \mathbf{p}^\top \mathbf{r}$, the result follows. \square

It is possible to show that the expected discounted total reward of a policy is equal to the expected undiscounted total reward with a geometrically distributed horizon (see exercise 23). As a corollary, it follows a Markov decision process with discounting is equivalent with one where there is no discounting, but a stopping probability $(1 - \gamma)$ at every step.

The value of a particular policy can be expressed as a linear equation. This is an important result, as it has led to a number of successful algorithms that employ linear theory.

Theorem 6.5.1. *For any stationary policy π , \mathbf{v}^π is the unique solution of*

$$\mathbf{v} = \mathbf{r} + \gamma \mathbf{P}_{\mu, \pi} \mathbf{v}. \quad (6.5.2)$$

In addition, the solution is:

$$\mathbf{v}^\pi = (\mathbf{I} - \gamma \mathbf{P}_{\mu, \pi})^{-1} \mathbf{r}, \quad (6.5.3)$$

where \mathbf{I} is the identity matrix.

To prove this we will need the following important theorem.

Theorem 6.5.2. *For any bounded linear transformation $\mathbf{A} : S \rightarrow S$ on a normed linear space S (i.e., there is $c < \infty$ s.t. $\|\mathbf{A}x\| := \sup_i \sum_j a_{i,j} \leq c\|x\|$ for all $x \in S$ with spectral radius $\sigma(\mathbf{A}) \triangleq \lim_{n \rightarrow \infty} \|\mathbf{A}^n\|^{1/n} < 1$), \mathbf{A}^{-1} exists spectral radius and is given by*

$$\mathbf{A}^{-1} = \lim_{T \rightarrow \infty} \sum_{n=0}^T (\mathbf{I} - \mathbf{A})^n. \quad (6.5.4)$$

Proof of Theorem 6.5.1. First note that by manipulating the infinite sum in Remark 6.5.1, one obtains $\mathbf{r} = (\mathbf{I} - \gamma \mathbf{P}_{\mu, \pi}) \mathbf{v}^\pi$. Since $\|\gamma \mathbf{P}_{\mu, \pi}\| < 1 \cdot \|\mathbf{P}_{\mu, \pi}\| = 1$, the inverse

$$(\mathbf{I} - \gamma \mathbf{P}_{\mu, \pi})^{-1} = \lim_{n \rightarrow \infty} \sum_{t=0}^n (\gamma \mathbf{P}_{\mu, \pi})^t$$

exists by Theorem 6.5.2. It follows that

$$\mathbf{v} = (\mathbf{I} - \gamma \mathbf{P}_{\mu, \pi})^{-1} \mathbf{r} = \sum_{t=0}^{\infty} \gamma^t \mathbf{P}_{\mu, \pi}^t \mathbf{r} = \mathbf{v}^\pi,$$

where the last step is by Remark 6.5.1 again. \square

It is important to note that the matrix $\mathbf{X} = (\mathbf{I} - \gamma \mathbf{P}_{\mu, \pi})^{-1}$ can be seen as the expected number of discounted cumulative visits to each state s , starting from state s' and following policy π . More specifically, the entries of the matrix are:

$$x(s, s') = \mathbb{E}_\mu^\pi \left\{ \sum_{t=0}^{\infty} \gamma^t \mathbb{P}_\mu^\pi(s_t = s' \mid s_t = s) \right\}. \quad (6.5.5)$$

This interpretation is quite useful, as many algorithms rely on an estimation of \mathbf{X} for approximating value functions.

6.5.3 Optimality equations

Let us now look at the backwards induction algorithms in terms of operators. We introduce the operator of a policy, which is the one-step backwards induction operation for a fixed policy, and the Bellman operator, which is the equivalent operator for the optimal policy. If a value function is optimal, then it satisfies the Bellman optimality equation.

Definition 6.5.2 (Policy and Bellman operator). The linear operator of a policy π is:

$$\mathcal{L}_\pi \mathbf{v} \triangleq \mathbf{r} + \gamma \mathbf{P}_\pi \mathbf{v} \quad (6.5.6)$$

Sby contract The (non-linear) Bellman operator in the space of value functions \mathcal{V} is defined as:

$$\mathcal{L} \mathbf{v} \triangleq \sup_\pi \{ \mathbf{r} + \gamma \mathbf{P}_\pi \mathbf{v} \}, \quad \mathbf{v} \in \mathcal{V} \quad (6.5.7)$$

We now show that the Bellman operator satisfies the following monotonicity properties with respect to an arbitrary value vector \mathbf{v} .

Theorem 6.5.3. *Let $\mathbf{v}^* \triangleq \sup_\pi \mathbf{v}^\pi$. Then for any bounded \mathbf{r} , it holds that for $\mathbf{v} \in \mathcal{V}$:*

- (1) *If $\mathbf{v} \geq \mathcal{L}\mathbf{v}$, then $\mathbf{v} \geq \mathbf{v}^*$.*
- (2) *If $\mathbf{v} \leq \mathcal{L}\mathbf{v}$, then $\mathbf{v} \leq \mathbf{v}^*$.*
- (3) *If $\mathbf{v} = \mathcal{L}\mathbf{v}$, then \mathbf{v} is unique and $\mathbf{v} = \sup_\pi \mathbf{v}^\pi$. Therefore, $\mathbf{v} = \mathcal{L}\mathbf{v}$ is called the Bellman optimality equation.*

Proof. We first prove (1). A simple proof by induction over n shows that for any π

$$\mathbf{v} \geq \mathbf{r} + \gamma \mathbf{P}_\pi \mathbf{v} \geq \sum_{k=0}^{n-1} \gamma^k \mathbf{P}_\pi^k \mathbf{r} + \gamma^n \mathbf{P}_\pi^n \mathbf{v}.$$

Since $\mathbf{v}^\pi = \sum_{t=0}^{\infty} \gamma^t \mathbf{P}_\pi^t \mathbf{r}$ it follows that

$$\mathbf{v} - \mathbf{v}^\pi \geq \gamma^n \mathbf{P}_\pi^n \mathbf{v} - \sum_{k=n}^{\infty} \gamma^k \mathbf{P}_\pi^k \mathbf{r}.$$

The first-term on the right-hand side can be bounded by arbitrary $\epsilon/2$ for large enough n . Also note that

$$\sum_{k=n}^{\infty} \gamma^k \mathbf{P}_{\pi}^k \mathbf{r} \geq -\frac{\gamma^n \mathbf{e}}{1-\gamma},$$

with \mathbf{e} being a unit vector, so this can be bounded by $\epsilon/2$ as well. So for any $\pi, \epsilon > 0$:

$$\mathbf{v} \geq \mathbf{v}^{\pi} - \epsilon,$$

so

$$\mathbf{v} \geq \sup_{\pi} \mathbf{v}^{\pi}.$$

An equivalent argument shows that

$$\mathbf{v} \leq \mathbf{v}^{\pi} + \epsilon,$$

proving (2). Putting together (1) and (2) gives (3). \square

We eventually want show that repeated application of the Bellman operator converges to the optimal value. As a preparation, we need the following theorem.

Theorem 6.5.4 (Banach Fixed-Point theorem). *Suppose \mathcal{S} is a Banach space (i.e. a complete normed linear space) and $T : \mathcal{S} \rightarrow \mathcal{S}$ is a contraction mapping (i.e. $\exists \gamma \in [0, 1)$ s.t. $\|Tu - Tv\| \leq \gamma \|u - v\|$ for all $u, v \in \mathcal{S}$). Then*

- there is a unique $u^* \in U$ s.t. $Tu^* = u^*$, and
- for any $u^0 \in \mathcal{S}$ the sequence $\{u^n\}$:

$$u^{n+1} = Tu^n = T^{n+1}u^0$$

converges to u^* .

Proof. For any $m \geq 1$

$$\begin{aligned} \|u^{n+m} - u^n\| &\leq \sum_{k=0}^{m-1} \|u^{n+k+1} - u^{n+k}\| = \sum_{k=0}^{m-1} \|T^{n+k}u^1 - T^{n+k}u^0\| \\ &\leq \sum_{k=0}^{m-1} \gamma^{n+k} \|u^1 - u^0\| = \frac{\gamma^n(1-\gamma^m)}{1-\gamma} \|u^1 - u^0\|. \end{aligned}$$

\square

Theorem 6.5.5. *For $\gamma \in [0, 1)$ the Bellman operator \mathcal{L} is a contraction mapping in \mathcal{V} .*

Proof. Let $\mathbf{v}, \mathbf{v}' \in \mathcal{V}$. Consider $s \in \mathcal{S}$ such that $\mathcal{L}\mathbf{v}(s) \geq \mathcal{L}\mathbf{v}'(s)$, and let

$$a_s^* \in \arg \max_{a \in \mathcal{A}} \left\{ r(s) + \sum_{j \in \mathcal{S}} \gamma p_{\mu}(j | s, a) \mathbf{v}(j) \right\}.$$

Using the fact that a_s^* is optimal for \mathbf{v} , but not necessarily for \mathbf{v}' , we have:

$$\begin{aligned} 0 &\leq \mathcal{L}\mathbf{v}(s) - \mathcal{L}\mathbf{v}'(s) \leq \sum_{j \in S} \gamma p(j | s, a_s^*) \mathbf{v}(j) - \sum_{j \in S} \gamma p(j | s, a_s^*) \mathbf{v}'(j) \\ &= \gamma \sum_{j \in S} p(j | s, a_s^*) [\mathbf{v}(j) - \mathbf{v}'(j)] \\ &\leq \gamma \sum_{j \in S} p(j | s, a_s^*) \|\mathbf{v} - \mathbf{v}'\| = \gamma \|\mathbf{v} - \mathbf{v}'\|. \end{aligned}$$

Repeating the argument for s such that $\mathcal{L}\mathbf{v}(s) \leq \mathcal{L}\mathbf{v}'(s)$, we obtain

$$|\mathcal{L}\mathbf{v}(s) - \mathcal{L}\mathbf{v}'(s)| \leq \gamma \|\mathbf{v} - \mathbf{v}'\|.$$

Taking the supremum over all possible s , the required result follows. \square

It is easy to show the same result for the \mathcal{L}_π operator, as a corollary to this theorem.

Theorem 6.5.6. *For discrete \mathcal{S} , bounded \mathbf{r} , and $\gamma \in [0, 1)$*

- (i) *there is a unique $\mathbf{v}^* \in \mathcal{V}$ such that $\mathcal{L}\mathbf{v}^* = \mathbf{v}^*$ and such that $\mathbf{v}^* = V_\mu^*$,*
- (ii) *for any stationary policy π , there is a unique $\mathbf{v} \in \mathcal{V}$ such that $\mathcal{L}_\pi \mathbf{v} = \mathbf{v}$ and $\mathbf{v} = V_\mu^\pi$.*

Proof. As the Bellman operator \mathcal{L} is a contraction by Theorem 6.5.5, application of the fixed-point Theorem 6.5.4 shows that there is a unique $\mathbf{v}^* \in \mathcal{V}$ such that $\mathcal{L}\mathbf{v}^* = \mathbf{v}^*$. This is also the optimal value function due to Theorem 6.5.5. The second part of the theorem follows from the first part when considering only a single policy π (which then is optimal). \square

6.5.4 MDP Algorithms

Let us now look at three basic algorithms for solving a known Markov decision process. The first, *value iteration*, is a simple extension of the backwards induction algorithm to the infinite horizon case.

Value iteration

In this version of the algorithm, we assume that rewards are dependent only on the state. An algorithm for the case where reward only depends on the state can be obtained by replacing $r(s, a)$ with $r(s)$.

Algorithm 6 Value iteration

```

Input  $\mu, \mathcal{S}$ .
Initialise  $\mathbf{v}_0 \in \mathcal{V}$ .
for  $n = 1, 2, \dots$  do
    for  $s \in \mathcal{S}_n$  do
         $\pi_n(s) = \arg \max_{a \in \mathcal{A}} \{r(s, a) + \gamma \sum_{s' \in \mathcal{S}} P_\mu(s' | s, a) \mathbf{v}_{n-1}(s')\}$ 
         $\mathbf{v}_n(s) = r(s, \pi_n(s)) + \gamma \sum_{s' \in \mathcal{S}} P_\mu(s' | s, \pi_n(s)) \mathbf{v}_{n-1}(s')$ 
    end for
    break if termination-condition is met
end for
Return  $\pi_n, V_n$ .

```

The value iteration algorithm is a direct extension of the backwards induction algorithm for an infinite horizon. However, since we know that stationary policies are optimal, we do not need to maintain the values and actions for all time steps. At each step, we can merely keep the previous value \mathbf{v}_{n-1} . However, since there is an infinite number of steps, we need to know whether the algorithm converges to the optimal value, and what is the error we make at a particular iteration.

Theorem 6.5.7. *The value iteration algorithm satisfies*

- $\lim_{n \rightarrow \infty} \|\mathbf{v}_n - \mathbf{v}^*\| = 0$.
- For each $\epsilon > 0$ there exists $N_\epsilon < \infty$ such that for all $n \geq N_\epsilon$

$$\|\mathbf{v}_{n+1} - \mathbf{v}_n\| \leq \epsilon(1 - \gamma)/2\gamma. \quad (6.5.8)$$

- For $n \geq N_\epsilon$ the policy π_ϵ that takes action

$$\arg \max_a r(s, a) + \gamma \sum_j p(j|s, a) \mathbf{v}_n(s')$$

is ϵ -optimal, i.e. $V_\mu^{\pi_\epsilon}(s) \geq V_\mu^*(s) - \epsilon$ for all states s .

- $\|\mathbf{v}_{n+1} - \mathbf{v}^*\| < \epsilon/2$ for $n \geq N_\epsilon$.

Proof. The first two statements follow from the fixed-point Theorem 6.5.4. Now note that

$$\|V_\mu^{\pi_\epsilon} - \mathbf{v}^*\| \leq \|V_\mu^{\pi_\epsilon} - \mathbf{v}_n\| + \|\mathbf{v}_n - \mathbf{v}^*\|$$

We can bound these two terms easily:

$$\begin{aligned} \|V_\mu^{\pi_\epsilon} - \mathbf{v}_{n+1}\| &= \|\mathcal{L}_{\pi_\epsilon} V_\mu^{\pi_\epsilon} - \mathbf{v}_{n+1}\| && \text{(by definition of } \mathcal{L}_{\pi_\epsilon} \text{)} \\ &\leq \|\mathcal{L}_{\pi_\epsilon} V_\mu^{\pi_\epsilon} - \mathcal{L} \mathbf{v}_{n+1}\| + \|\mathcal{L} \mathbf{v}_{n+1} - \mathbf{v}_{n+1}\| && \text{(triangle)} \\ &= \|\mathcal{L}_{\pi_\epsilon} V_\mu^{\pi_\epsilon} - \mathcal{L}_{\pi_\epsilon} \mathbf{v}_{n+1}\| + \|\mathcal{L} \mathbf{v}_{n+1} - \mathcal{L} \mathbf{v}_n\| && \text{(by definition)} \\ &\leq \gamma \|V_\mu^{\pi_\epsilon} - \mathbf{v}_{n+1}\| + \gamma \|\mathbf{v}_{n+1} - \mathbf{v}_n\|. && \text{(by contraction)} \end{aligned}$$

An analogous argument gives the same bound for the second term $\|\mathbf{v}_n - \mathbf{v}^*\|$. Then, rearranging we obtain

$$\|V_\mu^{\pi_\epsilon} - \mathbf{v}_{n+1}\| \leq \frac{\gamma}{1 - \gamma} \|\mathbf{v}_{n+1} - \mathbf{v}_n\|, \quad \|\mathbf{v}_{n+1} - \mathbf{v}^*\| \leq \frac{\gamma}{1 - \gamma} \|\mathbf{v}_{n+1} - \mathbf{v}_n\|,$$

and the third and fourth statements follow from the second statement. \square

The *termination condition* of value iteration has been left unspecified. However, the theorem above shows that if we terminate when (6.5.8) is true, then our error will be bounded by ϵ . However, better termination conditions can be obtained.

Now let us prove how fast value iteration converges.

Theorem 6.5.8 (Value iteration monotonicity). *Let \mathcal{V} be the set of value vectors with Bellman operator \mathcal{L} . Then:*

1. Let $\mathbf{v}, \mathbf{v}' \in \mathcal{V}$ with $\mathbf{v}' \geq \mathbf{v}$. Then $\mathcal{L} \mathbf{v}' \geq \mathcal{L} \mathbf{v}$.

termination condition

2. Let $\mathbf{v}_{n+1} = \mathcal{L}\mathbf{v}_n$. If there is an N s.t. $\mathcal{L}\mathbf{v}_N \leq \mathbf{v}_N$, then $\mathcal{L}\mathbf{v}_{N+k} \leq \mathbf{v}_{N+k}$ for all $k \geq 0$ and similarly for \geq .

Proof. Let $\pi \in \arg \max_{\pi} \mathbf{r} + \gamma \mathbf{P}_{\mu, \pi} \mathbf{v}$. Then

$$\mathcal{L}\mathbf{v} = \mathbf{r} + \gamma \mathbf{P}_{\mu, \pi} \mathbf{v} \leq \mathbf{r} + \gamma \mathbf{P}_{\mu, \pi} \mathbf{v}' \leq \max_{\pi'} \mathbf{r} + \gamma \mathbf{P}_{\mu, \pi'} \mathbf{v}',$$

where the first inequality is due to the fact that $\mathbf{P}\mathbf{v} \geq \mathbf{P}\mathbf{v}'$ for any \mathbf{P} . For the second part,

$$\mathcal{L}\mathbf{v}_{N+k} = \mathbf{v}_{N+k+1} = \mathcal{L}^k \mathcal{L}\mathbf{v}_N \leq \mathcal{L}^k \mathbf{v}_N = \mathbf{v}_{N+k}.$$

since $\mathcal{L}\mathbf{v}_N \leq \mathbf{v}_N$ by assumption and consequently $\mathcal{L}^k \mathcal{L}\mathbf{v}_N \leq \mathcal{L}^k \mathbf{v}_N$ by part one of the theorem. \square

Thus, value iteration converges monotonically to V_{μ}^* if the initial value $\mathbf{v}_0 \leq \mathbf{v}'$ for all \mathbf{v}' . If $r \geq 0$, it is sufficient to set $\mathbf{v}_0 = \mathbf{0}$. Then \mathbf{v}_n is always a lower bound on the optimal value function.

Theorem 6.5.9. *Value iteration converges with error in $O(\gamma^n)$. More specifically, for $r \in [0, 1]$ and $\mathbf{v}_0 = \mathbf{0}$,*

$$\|\mathbf{v}_n - V_{\mu}^*\| \leq \frac{\gamma^n}{1 - \gamma}, \quad \|V_{\mu}^{\pi_n} - V_{\mu}^*\| \leq \frac{2\gamma^n}{1 - \gamma}.$$

Proof. The first part follows from the contraction property (Theorem 6.5.5):

$$\|\mathbf{v}_{n+1} - \mathbf{v}^*\| = \|\mathcal{L}\mathbf{v}_n - \mathcal{L}\mathbf{v}^*\| \leq \gamma \|\mathbf{v}_n - \mathbf{v}^*\|. \quad (6.5.9)$$

Now divide by γ^n to obtain the final result. \square

Although value iteration converges exponentially fast, the convergence is dominated by the discount factor γ . When γ is very close to one, convergence can be extremely slow. In fact, Tseng [1990] showed that the number of iterations are on the order of $1/(1-\gamma)$, for bounded accuracy of the input data. The overall complexity is $\tilde{O}(|\mathcal{S}|^2 |\mathcal{A}| L(1-\gamma)^{-1})$, omitting logarithmic factors, where L is the total number of bits used to represent the input.¹

Policy iteration

Unlike value iteration, *policy iteration* attempts to iteratively improve a given policy, rather than a value function. At each iteration, it calculates the value of the current policy and then calculates the policy that is greedy with respect to this value function. For finite MDPs, the policy evaluation step can be performed with either linear algebra or backwards induction, while the policy improvement step is trivial. The algorithm described below can be extended to the case when the reward also depends on the action, by replacing \mathbf{r} with the policy-dependent reward vector \mathbf{r}_{π} .

¹Thus the result is *weakly* polynomial complexity, due to the dependence on the input size description.

Algorithm 7 Policy iteration

```

Input  $\mu, \mathcal{S}$ .
Initialise  $\mathbf{v}_0$ .
for  $n = 1, 2, \dots$  do
     $\pi_{n+1} = \arg \max_{\pi} \{ \mathbf{r} + \gamma \mathbf{P}_{\pi} \mathbf{v}_n \}$       // policy improvement
     $\mathbf{v}_{n+1} = V_{\mu}^{\pi_{n+1}}$           // policy evaluation
    break if  $\pi_{n+1} = \pi_n$ .
end for
Return  $\pi_n, \mathbf{v}_n$ .
```

The following theorem describes an important property of policy iteration, namely that the policies generated are monotonically improving.

Theorem 6.5.10. *Let $\mathbf{v}_n, \mathbf{v}_{n+1}$ be the value vectors generated by policy iteration. Then $\mathbf{v}_n \leq \mathbf{v}_{n+1}$.*

Proof. From the policy improvement step

$$\mathbf{r} + \gamma \mathbf{P}_{\pi_{n+1}} \mathbf{v}_n \geq \mathbf{r} + \gamma \mathbf{P}_{\pi_n} \mathbf{v}_n = \mathbf{v}_n$$

where the equality is due to the policy evaluation step for π_n . Rearranging, we get $\mathbf{r} \geq (\mathbf{I} - \gamma \mathbf{P}_{\pi_{n+1}}) \mathbf{v}_n$ and hence

$$(\mathbf{I} - \gamma \mathbf{P}_{\pi_{n+1}})^{-1} \mathbf{r} \geq \mathbf{v}_n,$$

noting that the inverse is positive. Since the left side equals \mathbf{v}_{n+1} by the policy evaluation step for π_{n+1} , the theorem follows. \square

We can use the fact that the policies are monotonically improving to show that policy iteration will terminate after a finite number of steps.

Corollary 6.5.1. *If \mathcal{S}, \mathcal{A} are finite, then policy iteration terminates after a finite number of iterations and returns an optimal policy.*

Proof. There is only a finite number of policies, and since policies in policy iteration are monotonically improving, the algorithm must stop after finitely many iterations. Finally, the last iteration satisfies

$$\mathbf{v}_n = \max_{\pi} \mathbf{r} + \gamma \mathbf{P}_{\pi} \mathbf{v}_n. \quad (6.5.10)$$

Thus \mathbf{v}_n solves the optimality equation. \square

However, it is easy to see that the number of policies is $|\mathcal{A}|^{|\mathcal{S}|}$, thus the above corollary only guarantees exponential-time convergence in the number of states. However, it is also known that the complexity of policy iteration is strongly polynomial Ye [2011], for any fixed γ , with the number of iterations required being $\frac{|\mathcal{S}|^2(|\mathcal{A}|-1)}{1-\gamma} \cdot \ln \left(\frac{|\mathcal{S}|^2}{1-\gamma} \right)$.

Policy iteration seems to have very different behaviour from value iteration. In fact, one can obtain families of algorithms that lie at the extreme ends of the spectrum between policy iteration and value iteration. The first member of this family is modified policy iteration, and the second member is temporal difference policy iteration.

Modified policy iteration

The astute reader will have noticed that it may be not necessary to fully evaluate the improved policy. In fact, we can take advantage of that to speed up policy iteration. Thus, a simple variant of policy iteration involves doing only a k -step update for the policy evaluation step. For $k = 1$, the algorithm becomes identical to value iteration, while for $k \rightarrow \infty$ the algorithm is equivalent to policy iteration, as $\mathbf{v}_n = V^{\pi_n}$.

Algorithm 8 Modified policy iteration

```

Input  $\mu, \mathcal{S}$ .
Initialise  $\mathbf{v}_0$ .
for  $n = 1, 2, \dots$  do
     $\pi_n = \arg \max_{\pi} \mathbf{r} + \gamma \mathbf{P}_{\pi} \mathbf{v}_{n-1}$       // policy improvement
     $\mathbf{v}_n = \mathcal{L}_{\pi_n}^k \mathbf{v}_{n-1}$           // partial policy evaluation
    break if  $\pi_n = \pi_{n+1}$ .
end for
Return  $\pi_n, \mathbf{v}_n$ .

```

Modified policy iteration can perform much better than either pure value iteration or pure policy iteration.

A geometric view

It is perhaps interesting to see the problem from a geometric perspective. This also gives rise to the so-called “temporal-difference” set of algorithms. First, we define the difference operator, which is the difference between a value function vector \mathbf{v} and its transformation via the Bellman operator.

difference operator

Definition 6.5.3. The *difference operator* is defined as

$$\mathcal{B}\mathbf{v} \triangleq \max_{\pi} \{ \mathbf{r} + (\gamma \mathbf{P}_{\pi} - \mathbf{I})\mathbf{v} \} = \mathcal{L}\mathbf{v} - \mathbf{v}. \quad (6.5.11)$$

Essentially, it is the change in the value function vector when we apply the Bellman operator. Thus the Bellman optimality equation can be rewritten as

$$\mathcal{B}\mathbf{v} = \mathbf{0}. \quad (6.5.12)$$

Now let us define the set of greedy policies with respect to a value vector $\mathbf{v} \in \mathcal{V}$ to be:

$$\Pi_{\mathbf{v}} \triangleq \arg \max_{\pi \in \Pi} \{ \mathbf{r} + (\gamma \mathbf{P}_{\pi} - \mathbf{I})\mathbf{v} \}.$$

We can now show the following inequality between the two different value function vectors.

Theorem 6.5.11. For any $\mathbf{v}, \mathbf{v}' \in \mathcal{V}$ and $\pi \in \Pi_{\mathbf{v}}$

$$\mathcal{B}\mathbf{v}' \geq \mathcal{B}\mathbf{v} + (\gamma \mathbf{P}_{\pi} - \mathbf{I})(\mathbf{v}' - \mathbf{v}). \quad (6.5.13)$$

Proof. By definition, $\mathcal{B}\mathbf{v}' \geq \mathbf{r} + (\gamma \mathbf{P}_{\pi} - \mathbf{I})\mathbf{v}'$, while $\mathcal{B}\mathbf{v} = \mathbf{r} + (\gamma \mathbf{P}_{\pi} - \mathbf{I})\mathbf{v}$. Subtracting the latter from the former gives the result. \square

Equation (6.5.13) is similar to the convexity of the Bayes-optimal utility (3.3.6). Geometrically, we can see from a look at Figure 6.5, that applying the Bellman operator on value function always improves it, yet may have a negative effect on the other value function. If the number of policies is finite, then the figure is also a good illustration of the policy iteration algorithm, where each value function improvement results in a new point on the horizontal axis, and the choice of the best improvement (highest line) for that point. In fact, we can write the policy iteration algorithm in terms of the difference operator.

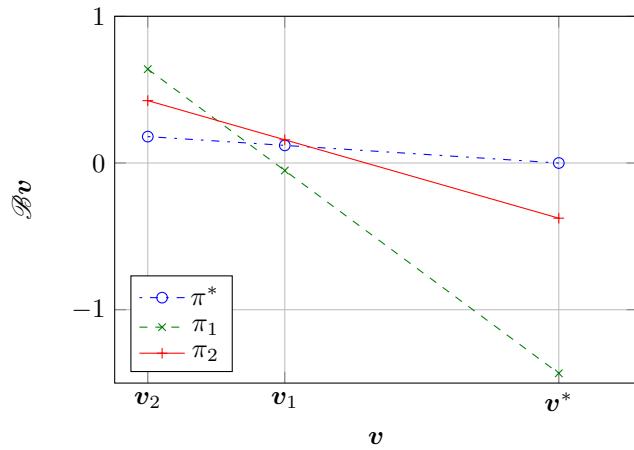


Figure 6.5: The difference operator. The graph shows the effect of the operator for the optimal value function v^* , and two arbitrary value functions, v_1, v_2 . Each line is the improvement effected by the greedy policy π^*, π_1, π_2 with respect to each value function v^*, v_1, v_2 .

Theorem 6.5.12. Let $\{\mathbf{v}_n\}$ be the sequence of value vectors obtained from policy iteration. Then for any $\pi \in \Pi_{\mathbf{v}_n}$,

$$\mathbf{v}_{n+1} = \mathbf{v}_n - (\gamma \mathbf{P}_\pi - \mathbf{I})^{-1} \mathcal{B}\mathbf{v}_n. \quad (6.5.14)$$

Proof. By definition, we have for $\pi \in \Pi_{\mathbf{v}_n}$

$$\begin{aligned} \mathbf{v}_{n+1} &= (\mathbf{I} - \gamma \mathbf{P}_\pi)^{-1} \mathbf{r} - \mathbf{v}_n + \mathbf{v}_n \\ &= (\mathbf{I} - \gamma \mathbf{P}_\pi)^{-1} [\mathbf{r} - (\mathbf{I} - \gamma \mathbf{P}_\pi) \mathbf{v}_n] + \mathbf{v}_n. \end{aligned}$$

Since $\mathbf{r} - (\mathbf{I} - \gamma \mathbf{P}_\pi) \mathbf{v}_n = \mathcal{B}\mathbf{v}_n$ the claim follows. \square

Temporal-Difference Policy Iteration

In *temporal-difference policy iteration*, similarly to the modified policy iteration algorithm, we replace the next-step value with an approximation \mathbf{v}_n of the n -th policy's value. Informally, this approximation is chosen so as to reduce the discrepancy of our value function over time.

At the n -th iteration of the algorithm, we use a policy improvement step to obtain the next policy π_{n+1} given our current approximation \mathbf{v}_n :

$$\mathcal{L}_{\pi_{n+1}} \mathbf{v}_n = \mathcal{L}\mathbf{v}_n. \quad (6.5.15)$$

To update the value from \mathbf{v}_n to \mathbf{v}_{n+1} we rely on the *temporal difference error*, *temporal difference error* defined as:

$$d_n(i, j) = [\mathbf{r}(i) + \gamma \mathbf{v}_n(j)] - \mathbf{v}_n(i). \quad (6.5.16)$$

This can be seen as the difference in the estimate when we move from state i to state j . In fact, it is easy to see that, if our value function estimate satisfies $\mathbf{v} = V^{\pi_n}$, then the expected error should be zero, as:

$$\sum_{j \in \mathcal{S}} d_n(i, j) p(j | i, \pi_n(i)) = \sum_{j \in \mathcal{S}} [\mathbf{r}(i) + \gamma \mathbf{v}_n(j)] p(j | i, \pi_n(i)) - \mathbf{v}_n(i).$$

Note the similarity to the difference operator in modified policy iteration. The idea of the temporal-difference policy iteration is to use adjust the current value \mathbf{v}_n , using the temporal differences mixed over an infinite number of steps:

$$\boldsymbol{\tau}_n(i) = \sum_{t=0}^{\infty} \mathbb{E}_{\pi_n} [(\gamma \lambda)^t d_n(s_t, s_{t+1}) | s_0 = i], \quad (6.5.17)$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \boldsymbol{\tau}_n. \quad (6.5.18)$$

Here the λ parameter is a simple way to mix together the different temporal difference errors. If $\lambda \rightarrow 1$, our error will be dominated by the terms far in the future, while if $\lambda \rightarrow 0$, our error $\boldsymbol{\tau}_n$, will be dominated by the short-term discrepancies in our value function. In the end, we shall adjust our value function in the direction of this error.

Putting all of those steps together, we obtain the following algorithm:

Algorithm 9 Temporal-Difference Policy Iteration

```

Input  $\mu, \mathcal{S}, \lambda$ .
Initialise  $\mathbf{v}_0$ .
for  $n = 0, 1, 2, \dots$  do
     $\pi_{n+1} = \arg \max_{\pi} \mathbf{r} + \gamma \mathbf{P}_{\pi} \mathbf{v}_n$  // policy improvement
     $\mathbf{v}_{n+1} = \mathbf{v}_n + \boldsymbol{\tau}_n$  // temporal difference update.
    break if  $\pi_{n+1} = \pi_n$ .
end for
Return  $\pi_n, \mathbf{v}_n$ .
```

In fact, \mathbf{v}_{n+1} is the unique fixed point of the following equation:

$$\mathcal{D}_n \mathbf{v} \triangleq (1 - \lambda) \mathcal{L}_{\pi_{n+1}} \mathbf{v}_n + \lambda \mathcal{L}_{\pi_{n+1}} \mathbf{v}. \quad (6.5.19)$$

That is, if we repeatedly apply the above operator to some vector \mathbf{v} , then at some point we shall obtain a fixed point $\mathbf{v}^* = \mathcal{D}_n \mathbf{v}^*$. It is interesting to see what happens at the two extreme choices of λ in this case. For $\lambda = 1$, this becomes identical to standard policy iteration, as the fixed point satisfies $\mathbf{v}^* = \mathcal{L}_{\pi_{n+1}} \mathbf{v}^*$, so then \mathbf{v}^* must be the value of policy π_{n+1} . For $\lambda = 0$, one obtains standard value iteration, as the fixed point is reached under one step and is simply $\mathbf{v}^* = \mathcal{L}_{\pi_{n+1}} \mathbf{v}_n$, i.e. the approximate value of the one-step greedy policy. In other words, the new value vector is moved only partially towards the direction of the Bellman update, depending on how we choose λ .

Linear programming

Perhaps surprisingly, we can also solve Markov decision processes through linear programming. The main idea is to reformulate the maximisation problem as a linear optimisation problem with linear constraints. The first step in our procedure is to recall that there is an easy way to determine whether a particular \mathbf{v} is an upper bound on the optimal value function \mathbf{v}^* , since if

$$\mathbf{v} \geq \mathcal{L}\mathbf{v}$$

then $\mathbf{v} \geq \mathbf{v}^*$. In order to transform this into a linear program, we must first define a scalar function to minimise. We can do this by selecting some arbitrary distribution on the states $\mathbf{y} \in \Delta^{|\mathcal{S}|}$. Then we can write the following linear program.

Primal linear program

$$\min_{\mathbf{v}} \mathbf{y}^\top \mathbf{v},$$

such that

$$\mathbf{v}(s) - \gamma \mathbf{p}_{s,a}^\top \mathbf{v} \geq r(s, a), \quad \forall a \in \mathcal{A}, s \in \mathcal{S},$$

where we use $\mathbf{p}_{s,a}$ to denote the vector of next state probabilities $p(j | s, a)$.

Note that the inequality condition is equivalent to $\mathbf{v} \geq \mathcal{L}\mathbf{v}$. Consequently, the problem is to find the smallest \mathbf{v} that satisfies this inequality. When \mathcal{A}, \mathcal{S} are finite, it is easy to see that this will be the optimal value function and the Bellman equation is satisfied.

It also pays to look at the dual linear program, which is in terms of a maximisation. This time, instead of finding the minimal upper bound on the value function, we find the maximal cumulative discounted state-action visits $x(s, a)$ that are consistent with the transition kernel of the process.

Dual linear program

$$\max_x \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} r(s, a) x(s, a)$$

such that $x \in \mathbb{R}_+^{|\mathcal{S} \times \mathcal{A}|}$ and

$$\sum_{a \in \mathcal{A}} x(j, a) - \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \gamma p(j | s, a) x(s, a) = y(j) \quad \forall j \in \mathcal{S}.$$

with $\mathbf{y} \in \Delta^{|\mathcal{S}|}$.

In this case, x can be interpreted as the discounted sum of state-action visits, as proved by the following theorem.

Theorem 6.5.13. *For any policy π ,*

$$x_\pi(s, a) = \mathbb{E}_{\pi, \mu} \left\{ \sum \gamma^n \mathbb{I}\{s_t = s, a_t = a \mid s_0 \sim y\} \right\}$$

is a feasible solution to the dual problem. On the other hand, if x is a feasible solution to the dual problem then $\sum_a x(s, a) > 0$. Finally, if we define the strategy

$$\pi(a | s) = \frac{x(s, a)}{\sum_{a' \in \mathcal{A}} x(s, a')}$$

then $x_\pi = x$ is a feasible solution.

The equality condition ensures that x is consistent with the transition kernel of the Markov decision process. Consequently, the program can be seen as search among all possible cumulative state-action distributions to find the one giving the highest total reward.

6.6 Summary

Markov decision processes can represent shortest path problems, stopping problems, experiment design problems, multi-armed bandit problems and reinforcement learning problems.

Bandit problems are the simplest type of Markov decision process, since they have a fixed, never-changing state. However, to solve them, one can construct a Markov decision processes in belief space, within a Bayesian framework. It is then possible to apply backwards induction to find the optimal policy.

Backwards induction is applicable more generally to arbitrary Markov decision processes. For the case of infinite-horizon problems, it is referred to as value iteration, as it converges to a fixed point. It is tractable when either the state space \mathcal{S} or the horizon T are small (finite).

When the horizon is infinite, policy iteration can also be used to find optimal policies. It is different from value iteration in that at every step, it fully evaluates a policy before the improvement step, while value iteration only performs a partial evaluation. In fact, at the n -th iteration, value iteration has calculated the value of an n -step policy.

We can arbitrarily mix between the two extremes of policy iteration and value iteration in two ways. Firstly, we can perform a k -step partial evaluation. When $k = 1$, we obtain value iteration, and when $k \rightarrow \infty$, we obtain policy iteration. The generalised algorithm is called modified policy iteration. Secondly, we can perform adjust our value function by using a temporal difference error of values in future time steps. Again, we can mix liberally between policy iteration and value iteration by focusing on errors far in the future (policy iteration) or on short-term errors (value iteration).

Finally, it is possible to solve MDPs through linear programming. This is done by reformulating the problem as a linear optimisation with constraints. In the primal formulation, we attempt to find a minimal upper bound on the optimal value function. In the dual formulation, our goal is to find a distribution on state-action visitations that maximises expected utility and is consistent with the MDP model.

6.7 Further reading

See the last chapter of [DeGroot, 1970] for further information on the MDP formulation of bandit problems in the decision theoretic setting. This was ex-

plored in more detail in Duff's PhD thesis [Duff, 2002]. When the number of (information) states in the bandit problem is finite, Gittins [1989] has proven that it is possible to formulate simple index policies. However, this is not generally applicable. Easily computable, near-optimal heuristic strategies for bandit problems will be given in Chapter 10. The decision-theoretic solution to the unknown MDP problem will be given in Chapter 9.

Further theoretical background on Markov decision processes, including many of the theorems in Section 6.5, can be found in [Puterman, 1994]. Chapter 2 of Bertsekas and Tsitsiklis [1996] gives a quick overview of MDP theory from the operator perspective. The introductory reinforcement learning book of Sutton and Barto [1998] also explains the basic Markov decision process framework.

6.8 Exercises

6.8.1 Medical diagnosis

EXERCISE 22 (Continuation of exercise ??). Now consider the case where you have the choice between tests to perform First, you observe S , whether or not the patient is a smoker. Then, you select a test to make: $d_1 \in \{\text{X-ray, ECG}\}$. Finally, you decide whether or not to treat for ASC: $d_2 \in \{\text{heart treatment, no treatment}\}$. An untreated ASC patient may die with probability 2%, while a treated one with probability 0.2%. Treating a non-ASC patient result in death with probability 0.1%.

1. Draw a decision diagram, where:
 - S is an observed random variable taking values in $\{0, 1\}$.
 - A is an hidden variable taking values in $\{0, 1\}$.
 - C is an hidden variable taking values in $\{0, 1\}$.
 - d_1 is a choice variable, taking values in $\{\text{X-ray, ECG}\}$.
 - r_1 is a result variable, taking values in $\{0, 1\}$, corresponding to negative and positive tests results.
 - d_2 is a choice variable, which depends on the test results, d_1 and on S .
 - r_2 is a result variable, taking values in $\{0, 1\}$ corresponding to the patient dying (0), or living (1).
2. Let $d_1 = \text{X-ray}$, and assume the patient suffers from ACS, i.e. $A = 1$. How is the posterior distributed?
3. What is the optimal decision rule for this problem?

6.8.2 Markov Decision Process theory

EXERCISE 23 (30). Show that the expected discounted total reward of any given policy is equal to the expected undiscounted total reward with a finite, but random horizon T . In particular, let T be distributed according to a geometric distribution on $\{1, 2, \dots\}$ with parameter $1 - \gamma$. Then show that:

$$\mathbb{E} \lim_{T \rightarrow \infty} \sum_{k=0}^T \gamma^k r_k = \mathbb{E} \left(\sum_{k=0}^T r_k \mid T \sim \text{Geom}(1 - \gamma) \right).$$

6.8.3 Automatic algorithm selection

Consider the problem of selecting algorithms for finding solutions to a sequence of problems. Assume you have n algorithms to choose from. At time t , you get a task and choose the i -th algorithm. Assume that the algorithms are randomised, so that the i -th algorithm will find a solution with some unknown probability. Our aim is to maximise the expected total number of solutions found. Consider the following specific cases of this problem:

EXERCISE 24 (120). In this case, we assume that the probability that the i -th algorithm successfully solves the t -th task is always p_i . Furthermore, tasks are in no way distinguishable from each other. In each case, assume that $p_i \in \{0.1, \dots, 0.9\}$ and a prior distribution $\xi_i(p_i) = 1/9$ for all i , with a complete belief $\xi(\mathbf{p}) = \prod_i \xi_i(p_i)$, and formulate the problem as a decision-theoretic n -armed bandit problem with reward

at time t being $r_t = 1$ if the task is solved and $r_t = 0$ if the problem is not solved. Whether or not the task at time t is solved or not, at the next time-step we go to the next problem. Our aim is to find a policy π mapping from the history of observations to selection of algorithms such that we maximise the total reward to time T in expectation

$$\mathbb{E}_{\xi, \pi} U_0^T = E_{\xi, \pi} \sum_{t=1}^T r_t.$$

1. Characterise the essential difference between maximising U_0^0 , U_0^1 , U_0^2 ?
2. For $n = 3$, calculate the maximum expected utility

$$\max_{\pi} \mathbb{E}_{\xi, \pi} U_0^T$$

using backwards induction for $T \in \{0, 1, 2, 3, 4\}$ and report the expected utility in each case. *Hint: Use the decision-theoretic bandit formulation to dynamically construct a Markov decision process which you can solve with backwards induction. See also the extensive decision rule utility from exercise set 3.*

3. Now utilise the backwards induction algorithm developed in the previous step in a problem where we receive a sequence of N tasks to solve and our utility is

$$U_0^N = \sum_{t=1}^N r_t$$

At each step $t \leq N$, find the optimal action by calculating $\mathbb{E}_{\xi, \pi} U_t^{t+T}$ for $T \in \{0, 1, 2, 3, 4\}$ take it. *Hint: At each step you can update your prior distribution using the same routine you use to update your prior distribution. You only need consider $T < N - t$.*

4. Develop a simple heuristic algorithm of your choice and compare its utility with the utility of the backwards induction. Perform 10^3 simulations, each experiment running for $N = 10^3$ time-steps and average the results. How does the performance improve? *Hint: If the program runs too slowly go only up to $T = 3$*

6.8.4 Scheduling

You are controlling a small processing network that is part of a big CPU farm. You in fact control a set of n processing nodes. At time t , you may be given a job of class $x_t \in X$ to execute. Assume these are identically and independently drawn such that $\mathbb{P}(x_t = k) = p_k$ for all t, k . With some probability p_0 , you are not given a job to execute at the next step. If you do have a new job, then you can either:

- (a) Ignore the job
- (b) Send the job to some node i . If the node is already active, then the previous job is lost.

Not all the nodes and jobs are equal. Some nodes are better at processing certain types of jobs. If the i -th node is running a job of type $k \in X$, then it has a probability of finishing it within that time step equal to $\phi_{i,k} \in [0, 1]$. Then the node becomes free, and can accept a new job.

For this problem, assume that there are $n = 3$ nodes and $k = 2$ types of jobs and that the completion probabilities are given by the following matrix:

$$\Phi = \begin{bmatrix} 0.3 & 0.1 \\ 0.2 & 0.2 \\ 0.1 & 0.3 \end{bmatrix}. \quad (6.8.1)$$

Also, we set $p_0 = 0.1, p_1 = 0.4, p_2 = 0.5$ to be the probabilities of not getting any job, and the probabilities of the two job types respectively. We wish to find the policy maximising the expected total reward given the MDP model π :

$$\mathbb{E}_{\mu, \pi} \sum_{t=0}^{\infty} \gamma^t r_t, \quad (6.8.2)$$

with $\gamma = 0.9$ and where we get a reward of 1 every time a job is completed.

More precisely, at each time step t , the following events happen:

1. A new job x_t appears
2. Each node either continues processing, or completes its current job and becomes free. You get a reward r_t equal to the number of nodes that complete their jobs within this step.
3. You decide whether to ignore the new job or add it to one of the nodes. If you add a job, then it immediately starts running for the duration of the time step. (If the job queue is empty then you cannot add a job to a node, obviously)

EXERCISE 25 (180). Solve the following problems:

1. Identify the state and action space of this problem and formulate it as a Markov decision process. *Hint: Use independence of the nodes to construct the MDP parameters.*
2. Solve the problem using value iteration, using the stopping criterion indicated in theorem 15, equation (5.5), in Chapter VII, with $\epsilon = 0.1$. Indicate the number of iterations needed to stop.
3. Solve the problem using policy iteration. Indicate the number of iterations needed to stop. *Hint: You can either modify the value iteration algorithm to perform policy evaluation, using the same epsilon, or you can use the linear formulation. If you use the latter, take care with the inverse!*
4. Now consider an alternative version of the problem, where we suffer a penalty of 0.1 (i.e. we get a negative reward) for each time-step that each node is busy. Are the solutions different?
5. Finally consider a version of the problem, where we suffer a penalty of 10 (i.e. we get a negative reward) each time we cancel an executing job. Are the solutions different?
6. Plot the value function for the optimal policy in each setting.

Hint: To verify that your algorithms work, test them first on a smaller MDP with known solutions. For example, <http://webdocs.cs.ulberta.ca/~sutton/book/ebook/node35.html>

6.8.5 General questions

- EXERCISE 26 (20!).
1. What in your view is the fundamental advantages and disadvantages of modelling problems as Markov decision processes?
 2. Is the algorithm selection problem of Exercise 24 solvable with policy iteration? If so, how? What are the fundamental similarities and differences between the decision-theoretic finite-horizon bandit setting of exercise 1 and the infinite-horizon MDP settings of exercise 2?

Chapter 7

Simulation-based algorithms

7.1 Introduction

In this chapter, we consider the general problem of reinforcement learning in dynamic environments. Up to now, we have only examined a solution method for bandit problems, which are only a special case. The Bayesian decision-theoretic solution is to *reduce* the bandit problem to a *Markov decision process* which can then be solved with backwards induction.

We also have seen that Markov decision processes can be used to *describe environments* in more general reinforcement learning problems. When our knowledge of the MDP describing these problems is perfect, then we can employ a number of standard algorithms to find the optimal policy. However, in the actual reinforcement learning problem, the model of the environment is *unknown*. However, as we shall see later, both of these ideas can be combined to solve the general reinforcement learning problem.

The main focus of this chapter is how to simultaneously learn about the underlying process and act to maximise utility in an *approximate* way. This can be done through approximate dynamic programming, where we replace the actual unknown dynamics of the Markov decision process with estimates. The estimates can be improved by drawing samples from the environment, either by acting within the real environment or using a simulator. In both cases we end up with a number of algorithms that can be used for reinforcement learning. Although may not be performing as well as the Bayes-optimal solution, these have a low enough computational complexity that they are worth investigating in practice.

It is important to note that the algorithms in this chapter can be quite far from optimal. They may converge eventually to an optimal policy, but they may not accumulate a lot of reward while still learning. In that sense, they are not solving the full reinforcement learning problem because their *online* performance can be quite low.

For simplicity, we shall first return to the example of bandit problems. As before, we have n actions corresponding to probability distributions P_i on the real numbers $\{P_i \mid i = 1, \dots, n\}$ and our aim is to maximise total reward (in expectation). Had we known the distribution, we could simply always the maximising action, as the expected reward of the i -th action can be easily calculated from P_i and the reward only depends on our current action.

As the P_i are unknown, we must use a history-dependent policy. In the remainder of this section, we shall examine algorithms which asymptotically converge to the optimal policy (which, in the case of bandits corresponds to pulling always pulling the best arm), but for which we cannot always guarantee a good initial behaviour.

7.1.1 The Robbins-Monro approximation

In this setting, we wish to replace the actual Markov decision process in which we are acting, with an estimate that will eventually converge to the true process. At the same time, we shall be taking actions which are nearly-optimal with respect to the estimate.

To approximate the process, we shall use the general idea of a Robbins-Monro stochastic approximation [Robbins and Monro, 1951]. This entails maintaining a *point estimate* of the parameter we want to approximate and perform *random*

steps that on average move towards the solution, in a way to be made more precise later. The stochastic approximation actually defines a large class of procedures, and it contains stochastic gradient descent as a special case.

Algorithm 10 Robbins-Monro bandit algorithm

```

1: input Step-sizes  $(\alpha_t)_t$ , initial estimates  $(\mu_{i,0})_i$ , policy  $\pi$ .
2: for  $t = 1, \dots, T$  do
3:   Take action  $a_t = i$  with probability  $\pi(i | a_1, \dots, a_{t-1}, r_1, \dots, r_{t-1})$ .
4:   Observe reward  $r_t$ .
5:    $\mu_{t,i} = \alpha_{i,t}r_t + (1 - \alpha_{i,t})\mu_{i,t-1}$  // estimation step
6:    $\mu_{t,i} = \mu_{j,t-1}$  for  $j \neq i$ .
7: end for
8: return  $\mu_T$ 

```

An bandit algorithm that uses a Robbins-Monro approximation is given in Algorithm 10. The input is a particular policy π , which defines probability distribution over the next actions given the observed history, a set of initial estates $\mu_{i,0}$ for the bandit means, and a sequence of step sizes α .

The algorithm can be separated in two parts. Taking actions according to the policy (step 3) and the observation of rewards with an update of the estimated values (steps 4-6). The policy itself is an input to the algorithm, but it will in practice only depend on $\mu_{t,i}$ and t ; we shall discuss appropriate policies later. Regarding the estimation itself, note that only the estimate for the arm which we have drawn is updated. As we shall see later, this particular update rule chosen in this case be seen as trying to minimise the expected squared error between the estimated reward, and the random reward obtained by each bandit. Consequently, the variance of the reward of each bandit plays an important role.

The step-sizes α must obey certain constraints in order for the algorithm to work, in particular it must decay neither too slowly, nor too fast. There is one particular choice, for which our estimates are in fact the mean estimate of the expected value of the reward for each action i , which is a natural choice if the bandits are stationary.

The other question is what policy to use to take actions. We must take all actions often enough, so that we have good estimates for the expected reward of every bandit. One simple way to do it is to play the apparently best bandit most of the time, but to sometimes select bandits randomly. This is called ϵ -greedy action selection. This ensures that all actions are tried a sufficient number of times.

Definition 7.1.1 (ϵ -greedy policy).

$$\hat{\pi}_\epsilon^* \triangleq (1 - \epsilon_t)\hat{\pi}_t^* + \epsilon_t \text{Unif}(\mathcal{A}), \quad (7.1.1)$$

$$\hat{\pi}_t^*(i) = \mathbb{I}\left\{i \in \hat{\mathcal{A}}_t^*\right\} / |\hat{\mathcal{A}}_t^*|, \quad \hat{\mathcal{A}}_t^* = \arg \max_{i \in \mathcal{A}} \mu_{t,i} \quad (7.1.2)$$

This is formally defined in Definition 7.1.1. We allow the randomness of the policy to depend on t . This is because, as our estimates converge to the true values, we wish to reduce randomness so as to converge to the optimal policy.

The main two parameters of the algorithm are the amount of randomness in the ϵ -greedy action selection and the step-size α in the estimation. Both of them have a significant effect in the performance of the algorithm. Although we could vary them with time, it is perhaps instructive to look at what happens for fixed values of ϵ, α . Figures 7.1 show the average reward obtained, if we keep the step size α or the randomness ϵ fixed, respectively, with initial estimates $\mu_{0,i} = 0$.

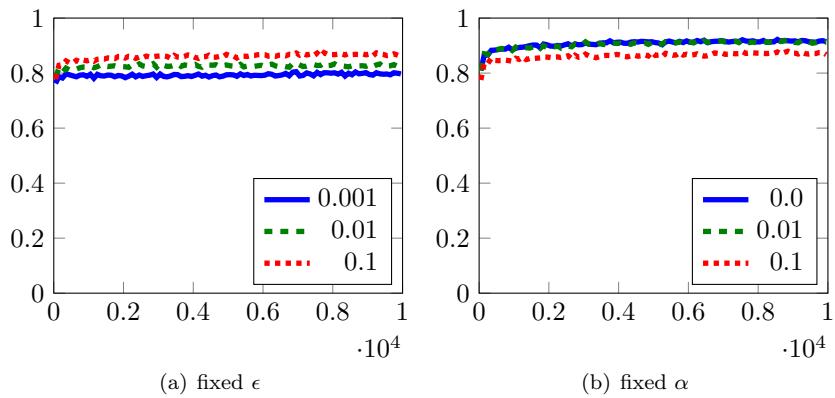


Figure 7.1: For the case of fixed $\epsilon_t = 0.1$, the step size is $\alpha \in \{0.01, 0.1, 0.5\}$. For the case of fixed α , the exploration rate is

For a fixed ϵ , we find that larger values of α tend to give a better result eventually, while smaller values have a better initial performance. This is a natural trade-off, since large α appears to ‘‘learn’’ fast, but it also ‘‘forgets’’ quickly. That is, for a large α , our estimates mostly depend upon the last few rewards observed.

Things are not so clear-cut for the choice of ϵ . We see that the choice of $\epsilon = 0$, is significantly worse than $\epsilon = 0.1$. So, that appears to suggest that there is an optimal level of exploration. How should that be determined? Ideally, we should be able to use the decision-theoretic solution seen earlier, but perhaps a good heuristic way of choosing ϵ may be good enough.

7.1.2 The theory of the approximation

Here we quickly review some basic results of stochastic approximation theory. Complete proofs can be found in Bertsekas and Tsitsiklis [1996]. The main question here is whether our estimates converge to the right values, and whether the complete algorithm itself converges to a optimal policy. We are generally not interested in how much reward we obtain during the optimisation process, but only on asymptotic convergence.

We first consider the core problem of stochastic approximation itself. In particular, we shall cast the approximation problem as a minimisation problem, i.e. we shall define a function f such that, if μ_t is our estimate of μ , then f is minimised at $f(\mu_t)$. Then, given the function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, we wish to develop an algorithm that generates a sequences of values μ_t which converges to some

μ^* that is a local minimum, or a stationary point for f . For strictly convex f , this would also be a global minimum.

In particular, we examine algorithms which maintain estimates μ_t over time, with the update equation:

$$\mu_{t+1} = \mu_t + \alpha_t z_{t+1}. \quad (7.1.3)$$

Here μ_t is our estimate, α_t is a step-size and z_t is a direction. In addition, we use $h_t \triangleq \{\mu_t, z_t, \alpha_t, \dots\}$ to denote the complete history of the algorithm.

The above algorithm can be shown to converge to a stationary point of f under certain assumptions. Sufficient conditions include continuity and smoothness properties of f and the update direction z . In particular, we shall assume the following about the function f that we wish to minimise.

Assumption 7.1.1. *Assume a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ such that:*

(i) $f(x) \geq 0$ for all $x \in \mathbb{R}^n$.

(ii) (Lipschitz derivative) f is continuously differentiable (i.e. the derivative ∇f exists and is continuous) and $\exists L > 0$ such that:

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|, \quad \forall x, y \in \mathbb{R}^n$$

(iii) (Pseudo-gradient) $\exists c > 0$ such that:

$$c \|\nabla f(\mu_t)\|^2 \leq -\nabla f(\mu_t)^\top \mathbb{E}(z_{t+1} | h_t), \quad \forall t.$$

(iv) $\exists K_1, K_2 > 0$ such that

$$\mathbb{E}(\|z_{t+1}\|^2 | h_t) \leq K_1 + K_2 \|\nabla f(\mu_t)\|^2$$

Condition (ii) is a very basic condition for convergence. It basically ensures that the function is well-behaved, so that gradient-following methods can easily find the minimum. Condition (iii) combines two assumptions in one. Firstly, that expected direction of update always decreases cost, and secondly that the squared norm of the gradient is not too large relative to the size of the update. Finally, condition (iv) ensures that update is bounded in expectation relative to the gradient. One can see how putting together the last two conditions ensures that the expected direction of our update is correct, and that its norm is bounded.

Theorem 7.1.1. *For the algorithm*

$$\mu_{t+1} = \mu_t + \alpha_t z_{t+1},$$

where $\alpha_t \geq 0$ satisfy

$$\sum_{t=0}^{\infty} \alpha_t = \infty, \quad \sum_{t=0}^{\infty} \alpha_t^2 < \infty, \quad (7.1.4)$$

and under Assumption 7.1.1, with probability 1:

1. The sequence $\{f(\mu_t)\}$ converges.
2. $\lim_{t \rightarrow \infty} \nabla f(\mu_t) = 0$.
3. Every limit point μ^* of μ_t satisfies $\nabla f(\mu^*) = 0$.

The above conditions are not necessary conditions. Alternative sufficient conditions relying on contraction properties are discussed in detail in Bertsekas and Tsitsiklis [1996]. The following example illustrates the impact of the choice of step size schedule on convergence.

Estimating the mean of a Gaussian distribution.

Consider a sequence of observations x_t , sampled from a Gaussian distribution with mean $1/2$ and variance 1 , in other words $x_t \sim \mathcal{N}(0.5, 1)$. We compare three different step-size schedules, with update direction:

$$z_{t+1} = x_{t+1} - \mu_t.$$

The first one, $\alpha_t = 1/t$, satisfies both assumptions. The second one, $\alpha_t = 1/\sqrt{t}$, reduces too slowly, and the third one, $\alpha_t = t^{-3/2}$, approaches zero too fast.

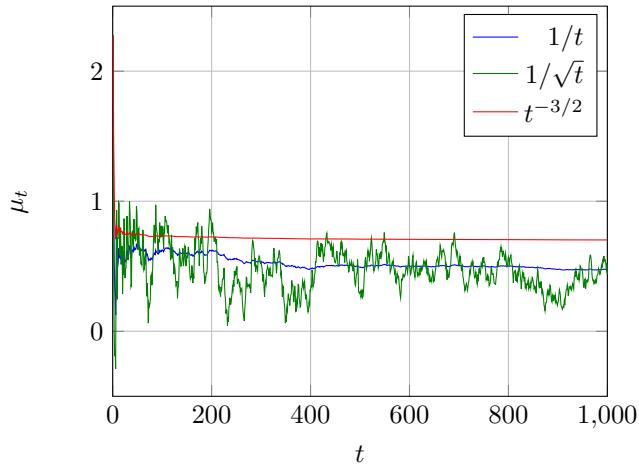


Figure 7.2: Estimation of the expectation of $x_t \sim \mathcal{N}(0.5, 1)$ using three step-size schedules.

Figure 7.2 demonstrates the convergence, or lack thereof, of our estimates μ_t to the expected value. In fact, the schedule $t^{-3/2}$ converges to a value quite far away from the expected value, while the slow schedule $1/\sqrt{t}$ oscillates.

EXAMPLE 33 (Robbins-Monroe conditions for Bernoulli bandits.). Let us now consider the conditions for convergence of the estimates of the bandit algorithm we examined before. Firstly, the function that we wish to minimise relates to the difference between our own estimates and the actual expected reward of the bandit arms. For that reason, we can write the function that we wish to approximate

7.2 Dynamic problems

It is possible to extend the ideas outlined in the previous section to dynamic settings. We simply need to have a policy that is greedy with respect to our estimates, and a way to update our estimates so that they converge to the actual Markov decision process we are acting in. However, the dynamic setting presents one essential difference. Our policy now affects which sequences of states we observe, while before it only affected the rewards. While in the bandit problem we could freely select an arm to pull, we might no longer be able to go to an arbitrary state.¹ Otherwise, the algorithmic structure remains the same and is described below.

Algorithm 11 Generic reinforcement learning algorithm

```

1: input Update-rule  $f : \Theta \times \mathcal{S}^2 \times \mathcal{A} \times \mathcal{R} \rightarrow \Theta$ , initial parameters  $\theta_0 \in \Theta$ ,  

   policy  $\pi : \mathcal{S} \times \Theta \rightarrow \Delta(\mathcal{A})$ .  

2: for  $t = 1, \dots, T$  do  

3:    $a_t \sim \pi(\cdot | \theta_t, s_t)$  // take action  

4:   Observe reward  $r_{t+1}$ , state  $s_{t+1}$ .  

5:    $\theta_{t+1} = f(\theta_t, s_t, a_t, r_{t+1}, s_{t+1})$  // update estimate  

6: end for
  
```

What should we estimate? For example, θ_t could be describing a posterior distribution over MDPs, or a distribution over parameters. What policy should we use? For example, we could try and use the Bayes-optimal policy with respect to θ , or some heuristic policy.

EXAMPLE 34 (The chain task). The chain task has two actions and five states, as shown in Fig. 7.3. The reward in the leftmost state is 0.2 and 1.0 in the rightmost state, and zero otherwise. The first action (dashed, blue) takes you to the right, while the second action (solid, red) takes you to the first state. However, there is a probability 0.2 with which the actions have the opposite effects. The value function of the chain task for a discount factor $\gamma = 0.95$ is shown in Table 7.1.

The chain task is a very simple, but well-known task, used to test the efficacy of reinforcement learning algorithms. In particular, it is useful for analysing how algorithms solve the exploration-exploitation trade-off, since in the short run simply moving to the leftmost state is advantageous. For a long enough horizon or large enough discount factor, algorithms should be incentivised to more fully explore the state space. A variant of this task, with action-dependent rewards (but otherwise equivalent) was used by [Dearden et al., 1998].

¹This actually depends on what the exact setting is. If the environment is a simulation, then we could try and start from an arbitrary state, but in the reinforcement learning setting this is not the case.

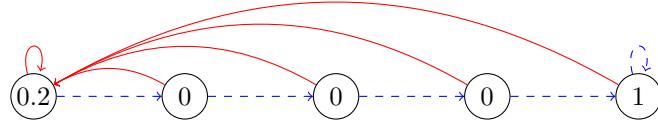


Figure 7.3: The chain task

s	s_1	s_2	s_3	s_4	s_5
$V^*(s)$	7.6324	7.8714	8.4490	9.2090	10.209
$Q^*(s, 1)$	7.4962	7.4060	7.5504	7.7404	8.7404
$Q^*(s, 2)$	7.6324	7.8714	8.4490	9.2090	10.2090

Table 7.1: The chain task's value function for $\gamma = 0.95$

7.2.1 Monte-Carlo policy evaluation and iteration

*reset action
simulation*

To make things as easy as possible, let us assume that we have a way to start the environment from any arbitrary state. That would be the case if the environment had a *reset action*, or if we were simply running an accurate *simulation*.

We shall begin with simplest possible problem, that of estimating the expected utility of each state for a specific policy. This can be performed with Monte-Carlo policy evaluation. In the standard setting, we can the value function for every state by approximating the expectation with the sum of rewards obtained over multiple trajectories starting from each state. The k -th trajectory starts from some initial state $s_0 = s$ and the next states are sampled as follows

$$a_t^{(k)} \sim \pi(a_t | h_t), r_t^{(k)} \sim \mathbb{P}_\mu(r_t | s_t^{(k)}, a_t^{(k)}) s_{t+1}^{(k)} \sim \mathbb{P}_\mu(s_{t+1} | s_t^{(k)}, a_t^{(k)}). \quad (7.2.1)$$

Then the value function satisfies

$$V_\mu^\pi(s) \triangleq \mathbb{E}_\mu^\pi(U | s_1 = s) \approx \frac{1}{K} \sum_{k=1}^K \sum_{t=1}^T r_t^{(k)},$$

where $r_t^{(k)}$ is the sequence of rewards obtained from the k -th trajectory.

Algorithm 12 Stochastic policy evaluation

```

1: input Initial parameters  $v_0$ , Markov policy  $\pi$ .
2: for  $s \in \mathcal{S}$  do
3:    $s_1 = s$ .
4:   for  $k = 1, \dots, K$  do
5:     Run policy  $\pi$  for  $T$  steps.
6:     Observe utility  $U_k = \sum_t r_t$ .
7:     Update estimate  $v_{k+1}(s) = v_k(s) + \alpha_k(U_k - v_k(s))$ 
8:   end for
9: end for
10: return  $v_K$ 

```

For $\alpha_k = 1/k$ and iterating over all \mathcal{S} , this is the same as Monte-Carlo policy evaluation.

7.2.2 Monte Carlo updates

Note that s_1, \dots, s_T contains s_k, \dots, s_T .

This suggests that we could update the value of all encountered states, as we also have the utility starting from each state. We call this algorithm

Algorithm 13 Every-visit Monte-Carlo update

```

1: input Initial parameters  $\mathbf{v}_k$ , trajectory  $s_1, \dots, s_T$ , rewards  $r_1, \dots, r_T$  visit
   counts  $n$ .
2: for  $t = 1, \dots, T$  do
3:    $U_t = \sum_{t=1}^T r_t$ .
4:    $n_t(s_t) = n_{t-1}(s_t) + 1$ 
5:    $\mathbf{v}_{t+1}(s_t) = \mathbf{v}_t(s_t) + \alpha_{n_t(s_t)}(s_t)(U_t - \mathbf{v}_t(s_t))$ 
6:    $n_t(s) = n_{t-1}(s)$ ,  $\mathbf{v}_t(s) = \mathbf{v}_{t-1}(s)$   $\forall s \neq s_t$ .
7: end for
8: return  $\mathbf{v}_K$ 
```

For a proper Monte-Carlo estimate, when the environment is stationary $\alpha_{n_t(s_t)}(s_t) = 1/n_t(s_t)$. Nevertheless, this type of estimate can be biased, as can be seen by the following example.

EXAMPLE 35. Consider a two-state chain with $\mathbb{P}(s_{t+1} = 1 | s_t = 0) = \delta$ and $\mathbb{P}(s_{t+1} = 1 | s_t = 1) = 1$, and reward $r(1) = 1$, $r(0) = 0$. Then the every-visit estimate is biased.

Let us consider the discounted setting. Then value of the second state is $1/(1-\gamma)$ and the value of the first state is $\sum_k (\delta\gamma)^k = 1/(1-\delta\gamma)$. Consider the every-visit Monte-Carlo update. The update is going to be proportional to the number of steps you spend in that state.

In order to avoid the bias, we must instead look at only the first visit to every state. This eliminates the dependence between states and is called the first visit Monte-Carlo update .

Algorithm 14 First-visit Monte-Carlo update

```

1: input Initial parameters  $\mathbf{v}_1$ , trajectory  $s_1, \dots, s_T$ , rewards  $r_1, \dots, r_T$ , visit
   counts  $n$ .
2: for  $t = 1, \dots, T$  do
3:    $U_t = \sum_{t=1}^T r_t$ .
4:    $n_t(s_t) = n_{t-1}(s_t) + 1$ 
5:    $\mathbf{v}_{t+1}(s_t) = \mathbf{v}_t(s_t) + \alpha_{n_t(s_t)}(s_t)(U_t - \mathbf{v}_t(s_t))$  if  $n_t(s_t) = 1$ .
6:    $n_t(s) = n_{t-1}(s)$ ,  $\mathbf{v}_t(s) = \mathbf{v}_{t-1}(s)$  otherwise
7: end for
8: return  $\mathbf{v}_{T+1}$ 
```

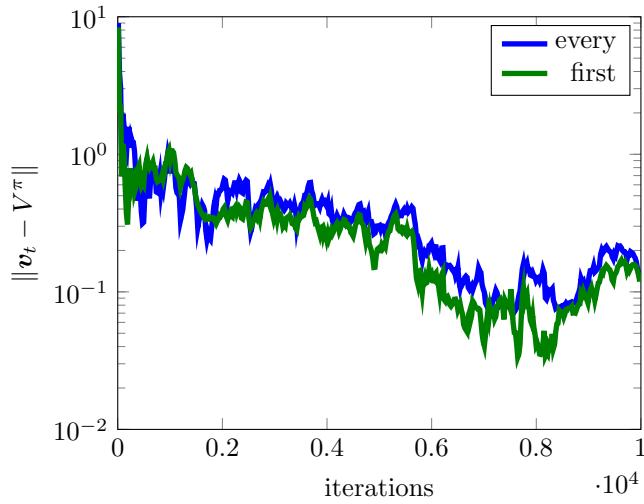


Figure 7.4: Error as the number of iterations n increases, for first and every visit Monte Carlo estimation.

7.2.3 Approximate policy iteration

A well-known algorithm for getting an optimal policy is policy iteration, Algorithm 7 in Section 6.5.4. This consists of estimating the value of a particular policy, and then trying to get an improved policy using this value. We can still apply the same principle for the case where we cannot exactly evaluate a policy. This is called approximate policy iteration. Unfortunately, approximate policy iteration does not necessarily converge without strong conditions on each approximation step.

Algorithm 15 Approximate policy iteration

```

1: input Initial parameters  $\mathbf{v}_0$ , initial Markov policy  $\pi_0$ , estimator  $f$ .
2: for  $i = 1, \dots, N$  do
3:   Get estimate  $\mathbf{v}_i = f(\mathbf{v}_{i-1}, \pi_{i-1})$ .
4:   Calculate new policy  $\pi_i = \arg \max_{\pi} \mathcal{L}\mathbf{v}_i$ .
5: end for
```

7.2.4 Temporal difference methods

The main idea of temporal differences is to use partial samples of the utility and replace the remaining sample from time t with an estimate of the expected utility after time t . Since there maybe no particular reason to choose a specific t , frequently an exponential distribution t 's is used.

Let us first look at the usual update when we have the complete utility sample U_k . The full stochastic update is of the form:

$$\mathbf{v}_{k+1}(s) = \mathbf{v}_k(s) + \alpha(U_k - \mathbf{v}_k(s)),$$

Using the *temporal difference error* $d(s_t, s_{t+1}) = \mathbf{v}(s_t) - [\mathbf{r}(s_t) + \gamma\mathbf{v}(s_{t+1})]$, we obtain the update:

$$\mathbf{v}_{k+1}(s) = \mathbf{v}_k(s) + \alpha \sum_t \gamma^t d_t, \quad d_t \triangleq d(s_t, s_{t+1}) \quad (7.2.2)$$

Stochastic, incremental, update:

$$\mathbf{v}_{t+1}(s) = \mathbf{v}_t(s) + \alpha \gamma^t d_t. \quad (7.2.3)$$

We have now converted the full stochastic update into an incremental update that is nevertheless equivalent to the old update. Let us see how we can generalise this to the case where we have a mixture of temporal differences.

Temporal difference algorithm with eligibility traces.

TD(λ).

Recall the temporal difference update when the MDP is given in analytic form.

$$\mathbf{v}_{n+1}(i) = \mathbf{v}_n(i) + \tau_n(i), \quad \tau_n(i) \triangleq \sum_{t=0}^{\infty} \mathbb{E}_{\pi_n, \mu} [(\gamma\lambda)^m d_n(s_t, s_{t+1}) \mid s_0 = i].$$

We can convert this to a stochastic update, which results in the well-known TD(λ) algorithm for policy evaluation.

$$\mathbf{v}_{n+1}(s_t) = \mathbf{v}_n(s_t) + \alpha \sum_{k=t}^{\infty} (\gamma\lambda)^{k-t} d_k. \quad (7.2.4)$$

Unfortunately, this algorithm is only possible to implement offline due to the fact that we are looking at future values.

This problem can be fixed by the backwards-looking Online TD(λ) algorithm. The main idea is to backpropagate changes in future states to previously encountered states. However, we wish to modify older states less than more recent states.

Algorithm 16 Online TD(λ)

```

1: input Initial parameters  $\mathbf{v}_k$ , trajectories  $(s_t, a_t, r_t)$ 
2:  $\mathbf{e}_0 = \mathbf{0}$ .
3: for  $t = 1, \dots, T$  do
4:    $d_t \triangleq d(s_t, s_{t+1})$  // temporal difference
5:    $\mathbf{e}_t(s_t) = \mathbf{e}_{t-1}(s_t) + 1$  // eligibility increase
6:   for  $s \in \mathcal{S}$  do
7:      $\mathbf{v}_{t+1}(s_t) = \mathbf{v}_t(s) + \alpha_t \mathbf{e}_t(s) d_t$ . // update all eligible states
8:   end for
9:    $\mathbf{e}_{t+1} = \lambda \mathbf{e}_t$ 
10: end for
11: return  $\mathbf{v}_T$ 

```

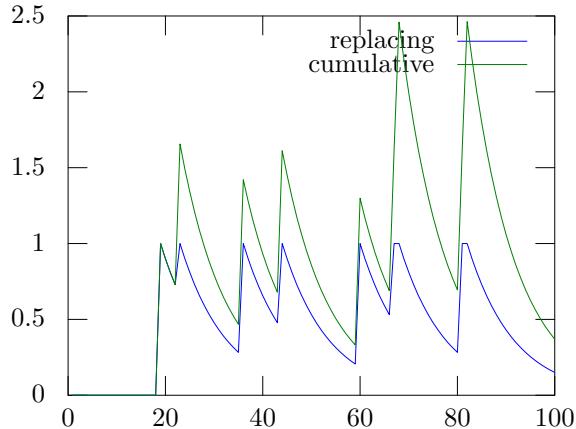


Figure 7.5: Eligibility traces, replacing and cumulative.

For replacing traces, use $e_t(s_t) = e_{t-1}(s_t) + 1$.

7.2.5 Stochastic value iteration methods

The main problem we had seen so far with Monte-Carlo based simulation is that we normally require a complete sequence of rewards before updating values. However, in value iteration, we can simply perform a backwards step from all the following states in order to obtain a utility estimate. This idea is explored in stochastic value iteration methods.

The standard value iteration algorithm performs a sweep over the complete state space at each iteration. However, could perform value iteration over an arbitrary sequence of states. For example, we can follow a sequence of states generated from a particular policy. This lends to the idea of *simulation-based* value iteration.

Such state sequences must satisfy various technical requirements. In particular, the policies that generate those state sequences must be *proper* for episodic problems. That is, that all policies should reach a terminating state with probability 1. For discounted non-episodic problems, this is easily achieved by using a geometric distribution for termination time. This ensures that all policies will be proper. Alternatively, of course, we could simply select starting states with an arbitrary schedule, as long as all states are visited infinitely often in the limit.

However, value iteration also requires the Markov decision process model. The question is whether it is possible to replace the MDP model with some arbitrary estimate. This estimate can itself be obtained via simulation. This leads to a whole new family of stochastic value iteration algorithms. The most important and well-known of these is Q -learning, which uses a trivial empirical MDP model.

Simulation-based value iteration

First, however, we shall discuss the extension of value iteration to the case where we obtain state data from simulation. This allows us to concentrate our estimates to the most useful states.

Algorithm 17 shows a generic simulation-based value iteration algorithm, with a uniform restart distribution $\text{Unif}(\mathcal{S})$ and termination probability ϵ .

Algorithm 17 Simulation-based value iteration

- 1: Input μ, \mathcal{S} .
 - 2: Initialise $s_t \in \mathcal{S}, \mathbf{v}_0 \in \mathcal{V}$.
 - 3: **for** $t = 1, 2, \dots$ **do**
 - 4: $s = s_t$.
 - 5: $\pi_t(s) = \arg \max_a r(s, a) + \gamma \sum_{s' \in \mathcal{S}} \mathbb{P}_\mu(s'|s, a) \mathbf{v}_{t-1}(s')$
 - 6: $\mathbf{v}_t(s) = r(s, a) + \gamma \sum_{s' \in \mathcal{S}} \mathbb{P}_\mu(s'|s, \pi_t(s)) \mathbf{v}_{t-1}(s')$
 - 7: $s_{t+1} \sim (1 - \epsilon) \cdot \mathbb{P}(s_{t+1} | s_t = a, \pi_t, \mu) + \epsilon \cdot \text{Unif}(\mathcal{S})$.
 - 8: **end for**
 - 9: Return π_n, V_n .
-

In the following figures, we can see the error in value function estimation in the chain task when using simulation-based value iteration. It is always a better idea to use an initial value \mathbf{v}_0 that is an upper bound on the optimal value function, if such a value is known. This is due to the fact that in that case, convergence is always guaranteed when using simulation-based value iteration, as long as the policy that we are using is proper.²

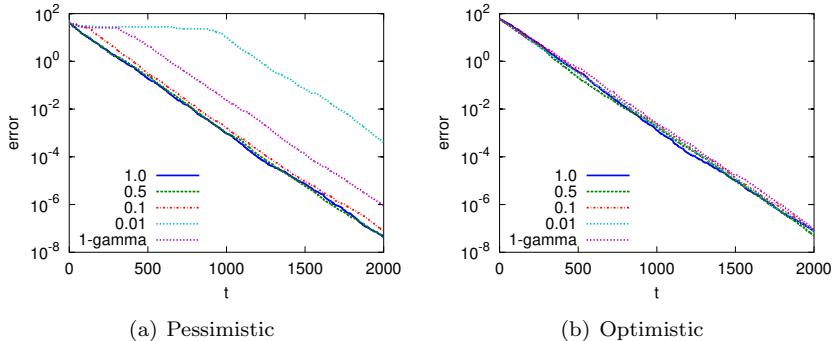


Figure 7.6: Simulation-based value iteration with pessimistic initial estimates ($\mathbf{v}_0 = 0$) and optimistic initial estimates ($\mathbf{v}_0 = 20 = 1/(1 - \gamma)$), for varying ϵ . Errors indicate $\|\mathbf{v}_t - V^*\|_1$.

As can be seen in Figure 7.6, the value function estimation error of simulation-based value iteration is highly dependent upon the initial value function estimate \mathbf{v}_0 and the exploration parameter ϵ . It is interesting to see uniform sweeps ($\epsilon = 1$) result in the lowest estimation error in terms of the value function L_1 norm.

Q-learning

Simulation-based value iteration can be suitably modified for the actual reinforcement learning problem. Instead of relying on a model of the environment,

²In the case of discounted non-episodic problems, this amounts to a geometric stopping time distribution, after which the state is drawn from the initial state distribution.

we replace arbitrary random sweeps of the state-space with the actual state sequence observed in the real environment. We also use this sequence as a simple way to estimate the transition probabilities.

Algorithm 18 Q-learning

- 1: Input $\mu, \mathcal{S}, \epsilon_t, \alpha_t$.
 - 2: Initialise $s_t \in \mathcal{S}, \mathbf{q}_0 \in \mathcal{V}$.
 - 3: **for** $t = 1, 2, \dots$ **do**
 - 4: $s = s_t$.
 - 5: $a_t \sim \hat{\pi}_{\epsilon_t}^*(a | s_t, \mathbf{q}_t)$
 - 6: $s_{t+1} \sim \mathbb{P}(s_{t+1} | s_t = s, a_t, \pi_t, \mu)$.
 - 7: $\mathbf{q}_{t+1}(s_t, a_t) = (1 - \alpha_t)\mathbf{q}_t(s_t, a_t) + \alpha_t[r(s_t) + \mathbf{v}_t(s_{t+1})]$, where $\mathbf{v}_t(s) = \max_{a \in \mathcal{A}} \mathbf{q}_t(s, a)$.
 - 8: **end for**
 - 9: Return π_n, V_n .
-

The result is Q -learning (Algorithm 18), one of the most well-known and simplest algorithms in reinforcement learning. In light of the previous theory, it can be seen as a stochastic value iteration algorithm, where at every step t , given the partial observation (s_t, a_t, s_{t+1}) you have an approximate transition model for the MDP which is as follows:

$$P(s' | s_t, a_t) = \begin{cases} 1, & \text{if } s_{t+1} = s' \\ 0, & \text{if } s_{t+1} \neq s'. \end{cases} \quad (7.2.5)$$

Even though this model is very simplistic, it still seems to work relatively well in practice, and the algorithm is simple to implement. In addition, since we cannot arbitrarily select states in the real environment, we replace the state-exploring parameter ϵ with a time-dependent exploration parameter ϵ_t for the policy we employ on the real environment.

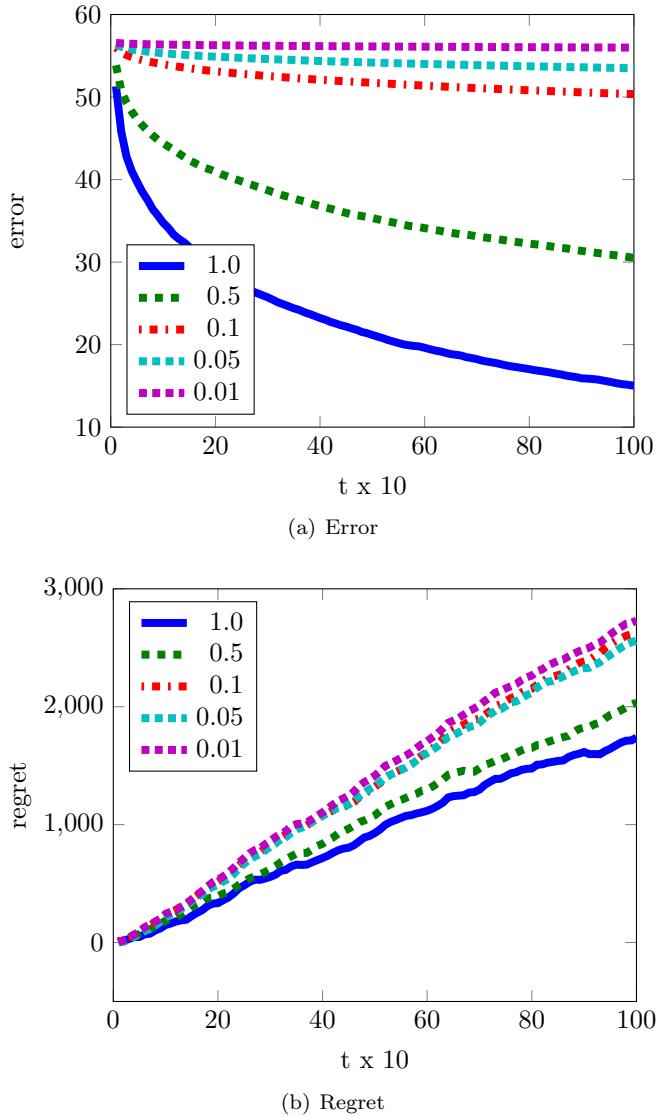


Figure 7.7: Q -learning with $v_0 = 1/(1 - \gamma)$, $\epsilon_t = 1/n_{st}$, $\alpha_t \in \alpha n_{st}^{-2/3}$.

Figure 7.7 shows the performance of the basic Q -learning algorithm for the Chain task, in terms of value function error and regret. In this particular implementation, we used a polynomially decreasing exploration parameter ϵ_t and step size α_t . Both of these depend on the number of visits to a particular state and so perform more efficient Q -learning.

Of course, one could get any algorithm in between pure Q -learning and pure stochastic value iteration. In fact, variants of the Q -learning algorithm using eligibility traces (see Section 7.2.4) can be formulated in this way.

Generalised stochastic value iteration Finally, we can generalise the above ideas to the following algorithm. This is an online algorithm, which can be

applied directly to a reinforcement learning problem and it includes simulation-based value iteration and Q -learning as special cases. There are three parameters associated with this algorithm. The first is ϵ_t , the exploration amount performed by the policy we follow. The second is α_t , the step size parameter. The third one is σ_t , the state-action distribution. The final parameter is the MDP estimator $\hat{\mu}_t$. This includes both an estimate of the transition probabilities $\mathbb{P}_{\hat{\mu}_t}(s' | s, a)$ and of the expected reward $r_{\hat{\mu}_t}(s, a)$.

Algorithm 19 Generalised stochastic value iteration

```

1: Input  $\hat{\mu}_0, \mathcal{S}, \epsilon_t, \alpha_t$ .
2: Initialise  $s_1 \in \mathcal{S}, \mathbf{q}_1 \in \mathcal{Q}, \mathbf{v}_0 \in \mathcal{V}$ .
3: for  $t = 1, 2, \dots$  do
4:    $a_t \sim f(\hat{\pi}_{\epsilon_t}^*(a | s_t, \mathbf{q}_t))$ 
5:   Observe  $s_{t+1}, r_{t+1}$ .
6:    $\hat{\mu}_t = \hat{\mu}_{t-1} | s_t, a_t, s_{t+1}, r_{t+1}$ . // update MDP estimate.
7:   for  $s \in \mathcal{S}, a \in \mathcal{A}$  do
8:     With probability  $\sigma_t(s, a)$  do:

$$\mathbf{q}_{t+1}(s, a) = (1 - \alpha_t)\mathbf{q}_t(s, a) + \alpha_t \left[ r_{\hat{\mu}_t}(s, a) + \gamma \sum_{s' \in \mathcal{S}} \mathbb{P}_{\hat{\mu}_t}(s' | s, a) \mathbf{v}_t(s') \right].$$

9:     otherwise  $\mathbf{q}_{t+1}(s, a) = \mathbf{q}_t(s, a)$ .
10:     $\mathbf{v}_{t+1}(s) = \max_{a \in \mathcal{A}} \mathbf{q}_{t+1}(s, a)$ ,
11:    end for
12:  end for
13: Return  $\pi_n, V_n$ .

```

It is instructive to examine special cases for these parameters. For the case when $\sigma_t = 1$, $\alpha_t = 1$, and when $\hat{\mu}_t = \mu$, we obtain standard value iteration.

For the case when $\sigma_t(s, a) = \mathbb{I}\{s_t = s \wedge a_t = a\}$ and

$$\mathbb{P}_{\hat{\mu}_t}(s_{t+1} = s' | s_t = s, a_t = a) = \mathbb{I}\{s_{t+1} = s' | s_t = s, a_t = a\},$$

it is easy to see that we obtain Q -learning.

Finally, if we set $\sigma_t(s, a) = e_t(s, a)$, then we obtain a stochastic eligibility-trace Q -learning algorithm similar to $Q(\lambda)$.

7.3 Discussion

Most of these algorithms are quite simple, and so clearly demonstrate the principle of learning by reinforcement. However, they do not aim to solve the reinforcement learning problem optimally. They have been mostly of use for finding near-optimal policies given access to samples from a simulator, as used for example to learn to play Atari games Mnih et al. [2015]. However, even in this case, a crucial issue is how much data is needed in the first place to approach optimal play. The second issue is using such methods for online reinforcement learning, i.e. in order to maximise expected utility while still learning.

Convergence. Even though it is quite simple, the convergence of Q -learning has been established in various settings. Tsitsiklis [1994] has provided an asymptotic proof based on stochastic approximation theory with less restrictive assumptions than the original paper Watkins and Dayan [1992]. Later Kearns and Singh [1999] proved finite sample convergence results under strong mixing assumptions on the MDP.

Q -learning can be seen as using a very specific type of approximate transition model. By modifying this, we can obtain more efficient algorithms, such as delayed Q -learning Strehl et al. [2006], which needs $\tilde{O}(|\mathcal{S}||\mathcal{A}|)$ samples to find an ϵ -optimal policy with high probability.

Exploration. In order to perform exploration efficiently, Q -learning does not attempt to perform optimal exploration. Another extension of Q -learning is using a population value function estimates. This was introduced in Dimitrakakis [2006b,a] through the use of random initial values and weighted bootstrapping and evaluated for bandit tasks. Recently, this idea has also been exploited in the context of deep neural networks by Osband et al. [2016] representations of value functions for the case of full reinforcement learning. We will examine this more closely in Chapter 8.

Bootstrapping and subsampling (App. B.6) use a single set of empirical data to obtain an empirical measure of uncertainty about statistics of the data. We wish to do the same thing for value functions, based on data from one or more trajectories. Informally, this variant maintains a collection of Q -value estimates, each one of which is trained on different segments³ of the data, with possible overlaps. In order to achieve efficient exploration, a random Q estimate is selected at every episode, or every few steps. This results in a bootstrap analogue of Thompson sampling. Figure 7.8 shows the use of weighted bootstrap estimates for the Double Chain problem introduced by Dearden et al. [1998]. Bootstrapping and subsampling (App. B.6) use a single

³If not dealing with bandit problems, it is important to do this with trajectories.

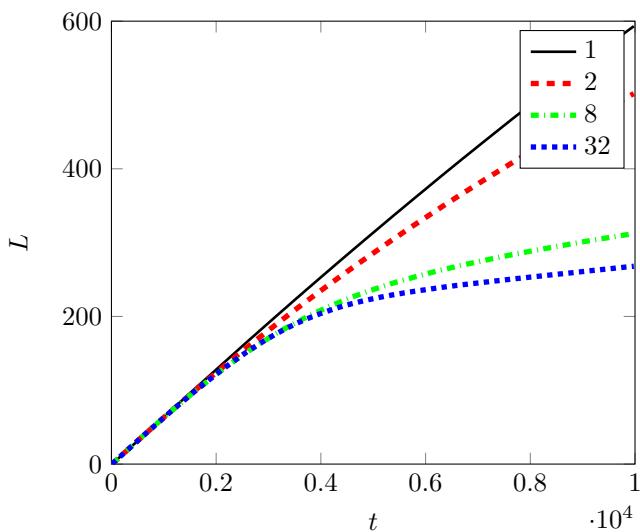


Figure 7.8: Cumulative regret of weighted Bootstrap Q -learning for various amount of bootstrap replicates (1 is equivalent to plain Q -learning). Generally speaking, an increased amount of replicates leads to improved exploration performance.

7.4 Exercises

EXERCISE 27 (180). This is a continuation of exercise 22. Create a reinforcement learning version of the diagnostic model from exercise 22. In comparison to that exercise, here the doctor is allowed to take zero, one, or two diagnostic actions.

View the treatment of each patient as a single episode and design an appropriate state and action space to apply the standard MDP framework: note that all episodes run for at least 2 steps, and there is a different set of actions available at each state: the initial state only has diagnostic actions, while any treatment action terminates the episode and returns us the result.

1. Define the state and action space for each state.
2. Create a simulation of this problem, according to the probabilities mentioned in Exercise 22.
3. Apply a simulation-based algorithm such as Q -learning to this problem. How much times does it take to perform well? Can you improve it so as to take into account the problem structure?

EXERCISE 28. It is well-known that the value function of a policy π for an MDP μ with state reward function r can be written as the solution of a linear equation $V_\mu^\pi = (I - \gamma P_\mu^\pi)^{-1} r$, where the term $\Phi_\mu^\pi \triangleq (I - \gamma P_\mu^\pi)^{-1}$ can be seen as a feature matrix. However, Sarsa and other simulation-based algorithms only approximate the value function directly rather than Φ_μ^π . This means that, if the reward function changes, they have to be restarted from scratch. Is there a way to rectify this?⁴

- 3h Develop and test a simulation-based algorithm (such as Sarsa) for estimating Φ_μ^π , and prove its asymptotic convergence. *Hint: focus on the fact that you'd like to estimate a value function for all possible reward functions.*
- ? Consider a model-based approach, where we build an empirical transition kernel P_μ^π . How good are our value function estimates in the first versus the second approach? Why would you expect either one to be better?
- ? Can the same idea be extended to Q -learning?

⁴This exercise stems from a discussion with Peter Auer in 2012 about this problem.

Chapter 8

Approximate representations

8.1 Introduction

In this chapter, we consider approximation algorithms, which are necessary when the value function, policy, or transition kernel can only be approximately represented. This is the case when the state or policy space are large, which force ourselves to use some parameterisation that may not include the true value function, policy, or transition kernel. In general, we shall assume the existence of either some approximate value function space \mathcal{V}_Θ or some approximate policy space Π_Θ , which are the set of allowed value functions and policies respectively. For the purposes of this chapter, we will assume that we have access to some simulator or approximate model of the transition probabilities, wherever necessary. Model-based reinforcement learning where the transition probabilities are explicitly estimated will be examined in the next two chapters.

As an introduction, let us start with the case we have a value function space \mathcal{V} and some value function $\mathbf{u} \in \mathcal{V}$ that is our best approximation to the optimal value function in \mathcal{V} . Then we can define the greedy policy with respect to \mathbf{u} as follows:

Definition 8.1.1 (\mathbf{u} -greedy policy and value function).

$$\pi_{\mathbf{u}}^* \in \arg \max_{\pi \in \Pi} \mathcal{L}_\pi \mathbf{u}, \quad \mathbf{v}_{\mathbf{u}}^* = \mathcal{L} \mathbf{u}, \quad (8.1.1)$$

where $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ maps from states to action distributions.

Although the greedy policies do not need to be stochastic, here we are explicitly considering stochastic policies, because this sometimes facilitates finding a good approximation. If \mathbf{u} is the optimal value function V^* , then the greedy policy is going to be optimal.

However, when we are trying to approximate a value function, we are constrained to look for it in a parameterised set of value functions \mathcal{V}_Θ , where Θ is the parameter space. Hence, it might be the case that the optimal value function may not lie within \mathcal{V}_Θ . Similarly, the policies that we can use lie in a space Π_Θ , which may not include the greedy policy itself. This is usually because it is not possible to represent all possible value functions and policies in complex problems.

Another difficulty is that we cannot look for a *uniformly good* approximation to a value function or policy. Instead, we define ϕ , a distribution on \mathcal{S} , which specifies on which parts of the state space we want to have a good approximation, by placing higher weight on the most important states.

A simple case is when ϕ does not support \mathcal{S} , meaning only takes positive values for some states $s \in \mathcal{S}$. Frequently, ϕ only has a *finite* support, meaning that we only measure the approximation error over a finite set of states. In the sequel, we shall always define the quality of an approximate value or policy with respect to ϕ .

In the remainder of this chapter, we shall examine a number of approximate dynamic programming algorithms. What all of these algorithms have in common is the requirement to calculate an approximate value function or policy. The two next sections given an overview of the basic problem of fitting an approximate value function or policy to a target.

8.1.1 Fitting a value function.

Let us begin by considering the problem of finding the value function $\mathbf{v}_\theta \in \mathcal{V}_\Theta$ that best matches a target value function \mathbf{u} . This can be done by minimising the difference between the target value \mathbf{u} and the approximation \mathbf{v}_θ :

$$\|\mathbf{v}_\theta - \mathbf{u}\|_\phi = \int_{\mathcal{S}} |\mathbf{v}_\theta(s) - \mathbf{u}(s)| d\phi(s), \quad (8.1.2)$$

with respect to some measure ϕ on \mathcal{S} . If $\mathbf{u} = V^*$, i.e. the optimal value function, then we end up getting the best possible value function with respect to the distribution ϕ . The high-level for fitting an approximate value function to a target is given in Figure 8.1. However, the algorithm remains quite abstract

Approximate value function fit

$$\mathcal{V}_\Theta = \{\mathbf{v}_\theta \mid \theta \in \Theta\}, \quad \theta^* \in \arg \min_{\theta \in \Theta} \|\mathbf{v}_\theta - \mathbf{u}\|_\phi \quad (8.1.3)$$

where $\|\cdot\|_\phi \triangleq \int_{\mathcal{S}} |\cdot| d\phi$.

Figure 8.1: Fitting a value function \mathbf{u} with the best member of an approximate value function space \mathcal{V}_Θ with error metric $\|\cdot\|_\phi$.

as is and the minimisation problem can be difficult to solve in general. A particularly simple case is when the set of approximate functions is small enough for the minimisation to be performed via enumeration.

EXAMPLE 36 (Fitting a finite number of value functions). Consider a finite space of value functions $\mathcal{V}_\Theta = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$, which wish to fit to a target value function \mathbf{u} . In this particular scenario, $\mathbf{v}_1(x) = \sin(0.1x)$, $\mathbf{v}_2(x) = \sin(0.5x)$, $\mathbf{v}_3(x) = \sin(x)$, while

$$\mathbf{u}(x) = 0.5 \sin(0.1x) + 0.3 \sin(0.1x) + 0.1 \sin(x) + 0.1 \sin(10x).$$

Clearly, none of the given functions is a perfect fit. In addition, finding the best overall fit requires minimising an integral. So, for this problem we choose a random set of points $X = \{x_t\}$ on which to evaluate the fit, with $\phi(x_t) = 1$ for every point $x_t \in X$. This is illustrated in Figure 8.2, which shows the error of the functions at the selected points, as well as their cumulative error.

In the example above, the approximation space \mathcal{V}_Θ does not have a member that is sufficiently close to the target value function. It could be that a larger function space contains a better approximation. However, it may be difficult to find the best fit in an arbitrary set \mathcal{V}_Θ .

8.1.2 Fitting a policy.

The problem of fitting a policy is not significantly different from that of fitting a value function, especially when the action space is continuous. Once more, we define an appropriate normed vector space so it makes sense to talk about

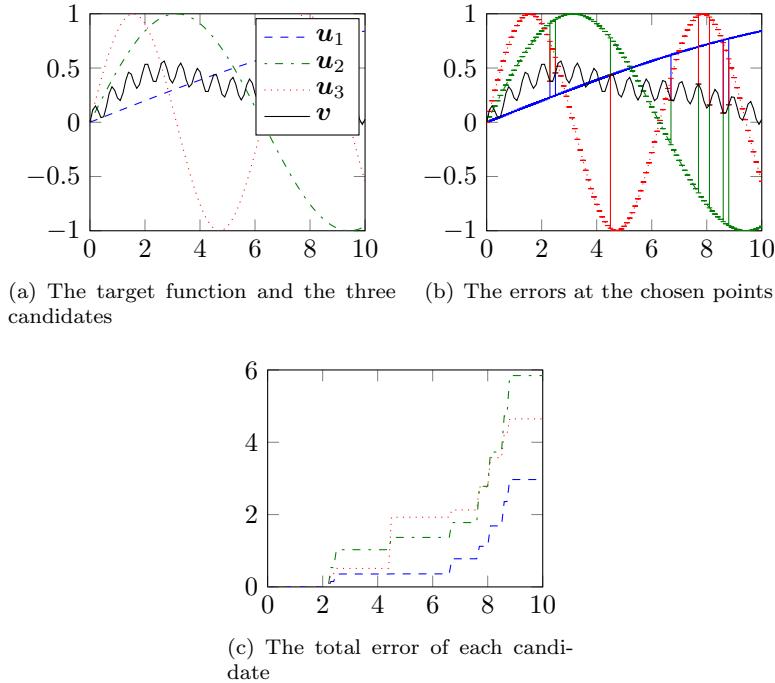


Figure 8.2: Fitting a value function in $\mathcal{V}_\Theta = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ to a target value function \mathbf{u} , over a finite number of points. While none of the tree candidates is a perfect fit, we clearly see that \mathbf{v}_1 has the lowest cumulative error over the measured set of points.

the normed difference between policies with respect to some measure ϕ on the states. In particular, we use the following error between two policies π, π' :

$$\|\pi - \pi'\|_\phi = \int_{\mathcal{S}} \|\pi(\cdot | s) - \pi'(\cdot | s)\| d\phi(s), \quad (8.1.4)$$

where the norm within the integral is usually the L_1 norm. For a finite action space, this corresponds to $\|\pi(\cdot | s) - \pi'(\cdot | s)\| = \sum_{a \in \mathcal{A}} |\pi(a | s) - \pi'(a | s)|$, but certainly other norms may be used and are sometimes more convenient. Figure 8.3 shows the basic algorithm for fitting an approximate policy from a set of policies Π_Θ to a target policy π . Once more, the minimisation problem may not be trivial, but there are some cases where it is particularly easy. One of these is when the policies can be efficiently enumerated, as in the example below.

EXAMPLE 37 (Fitting a finite space of policies). For simplicity, consider the space of deterministic policies, with a binary action space $\mathcal{A} = \{0, 1\}$. Then each policy can be represented as a simple mapping $\pi : \mathcal{S} \rightarrow \{0, 1\}$, corresponding to a binary partition of the state space. In this example, the state space is the 2-dimensional unit cube, $\mathcal{S} = [0, 1]^2$. Figure 8.4 shows an example policy, where the light red and light green areas represent it taking action 1 and 0 respectively. The measure ϕ has support only on the crosses and circles, which indicate the action taken at that location. Consider a policy space Π_Θ consisting of just four policies. Each set of two policies is indicated by

Approximate policy fit

$$\Pi_\Theta = \{\pi_\theta \mid \theta \in \Theta\}, \quad \theta^* \in \arg \min_{\theta \in \Theta} \|\pi_\theta - \pi_u^*\|_\phi \quad (8.1.5)$$

where $\pi_u^* = \arg \max_{\pi \in \Pi} \mathcal{L}_\pi u$

Figure 8.3: Approximating a policy

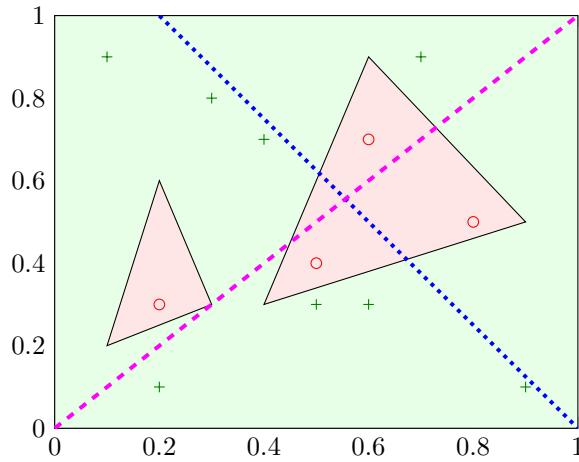


Figure 8.4: An example policy. The red areas indicate action 1 been taken, and the green areas action 0. The ϕ measure has finite support, indicated by the crosses and circles. The blue and magenta lines indicate two possible policies that separate the state space with a hyperplane.

the magenta (dashed) and blue (dotted) lines in Figure 8.4. Each line corresponds to two possible policies, one selecting action 1 in the high region, and the other selecting action 0 instead. In terms of our error metric, the best policy is the one that makes the fewest mistakes. Consequently, the best policy in this set to use the blue line and play action 1 (red) in the top-right region.

8.1.3 Features

Frequently, when dealing with large, or complicated spaces, it pays to project the state and action observations onto a feature space \mathcal{X} . In that way, we can make problems much more manageable. Generally speaking, a feature mapping is defined as follows.

Feature mapping $f : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{X}$.

For $\mathcal{X} \subset \mathbb{R}^n$, the feature mapping can be written in vector form:

$$f(s, a) = \begin{bmatrix} f_1(s, a) \\ \vdots \\ f_n(s, a) \end{bmatrix} \quad (8.1.6)$$

What sort of functions should we use? A common idea is to use a set of smooth functions, that are focused around a single point. One of the most usual examples are radial basis functions.

EXAMPLE 38 (Radial Basis Functions). Let d be a metric on $\mathcal{S} \times \mathcal{A}$ and $\{(s_i, a_i) \mid i = 1, \dots, n\}$. Then we define each element of f as:

$$f_i(s, a) \triangleq \exp \{-d[(s, a), (s_i, a_i)]\}. \quad (8.1.7)$$

These function are sometimes called *kernels*.

Another common type of functions are binary functions. These effectively discretise a continuous space through either a cover or a partition.

Definition 8.1.2. The collection of sets \mathcal{G} is a *cover* of X iff $\bigcup_{S \in \mathcal{G}} S \supset X$.

Definition 8.1.3. The collection of sets \mathcal{G} is a *partition* of X iff

1. \mathcal{G} is a cover of X
2. If $S \neq R \in \mathcal{G}$ then $S \cap R = \emptyset$.
3. $\bigcup_{S \in \mathcal{G}} S = X$.

In reinforcement learning, these types of feature functions corresponding to partitions are usually referred to as *tilings*.

EXAMPLE 39 (Tilings). Let $\mathcal{G} = \{X_1, \dots, X_n\}$ be a *partition* of $\mathcal{S} \times \mathcal{A}$ of size n . Then:

$$f_i(s, a) \triangleq \mathbb{I}\{(s, a) \in X_i\}. \quad (8.1.8)$$

Multiple tilings create a cover. These can be used without many difficulties with most discrete reinforcement learning algorithms.

For a more realistic example, consider the well-known inverted pendulum problem, where a controller must balance a rod upside-down. The state information is the rotational velocity and position of the pendulum. In the value function approximation shown in Figure 8.6, we consider the value of the uniformly random policy where there are three allowed actions: maximum torque left, maximum torque right, and no torque. This leads to higher values near the central balancing point and a smooth drop-off. However, the approximation quality strongly depends on the statistical method we use. Linear-Gaussian models are smooth, but are inadequate for modelling the value function in the 2-dimensional state space. However, a high-dimensional non-linear projection using RBF kernels results in a smooth and accurate value function representation. Non-parametric models such as k -nearest neighbours do not suffer from this problem, and their behaviour is not very sensitive to the input representation.

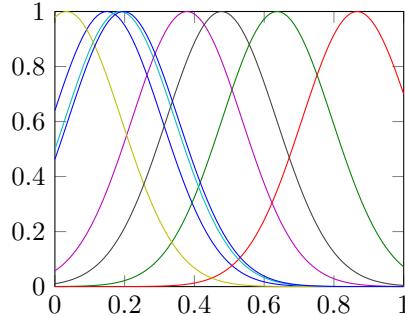


Figure 8.5: Radial Basis Functions

8.2 Approximate policy iteration (API)

Approximate policy iteration

The main idea of approximate policy iteration is to replace the exact Bellman operator \mathcal{L} with an approximate version $\hat{\mathcal{L}}$ and the exact value of the policy with an approximate version. In fact, in the policy improvement step, we simply try to get as close as possible to the best possible improvement, in a restricted set of policies, using an approximate operator. Similarly, in the policy evaluation step, we try to get as close as possible to the actual value of the improved policy.

Algorithm 20 Generic approximate policy iteration algorithm

```

input Initial value function  $v_0$ , approximate Bellman operator  $\hat{\mathcal{L}}$ , approximate value estimator  $\hat{V}$ .
for  $k = 1, \dots$  do
     $\pi_k = \arg \min_{\pi \in \hat{\Pi}} \|\hat{\mathcal{L}}_\pi v_{k-1} - \mathcal{L} v_{k-1}\|$  // policy improvement
     $v_k = \arg \min_{v \in \hat{V}} \|v - V_\mu^{\pi_k}\|$  // policy evaluation
end for

```

More precisely, at the k -th iteration, we use the approximate value v_{k-1} of the previous policy, π_{k-1} , to obtain an improved policy π_k . However, we may not be able to implement the policy $\max_\pi \mathcal{L}_\pi v_{k-1}$ for two reasons. Firstly, because our policy space may not include all possible policies, due to the policy parameterisation. Secondly, because the Bellman operator we have available may only be approximate. Next, we'd like to find the value function v that is the closest to the true value function of policy π_k for the MDP μ . However, even if our value function space is rich enough to do that, the minimisation is done over a norm that integrates over a finite subset of the state space. The following section discusses what the errors in those two approximations imply for the convergence of approximate policy iteration.

8.2.1 Error bounds for approximate value functions

If the approximate value function u is close to V^* then the greedy policy with respect to u is close to optimal. For a finite state and action space, the following holds.

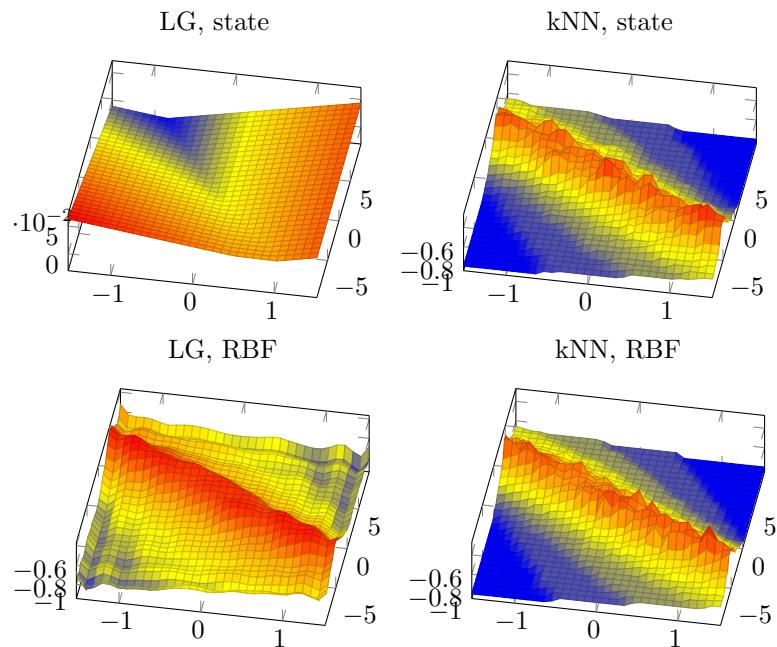


Figure 8.6: Estimated value function of a uniformly random policy on the pendulum problem. Results are shown for a k -nearest neighbour model (kNN) with $k = 3$ and a Bayesian linear-Gaussian model (LG), for either the case when the model uses the plain state information (state) or an 256-dimensional RBF embedding (RBF).

Theorem 8.2.1. Consider a finite MDP μ with discount factor $\gamma < 1$ and a vector $\mathbf{u} \in \mathcal{V}$ such that $\|\mathbf{u} - V_\mu^*\|_\infty = \epsilon$. If π is the \mathbf{u} -greedy policy then

$$\|V_\mu^\pi - V_\mu^*\|_\infty \leq \frac{2\gamma\epsilon}{1-\gamma}.$$

In addition, $\exists \epsilon_0 > 0$ s.t. if $\epsilon < \epsilon_0$, then π is optimal.

Proof. Recall that \mathcal{L} is the one-step Bellman operator and \mathcal{L}_π is the one-step policy operator on the value function. Then

$$\begin{aligned} \|V^\pi - V^*\|_\infty &= \|\mathcal{L}_\pi V^\pi - V^*\|_\infty \\ &\leq \|\mathcal{L}_\pi V^\pi - \mathcal{L}_\pi \mathbf{u}\|_\infty + \|\mathcal{L}_\pi \mathbf{u} - V^*\|_\infty \\ &\leq \gamma \|V^\pi - \mathbf{u}\|_\infty + \|\mathcal{L} \mathbf{u} - V^*\|_\infty \\ &\leq \gamma \|V^\pi - V^*\|_\infty + \gamma \|V^* - \mathbf{u}\|_\infty + \gamma \|\mathbf{u} - V^*\|_\infty \\ &\leq \gamma \|V^\pi - V^*\|_\infty + 2\gamma\epsilon. \end{aligned}$$

This proves the first part.

For the second part, note that the state and action sets are finite. Consequently, the set of policies is finite. Thus, there is some $\epsilon_0 > 0$ such that the best sub-optimal policy is ϵ_0 -close to the optimal policy in value. So, if $\epsilon < \epsilon_0$, the obtained policy must be optimal. \square

Building on this result, we can prove some simple bounds for approximate policy iteration. These are based on the following assumption.

Assumption 8.2.1. Consider a discounted problem with discount factor γ and iterates \mathbf{v}_k, π_k such that:

$$\|\mathbf{v}_k - V^{\pi_k}\|_\infty \leq \epsilon, \quad \forall k \tag{8.2.1}$$

$$\|\mathcal{L}_{\pi_{k+1}} \mathbf{v}_k - \mathcal{L} \mathbf{v}_k\|_\infty \leq \delta, \quad \forall k \tag{8.2.2}$$

This assumption uniformly bounds the error in approximating the value of a policy by ϵ . It also demands that our approximate Bellman operator is δ -close to \mathcal{L} . Even though these assumptions are quite strong, we still only can obtain this rather weak asymptotic convergence result.¹

Theorem 8.2.2 (Bertsekas and Tsitsiklis [1996], proposition 6.2). *Under Assumption 8.2.1*

$$\limsup_{k \rightarrow \infty} \|V^{\pi_k} - V^*\|_\infty \leq \frac{\delta + 2\gamma\epsilon}{(1-\gamma)^2}. \tag{8.2.3}$$

8.2.2 Estimation building blocks

Look-ahead policies

Given an approximate value function \mathbf{u} , the transition model of the MDP P_μ and expected rewards r_μ we can always find the improving policy given in Def. 8.1.1 via the following single-step look-ahead.

¹The keen reader will note that, when $\delta = 0$, this is identical to the result for ϵ -equivalent MDPs by Even-Dar and Mansour [2003] for $\delta = 0$.

Single-step look-ahead

$$\pi_{\mathbf{q}}(a \mid i) > 0 \quad \text{iff } a \in \arg \max_{a' \in \mathcal{A}} q(i, a') \quad (8.2.4)$$

$$q(i, a) \triangleq r_\mu(i, a) + \gamma \sum_{j \in \mathcal{S}} P_\mu(j \mid i, a) \mathbf{u}(j). \quad (8.2.5)$$

We are however not necessarily limited to the first-step. By looking T steps forward into the future we can improve both our value function and policy estimates.

 T -step look-ahead

$$\pi(i; \mathbf{q}_T) = \arg \max_{a \in \mathcal{A}} q_T(i, a), \quad (8.2.6)$$

where \mathbf{u}_k is recursively defined as:

$$q_k(i, a) = r_\mu(i, a) + \gamma \sum_{j \in \mathcal{S}} P_\mu(j \mid i, a) \mathbf{u}_{k-1}(j) \quad (8.2.7)$$

$$\mathbf{u}_k(i) = \max \{q_k(i, a) \mid a \in \mathcal{A}\} \quad (8.2.8)$$

and $\mathbf{u}_0 = \mathbf{u}$.

In fact, taking $\mathbf{u} = \mathbf{0}$, this recursion is identical to solving the k -horizon problem and at the limit we obtain solution to the original problem. In the general case, our value function estimation error is bounded by $\gamma^k \|\mathbf{u} - V^*\|$.

Rollout policies

As we have seen in Section 6.4.2 one way to obtain an the approximate value function of an arbitrary policy π is to use Monte Carlo estimation. That is, to simulate K sequences of state-action-reward tuples by running the policy on the MDP. More specifically, we have the following rollout estimate.

Rollout estimate of the q -factor

$$q(i, a) = \frac{1}{K_i} \sum_{k=1}^{K_i} \sum_{t=0}^{T_k-1} r(s_{t,k}, a_{t,k}),$$

where $s_{t,k}, a_{t,k} \sim \mathbb{P}_\mu^\pi(\cdot \mid s_0 = i, a_0 = a)$, and $T_k \sim \text{Geom}(1 - \gamma)$.

This results in a set of samples of q -factors. We now find a parametric policy that approximates the optimal policy with respect to our samples, $\pi_{\mathbf{q}}^*$. For a

finite number of actions, this fitting can be seen as a classification problem. Once more, we define a distribution ϕ on the states, over which we wish to perform the minimisation.

Rollout policy estimation.

Given a set of samples $q(i, a)$ for $i \in \hat{S}$, we estimate

$$\min_{\boldsymbol{\theta}} \|\pi_{\boldsymbol{\theta}} - \pi_{\mathbf{q}}^*\|_{\phi},$$

for some ϕ on \hat{S} .

In this setting, when the policy is over a discrete action space, the minimisation is essentially a classification problem. For continuous actions, it becomes a regression problem.

8.2.3 The value estimation step

We can now attempt to fit a parametric approximation to a given value function \mathbf{v} or \mathbf{q} . The simplest way to do so is via a generalised linear model. A natural parameterisation for the value function is to use a generalised linear model on a set of features. Then the value function is a linear function of the features with parameters $\boldsymbol{\theta}$. More precisely, we can define the following model.

Generalised linear model using features (or kernel)

Feature mapping $f : \mathcal{S} \rightarrow \mathbb{R}^n$, parameters $\boldsymbol{\theta} \in \mathbb{R}^n$.

$$\mathbf{v}_{\boldsymbol{\theta}}(s) = \sum_{i=1}^n \theta_i f_i(s) \quad (8.2.9)$$

In order to fit a value function, we first pick a set of *representative states* \hat{S} to fit our value function $\mathbf{v}_{\boldsymbol{\theta}}$ to \mathbf{v} . We can then estimate the optimal parameters via gradient descent.

Fitting a value function to a target.

$$c(\boldsymbol{\theta}) = \sum_{s \in \hat{S}} c_s(\boldsymbol{\theta}), \quad c_s(\boldsymbol{\theta}) = \phi(s) \|\mathbf{v}_{\boldsymbol{\theta}}(s) - \mathbf{v}(s)\|_p^\kappa. \quad (8.2.10)$$

This type of estimation can be seen as a simple regression. Indeed, it simply to implement a concrete example using gradient descent, as seen below.

EXAMPLE 40 (Gradient descent for $p = 2$, $\kappa = 2$). In this case the square root and κ cancel out and we obtain

$$\nabla_{\boldsymbol{\theta}} c_s = \phi(s) \sum_{a \in \mathcal{A}} \nabla_{\boldsymbol{\theta}} [\mathbf{v}_{\boldsymbol{\theta}}(s) - \mathbf{v}(s)]^2 = 2[\mathbf{v}_{\boldsymbol{\theta}}(s) - \mathbf{v}(s)] \nabla_{\boldsymbol{\theta}} \mathbf{v}_{\boldsymbol{\theta}},$$

where $\nabla_{\boldsymbol{\theta}} \mathbf{v}_{\boldsymbol{\theta}}(s) = f(s)$. Taking partial derivatives $\partial/\partial\theta_j$, leads to the update rule:

$$\theta'_j = \theta_j - 2\alpha\phi(s)[\mathbf{v}_{\boldsymbol{\theta}}(s) - \mathbf{v}(s)]f_j(s). \quad (8.2.11)$$

However, the value function is not necessarily self-consistent, meaning that for the given policy π , (which may be the optimal policy), we do not have the identity $\mathbf{v}_{\boldsymbol{\theta}}(s) = \mathbf{r}(s) + \int_{\mathcal{S}} \mathbf{v}_{\boldsymbol{\theta}}(s') dP s' sa$. For that reason, we can instead choose a parameter that minimises this error:

Minimising the Bellman error.

$$\inf_{\boldsymbol{\theta}} \left\| \rho(s) + \gamma \int_{\mathcal{S}} \mathbf{v}_{\boldsymbol{\theta}}(s') d\hat{P}(s' | s, a) - \mathbf{v}_{\boldsymbol{\theta}}(s) \right\|_{\phi}. \quad (8.2.12)$$

Here \hat{P} is not necessarily the true transition kernel. It can be a model or an empirical approximation (in which case the integral would only be over the empirical support). The summation itself is performed with respect to the measure ϕ

In this chapter, we will look at two methods for approximately minimising the Bellman error. The first, least square policy iteration is a batch algorithm for approximate policy iteration and finds the least-squares solution to the problem using the empirical matrix. The second is a gradient based method, which is flexible enough to use either a model or the empirical transition kernel.

8.2.4 Policy estimation

A natural parameterisation for the policy is to use a generalised linear model on a set of features. Then the policy can be described (up to scaling) as a linear function of the features with parameters $\boldsymbol{\theta}$. More precisely, we can define the following model.

Generalised linear model using features (or kernel).

Feature mapping $f : \mathcal{S} \rightarrow \mathbb{R}^n$, parameters $\boldsymbol{\theta} \in \mathbb{R}^n$.

$$\pi_{\boldsymbol{\theta}}(a | s) = \frac{g(s, a)}{h(s)}, \quad g(s, a) = \sum_{i=1}^n \theta_i f_i(s, a), \quad h(s) = \sum_{b \in \mathcal{A}} g(s, b) \quad (8.2.13)$$

We are performing the intermediate step of estimating g first, because we need to make sure that the policy is a distribution over actions. An alternative method would be to directly constrain the policy parameters so the result is always a distribution, but that would require a more complex optimisation method.

In order to fit a policy, we first pick a set of representative states $\hat{\mathcal{S}}$ and then we find a $\pi_{\boldsymbol{\theta}}$ that approximates π . In order to do so, we can define an appropriate cost function and then estimate the optimal parameters via some arbitrary optimisation method.

Fitting a policy through a cost function.

$$c(\boldsymbol{\theta}) = \sum_{s \in \hat{S}} c_s(\boldsymbol{\theta}), \quad c_s(\boldsymbol{\theta}) = \phi(s) \|\pi_{\boldsymbol{\theta}}(\cdot | s) - \pi(\cdot | s)\|_p^{\kappa}. \quad (8.2.14)$$

The function $\phi : \mathcal{S} \rightarrow \mathbb{R}_+$ is a weighting on the state space, such that we put more weight in more “important” states. Choosing the weights and the set of representative states \hat{S} is an interesting problem. A good choice is to relate those to the state distribution under different policies. One method to minimise the cost function is to use gradient descent. The gradient of this cost function is $\nabla_{\boldsymbol{\theta}} c = \sum_{s \in \hat{S}} \nabla_{\boldsymbol{\theta}} c_s$. We obtain different results for different norms, but there are three cases of main interest: $p = 1, p = 2, p \rightarrow \infty$. We present the first one here, and leave the others as an exercise.

The case $p = 1, \kappa = 1$.

EXAMPLE 41. The derivative can be written as:

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} c_s &= \phi(s) \sum_{a \in \mathcal{A}} \nabla_{\boldsymbol{\theta}} |\pi_{\boldsymbol{\theta}}(a | s) - \pi(a | s)|, \\ \nabla_{\boldsymbol{\theta}} |\pi_{\boldsymbol{\theta}}(a | s) - \pi(a | s)| &= \nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(a | s) \operatorname{sgn}[\pi_{\boldsymbol{\theta}}(a | s) - \pi(a | s)] \end{aligned}$$

The policy derivative in turn is

$$\pi_{\boldsymbol{\theta}}(a | s) = \frac{h(s) \nabla_{\boldsymbol{\theta}} g(s, a) - \nabla_{\boldsymbol{\theta}} h(s) g(s, a)}{h(s)^2},$$

with $\nabla_{\boldsymbol{\theta}} h(s) = (\sum_{b \in \mathcal{A}} f_i(s, b))_i$ and $\nabla_{\boldsymbol{\theta}} g(s, a) = f(s, a)$. Taking partial derivatives $\partial/\partial \theta_j$, leads to the update rule:

$$\theta'_j = \theta_j - \alpha \phi(s) \left(\pi_{\boldsymbol{\theta}}(a | s) \sum_{b \in \mathcal{A}} f_j(s, b) - f_j(s, a) \right). \quad (8.2.15)$$

Iterating over (s, a) pairs with a decreasing step-size α according to the stochastic approximation assumptions, should ensure convergence.

Alternative cost functions. It is frequently a good idea to add a *penalty term* to the cost function. The purpose of this is to prevent overfitting of the parameters to a small number of observations. Frequently, this is done by constraining the parameters to be small, via a penalty term of the form $\|\boldsymbol{\theta}\|^q$.

8.2.5 Rollout-based policy iteration methods

One idea for estimating the value function is to simply perform rollouts, while the policy itself is estimated in parametric form, as suggested in Bertsekas and Tsitsiklis [1996]. The first practical algorithm in this direction was Rollout

Sampling Approximate Policy iteration Dimitrakakis and Lagoudakis [2008b]. The main idea is to concentrate rollouts in interesting parts of the state space.

Algorithm 21 Rollout Sampling Approximate Policy Iteration.

```

for  $k = 1, \dots$  do
    Select a set of representative states  $\hat{S}_k$ 
    for  $n = 1, \dots$  do
        Select a state  $s_n \in \hat{S}_k$  maximising  $U_n(s)$  and perform a rollout.
        If  $\hat{a}^*(s_n)$  is optimal w.p.  $1 - \delta$ , put  $s_n$  in  $\hat{S}_k(\delta)$  and remove it from  $\hat{S}_k$ .
    end for
    Calculate  $\mathbf{q}_k \approx Q^{\pi_k}$  from the rollouts.
    Train a classifier  $\pi_{\theta_{k+1}}$  on the set of states  $\hat{S}_k(\delta)$  with actions  $\hat{a}^*(s)$ .
end for

```

The main idea is to concentrate rollouts on promising states. We can use the empirical state distribution to select starting states. We always choose the state s with the highest upper bound $U_n(s)$. More specifically, we employ a Hoeffding bound to select the state with the largest gap between actions. We stop rolling out states where we are certain to have found the best action. This is done by applying the Hoeffding bound to gaps between actions.

8.2.6 Least Squares Methods

The main idea is to formulate the problem in linear form, using a feature mapping that projects individual states (or state action pairs) onto a high-dimensional space. Then the value function is a linear function of the parameters and this mapping, minimising a squared error over the observed trajectories.

To get an intuition for these methods, recall that the solution of

$$\mathbf{v} = \mathbf{r} + \gamma \mathbf{P}_{\mu, \pi} \mathbf{v} \quad (8.2.16)$$

is the value function of π and can be obtained via

$$\mathbf{v} = (\mathbf{I} - \gamma \mathbf{P}_{\mu, \pi})^{-1} \mathbf{r}. \quad (8.2.17)$$

However, in this setting, we do not have access to the transition matrix. In addition, when the state space is continuous (e.g. $\mathcal{S} \subset \mathbb{R}^n$), the transition matrix becomes a general transition kernel. In addition, while up to now the set of value functions \mathcal{V} was a Euclidean subset, now \mathcal{V} becomes a Hilbert space.

In general, we deal with this case via projections. We project down from the infinite-dimensional Hilbert space to one with finite-dimensions. We assume that there is a projection that is complex enough for us to be able to recover the original value function sufficiently well.

Projection.

Setting $\mathbf{v} = \Phi \boldsymbol{\theta}$ where Φ is a feature matrix and $\boldsymbol{\theta}$ is a parameter vector we have

$$\Phi \boldsymbol{\theta} = \mathbf{r} + \gamma \mathbf{P}_{\mu, \pi} \Phi \boldsymbol{\theta} \quad (8.2.18)$$

$$\boldsymbol{\theta} = [(\mathbf{I} - \gamma \mathbf{P}_{\mu, \pi}) \Phi]^{-1} \mathbf{r} \quad (8.2.19)$$

However, generally the value function space generated by the features and the linear parameterisation does not allow us to obtain exact value functions. For this reason we replace the inverse with the *pseudo-inverse*,

$$\tilde{\mathbf{A}}^{-1} \triangleq \mathbf{A}^\top (\mathbf{A}\mathbf{A}^\top)^{-1}, \quad \mathbf{A} = (\mathbf{I} - \gamma \mathbf{P}_{\mu,\pi})\boldsymbol{\Phi}$$

which gives us an estimate for the parameters. If the inverse exists, then it is equal to the pseudo-inverse. The main idea that makes this work is to calculate everything on the empirical transition matrix, the empirical rewards and the empirical feature vectors.

Empirical constructions.

Given a set of data points $\{(s_i, a_i, r_i, s'_i) \mid i = 1, \dots, n\}$, which may not be consecutive, we define:

1. $\mathbf{r} = (r_i)_i$.
2. $\boldsymbol{\Phi}_i = f(s_i, a_i)$, $\boldsymbol{\Phi} = (\boldsymbol{\Phi}_i)_i$.
3. $\mathbf{P}_{\mu,\pi} = \mathbf{P}_\mu \mathbf{P}_\pi$, $\mathbf{P}_{\mu,\pi}(i, j) = \mathbb{I}\{j = i + 1\}$

Let us now describe LSTDQ, an algorithm that estimates an approximate value function for some policy π given some data D and a feature mapping f .

Algorithm 22 LSTDQ - Least Squares Temporal Differences on \mathbf{q} -factors

input data $D = \{(s_i, a_i, r_i, s'_i) \mid i = 1, \dots, n\}$, feature mapping f , policy π
 $\boldsymbol{\theta} = (\boldsymbol{\Phi}(\mathbf{I} - \widetilde{\gamma \mathbf{P}_{\mu,\pi}}))^{-1} \mathbf{r}$

In LSTDQ, shown in Algorithm 22, we maintain \mathbf{q} -factors, so that $\mathbf{q}(s, a) = f(s, a)\boldsymbol{\theta}$, and use the empirical transition matrix defined above. This algorithm is sufficient for performing approximate policy iteration by plugging it into the generic API algorithm to estimate a value function. Since LSTDQ returns \mathbf{q} -factors, our next policy can simply be greedy with respect to the value estimates.

Algorithm 23 LSPI - Least Squares Policy Iteration

input data $D = \{(s_i, a_i, r_i, s'_i) \mid i = 1, \dots, n\}$, feature mapping f
Set π_0 arbitrarily.
for $k = 1, \dots$ **do**
 $\boldsymbol{\theta}_k = LSTDQ(D, f, \pi_{k-1})$.
 $\pi_k = \pi_{\boldsymbol{\Phi}\boldsymbol{\theta}_k}^*$.
end for

8.3 Approximate Value Iteration

Approximate algorithms can also be defined for backwards induction. The general algorithmic structure remains the same. We only need to replace the exact steps with approximations. Usually this is necessary when the value function

cannot be updated everywhere exactly, possibly because our value function representations are not complex enough to capture the true value function.

8.3.1 Approximate backwards induction

The first algorithm is approximate backwards induction. Let us start with the basic backwards induction algorithm:

$$V_t^*(s) = \max_{a \in \mathcal{A}} \{ r(s, a) + \gamma \mathbb{E}_\mu (V_{t+1}^* | s_t = s, a_t = a) \} \quad (8.3.1)$$

This is essentially the same both for finite and infinite-horizon problems. Now assume that the set of functions \mathcal{V} that you can use to approximate the value functions is not rich enough, so none of its members will correspond to the left side of (8.3.1). Consider then the following value function approximation.

Let our estimate at time t be $\mathbf{v}_t \in \mathcal{V}$, with \mathcal{V} being a set of parameterised functions. Let \hat{V}_t be our one-step update given the value function approximation at the next step, \mathbf{v}_{t+1} . Then \mathbf{v}_t will be the closest approximation in that set.

Iterative approximation

$$\hat{V}_t(s) = \max_{a \in \mathcal{A}} \left\{ r(s, a) + \gamma \sum_{s'} \mathbf{P}_\mu(s' | s, a) \mathbf{v}_{t+1}(s') \right\} \quad (8.3.2)$$

$$\mathbf{v}_t = \arg \min \left\{ \|\mathbf{v} - \hat{V}_t\| \mid \mathbf{v} \in \mathcal{V} \right\} \quad (8.3.3)$$

Any algorithm can be used to perform the above minimisation, including gradient descent. Now consider the case where \mathbf{v} is a parameterised function with parameters $\boldsymbol{\theta}$. Then it is sufficient for us to maintain the parameter $\boldsymbol{\theta}_t$ at time t . These can be updated with a gradient scheme at every step. In the online case, our next-step estimates can be given by gradient descent:

Online gradient estimation

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha_t \nabla_{\boldsymbol{\theta}} \|\mathbf{v}_t - \hat{V}_t\| \quad (8.3.4)$$

This gradient descent algorithm can also be made stochastic, if we sample from the probability distribution given in the iterative approximation. The next sections give some examples.

8.3.2 State aggregation

Partitions, or tiling of the state space, inevitably lead to what is called *state aggregation*. That is, multiple different states are seen as identical by the algorithm. Unfortunately, it is very rarely the case that aggregated states really

are identical. Nevertheless, as we can see in the example below, aggregation significantly simplifies the estimation problems.

Aggregated estimate.

Let $\mathcal{G} = \{S_0, S_1, \dots, S_n\}$ be a partition of \mathcal{S} , with $S_0 = \emptyset$ and $\boldsymbol{\theta} \in \mathbb{R}^n$ and let $f_k(s_t) = \mathbb{I}\{s_t \in S_k\}$. Then the approximate value function is

$$\mathbf{v}(s) = \boldsymbol{\theta}(k), \quad \text{if } s \in S_k, k \neq 0. \quad (8.3.5)$$

That is, the value of every state corresponds to the value of the k -th set in the partition. Of course, this is only a very rough approximation if the sets S_k are very large. However, this is a very nice approach to use for gradient descent updates, as only one parameter needs to be updated at every step.

Online gradient estimate.

Consider the case $\|\cdot\| = \|\cdot\|_2^2$. For $s_t \in S_k$:

$$\boldsymbol{\theta}_{t+1}(k) = (1 - \alpha)\boldsymbol{\theta}_t(k) + \alpha \max_{a \in \mathcal{A}} r(s_t, a) + \gamma \sum_j P(j | s_t, a) \mathbf{v}_t(s) \quad (8.3.6)$$

For $s_t \notin S_k$:

$$\boldsymbol{\theta}_{t+1}(k) = \boldsymbol{\theta}(k). \quad (8.3.7)$$

Of course, whenever we perform the estimation online, we are limited to estimation on the sequence of states s_t that we visit. Consequently, estimation on other states may not be very good. It is indeed possible that we will suffer from oscillation problems.

8.3.3 Representative state approximation

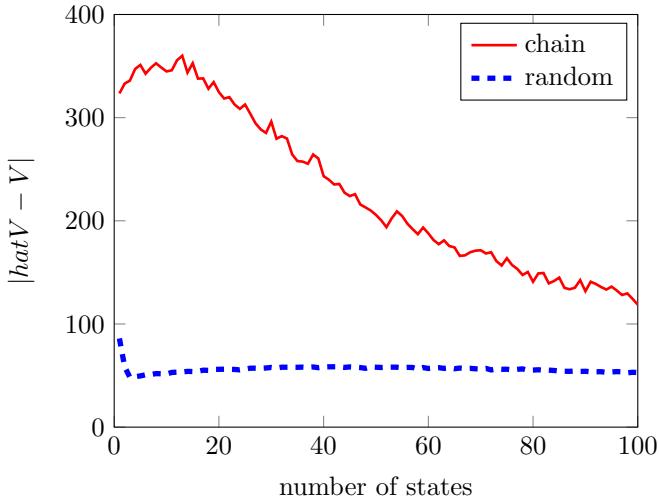
A rather different idea is to choose only some representative states on which to perform the approximation. The main assumption is that the value of all other states can be represented as a convex combination of the value of the representative states.

Representative states approximation.

Let $\hat{\mathcal{S}}$ be a set of n representative states and $\boldsymbol{\theta} \in \mathbb{R}^n$ and a feature mapping f :

$$\sum_{i=1}^n f_i(s) = 1, \quad \forall s \in \mathcal{S}.$$

The feature mapping is used to perform the convex combination. For any given state s , it has higher value for representative states i which are “closer”



to it. In general, the feature mapping is fixed, and we just want to find a set of parameters for the values of the representative states.

We focus here on the online estimate. At time t , for each representative state i , we obtain a new estimate of its value function and plug it back in.

Representative state update.

For $i \in \hat{S}$:

$$\boldsymbol{\theta}_{t+1}(i) = \max_{a \in \mathcal{A}} \left\{ r(i, a) + \gamma \int \mathbf{v}_t(s) dP(s | i, a) \right\} \quad (8.3.8)$$

with

$$\mathbf{v}_t(s) = \sum_{i=1}^n f_i(s) \boldsymbol{\theta}_t(i). \quad (8.3.9)$$

When the summation is not possible, we may instead approximate the expectation with a Monte-Carlo method. One particular problem with this method arises when the transition kernel is very sparse. Then we are basing our estimates on approximate values of other states, which may be very far from any other representative state.

8.3.4 Bellman error methods

The problems with the representative state update can be alleviated through Bellman error minimisation. The idea here is to obtain as a *consistent* value function as possible. The basic Bellman error minimisation is as follows:

$$\min_{\boldsymbol{\theta}} \|\mathbf{v}_{\boldsymbol{\theta}} - \mathcal{L}\mathbf{v}_{\boldsymbol{\theta}}\| \quad (8.3.10)$$

This is different from the approximate backwards induction algorithm we saw previously, since the same parameter $\boldsymbol{\theta}$ appears in both sides of the equality.

Furthermore, if the norm has support in all of the state space and the approximate value function space contains the actual set of value functions then the minimum is 0 and we obtain the optimal value function.

Gradient update.

When the norm is

$$\|\mathbf{v}_\theta - \mathcal{L}\mathbf{v}_\theta\| = \sum_{s \in \hat{\mathcal{S}}} D_\theta(s)^2, \quad D_\theta(s) = \mathbf{v}_\theta(s) - \max_{a \in \mathcal{A}} \int_{\mathcal{S}} \mathbf{v}_\theta(j) dP(j | s, a). \quad (8.3.11)$$

then the gradient update becomes

$$\theta_{t+1} = \theta_t - \alpha D_{\theta_t}(s_t) \nabla_\theta D_{\theta_t}(s_t) \quad (8.3.12)$$

$$\nabla_\theta D_{\theta_t}(s_t) = \nabla_\theta \mathbf{v}_{\theta_t}(s_t) - \int_{\mathcal{S}} \nabla_\theta \mathbf{v}_{\theta_t}(j) dP(j | s_t, a_t^*) \quad (8.3.13)$$

$$a_t^* = \arg \max_{a \in \mathcal{A}} \left\{ r(s_t, a) + \gamma \int_{\mathcal{S}} \mathbf{v}_\theta(j) dP(j | s_t, a) \right\} \quad (8.3.14)$$

We can also construct a Q -factor approximation for the case where no model is available. This is going to be simply done by replacing P with the empirical transition observed at time t .

Consider the inverted pendulum example again. Now, however, we are attempting to find the optimal value function. In Figure 8.7, we see an estimate of the optimal value function for the

8.4 Policy gradient

In the previous section, we saw how we could use gradient methods for value function approximation. However, it is also possible to do the same thing for estimating policies – the only necessary ingredient is a policy representation, and a way to evaluate a policy. The representation is usually parametric, but non-parametric representations are also possible. A common choice for parameterised policies is to use a feature function $f : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{X}$ and a linear parameterisation leading to a Gibbs distribution

$$\pi(a | s) = \frac{e^{F(s,a)}}{\sum_{a' \in \mathcal{A}} e^{F(s,a')}}, \quad F(s, a) \triangleq \theta^\top f(s, a). \quad (8.4.1)$$

As usual, we would like to find a policy maximising expected utility. Policy gradient algorithms employ gradient ascent on the expected utility to find a locally maximising policy. For the *average reward criterion*, the utility is

average reward criterion

$$U = \frac{1}{T} \sum_{t=0}^{T-1} r_t.$$

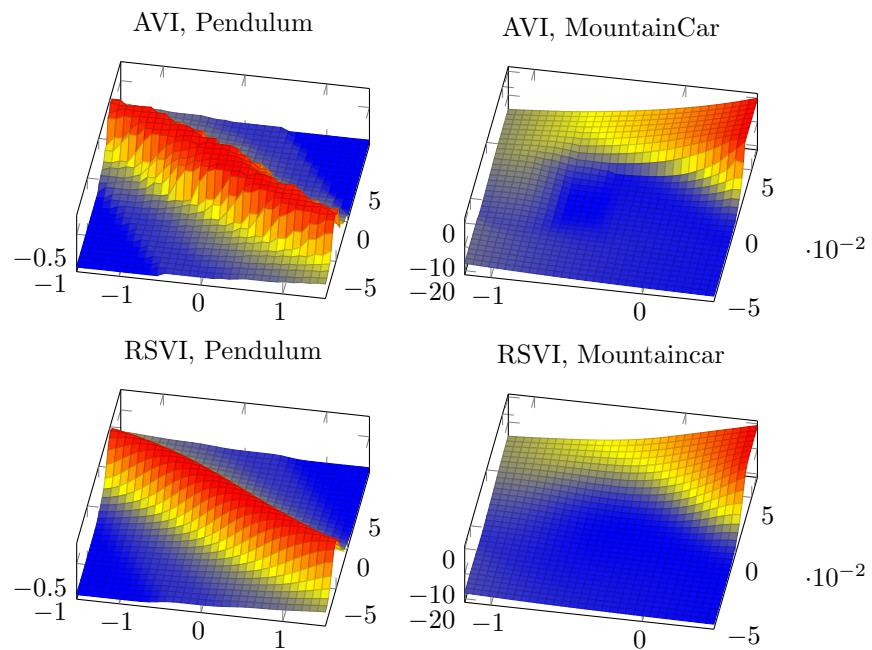


Figure 8.7: Estimated optimal value function for the pendulum problem. Results are shown for approximate value iteration (AVI) with a Bayesian linear-Gaussian model, and a representative state representation (RSVI) with an RBF embedding. Both the embedding and the states where the value function is approximated are a 16×16 uniform grid over the state space.

For the discounted reward criterion, the utility is defined as usual to be:

$$U = \sum_{t=0}^{\infty} \gamma^t r_t,$$

and we are interested in the expected utility from a starting state distribution \mathbf{y} . In either case, there are many simple expressions for the gradient of the expected utility, as can be seen below.

Policy gradient theorem

Theorem 8.4.1. *For any θ -parameterised policy space Π , the gradient of the utility from starting state distribution \mathbf{y} can be equivalently written in the three following forms:*

$$\nabla_{\theta} \mathbb{E}_{\mathbf{y}}^{\pi} U = \mathbf{y}^{\top} \gamma (\mathbf{I} - \gamma \mathbf{P}_{\mu}^{\pi})^{-1} \nabla_{\theta} \mathbf{P}_{\mu}^{\pi} (\mathbf{I} - \gamma \mathbf{P}_{\mu}^{\pi})^{-1} \mathbf{r} \quad (8.4.2)$$

$$= \sum_s x_{\mu, \mathbf{y}}^{\pi}(s) \sum_a \nabla_{\theta} \pi(a \mid s) Q_{\mu}^{\pi}(s, a) \quad (8.4.3)$$

$$= \sum_h U(h) \mathbb{P}_{\mu}^{\pi}(h) \nabla \ln \mathbb{P}_{\mu}^{\pi}(h), \quad (8.4.4)$$

where $h \in (\mathcal{S} \times \mathcal{A})^*$ is a state-action history and $\mathbb{P}_{\mu}^{\pi}(h)$ its probability under the policy π and MDP μ and $U(h)$ is the utility of history h .

Proof. We begin by proving the claim (8.4.2). Note that the

$$\mathbb{E}_{\mu}^{\pi} U = \mathbf{y}^{\top} (\mathbf{I} - \gamma \mathbf{P}_{\mu}^{\pi})^{-1} \mathbf{r}$$

where \mathbf{y} is a starting state distribution vector and \mathbf{P}_{μ}^{π} is the transition matrix resulting from applying policy π to μ . Then we can calculate the derivative of the above expression using matrix calculus, that is

$$\nabla_{\theta} \mathbb{E} U = \mathbf{y}^{\top} \nabla_{\theta} (\mathbf{I} - \gamma \mathbf{P}_{\mu}^{\pi})^{-1} \mathbf{r}$$

and then

$$\nabla_{\theta} (\mathbf{I} - \gamma \mathbf{P}_{\mu}^{\pi})^{-1} = -(\mathbf{I} - \gamma \mathbf{P}_{\mu}^{\pi})^{-1} \nabla_{\theta} (\mathbf{I} - \gamma \mathbf{P}_{\mu}^{\pi}) (\mathbf{I} - \gamma \mathbf{P}_{\mu}^{\pi})^{-1} \quad (8.4.5)$$

$$= \gamma (\mathbf{I} - \gamma \mathbf{P}_{\mu}^{\pi})^{-1} \nabla_{\theta} \mathbf{P}_{\mu}^{\pi} (\mathbf{I} - \gamma \mathbf{P}_{\mu}^{\pi})^{-1} \quad (8.4.6)$$

This concludes the proof of the first claim. We can also expand the \mathbf{P}_{μ}^{π} term, thus obtaining a formula that only has a derivative for π :

$$\frac{\partial}{\partial \theta_i} \mathbb{P}_{\mu}^{\pi}(s' \mid s) = \sum_a \mathbb{P}_{\mu}(s' \mid s, a) \frac{\partial}{\partial \theta_i} \pi(a \mid s). \quad (8.4.7)$$

Define the state visitation matrix $\mathbf{X} \triangleq (\mathbf{I} - \gamma \mathbf{P})^{-1}$ for simplicity, and obtain:

$$\nabla \mathbb{E}_{\mu}^{\pi} U = \gamma \mathbf{y}^{\top} \mathbf{X} \nabla_{\theta} \mathbf{P}_{\mu}^{\pi} \mathbf{X} \mathbf{r}. \quad (8.4.8)$$

We are now ready to prove claim (8.4.3). It is easy to see that $\mathbf{x} = \mathbf{y}^\top \mathbf{X}$, hence the above becomes:

$$\nabla \mathbb{E}_\mu^\pi U = \gamma \mathbf{y}^\top \nabla_\theta \mathbf{P}_\mu^\pi \mathbf{X} \mathbf{r} \quad (8.4.9)$$

$$= \gamma \sum_s y(s) \sum_{a,s'} \mathbf{P}_\mu(s' | s, a) \nabla_\theta \pi(a | s) V(s') \quad (8.4.10)$$

$$= \gamma \sum_s y(s) \sum_a \nabla_\theta \pi(a | s) \sum_{s'} \mathbf{P}_\mu(s' | s, a) V(s') \quad (8.4.11)$$

$$= \gamma \sum_s y(s) \sum_a \nabla_\theta \pi(a | s) Q(s, a). \quad (8.4.12)$$

We thus obtain the first well-known policy gradient theorem. Note here that if the policy is deterministic, we can simply drop the marginalisation over actions.

The last claim (8.4.4) is straightforward.

$$\nabla \mathbb{E} U = \sum_h U(h) \nabla \mathbb{P}(h) = \sum_h U(h) \mathbb{P}(h) \nabla \ln \mathbb{P}(h), \quad (8.4.13)$$

as $\nabla \ln \mathbb{P}(h) = \frac{1}{\mathbb{P}(h)} \nabla \mathbb{P}(h)$. \square

8.4.1 Specific instantiations.

The meaning of some of these variables depends on the definition of the utility. For the average reward criterion, \mathbf{x}_π is the stationary state distribution:

$$x_\pi(s) = \lim_{t \rightarrow \infty} \sum_{s'} \mathbb{P}_\mu^\pi(s_t = s | s_0 = s') y(s'),$$

For the discounted reward criterion, \mathbf{x}_π is the cumulative discounted state occupancy vector:

$$x_\pi(s) = \sum_{s'} \sum_{t=0}^{\infty} \gamma^t \mathbb{P}_\mu^\pi(s_t = s | s_0 = s') y(s').$$

We can obtain \mathbf{x}_π through state occupancy matrix (6.5.5) by left multiplying the latter with the initial state distribution \mathbf{y} . However, in the context of gradient methods, it makes more sense to use a stochastic estimate of \mathbf{x}_π to calculate the gradient, since:

$$\nabla_\theta \mathbb{E}^\pi U = \mathbb{E}_y^\pi \sum_a \nabla_\theta \pi(a | s) Q^\pi(s, a). \quad (8.4.14)$$

For the average reward criterion, the expectation is taken for the stationary state distribution. Hence we need a sufficiently long chain to approximate this with a Monte Carlo estimate. For the discounted reward criterion, we can easily obtain unbiased samples through geometric stopping (see exercise 23).

Importance sampling.

The last formulation is especially useful as it allows us to use importance sampling rather easily. First note that for any history $h = (s^T, a^T)$ we have,

$$\mathbb{P}_\mu^\pi(h) = \prod_{t=1}^T \mathbb{P}_\mu(s_t | s^{t-1}, a^{t-1}) \mathbb{P}^\pi(a_t | s^t, a^{t-1}) \quad (8.4.15)$$

without any Markovian assumptions on the model or policy. By importance sampling, we can write (8.4.13) as

$$\nabla \mathbb{E}_\mu^\pi U = \mathbb{E}_{\mu'}^{\pi'} \left(U(h) \nabla \ln \mathbb{P}_\mu^\pi(h) \frac{\mathbb{P}_\mu^\pi(U)}{\mathbb{P}_\mu^{\pi'}(U)} \right) \quad (8.4.16)$$

$$= \mathbb{E}_{\mu'}^{\pi'} \left(U(h) \nabla \ln \mathbb{P}_\mu^\pi(h) \prod_{t=1}^T \frac{\mathbb{P}^\pi(a_t | s^t, a^{t-1})}{\mathbb{P}^{\pi'}(a_t | s^t, a^{t-1})} \right), \quad (8.4.17)$$

as the μ -dependent terms in (8.4.15) cancel out. In practice the expectation would be approximated through sampling trajectories h . Finally, since \mathbb{P}_μ does not depend on π , we obtain:

$$\nabla \ln \mathbb{P}_\mu^\pi(h) = \sum_t \nabla \ln \mathbb{P}^\pi(a_t | s^t, a^{t-1}) = \sum_t \frac{\nabla \mathbb{P}^\pi(a_t | s^t, a^{t-1})}{\mathbb{P}^\pi(a_t | s^t, a^{t-1})}.$$

8.4.2 Practical considerations.

The main problem we face is how to parameterise the policy. For the discrete case, a common parameterisation is to have a separate and independent parameter for each state-action pair, i.e. $\theta_{s,a} = \pi(a|s)$. This leads to a particularly simple expression for the second form, which is $\partial/\partial\theta_{s,a} \mathbb{E}_\mu^\pi U = y(s)Q(s,a)$. However, it is easy to see that in this case the parameterisation will lead to all parameters increasing if rewards are positive. This can be avoided by either a Softmax parameterisation or by subtracting a bias term² from the derivative. Nevertheless, this parameterisation implies stochastic discrete policies.

We could also suitably parameterise continuous policies. For example, we can consider a linear policy. Apart from the summations, there is nothing in the above equations that limits us to discrete spaces. Most of the derivation carries over to Euclidean state-action spaces. The only question is how to parameterise the policy, but nothing precludes the use of continuous policies. In particular, the second form is also suitable for deterministic policies.

Finally, in practice, we may not need to explicitly calculate the expectations. Sample trajectories are sufficient to update the gradient in a meaningful way, especially for the third form, as we can naturally sample from the distribution of trajectories. However, the fact that this form doesn't need a Markovian assumption also means that it cannot take advantage of Markovian environments.

Policy gradient methods are useful, especially in cases where the environment model or value function can be extremely complicated, while the optimal policy itself might be quite simple. The main difficulty lies in obtaining an appropriate estimate of the gradient itself, but convergence is generally good as long as we are adjusting the parameters in a gradient-related direction.

8.5 Further reading

Among value function approximation methods, the two most well known are fitted Q-iteration Antos et al. [2008b], and fitted value iteration, which has been analysed in Munos and Szepesvári [2008]. Minimising the Bellman error Antos

²e.g. $Q_\mu^\pi(s, a_1)$

et al. [2008a], Dimitrakakis [2013], Ghavamzadeh and Engel [2006] is generally a good way to ensure that approximate value iteration is stable.

In approximate policy iteration methods, one needs to approximate both the value function and policy. In rollout sampling policy iteration Dimitrakakis and Lagoudakis [2008b,a], an empirical approximation of the value function is maintained. However, one can employ least-squares methods Bradtke and Barto [1996], Boyan [2002], Lagoudakis and Parr [2003] for example.

The general technique of state aggregation Singh et al. [1995], Bernstein [2007] is applicable to a variety of reinforcement learning algorithms. While the more general question of selecting features appropriately is open, there has been some progress in the domain of feature reinforcement learning Hutter [2009]. In general, learning internal representations (i.e. features) has been a prominent aspect of neural network research Rumelhart et al. [1987]. Even if it is unclear to what extent recently proposed approximations architectures that employ deep learning actually learn any useful representations, they have been successfully used in combination with simple reinforcement learning algorithms Mnih et al. [2015]. Another interesting direction is links between features and approximately sufficient statistics Dimitrakakis and Tziortziotis [2013, 2014].

Finally, The policy gradient theorem in the state visitation form was first proposed by Sutton et al. [1999], while Williams [1992] was the first to use the log-ratio trick in reinforcement learning. To our knowledge, the analytical gradient has not actually been applied (or indeed, described) in prior literature. Extensions of the policy gradient idea are also natural. They have also been used in a Bayesian setting by Ghavamzadeh and Engel [2006], while the natural gradient has been proposed by Kakade [2002]. A survey of policy gradient methods can be found in Peters and Schaal [2006].

8.6 Exercises

EXERCISE 29 (Enlarging the function space.). Consider the problem in example 36. What would be a simple way to extend the space of value functions from the three given candidates to an infinite number of value functions? How could we get a good fit?

EXERCISE 30 (Enlarging the policy.). Consider example 37. This represents and example of a linear deterministic policies. In which two ways can this policy space be extended and how?

EXERCISE 31. Find the derivative for the two other cases, specifically:

1. $p = 2, \kappa = 2$.
2. $p \rightarrow \infty, \kappa = 1$.

Solution. For $p = 2, \kappa = 2$, the derivative can be written as:

□

Chapter 9

Bayesian reinforcement learning

9.1 Introduction

Bayesian reinforcement learning connects all elements previously seen in the book. Firstly, how to express uncertainty and preferences via probabilities and utilities. Secondly, how to make decisions under uncertainty, and in particular how to maximise expected utility. Thirdly, how to adjust our subjective belief in the face of new evidence. Fourthly, optimal experiment design: how to make decisions in problems where our decisions can affect the evidence we obtain. These problems can be modelled as Markov decision processes. We also consider the problem of finding optimal policies for Markov decision processes.

In the previous two chapters, we have considered stochastic algorithms for acting within Markov decision processes and approximation algorithms for value functions and policies. These stochastic analogues of exact deterministic MDP algorithms can also be used in the context of estimating the optimal policy while acting in the MDP itself, even if the MDP parameters are not known. This involves just running the algorithm in the real environment, rather than a simulation. In the case where the MDP is very large, or the state/action spaces are continuous, it is necessary to use an approximate representation for the value function, the policy, or both. These can be used in conjunction with stochastic approximations, but we no longer have guarantee of convergence to the optimal policy, even asymptotically.

In this chapter, we will come full circle to the setting of subjective probability and utility, by formalising the reinforcement learning problem as a Bayesian decision problem and solving it directly. In the Bayesian setting, we are acting in an MDP which is not known, but we have a subjective belief about what the environment is. We shall first consider the case of acting in unknown MDPs, which is the focus of the reinforcement learning problem. We will examine a few different heuristics for maximising expected utility in the Bayesian setting, and contrast them with tractable approximations to the Bayes-optimal solution. In Section 9.3, we will connect this problem to partially observable MDPs. Finally, we shall present extensions of these ideas to continuous domains.

9.2 Acting in unknown MDPs

The reinforcement learning problem can be formulated as the problem of learning to act in an unknown environment, only by interaction and reinforcement. All of those elements of the definition are important. Firstly and foremost it is a *learning* problem. Consequently, we have only partial prior knowledge about the environment we are acting in. This knowledge is arrived at via *interaction* with the environment. We do not have a fixed set of data to work with, but we must actively explore the environment to understand how it works. Finally, there is an intrinsic *reinforcement* that punishes some behaviours and rewards others. We can formulate some of these problems as Markov decision processes.

Let us begin by using an MDP μ to represent an environment, where at each time step t , we observe the environment's state $s_t \in \mathcal{S}$, take action $a_t \in \mathcal{A}$ and receive reward $r_t \in \mathbb{R}$. In this setting, the environment state and our action fully determines the distribution of the immediate reward, as well as that of the next state, as described in Definition 6.3.1. For a specific MDP μ is known, the probability of the immediate reward is given by $P_\mu(r_t | s_t, a_t)$ and that of

next state by $P_\mu(s_{t+1} | s_t, a_t)$. If these quantities are known analytically, or if we can at least draw samples from these distributions, it is possible to employ stochastic approximation and approximate dynamic programming to estimate the optimal policy and value function for the MDP.

More precisely, when μ is known, we wish to find a *policy* $\pi : \mathcal{S} \rightarrow \mathcal{A}$ maximising the *utility* in expectation. This requires us to solve the maximisation problem $\max_{\pi} \mathbb{E}_{\mu}^{\pi} U$, where the utility is an additive function of rewards, $U = \sum_{t=1}^T r_t$. When μ is *known*, we can use standard algorithms, such as value or policy iteration. However, knowing μ is contrary to the problem definition.

In Chapter 7 we have seen a number of stochastic approximation algorithms which allow us to learn the optimal policy for a given MDP eventually. However, these generally give few guarantees on the performance of the policy while learning. How can we create an algorithm for optimal learning MDPs? This should trade off exploring the environment to obtain further knowledge, and simultaneously exploiting its knowledge.

The solution is rather simple, conceptually. Within the subjective probabilistic framework, we only need to define a prior belief ξ on the set of MDPs \mathcal{M} , and then find the policy that maximises the expected utility with respect to the prior. The value of information is automatically taken into account in this model. The structure of the unknown MDP process is shown in Figure 9.1 below.

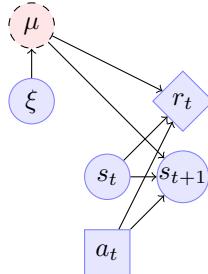


Figure 9.1: The unknown Markov decision process

The fact that we can recast the unknown MDP problem as a standard Bayesian decision problem should not be too surprising, as we have previously seen it in two Bayes-optimal constructions. The first was the simple optimal stopping procedure in Section 5.2.2, which introduced the backwards induction algorithm. The second was the optimal experiment design problem, which resulted in the bandit Markov decision process of Section 6.2. Let us now formulate the reinforcement learning problem as a Bayesian maximisation problem.

Let ξ be a prior over \mathcal{M} and Π be a set of policies. Then the expected utility of the optimal policy is:

$$U_{\xi}^* \triangleq \max_{\pi \in \Pi} \mathbb{E}(U | \pi, \xi) = \max_{\pi \in \Pi} \int_{\mathcal{M}} \mathbb{E}(U | \pi, \mu) d\xi(\mu) \quad (9.2.1)$$

Finding the optimal policy is not easy, as in general the optimal policy π must now map from *complete histories* to actions. It is a history-dependent policy, meaning that any action we take at step $t+k$ must depend on what we will have observed in steps $t, t+1, \dots, t+k$. Consequently, as *planning* means specifying

a policy from t to $t + k$, this policy must take into account the learning that will take place in this interval.

As our policies can be more complicated in this setting, it is useful to differentiate between different policy types. We use Π to denote the set of all policies. We use Π_k to denote the set of k -order Markov policies. Important special cases are the set of *blind* policies Π_0 and the set of *memoryless* policies Π_1 . A policy in $\pi \in \bar{\Pi}_k \subset \Pi_k$ is *stationary*, when $\pi(A | s_{t-k+1}^t, a_{t-k+1}^{t-1}) = \pi(A | s^k, a^{k-1})$ for all t . Finally, policies may be indexed by some parameter set Θ , in which case the set of parameterised policies is given by Π_Θ .

Generally speaking, the Bayes-optimal policies have to be history-dependent, as shown by the following counterexample.

EXAMPLE 42. Consider two MDPs, μ_1, μ_2 with states $\mathcal{S} = \{1\}$ and actions $\mathcal{A} = \{1, 2\}$. In the i -th MDP, whenever you take action $a_t = i$, you obtain reward $r_t = 1$, otherwise you obtain reward 0. The expected utility of a memoryless policy taking action i with probability $\pi(i)$ would be

$$\mathbb{E}_\xi^\pi U = T \sum_i \xi(\mu_i) \pi(i),$$

for horizon T . Consequently, if your prior is not uniform, you select the action corresponding to the MDP with the highest prior probability. Then, the maximal expected utility is:

$$\max_{\pi \in \Pi_1} \mathbb{E}_\xi^\pi U = T \max_i \xi(\mu_i).$$

In this case, we are certain which one is the right MDP as soon as we take one action. We can then follow the policy which selects the apparently best action at first, and then switches to the best action for the MDP we have seen. Then, our utility is simply $\max_i \xi(\mu_i) + (T - 1)$.

Can something be said about Bayes-optimal policies more generally? Given that they must be history-based, how much of the history must be retained? This must of course depend on the MDP set \mathcal{M} we are considering. To make a general statement, it is useful to consider policies that are based on statistics.

Theorem 9.2.1. *Let \mathcal{H} be the set of all histories. Consider the set of policies Π_ϕ , with each policy $\pi : \mathcal{H} \rightarrow \mathcal{A}$ factorised as $\pi = p\phi$, with $p : \Phi \rightarrow \mathcal{A}$ and $\phi : \mathcal{H} \rightarrow \Phi$ being a statistic. If the statistic is not sufficient, i.e. if there are histories $h, h' \in \mathcal{H}$ such that*

$$\phi(h) = \phi(h'), \quad \xi(\mu | h) \neq \xi(\mu | h')$$

for some ξ, μ , then $\max_{\pi \in \Pi} \mathbb{E}_\xi^\pi > \max_{\pi \in \Pi_\phi} \mathbb{E}_\xi^\pi$, unless there exists a set of policies that are optimal across all MDPs.

Proof. First we show that, for any non-trivial set of MDPs \mathcal{M} , there exist two histories h, h' such that:

$$\mathbb{E}_\xi^{\pi^*(h)}(U | h) > \mathbb{E}_\xi^{\pi^*(h')}(U | h).$$

To see this, note that otherwise there must be a common optimal policy set for all $\mu \in \mathcal{M}$. More specifically, if the above is an equality for all history pairs (h, h') , then $\pi^*(h') \in \arg \max_\pi \mathbb{E}_\xi^\pi(U | h) = \bigcup_{h \in \mathcal{H}} \{\pi^*(h)\}$. For a finite set of histories \mathcal{H} , we have

$$\max_{\pi \in \Pi} \mathbb{E}_\xi^\pi(U) = \max_{\pi \in \Pi} (\mathbb{E}_\xi^\pi(U | \neg h) P_\xi^\pi(\mathcal{H} \setminus h) + P_\xi^\pi(h) \mathbb{E}_\xi^\pi(U | h))$$

with the rightmost term satisfying $\max_{\pi \in \Pi} \mathbb{E}_\xi^\pi(U \mid h) > \max_{\pi \in \Pi_\phi} \mathbb{E}_\xi^\pi(U \mid h)$. \square

Let us now turn to the problem of how to construct an optimal policy. As the optimal policy must include learning, we must first examine how to update the belief. Given that, we shall examine methods for exact and approximate methods of policy optimisation.

9.2.1 Updating the belief

Strictly speaking, in order to update our belief, we must condition the prior distribution on all the information. This includes the sequence of observations up to this point in time, including the states s^t , actions a^{t-1} , and rewards r^{t-1} , as well as the policy π that we followed. Let $D_t = \langle s^t, a^{t-1}, r^{t-1} \rangle$ be the observed data to time t . Then the posterior measure is:

$$\xi(B \mid D_t, \pi) = \frac{\int_B \mathbb{P}_\mu^\pi(D_t) d\xi(\mu)}{\int_M \mathbb{P}_\mu^\pi(D_t) d\xi(\mu)}. \quad (9.2.2)$$

However, as we shall see in the following remark, we can usually¹ ignore the policy itself in the when calculating the posterior.

Remark 9.2.1. The dependence on the policy can be removed, since the posterior is the same for all policies that put non-zero mass on the observed data: Let $D_t \sim \mathbb{P}_\mu^\pi$. Then it is easy to see that $\forall \pi' \neq \pi$ such that $\mathbb{P}_\mu^{\pi'}(D_t) > 0$,

$$\xi(B \mid D_t, \pi) = \xi(B \mid D_t, \pi').$$

The proof is left as an exercise for the reader. In the specific case of MDPs, the posterior calculation is easy to perform incrementally. This also more clearly demonstrates why there is no dependence on the policy. Let ξ_t be the (random) posterior at time t . Then, the next-step belief is going to be:

$$\xi_{t+1}(B) \triangleq \xi(B \mid D_{t+1}) = \frac{\int_B \mathbb{P}_\mu^\pi(D_t) d\xi(\mu)}{\int_M \mathbb{P}_\mu^\pi(D_t) d\xi(\mu)} \quad (9.2.3)$$

$$= \frac{\int_B \mathbb{P}_\mu(s_{t+1}, r_t \mid s_t, a_t) \pi(a_t \mid s^t, a^{t-1}, r^{t-1}) d\xi(\mu \mid D_t)}{\int_M \mathbb{P}_\mu(s_{t+1}, r_t \mid s_t, a_t) \pi(a_t \mid s^t, a^{t-1}, r^{t-1}) d\xi(\mu \mid D_t)} \quad (9.2.4)$$

$$= \frac{\int_B \mathbb{P}_\mu(s_{t+1}, r_t \mid s_t, a_t) d\xi_t(\mu)}{\int_M \mathbb{P}_\mu(s_{t+1}, r_t \mid s_t, a_t) d\xi_t(\mu)} \quad (9.2.5)$$

The above calculation is easy to perform for arbitrarily complex MDPs when the set M is finite. The posterior calculation is also simple under certain conjugate priors, such as the Dirichlet-multinomial prior for transition distributions.

¹The exception involves any type of inference that uses simulated trajectories from past policies.

9.2.2 Finding Bayes-optimal policies

The problem of policy optimisation in the Bayesian case is much harder than in the known-MDP case. This is simply because of the history dependence, which has two effects. Firstly, it makes the policy space much larger, as we need to consider history dependent policies. However, even we consider only memoryless policies, it does not make dynamic programming easier.

In this section, we first consider two simple heuristics for finding optimal policies. Then we examine policies which try and construct upper and lower bounds on the expected utility. Finally, we consider finite look ahead backwards induction, that uses the same upper and lower bounds to perform efficient tree search.

The expected MDP heuristic

One simple heuristic is to simply calculate the expected MDP $\hat{\mu}_\xi \triangleq \mathbb{E}_\xi \mu$ for the belief ξ . In particular, the transition kernel of the expected MDP is simply the expected transition kernel:

$$\mathbb{P}_{\hat{\mu}_\xi}(s'|s, a) = \int_{\mathcal{M}} \mathbb{P}_\mu(s'|s, a) d\xi(\mu).$$

Then, we simply calculate the optimal memoryless policy for $\hat{\mu}_\xi$:

$$\pi^*(\hat{\mu}_\xi) \in \arg \max_{\pi \in \Pi_1} V_{\hat{\mu}_\xi}^\pi,$$

where $\Pi_1 = \{\pi \in \Pi \mid \mathbb{P}^\pi(a_t | s^t, a^{t-1}) = \mathbb{P}^\pi(a_t | s_t)\}$ is the set of Markov policies. Finally, we execute $\pi^*(\hat{\mu}_\xi)$ on the real MDP. The algorithm can be written as follows. Algorithm 24 gives the pseudocode for this heuristic. One important

Algorithm 24 The expected MDP heuristic

```

for  $k = 1, \dots$  do
     $\mu_k \triangleq \mathbb{E}_{\xi_{t_k}} \mu.$ 
     $\pi_k \approx \arg \max_\pi \mathbb{E}_{\mu_k}^\pi U.$ 
    for  $t = 1 + T_{k-1}, \dots, T_k$  do
        Observe  $s_t$ .
        Update belief  $\xi_t(\cdot) = \xi_{t-1}(\cdot | s_t, a_{t-1}, r_{t-1}, s_{t-1})$ .
        Take action  $a_t \sim \pi_k(a_t | s_t)$ .
        Observe reward  $r_t$ .
    end for
end for

```

detail is that we are only updating the k -th policy after T_k steps. This is sometimes useful to ensure policies remain consistent. It is natural to use T_k in the order of $1/(1 - \gamma)$ for discounted problems, or simply the length of the episode for episodic problems. In the undiscounted case, switching policies whenever sufficient information has been obtained to significantly change the belief gives good regret guarantees, as we shall see in Chapter 10.

Unfortunately, the policy returned with this heuristic may be far from the Bayes-optimal policy in Π_1 , as shown by the following counterexample.

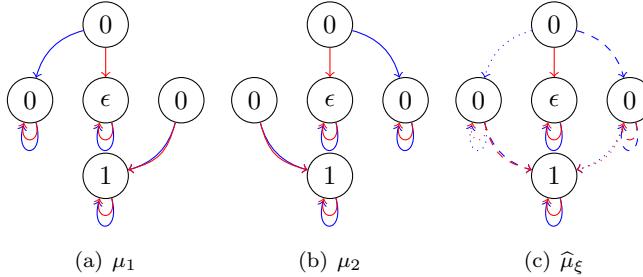


Figure 9.2: The two MDPs and the expected MDP from example 43

EXAMPLE 43 (Counterexample²). In this example, illustrated in Figure 9.2, $\mathcal{M} = \{\mu_1, \mu_2\}$ is the set of MDPs, and the belief is $\xi(\mu_1) = \theta$, $\xi(\mu_2) = 1 - \theta$. All transitions are deterministic, and there are two actions, the blue and the red action. When we calculate the expected MDP, we see that now the state with reward 1 is reachable.

Consequently, when $T \rightarrow \infty$, the $\hat{\mu}_\xi$ -optimal policy is not optimal in Π_1 if:

$$\epsilon < \frac{\gamma\theta(1-\theta)}{1-\gamma} \left(\frac{1}{1-\gamma\theta} + \frac{1}{1-\gamma(1-\theta)} \right)$$

In this example, $\hat{\mu}_\xi \notin \mathcal{M}$.

9.2.3 The maximum MDP heuristic

An alternative idea is to simply pick the maximum-probability MDP, as shown in Algorithm 25. This at least guarantees that the MDP that you are acting optimally for is actually within the set of MDPs. However, it may still be the case that the resulting policy is sub-optimal, as shown by the following counterexample.

Algorithm 25 The maximum MDP heuristic

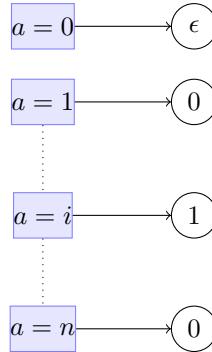
```

for  $k = 1, \dots$  do
     $\mu_k \triangleq \arg \max_\mu \xi_{t_k}(\mu).$ 
     $\pi_k \approx \arg \max_\pi \mathbb{E}_{\mu_k}^\pi U.$ 
    for  $t = 1 + T_{k-1}, \dots, T_k$  do
        Observe  $s_t$ .
        Update belief  $\xi_t(\cdot) = \xi_{t-1}(\cdot \mid s_t, a_{t-1}, r_{t-1}, s_{t-1}).$ 
        Take action  $a_t \sim \pi_k(a_t \mid s_t)$ .
        Observe reward  $r_t$ .
    end for
end for

```

EXAMPLE 44 (Counterexample for $\hat{\mu}_\xi^* \triangleq \arg \max_\mu \xi(\mu)$). Let the MDP set be $\mathcal{M} = \{\mu_i \mid i = 1, \dots, n\}$ with $\mathcal{A} = \{0, \dots, n\}$. In all MDPs, a_0 gives a reward of ϵ and the MDP terminates. In the i -th MDP, all other actions give you a reward of 0 apart from the i -th action which gives you a reward of 1. Then the MDP terminates. The MDP is visualised in Figure 9.3.

²Based on one by Remi Munos

Figure 9.3: The MDP μ_i from example 44

For this problem, the ξ -optimal policy takes action i iff $\xi(\mu_i) \geq \epsilon$, otherwise takes action 0. On the other hand, the $\widehat{\mu}_\xi^*$ -optimal policy takes $a = \arg \max_i \xi(\mu_i)$. Thus, this policy is sub-optimal if $\max_i \xi(\mu_i) < \epsilon$.

For smooth beliefs, $\widehat{\mu}_\xi$ is close to $\widehat{\mu}_\xi^*$, and in this case, those heuristics might be reasonable. However, they can be shown to be sub-optimal even for very simple stopping problems.

9.2.4 Bounds on the expected utility

Given that these heuristics are incorrect, what can we actually do? The first thing to try is to calculate the expected utility of some arbitrary policy. As it turns out, this operation is relatively simple in the Bayesian case, even when the set of MDPs is infinite.

Policy evaluation is particularly simple in Bayesian MDP problems for any fixed policy. We simply apply the basic utility theory definitions in order to calculate the expected utility of the policy under our belief. In the following, we will find it useful to also define the Bayes-value function of a policy π as the conditional expected utility under that policy and our belief ξ . Analogously to an MDP value function, we define.

Definition 9.2.1 (Bayesian value function π for a belief ξ).

$$V_\xi^\pi(s) \triangleq \mathbb{E}_\xi^\pi(U \mid s_t = s) \quad (9.2.6)$$

It is easy to see that the Bayes value function of a policy is simply the expected value functions under ξ :

$$V_\xi^\pi(s) = \int_{\mathcal{M}} \mathbb{E}_\mu^\pi(U \mid s_t = s) d\xi(\mu) = \int_{\mathcal{M}} V_\mu^\pi(s) d\xi(\mu) \quad (9.2.7)$$

However, the Bayes-optimal value function is not equal to the expected value function of the optimal policy for each MDP. In fact, the Bayes-value of any policy is a natural lower bound on the Bayes-optimal value function, as the Bayes-optimal policy is the maximum by definition. We can however use the

Algorithm 26 Bayesian Monte-Carlo policy evaluation

```

input policy  $\pi$ , belief  $\xi$ 
for  $k = 1, \dots, K$  do
     $\mu_k \sim \xi.$ 
     $v_k = V_{\mu_k}^\pi$ 
end for
 $u = \frac{1}{K} \sum_{k=1}^K v_k.$ 
return  $u.$ 

```

expected optimal value function as an upper bound on the Bayes-optimal value:

$$V_\xi^* \triangleq \sup_{\pi} \mathbb{E}_\xi^\pi(U) = \sup_{\pi} \int_{\mathcal{M}} \mathbb{E}_\mu^\pi(U) d\xi(\mu) \quad (9.2.8)$$

$$\leq \int_{\mathcal{M}} \sup_{\pi} V_\mu^\pi d\xi(\mu) = \int_{\mathcal{M}} V_\mu^* d\xi(\mu) \triangleq V_\xi^+ \quad (9.2.9)$$

Algorithm 27 Bayesian Monte-Carlo upper bound

```

input policy  $\pi$ , belief  $\xi$ 
for  $k = 1, \dots, K$  do
     $\mu_k \sim \xi.$ 
     $v_k = V_{\mu_k}^*$ 
end for
 $u^* = \frac{1}{K} \sum_{k=1}^K v_k.$ 
return  $u^*.$ 

```

Given the previous development, it is easy to see that the following inequalities always hold.

Bounds on $V_\xi^* \triangleq \max_{\pi} \mathbb{E}(U \mid \pi, \xi)$

$$V_\xi^\pi \leq V_\xi^* \leq V_\xi^+, \quad \forall \pi. \quad (9.2.10)$$

These bounds are geometrically demonstrated in Fig. 9.4. They are entirely analogous to the Bayes bounds of Sec. 3.3.1, with the only difference being that we are now considering complete policies rather than simple decisions.

9.2.5 Tighter lower bounds

One idea to get a better lower bound is to simply find better policies. This idea was explored in Dimitrakakis [2011].

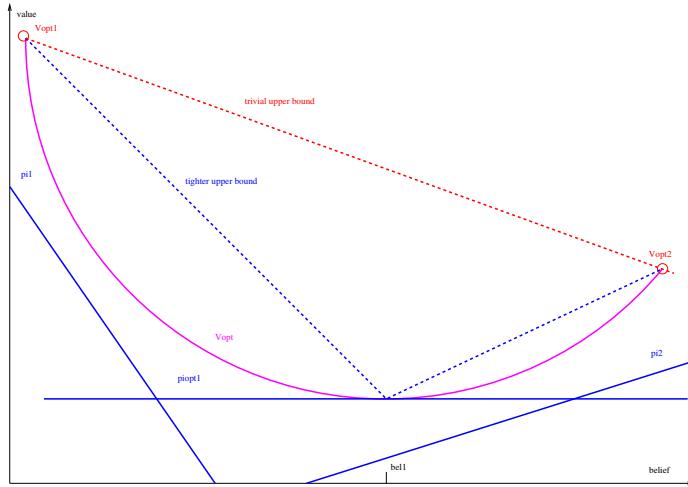
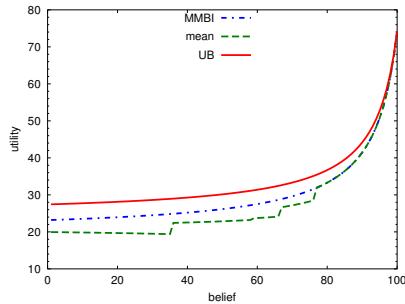


Figure 9.4: A geometric view of the bounds



The main idea was to maximise try and find the best memoryless policies. This can be done approximately by assuming that the belief is nearly constant over time, and performing backwards induction on n MDPs simultaneously. While this greedy procedure might not find the optimal memoryless policy, it still improves the lower bounds considerably.

The multi-MDP backwards induction procedure simply involves calculating the expected utility of a particular policy over all MDPs.

$$Q_{\xi,t}^{\pi}(s,a) \triangleq \int_{\mathcal{M}} \left\{ \bar{R}_{\mu}(s,a) + \gamma \int_{\mathcal{S}} V_{\mu,t+1}^{\pi}(s') d\mathcal{T}_{\mu}^{s,a}(s') \right\} d\xi(\mu) \quad (9.2.11)$$

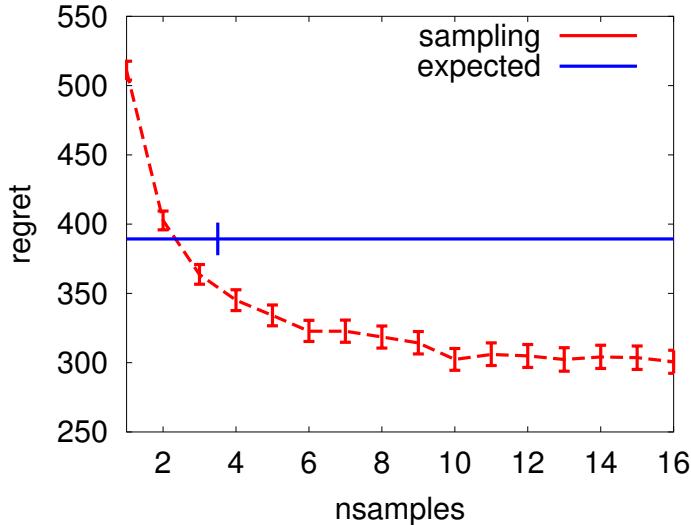
The algorithm greedily performs backwards induction as shown in Algorithm 28. However, this is not an optimal procedure, since the belief at any time-step t is not constant. Indeed, even though the policy is memoryless, $\xi(\mu | s_t, \pi) \neq \xi(\mu | s_t, \pi')$. This is because the probability of being at a particular state is different under different policies and at different time-steps (e.g. if you consider periodic MDPs). For the same reason, this type of backwards induction may not converge in the manner of value iteration.

Algorithm 28 Multi-MDP backwards induction

```

1: MMBIM,  $\xi, \gamma, T$ 
2: Set  $V_{\mu, T+1}(s) = 0$  for all  $s \in \mathcal{S}$ .
3: for  $t = T, T - 1, \dots, 0$  do
4:   for  $s \in \mathcal{S}, a \in \mathcal{A}$  do
5:     Calculate  $Q_{\xi, t}(s, a)$  from (9.2.11) using  $\{V_{\mu, t+1}\}$  .
6:   end for
7:   for  $s \in \mathcal{S}$  do
8:      $a_{\xi, t}^*(s) \in \arg \max_{a \in \mathcal{A}} Q_{\xi, t}(s, a)$ .
9:     for  $\mu \in \mathcal{M}$  do
10:       $V_{\mu, t}(s) = Q_{\mu, t}(s, a_{\xi, t}^*(s))$ .
11:    end for
12:   end for
13: end for

```



```

for Epochs  $i = 1, \dots$  do
  At the start-time  $t_i$  of the epoch, sample  $n$  MDPs  $\mu_1, \dots, \mu_n$  from  $\xi_{t_i}$ .
  Calculate the best memoryless policy  $\pi_i \approx \arg \max_{\pi \in \Pi_1} \sum_{k=1}^n V_{\mu}^{\pi}$  wrt the
  sample.
  Execute  $\pi_i$  until  $t = t_{i+1}$ .
end for

```

9.2.6 Further sampling methods

For $n = 1$, this method is equivalent to Thompson sampling [Thompson, 1933], which was first used in the context of Bayesian reinforcement learning by Strens [2000]. Even though Thompson sampling is good exploration heuristic Kaufmanna et al. [2012], Osband et al. [2013], it is not optimal. Other methods, such as BOSS [Asmuth et al., 2009], combine samples optimistically and hence have

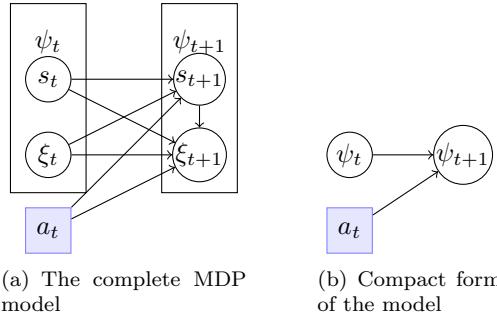


Figure 9.5: Belief-augmented MDP

looser bounds as n increases. In the case of BOSS, this is done by constructing the most optimistic MDP possible in this set. However, all of these methods suffer from the problem that they essentially consider an unchanging belief.

Instead of sampling MDPs, one could sample beliefs. This would lead to a finite hyper-state approximate of the complete belief MDP. One such approach is BEETLE [Poupart et al., 2006, Poupart and Vlassis, 2008] is a belief-sampling approach. It examines a set of possible future beliefs and approximates the value of each belief with a lower bound. In essence, it then creates the set of policies which are optimal with respect to these bounds.

Another idea is to take advantage of the expectation-maximisation view of reinforcement learning [Toussaint et al., 2006]. This allows us to apply a host of different probabilistic inference algorithms. This approach was investigated by Furmston and Barber [2010].

9.2.7 The Belief-augmented MDP

The most direct way to actually solve the general reinforcement learning problem is to cast it as a yet another MDP. We already saw how this can be done with bandit problems in Section 6.2.2.

The augmented MDP

We are given an initial belief ξ_0 on a set of MDPs \mathcal{M} . Each $\mu \in \mathcal{M}$ is a tuple $(\mathcal{S}, \mathcal{A}, P_\mu, \mathbf{r})$, with state space \mathcal{S} , action space \mathcal{A} , transition kernel P_μ and reward vector \mathbf{r} . We now construct the following augmented Markov decision process: $(\mathcal{S} \times \Xi, \mathcal{A}, P, \mathbf{r})$, with factorised transition probabilities: The optimal policy for the augmented MDP is the ξ -optimal for the original problem.

$$P(s_{t+1} \in S \mid \xi_t, s_t, a_t) \triangleq \int_S P_\mu(s_{t+1} \in S \mid s_t, a_t) d\xi_t(\mu) \quad (9.2.12)$$

$$\xi_{t+1}(\cdot) = \xi_t(\cdot \mid s_{t+1}, s_t, a_t) \quad (9.2.13)$$

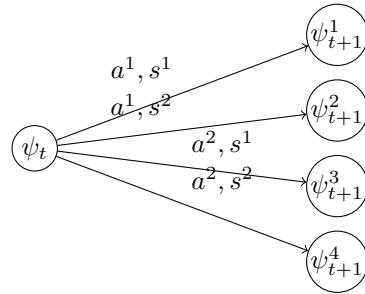
and reward $r_t = \rho(s_t, a_t)$. In the above,

- ξ_t our belief over MDPs $\mu \in \mathcal{M}$ at time t .

- s_t is the observed state of the unknown MDP at time t .
- P_μ is the transition kernel of the MDP μ .
- a_t is our action at time t .
- For simplicity, we assume that r_t be known.

9.2.8 The belief-augmented MDP tree structure

Given a belief over MDPs, we can create an *augmented* MDP with state space $\Psi = \mathcal{S} \times \Xi$. This has a pseudo-tree structure (since belief states might repeat). As an example, consider an MDP family \mathcal{M} with $\mathcal{A} = \{a^1, a^2\}$, $\mathcal{S} = \{s^1, s^2\}$. Then, for any hyper-state $\psi_t = (s_t, \xi_t)$, we can write the following expansion where each possible action-state transition results in one specific hyper-state.



When the branching factor is very large, or when we need to deal with very large tree depths, it becomes necessary to approximate the MDP structure.

9.2.9 Stochastic branch and bound

Branch and bound is a general technique for solving large problems. It can be applied in all cases where upper and lower bounds on the value of solution sets can be found.

Value bounds

Let upper and lower bounds \mathbf{q}^+ and \mathbf{q}^- such that:

$$\mathbf{q}^+(\psi, a) \geq Q^*(\psi, a) \geq \mathbf{q}^-(\psi, a) \quad (9.2.14)$$

$$\mathbf{v}^+(\psi) = \max_{a \in \mathcal{A}} Q^+(\psi, a), \quad \mathbf{v}^-(\psi) = \max_{a \in \mathcal{A}} Q^-(\psi, a). \quad (9.2.15)$$

Then the idea is to calculate upper \mathbf{q}^+ and lower bounds \mathbf{q}^- on the value of

any hyper-state–action pair (ψ, a) by performing backwards induction.

$$\mathbf{q}^+(\psi, a) = \sum_{\psi'} p(\psi' | \psi, a) [r(\psi, a, \psi') + \mathbf{v}^+(\psi')] \quad (9.2.16)$$

$$\mathbf{q}^-(\psi, a) = \sum_{\psi'} p(\psi' | \psi, a) [r(\psi, a, \psi') + \mathbf{v}^-(\psi')] \quad (9.2.17)$$

We can then use the upper bounds expand the tree, while the lower bounds can be used to select final policy. Sub-optimal branches can be discarded once their upper bounds become lower than the lower bound of some other branch.

Remark 9.2.2. If $\mathbf{q}^-(\psi, a) \geq \mathbf{q}^+(\psi, b)$ then b is sub-optimal at ψ .

However, the algorithm is only exact when the number of possible MDPs is finite. We can generalise this to the infinite case, by applying *stochastic* branch and bound methods Dimitrakakis [2010b, 2008]. This involves estimating upper and lower bounds on the values of leaf nodes through Monte-Carlo sampling.

9.2.10 Further reading.

One of the first treatments of this idea was due to Bellman [1957]. Although the idea was well-known in the statistical community [DeGroot, 1970], the popularisation of the idea in reinforcement learning was achieved with Duff’s thesis [Duff, 2002]. Most recent advances in this area involve the use of intelligent methods for exploring the tree, such as sparse sampling [Wang et al., 2005] and Monte-Carlo tree search [Veness et al., 2009].

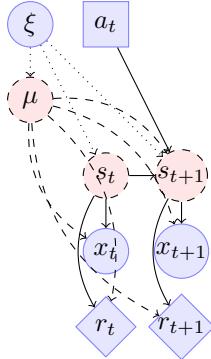
9.3 Partially observable Markov decision processes

In most real applications s_t , the state of the system at time t , is not observed. Instead, we obtain some observation x_t , which depends on the state of the system. While it does give us some information about the system state, it nevertheless is not sufficient to pin-point this exactly.

Partially observable Markov decision processes (POMDP)

When acting in μ , each time step t :

- The system *state* $s_t \in \mathcal{S}$ is not observed.
- We receive an *observation* $x_t \in \mathcal{X}$ and a *reward* $r_t \in \mathcal{R}$.
- We take *action* $a_t \in \mathcal{A}$.
- The system transits to state s_{t+1} .



Definition 9.3.1. Partially observable Markov decision process (POMDP) A POMDP $\mu \in \mathcal{M}_P$ is a tuple $(\mathcal{X}, \mathcal{S}, \mathcal{A}, P)$ where \mathcal{X} is an observation space, \mathcal{S} is a state space, \mathcal{A} is an action space, and P is a conditional distribution on observations, states and rewards. The reward, observation and next state are Markov with respect to the current state and action. In this book, we shall assume the following dependencies:

$$\mathbb{P}_\mu(s_{t+1}, r_t, x_t | s_t, a_t, \dots) = P(s_{t+1} | s_t, a_t)P(x_t | s_t)P(r_t | s_t). \quad (9.3.1)$$

$P(s_{t+1} | s_t, a_t)$ is the *transition distribution*, giving the probabilities of next states given the current state and action. $P(x_t | s_t)$ is the *observation distribution*, giving the probabilities of different observations given the current state. Finally, $P(r_t | s_t)$ is the *reward distribution*, which we make dependent only on the current state for simplicity. Different dependencies are possible, but they are all equivalent to the one given here.

transition distribution

observation distribution

reward distribution

9.3.1 Solving known POMDPs

When we know a POMDP's parameters, that is to say, when we know the transition, observation and reward distributions, the problem is formally the same as solving an unknown MDP. In particular, we can similarly define a *belief state* summarising our knowledge. This takes the form of a probability distribution on the hidden state variable s_t , rather than on the model μ . If μ defines starting state probabilities, then the belief is not subjective, as it only relies on the actual POMDP parameters. The transition distribution on states given our belief is as follows.

Belief ξ

For any distribution ξ on \mathcal{S} , we define:

$$\xi(s_{t+1} | a_t, \mu) \triangleq \int_{\mathcal{S}} P_\mu(s_{t+1} | s_t a_t) d\xi(s_t) \quad (9.3.2)$$

When there is no ambiguity, we shall use ξ to denote arbitrary marginal distributions on states and state sequence given the belief ξ .

When the model μ is given, calculating a belief update is not particularly difficult, but we must take care to properly use the time index t . Starting from Bayes' theorem, it is easy to derive the belief update from ξ_t to ξ_{t+1} as follows,

Belief update

$$\xi_{t+1}(s_{t+1} | \mu) \triangleq \xi_t(s_{t+1} | x_{t+1}, r_{t+1}, a_t, \mu) \quad (9.3.3)$$

$$= \frac{P_\mu(x_{t+1}, r_{t+1} | s_{t+1}) \xi_t(s_{t+1} | a_t, \mu)}{\xi_t(x_{t+1} | a_t, \mu)} \quad (9.3.4)$$

$$\xi_t(s_{t+1} | a_t, \mu) = \int_{\mathcal{S}} P_\mu(s_{t+1} | s_t, a_t, \mu) d\xi_t(s_t) \quad (9.3.5)$$

$$\xi_t(x_{t+1} | a_t, \mu) = \int_{\mathcal{S}} P_\mu(x_{t+1} | s_{t+1}) d\xi_t(s_{t+1} | a_t, \mu) \quad (9.3.6)$$

A particularly attractive example is when the model is finite. Then the sufficient statistic also has finite dimension and all updates are in closed form.

Remark 9.3.1. If $\mathcal{S}, \mathcal{A}, \mathcal{X}$ are finite, and then we can define the sequence of vectors $\mathbf{p}_t \in \mathbb{A}^{|\mathcal{S}|}$, matrices \mathbf{A}_t

- $\mathbf{p}_t(j) = P(x_t | s_t = j)$
- $\mathbf{A}_t(i, j) = P(s_{t+1} = j | s_t = i, a_t)$.
- $\mathbf{b}_t(i) = \xi_t(s_t = i)$

We can then use Bayes theorem:

$$\mathbf{b}_{t+1} = \frac{\text{diag}(\mathbf{p}_{t+1}) \mathbf{A}_t \mathbf{b}_t}{\mathbf{p}_{t+1}^\top \mathbf{A}_t \mathbf{b}_t}, \quad (9.3.7)$$

Even though inference is tractable in finite models, there is a small number of cases

9.3.2 Solving unknown POMDPs

This is a much harder problem, unfortunately. Let us take a look at the basic update equation, where we need to define a joint belief on both possible states and possible models.

$$\xi(\mu, s^t | x^t, a^t) \propto P_\mu(x^t | s^t) P_\mu(s^t | a^t) \xi(\mu) \quad (9.3.8)$$

Unfortunately, even for the simplest possible case of two possible models μ_1, μ_2 and binary observations, there is no finite-dimensional representation of the belief at time t .

Strategies for solving unknown POMDPs include solving the full Bayesian decision problem, but this requires exponential inference and planning for exact solutions Ross et al. [2008]. For this reason, we must use approximations.

One very simple approximation involves replacing a partially observable Markov process with a *variable order Markov decision process*. Fortunately, inference in variable order Markov processes has only logarithmic computational complexity Dimitrakakis [2010a]. Of course, the memory complexity is still linear.

In general, finding optimal controllers for POMDPs is hard even for restricted classes of policies Vlassis et al. [2012]. However, approximations Spaan and Vlassis [2005] and stochastic methods and policy search methods Baxter and Bartlett [2000], Toussaint et al. [2006] work quite well in practice.

9.4 Bayesian methods in continuous spaces

Formally, Bayesian reinforcement learning in continuous state spaces is not significantly different from the discrete case. Typically, we assume that the agent acts within a fully observable discrete-time Markov decision process (MDP), with a metric state space \mathcal{S} , for example $\mathcal{S} \subset \mathbb{R}^d S$. At time t , the agent observes the current environment state $s_t \in \mathcal{S}$, takes an action $a_t \in \mathcal{A}$. In many applications in continuous state space \mathcal{A} remains discrete, but it can also be continuous. The transition kernel

$$P_\mu(S | s, a) \triangleq \mathbb{P}_\mu(s_{t+1} \in S | s_t = s, a_t = a)$$

is now defined over this continuous state space. This is important for methods that estimate it as part of the learning process.

9.4.1 Transition models.

These types of approaches define an explicit distribution over transition models from the data. The simplest is a linear-Gaussian model, which also results in a closed form posterior calculation due to the conjugate prior. In our model we assume that, for a state set \mathcal{S} there exists a mapping $f : \mathcal{S} \rightarrow \mathcal{X}$ to a k -dimensional vector space \mathcal{X} such that the transformed state at time t is $x_t \triangleq f(s_t)$. The next state s_{t+1} is given by the output of a function $g : \mathcal{X} \times \mathcal{A} \rightarrow \mathcal{S}$ of the transformed state, the action and some additive noise:

$$s_{t+1} = g(x_t, a_t) + \varepsilon_t. \quad (9.4.1)$$

In this paper, we model the noise ε_t and the function g as a multivariate linear-Gaussian model. This is parameterised via a set of $k \times k$ *design* matrices $\{\mathbf{A}_i | i \in \mathcal{A}\}$, such that $g(x_t, a_t) = \mathbf{A}_{a_t} x_t$ and a set of *covariance* matrices $\{\mathbf{V}_i | i \in \mathcal{A}\}$ for the noise. Then, the next state distribution is:

$$s_{t+1} | x_t = x, a_t = i \sim \mathcal{N}(\mathbf{A}_i x, \mathbf{V}_i). \quad (9.4.2)$$

In order to model our uncertainty with a (subjective) prior distribution ξ , we have to specify the model structure. In our model, we do not assume independence between the output dimensions, something which could potentially make inference difficult. Fortunately, in this particular case, a conjugate prior exists in the form of the *matrix-normal distribution* for \mathbf{A} and the *inverse-Wishart distribution* for \mathbf{V} . Given \mathbf{V}_i , the distribution for \mathbf{A}_i is matrix-normal, while the marginal distribution of \mathbf{V}_i is inverse-Wishart. More specifically,

$$\mathbf{A}_i | \mathbf{V}_i = \widehat{\mathbf{V}} \sim \phi(\mathbf{A}_i | \mathbf{M}, \mathbf{C}, \widehat{\mathbf{V}}) \quad (9.4.3)$$

$$\mathbf{V}_i \sim \psi(\mathbf{V}_i | \mathbf{W}, n), \quad (9.4.4)$$

where ϕ_i is the prior distribution on dynamics matrices conditional on the covariance and two prior parameters: \mathbf{M} , which is the prior mean and \mathbf{C} which

is the prior output (dependent variable) covariance. Finally, ψ is the marginal prior on covariance matrices, which has an inverse-Wishart distribution with \mathbf{W} and n . More precisely, the distributions are:

$$\phi(\mathbf{A}_i | \mathbf{M}, \mathbf{C}, \widehat{\mathbf{V}}) \propto e^{-\frac{1}{2} \text{trace}[\mathbf{P}(\mathbf{A}_i - \mathbf{M}) \mathbf{V}_i^{-1} (\mathbf{A}_i - \mathbf{M}) \mathbf{C}]}, \quad (9.4.5)$$

$$\psi(\mathbf{V}_i | \mathbf{W}, n) \propto |\mathbf{V}^{-1} \mathbf{W} / 2|^{n/2} e^{-\frac{1}{2} \text{trace}(\mathbf{V}^{-1} \mathbf{W})}. \quad (9.4.6)$$

Essentially, the model is an extension of the univariate Bayesian linear regression model (see for example DeGroot [1970]) to the multivariate case via vectorisation of the mean matrix. Since the prior is conjugate, it is relatively simple to calculate posterior values of the parameters after each observation. While we omit the details, a full description of inference using this model is given in Minka [2001b].

Gaussian processes

Further reading. More complex transition models include the non-parametric extension of the above model, *Gaussian processes* (GP) Rasmussen and Williams [2006]. For an n -dimensional state space, GPs are typically applied by use one GP to predict each state dimension, i.e. $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$. As this completely decouples the state dimensions, it's best to consider a joint model, but this requires various approximations (e.g. Alvarez et al. [2011]). A well-known method for model-based Gaussian process reinforcement learning is GP-RmaxJung and Stone [2010], which has been recently shown by Grande et al. [2014] to be KWIK-learnable.³

Another straightforward extension of linear models are piecewise linear models, which can be described in a Bayesian non-parametric framework Tziortziotis et al. [2014]. This avoids the computational complexity of GPs.

9.4.2 Approximate dynamic programming

Bayesian methods are also frequently used as part of a dynamic programming approach. Typically, we wish to maintain a distribution over value functions in some sense. This generally involves defining some generative model over which inference can be performed. For continuous state spaces particularly, one can assume that the value function \mathbf{v} is drawn from a Gaussian process. However, to perform inference we also need to specify some generative model for the observations.

Temporal differences. Engel et al. [2003] consider temporal differences from a Bayesian perspective in conjunction with a GP model, so that the rewards are distributed as follows:

$$r_t | \mathbf{v}, s_t, s_{t+1} \sim \mathcal{N}(\mathbf{v}(s_{t+1}) - \gamma \mathbf{v}(s_t), \sigma), \quad (9.4.7)$$

which essentially gives a simple model for $P(r^T | \mathbf{v}, s^T)$. We can now write the posterior as $\xi(val | r^T, s^T) \propto P(r^T | \mathbf{v}, s^T) \xi(\mathbf{v})$, where the dependence $\xi(\mathbf{v}|s^T)$

³Informally, a class is KWIK learnable if the number of mistakes made by the algorithm is polynomially bounded in the problem parameters. In the context of reinforcement learning this would be the number of steps for which no guarantee of utility can be provided.

is suppressed. This model was later updated in Engel et al. [2005], with the following reward distribution

$$r_t \mid \mathbf{v}, s_t, s_{t+1} \sim \mathcal{N}(\mathbf{v}(s_t) - \gamma \mathbf{v}(s_{t+1}), N(s_t, s_{t+1})). \quad (9.4.8)$$

The main part of the model is $N(s, s') \triangleq \Delta_U(s) - \gamma \Delta_U(s')$, where the $\Delta_U(s) \triangleq U(s) - \mathbf{v}(s)$ denotes the distribution of the residual, i.e. the utility when starting from s minus its expectation. The correlation between $U(s)$ and $U(s')$ is captured via N , and the residuals are modelled as a Gaussian process. While the model is still an approximation, it is equivalent to performing GP regression using Monte-Carlo samples of the discounted return.

Bayesian finite-horizon dynamic programming for deterministic systems. Deisenroth et al. [2009] on the other hand employ a series of GPs, each for one dynamic programming stage, under the assumption that the dynamics are deterministic and the rewards are Gaussian-distributed. It is possible to extend this approach to the case of non-deterministic transitions, at the cost of requiring additional approximations. However, since a lot of real-world problems do in fact have deterministic dynamics, the approach is sound.

Bayesian least-squares temporal differences. Tziortziotis and Dimitrakakis [2017] instead considers a model for the value function itself, where the random quantity is the empirical transition matrix \hat{P} rather than the reward:

$$\hat{P}\mathbf{v} \mid \mathbf{v}.P \sim \mathcal{N}(P\mathbf{v}, \beta I). \quad (9.4.9)$$

This model makes a different trade-off in its distributional assumptions. It allows us to model the uncertainty about P in a Bayesian manner, but instead of explicitly modelling this as a distribution on P itself, we are modelling a distribution on the resulting Bellman operator.

Gradient methods. Generally speaking, if we are able to sample from the posterior distribution, we can leverage stochastic gradient descent methods to extend any gradient algorithm for reinforcement learning with a given model to the Bayesian setting. More precisely, if we have a utility gradient $\nabla_\theta U(\mu, \theta)$ for model μ , then by linearity we obtain that $\nabla_\theta \mathbb{E}_\xi^\theta U = \int_\Theta \nabla_\mu U(\mu, \theta) d\xi(\mu)$ and stochastic gradient descent can be implemented simply by sampling $\mu \sim \xi$ and updating the parameters using the gradient of the sampled MDP. A few examples of this approach include Dimitrakakis [2013], Ghavamzadeh and Engel [2006].

9.5 Exercises

EXERCISE 32. Consider the algorithms we have seen in Chapter 8. Are any of those applicable to belief-augmented MDPs? Outline a strategy for applying one of those algorithms to the problem. What would be the most formidable obstacle we would have to overcome in your specific example?

EXERCISE 33. Prove Remark 9.2.1

EXERCISE 34. A practical case is when we have an independent belief over the transition probabilities of each state-action pair. Consider the case where we have n states and k actions. Similar to the product-prior in the bandit case in Section 6.2, we assign a probability (density) $\xi_{s,a}$ to the probability vector $\theta_{(s,a)} \in \Delta^n$. We can then define our joint belief on the $(nk) \times n$ matrix Θ to be

$$\xi(\Theta) = \prod_{s \in \mathcal{S}, a \in \mathcal{A}} \xi_{s,a}(\theta_{(s,a)}).$$

- (i) Derive the updates for a product-Dirichlet prior on transitions.
- (ii) Derive the updates for and a product-Normal-Gamma prior on rewards.
- (iii) What would be the meaning of using a Normal-Wishart prior on rewards?

EXERCISE 35. Consider the Gaussian process model of (9.4.7). What is the implicit assumption made about the transition model? If this assumption is satisfied, what does the corresponding posterior distribution represent?

Chapter 10

Distribution-free reinforcement learning

10.1 Introduction

The Bayesian framework requires specifying a prior distribution. For many reasons, we may frequently be unable to specify such a prior distribution. In addition, as we have seen, the Bayes-optimal solution is frequently intractable. Here we shall take a look at algorithms that do not require specifying a prior distribution. Instead, they employ the heuristic of “optimism under uncertainty” to select policies. This idea is very similar to heuristic search algorithms, such as A^* . [ro: Is this explained somewhere? Enter reference?] All these algorithms assume the best possible model that is consistent with the observations so far and choose the optimal policy in this “optimistic” model. Intuitively, this means that for each possible policy we maintain an upper bound on the value/utility we can reasonably expect from it. In general we want this upper bound to

1. be as tight as possible (i.e., to be close to the true value),
2. still hold with high probability.

We begin with an introduction to these ideas in bandit problems, when the objective is to maximise total reward. We then expand this discussion to structured bandit problems, which have many applications in optimisation. Finally, we look at the case of maximising total reward in unknown MDPs. The same main ideas can be used, but the very definition of an optimal MDP policy is not trivial when we wish to maximise total reward. [ro: Update the following in case...] For this reason, we shall go over the various optimality criteria we can use. We then briefly discuss a nearly-optimal reinforcement learning algorithm.

10.2 Finite Stochastic Bandit problems

First of all, let us remind the reader of the stochastic bandit problem. The learner in each time step t chooses an *arm* a_t from a given set $\mathcal{A} = \{1, \dots, K\}$ of K arms. The expected reward for choosing the arm i is $\mu_i = \mathbb{E}(r_t|a_t = i)$ independent of the step t and unknown to the learner. Further, we assume that the rewards are bounded, i.e. that $r_t \in \mathcal{R} \subset \mathbb{R}$. For the sake of simplicity, we rescale rewards so that they are always in $[0, 1]$. The goal is to maximise the total reward $\sum_{t=1}^T r_t$ after T time steps. [ro: skipped that T may be random, as this makes problems with the expected regret below]

Let $\mu^* \triangleq \max_i \mu_i$ be the highest average reward that can be achieved. Obviously, the optimal policy π^* in each time step chooses the arm giving the highest average reward μ^* . The learner who does not know which arm is optimal will choose at each time step t an arm a_t from \mathcal{A} , or more generally, a probability distribution π_t over the arms from which a_t then is drawn. It is important to notice that maximising the total reward is equivalent to minimising total regret with respect to that policy.

Definition 10.2.1 (Total regret). The (*total*) *regret* of a policy π relative to the optimal fixed policy π^* after T steps is

$$L_T(\pi) \triangleq \sum_{t=1}^T (r_t^* - r_t^\pi), \quad (10.2.1)$$

where r_t^π is the reward obtained by the policy π at step t and $r_t^* \triangleq r_t^{\pi^*}$. Accordingly, the *expected (total) regret* is

$$\mathbb{E} L_T(\pi) \triangleq T\mu^* - \mathbb{E}_\pi \sum_{t=1}^T r_t. \quad (10.2.2)$$

The regret compares the collected rewards to those of the best fixed policy. Comparing instead to the best rewards obtained by the arms at each time would be too hard, as these rewards are by definition unpredictable, which would make learning impossible.

[ro: I've skipped the proto-UCB algorithm you gave here, as I think it's not very interesting. In particular, the analysis for the deterministic case is not very meaningful in my view, as in this case the silly 'choose arm with best estimate'-algorithm would work even better.]

10.2.1 The UCB1 algorithm

Obviously, each learning algorithm will use the empirical average rewards obtained for each arm so far.

Empirical average

$$\hat{\mu}_{t,i} \triangleq \frac{1}{N_{t,i}} \sum_{k=1}^t r_{k,i} \mathbb{I}\{a_k = i\}, \text{ where } N_{t,i} \triangleq \sum_{k=1}^t \mathbb{I}\{a_k = i\}$$

and $r_{k,i}$ denotes the (random) reward the learner receives upon choosing arm i at step k .

Simply always choosing the arm with best the empirical average reward so far is not a very good idea, because you might get stuck with a sub-optimal arm: If the optimal arm underperforms at the beginning, so that its empirical average is far below the true mean of a suboptimal arm, it will never be chosen again. A better idea is to choose arms optimistically. Intuitively, as long as an arm has a significant chance of being the best, you play it every now and then. One simple way to implement this is shown in the following UCB1 algorithm Auer et al. [2002a].

Algorithm 29 UCB1

```

Input  $\mathcal{A}$ 
 $\hat{\mu}_{0,i} = 1, \forall i$ 
for  $t = 1, \dots$  do
     $a_t = \arg \max_{i \in \mathcal{A}} \left\{ \hat{\mu}_{t-1,i} + \sqrt{\frac{2 \ln t}{N_{t-1,i}}} \right\}$ 
end for
```

Thus, the algorithm adds a bonus value of order $O(\sqrt{\ln t / n_i})$ to the empirical value of each arm and chooses the arm with maximal *upper confidence bound* value $\hat{\mu}_i + O(\sqrt{\ln t / n_i})$. The bonus value is chosen such that by the Hoeffding bound (4.5.5) with high probability the true mean reward of each arm will be below the upper confidence bound value.

Theorem 10.2.1 (Auer et al Auer et al. [2002a]). *The expected regret of UCB1 after T rounds is at most*

$$\sum_{i:\mu_i < \mu^*} \frac{8 \ln T}{\mu^* - \mu_i} + 5 \sum_i (\mu^* - \mu_i).$$

Proof. By Wald's identity (5.2.14) the expected regret can be written as

$$\mathbb{E} L_T(\pi) = \mathbb{E} \sum_{t=1}^T (\mu^* - r_t) = \sum_i \mathbb{E} N_{T,i} (\mu^* - \mu_i), \quad (10.2.3)$$

so that we focus on bounding $\mathbb{E} N_{t,i}$. Thus, let i be an arbitrary suboptimal arm and consider when it will be chosen by the algorithm. Write $B_{t,s} = \sqrt{(2 \ln t)/s}$ for the “bonus” value at step t after s observations. Note that by the Hoeffding bound (4.5.5) for fixed values of $t, s, s_i \in \mathbb{N}$ under the assumption that $N_{t,i} = s_i$ and (the count of the optimal action) $N_{t,*} = s$, we have that

$$\mathbb{P}(\hat{\mu}_i \geq \mu_{t,i} + B_{t,s_i}) \leq e^{-4 \ln t} = t^{-4}, \quad (10.2.4)$$

$$\mathbb{P}(\hat{\mu}_i^* \leq \mu_t^* - B_{t,s}) \leq e^{-4 \ln t} = t^{-4}, \quad (10.2.5)$$

so that we may assume (we take care of the contribution of the error probabilities to $\mathbb{E} N_{t,i}$ below)

$$\hat{\mu}_{t,i} < \mu_i + B_{t,N_{t,i}}, \quad (10.2.6)$$

$$\mu^* < \hat{\mu}_t^* + B_{t,N_{t,*}}. \quad (10.2.7)$$

Now note that for $s \geq \lceil (8 \ln T) / (\mu^* - \mu_i)^2 \rceil$ it holds that

$$2B_{t,s} \leq (\mu^* - \mu_i), \quad (10.2.8)$$

so that after arm i has been chosen $\lceil (8 \ln T) / (\mu^* - \mu_i)^2 \rceil$ times we get from (10.2.6), (10.2.8), and (10.2.7) that

$$\begin{aligned} \hat{\mu}_{t,i} + B_{t,N_{t,i}} &< \mu_i + 2B_{t,N_{t,i}} \leq \mu^* \\ &< \hat{\mu}_t^* + B_{t,N_{t,*}}, \end{aligned}$$

showing that the algorithm won't choose arm i . Taking into account the error probabilities for (10.2.6) and (10.2.7) we might play arm i once whenever either equation does not hold. Thus, summing over all possible values for $t, N_{t,i}$ and $N_{t,*}$ this shows that

$$\mathbb{E} N_{t,i} \leq \left\lceil \frac{8 \ln T}{(\mu^* - \mu_i)^2} \right\rceil + \sum_{\tau \geq 1} \sum_{s \leq \tau} \sum_{s_i \leq \tau} 2\tau^{-4}.$$

Combining this with (10.2.3) and noting that the sum converges to a value < 4 , proves the regret bound. \square

[ro: also include sqrt-bound?]

The UCB1 algorithm is actually not the first algorithm employing *optimism in the face of uncertainty* to deal with the exploration-exploitation dilemma, nor the first that uses confidence intervals for that purpose. This idea goes back

to the seminal work of [Lai and Robbins, 1985] that used the same approach, however in a more complicated form. In particular, the whole history is used for computing the arm to choose. The derived bounds of Lai and Robbins [1985] show that after T steps each suboptimal arm is played at most $(\frac{1}{D_{KL}} + o(1)) \log T$ times in expectation, where D_{KL} measures the distance between the reward distributions of the optimal and the suboptimal arm by the Kullback-Leibler divergence, and $o(1) \rightarrow 0$ as $T \rightarrow \infty$. This bound was also shown to be asymptotically optimal by [Lai and Robbins, 1985]. A lower bound logarithmic in T for any finite T that is close to matching the bound of Theorem 10.2.1 can be found in Mannor and Tsitsiklis [2004]. Improvements that get closer to the lower bound (and are still based on the UCB1 idea) can be found in Auer and Ortner [2010].

For so-called distribution-independent bounds that do not depend on problem parameters like the ‘gaps’ $\mu^* - \mu_i$, see e.g. Audibert and Bubeck [2009]. In general, these bounds cannot be logarithmic in T anymore (as the gaps may be of order $1/\sqrt{T}$) and are $O(\sqrt{T})$.

10.2.2 Non iid Rewards

The stochastic setting just considered is only one among several variants of the multi-armed bandit setting. While it is impossible to cover them all, we give a brief overview of the most common scenarios and refer to Bubeck and Cesa-Bianchi [2012] for a more complete overview.

What is common to most variants of the classic stochastic setting is that the assumption of receiving iid rewards when sampling a fixed arm is loosened. The most extreme case is the so-called *nonstochastic*, sometimes also called *adversarial bandit* setting, where the reward sequence for each arm is assumed to be fixed in advance (and thus not random at all). In this case, the reward is maximised when choosing in each time step the arm that maximises the reward at this step. Obviously, since the reward sequences can be completely arbitrary, no learner can stand a chance to perform well with respect to this optimal policy. Thus, one confines oneself to consider the regret with respect to the best *fixed* arm in hindsight, that is, $\arg \max_i \sum_{t=1}^T r_{t,i}$ where $r_{t,i}$ is the reward of arm i at step t . It is still not clear that this is not too much to ask for, but it turns out that one can achieve regret bounds of order $O(\sqrt{KT})$ in this setting. Clearly, algorithms that choose arms deterministically can always be tricked by an adversarial reward sequence. However, algorithms that at each time step choose an arm from a suitable distribution over the arms (that is updated according to the collected rewards) like the Exp3 algorithm of [Auer et al., 2002b] or similar algorithms that use an exponential weighting scheme meet the mentioned upper bound on the regret, which can be shown to be optimal.

[**ro:** include prediction with expert advice?]

In *contextual bandits* the learner receives some additional side information called the *context*. The reward for choosing an arm is assumed to depend on the context as well as the chosen arm and can be either stochastic or adversarial. The learner usually competes against the best policy that maps contexts to arms. There is a notable amount of literature dealing with various settings that are usually also interesting for applications like web advertisement where

user data takes the role of provided side information. For an overview see e.g. Chapter 4 of Bubeck and Cesa-Bianchi [2012].

In other settings the iid assumption about the rewards of a fixed arm is replaced by more general assumptions, such as that underlying each arm there is a Markov chain and rewards depend on the state of the Markov chain when sampling the arm. This is called the *restless bandits* problem, that is already quite close to the general reinforcement learning setting with an underlying Markov decision process (see Section 10.4.3 below). Regret bounds in this setting can be shown to be $\tilde{O}(\sqrt{T})$ even if at each time step the learner can observe only the state of the arm he chooses, see Ortner et al. [2014].

10.3 Structured bandit problems

Bandits and optimisation

- Continuous stochastic functions Kocsis and Szepesvári [2006], Auer et al. [2007], Bubeck et al. [2011]
- Constrained deterministic distributed functions Ottens et al. [2012]

Solve a sequence of discrete bandit problems.

At epoch i , we have some interval A_i

- Split the interval A_i in k regions $A_{i,j}$
- Run UCB on the k -armed bandit problem.
- When a region is sub-optimal with high probability, remove it!

First idea Auer et al. [2007]

Tree bandits Bubeck et al. [2011] Create a tree of coverings, with (h, i) being the i -th node at depth h . \mathcal{D} are the descendants and \mathcal{C} the children of a node.

At time t we pick node H_t, I_t . Each node is picked at most once.

$$\begin{aligned} n_{h,i}(T) &\triangleq \sum_{t=1}^T \mathbb{I}\{(H_t, I_t) \in \mathcal{D}(h, i)\} && \text{(visits of } (h, i)) \\ \hat{\mu}_{h,i}(T) &\triangleq \frac{1}{n_{h,i}(T)} \sum_{t=1}^T r_t \mathbb{I}\{(H_t, I_t) \in \mathcal{C}(h, i)\} && \text{(reward from } (h, i)) \\ C_{h,i}(T) &\triangleq \hat{\mu}_{h,i}(T) + \sqrt{\frac{2 \ln T}{n_{h,i}(T)}} + n u_1 \rho^h && \text{(confidence bound)} \\ B_{h,i}(T) &\triangleq \min \left\{ C_{h,i}(T), \max_{(h+1,j) \in \mathcal{C}(h,i)} B_{h+1,j} \right\} && \text{(child bound)} \end{aligned}$$

10.4 Reinforcement learning problems

[**ro:** Shouldn't the following subsection not rather go the chapter about MDPs? (I've ignored it for the moment.)]

10.4.1 Optimality Criteria

In all previous cases, we assumed a specific discount rate, or horizon for our problem. Now we shall examine different choices and how they affect the existence of an optimal policy.

As mentioned previously, the following two views of discounted reward processes are equivalent.

Infinite horizon, discounted

Discount factor γ such that

$$U_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \quad \Rightarrow \mathbb{E} U_t = \sum_{k=0}^{\infty} \gamma^k \mathbb{E} r_{t+k} \quad (10.4.1)$$

Geometric horizon, undiscounted

At each step t , the process terminates with probability $1 - \gamma$:

$$U_t^T = \sum_{k=0}^{T-t} r_{t+k}, \quad T \sim \text{Geom}(1 - \gamma) \quad \Rightarrow \mathbb{E} U_t = \sum_{k=0}^{\infty} \gamma^k \mathbb{E} r_{t+k} \quad (10.4.2)$$

$$V_{\gamma}^{\pi}(s) \triangleq \mathbb{E}(U_t \mid s_t = s)$$

The expected total reward criterion

$$V_t^{\pi,T} \triangleq \mathbb{E}_{\pi} U_t^T, \quad V^{\pi} \triangleq \lim_{T \rightarrow \infty} V^{\pi,T} \quad (10.4.3)$$

Dealing with the limit

- Consider μ s.t. the limit exists $\forall \pi$.

$$V_+^{\pi}(s) \triangleq \mathbb{E}_{\pi} \left(\sum_{t=1}^{\infty} r_t^+ \mid s_t = s \right), \quad V_-^{\pi}(s) \triangleq \mathbb{E}_{\pi} \left(\sum_{t=1}^{\infty} r_t^- \mid s_t = s \right) \quad (10.4.4)$$

$$r_t^+ \triangleq \max\{-r, 0\}, \quad r_t^- \triangleq \max\{r, 0\}. \quad (10.4.5)$$

- Consider μ s.t. $\exists \pi^*$ for which V^{π^*} exists and

$$\lim_{T \rightarrow \infty} V^{\pi^*, T} = V^{\pi^*} \geq \limsup_{T \rightarrow \infty} V^{\pi, T}.$$

- Use optimality criteria sensitive to the divergence rate.

The gain g

$$g^\pi(s) \triangleq \lim_{T \rightarrow \infty} \frac{1}{T} V^{\pi, T}(s) \quad (10.4.6)$$

The average reward (gain) criterion

$$g_+^\pi(s) \triangleq \limsup_{T \rightarrow \infty} \frac{1}{T} V^{\pi, T}(s), \quad g_-^\pi(s) \triangleq \liminf_{T \rightarrow \infty} \frac{1}{T} V^{\pi, T}(s) \quad (10.4.7)$$

If $\lim_{T \rightarrow \infty} \mathbb{E}(r_T | s_0 = s)$ exists then it equals $g^\pi(s)$.

Let Π be the set of all history-dependent, randomised policies.

Using our overloaded symbols, we have that π^* is *total reward optimal* if

$$V^{\pi^*}(s) \geq V^\pi(s) \quad \forall s \in \mathcal{S}, \pi \in \Pi.$$

π^* is *discount optimal* for $\gamma \in [0, 1)$ if

$$V_\gamma^{\pi^*}(s) \geq V_\gamma^\pi(s) \quad \forall s \in \mathcal{S}, \pi \in \Pi.$$

π^* is *gain optimal* if

$$g^{\pi^*}(s) \geq g^\pi(s) \quad \forall s \in \mathcal{S}, \pi \in \Pi.$$

Overtaking optimality

π^* is *overtaking optimal* if

$$\liminf_{T \rightarrow \infty} \left[V^{\pi^*, T}(s) - V^{\pi, T}(s) \right] \geq 0 \quad \forall s \in \mathcal{S}, \pi \in \Pi.$$

However, no overtaking optimal policy may exist.

π^* is *average-overtaking optimal* if

$$\liminf_{T \rightarrow \infty} \frac{1}{T} \left[V^{\pi^*, T}(s) - V_+^\pi(s) \right] \geq 0 \quad \forall s \in \mathcal{S}, \pi \in \Pi.$$

Sensitive discount optimality π^* is *n-discount optimal* for $n \in \{-1, 0, 1, \dots\}$ if

$$\liminf_{\gamma \uparrow 1} (1 - \gamma)^{-n} \left[V_\gamma^{\pi^*}(s) - V_\gamma^\pi(s) \right] \geq 0 \quad \forall s \in \mathcal{S}, \pi \in \Pi.$$

A policy is *Blackwell optimal* if $\forall s, \exists \gamma^*(s)$ such that

$$V_\gamma^{\pi^*}(s) - V_\gamma^\pi(s) \geq 0, \quad \forall \pi \in \Pi, \gamma^*(s) \gamma < 1.$$

Lemma 10.4.1. *If a policy is m-discount optimal then it is n-discount optimal for all $n \leq m$.*

Lemma 10.4.2. *Gain optimality is equivalent to -1-discount optimality.*

10.4.2 Introduction

Now we want to take a step further from the bandit problems of the previous sections to the general reinforcement learning setting with an underlying MDP unknown to the learner. Note that the stochastic bandit problem corresponds to a single state MDP.

Thus, consider an MDP μ^* with state space \mathcal{S} , action space \mathcal{A} , and let $r(s, a) \in [0, 1]$ and $P(\cdot|s, a)$ be the mean reward and the transition probability distribution on \mathcal{S} for each state $s \in \mathcal{S}$ and each action $a \in \mathcal{A}$, respectively. For the moment we assume that \mathcal{S} and \mathcal{A} are finite. As we have seen in Section 10.4.1 there are various optimality criteria for MDPs. In the spirit of the bandit problems considered so far we consider undiscounted rewards and examine the regret after any T steps with respect to an optimal policy.

Since the optimal T -step policy in general will be non-stationary and different for different horizons T and different initial states, we will compare to a *gain optimal* policy π^* . Further, we assume that the MDP is *communicating*, that is, for any two states s, s' there is a policy $\pi_{s,s'}$ that with positive probability reaches s' when starting in s and playing actions according to $\pi_{s,s'}$. This also means that when learning in the MDP we can always recover when making a mistake. Note that in MDPs that are not communicating one wrong step may lead to a suboptimal region of the state space that cannot be left anymore, which makes competing to an optimal policy in a learning setting impossible. For communicating MDPs we can define the diameter to be the maximal expected time it takes to connect any two states.

Definition 10.4.1. Let $T(\pi, s, s')$ the expected number of steps it takes to reach state s' when starting in s and playing policy π . Then the *diameter* is defined as

$$D \triangleq \max_{s, s'} \min_{\pi} T(\pi, s, s').$$

Given that our rewards are assumed to be bounded in $[0, 1]$, intuitively, when we make one wrong step in some state s , in the long run we won't lose more than D . After all, in D steps we can go back to s and continue optimally.

Under the assumption that the MDP is communicating, the gain g^* can be shown to be independent of the initial state, that is, $g^*(s) = g^*$ for all states s .

Then we define the T -step regret of a learning algorithm as

$$L_T \triangleq \sum_{t=1}^T (g^* - r_t),$$

where r_t is the reward collected by the algorithm at step t . Note that in general (and depending on the initial state) the value Tg^* we compare to will differ from the optimal T -step reward. However, this difference can be shown to be upper bounded by the diameter and is therefore negligible when considering the regret.

10.4.3 An upper-confidence bound algorithm

Now we would like to extend the idea underlying the UCB1 algorithm to the general reinforcement learning setting. Again, we would like to have for each (stationary) policy π an upper bound on the gain that is reasonable to expect.

Note that simply taking each policy to be the arm of a bandit problem does not work well. First, to approach the true gain of a chosen policy, it will not be sufficient to choose it just once, but for a sufficiently high number of consecutive steps. Without knowledge of some characteristics of the underlying MDP like mixing times, it might be however difficult to determine how long a policy shall be played (see however the respective mechanism of the UCRL2 algorithm below). Further, due to the large number of stationary policies, which is $|\mathcal{A}|^{|S|}$, the regret bounds that would result from such an approach would be exponential in the number of states. Thus, we rather maintain confidence regions for the rewards and transition probabilities of each state-action pair s, a . Then, at each step t , these confidence regions implicitly define a confidence region for the true underlying MDP, that is, a set M_t of *plausible* MDPs. For suitably chosen confidence intervals for the rewards and transition probabilities one can obtain that

$$\mathbb{P}(\mu^* \notin M_t) < \delta. \quad (10.4.8)$$

Given this confidence region M_t , one can define the optimistic value for any policy π to be

$$g_+^\pi(M_t) \triangleq \max \{g_\mu^\pi \mid \mu \in M_t\}. \quad (10.4.9)$$

Note that similar to the bandit setting this estimate is optimistic for each policy, as due to (10.4.8) it holds that $g_+^\pi(M_t) \geq g_{\mu^*}^\pi$ with high probability. Analogously to UCB1 we would like to make an optimistic choice among the possible policies, that is, we choose a policy π that maximises $g_+^\pi(M_t)$.

However, unlike in the bandit setting where we immediately receive a sample from the reward of the chosen arm, in the MDP setting we only obtain information about the reward in the current state, which is however random as well. Thus, we should not play the chosen optimistic policy just for one but a sufficiently large number of steps. As already mentioned before, without prior knowledge it is hard to tell what would be a sufficient number. However, an easy trick is to play policies in episodes of increasing length, such that sooner or later each policy is played for a sufficient number of steps. See below for details.

Summarized, we obtain an algorithm as shown below.

UCRL2 Jaksch et al. [2010] outline

In episodes $k = 1, 2, \dots$

- At the first step t_k of episode k , update the confidence region M_{t_k} .
- Compute an optimistic policy $\tilde{\pi}_k \in \arg \max_\pi g_+^\pi(M_{t_k})$.
- Execute $\tilde{\pi}_k$, observe rewards and transitions until t_{k+1} .

Technical details for UCRL2

To make the algorithm complete, we have to fill in some technical details. In the following, let S be the number of states and A the number of actions of the underlying MDP μ^* . Further, $\delta > 0$ is a confidence parameter of the algorithm.

The confidence region First, concerning the confidence regions, for the rewards it is sufficient to use confidence intervals similar to those for UCB1. Similarly, for the transition probabilities we consider all those transition probability distributions to be plausible if their $\|\cdot\|_1$ -norm is close to the empirical distribution $\hat{\mathbf{P}}_t(\cdot | s, a)$. That is, the confidence region M_t at step t used to compute the optimistic policy in each episode can be defined as the set of MDPs with mean rewards $r(s, a)$ and transition probabilities $\mathbf{P}(\cdot | s, a)$ such that

$$|r(s, a) - \hat{r}(s, a)| \leq \sqrt{\frac{7 \log(2SAT/\delta)}{2N_t(s, a)}}, \quad (10.4.10)$$

$$\left\| \mathbf{P}(\cdot | s, a) - \hat{\mathbf{P}}_t(\cdot | s, a) \right\|_1 \leq \sqrt{\frac{14S \log(2At/\delta)}{N_t(s, a)}}, \quad (10.4.11)$$

where $\hat{r}(s, a)$ and $\hat{\mathbf{P}}_t(\cdot | s, a)$ are the estimates for the rewards and the transition probabilities, and $N_t(s, a)$ denotes the number of samples of action a in state s (at time step t).

One can show via a bound due to [Weissman et al., 2003] that given n samples of the transition probability distribution $\mathbf{P}(\cdot | s, a)$, one has

$$\mathbb{P}\left(\left\| \mathbf{P}(\cdot | s, a) - \hat{\mathbf{P}}_t(\cdot | s, a) \right\|_1 \geq \varepsilon\right) \leq 2^S \exp\left(-\frac{n\varepsilon}{2}\right). \quad (10.4.12)$$

Using this together with standard Hoeffding bounds for the reward estimates, it can be shown that the confidence region contains the true underlying MDP with high probability.

Lemma 10.4.3. $\mathbb{P}(\mu^* \in M_t) > 1 - \frac{\delta}{15t^6}$.

Episode lengths Concerning the termination of episodes, as already mentioned, we would like to have episodes that are long enough so that we can estimate the gain of the played policy, but not too long to suffer large regret when playing a suboptimal policy. Intuitively, it only pays off to recompute the optimistic policy when the estimates/confidence intervals have changed sufficiently. One option is e.g. to terminate an episode when the confidence interval for one state-action pair has shrunked by some factor. Even simpler, one can terminate an episode when a state-action pair has been sampled often (compared to the samples one had before the episode has started), e.g. when one has doubled the number of visits in some state-action pair. This also allows to bound the total number of episodes up to step T .

Lemma 10.4.4. *If an episode of UCRL2 is terminated when the number of visits in some state-action pair has been doubled, the total number of episodes up to step T is upper bounded by $S A \log_2 \frac{8T}{SA}$.*

The episode termination criterion also allows to bound the sum over all fractions of the form $\frac{v_k(s, a)}{\sqrt{N_k(s, a)}}$, where $v_k(s, a)$ is the number of times action a has been chosen in state s during episode k , while $N_k(s, a)$ is the respective count of visits *before* episode k . The evaluation of this sum will turn out to be important to bound the sum over all confidence intervals over the visited state-action pairs in the regret analysis below.

Lemma 10.4.5. $\sum_k \sum_{s, a} \frac{v_k(s, a)}{\sqrt{N_k(s, a)}} \leq (\sqrt{2} + 1)\sqrt{SAT}$.

Calculating the optimistic policy It is important to note that the computation of the optimistic policy can be performed efficiently by using a modification of value iteration. Intuitively, for each policy π the optimistic value $V_+^\pi(M_t)$ maximises the gain over all possible values in the confidence intervals for the rewards and the transition probabilities for π . This is an optimisation problem over a compact space that can be easily solved. In order to find $\arg \max_\pi V_+^\pi(M_t)$, for each considered policy one additionally has to determine the precise values for rewards and transition probabilities within the confidence region. This corresponds to finding the optimal policy in an MDP with compact action space, which can be solved by an extension of value iteration that in each iteration now not only maximises over the original action space but also within the confidence region of the respective action. Noting that $V_+^\pi(M_t)$ is maximised when the rewards are set to their upper confidence values, this results in the following value iteration scheme:

1. Set the optimistic rewards $\tilde{r}(s, a)$ to the upper confidence values for all states s and all actions a .
2. Set $u_0(s) := 0$ for all s .
3. For all $i > 0$ set

$$u_{i+1}(s) := \max_a \left\{ \tilde{r}(s, a) + \max_{P \in \mathcal{P}(s, a)} \left\{ \sum_{s'} P(s') u_i(s') \right\} \right\} \quad (10.4.13)$$

where $\mathcal{P}(s, a)$ is the set of all plausible transition probabilities for choosing a in s .

This scheme can be shown to converge, that is, $\max_s \{u_{i+1}(s) - u_i(s)\} - \min_s \{u_{i+1}(s) - u_i(s)\} \rightarrow 0$ and also

$$u_{i+1}(s) \rightarrow u_i(s) + g_+^{\tilde{\pi}} \text{ for all } s. \quad (10.4.14)$$

After convergence the maximizing actions constitute the optimistic policy $\tilde{\pi}$, and the maximizing transition probabilities are the respective optimistic transition values \tilde{P} .

One can also show that the $\text{span } \max_s u_i(s) - \min_s u_i(s)$ of the converged value vector u_i is upper bounded by the diameter. This follows by optimality of the vector u_i . Intuitively, if the span would be larger than D one could increase the collected reward in the lower value state s^- by going (as fast as possible) to the higher value state s^+ . (Here we use the fact that the true MDP w.h.p. is plausible, so that we may take the true transitions to go from s^- to s^+ .)

Lemma 10.4.6. *Let $u_i(s)$ the converged value vector. Then*

$$\max_s u_i(s) - \min_s u_i(s) \leq D.$$

[ro: I left the following in the text, but not sure what you wanted to include here.]

High-probability value function bound

$$V_+^* = \max \{ V_\mu^* \mid \mu \in M_t \}, \quad \mathbb{P}(\mu^* \in M_t) \geq 1 - \delta.$$

Highly credible value function bound

$$V_+^* = \max \{ V_\mu^* \mid \mu \in M_t \}, \quad \xi_t(M_t) \geq 1 - \delta.$$

Bayesian value function bound (e.g. Dimitrakakis [2011])

$$V_+^* = \int_{\mathcal{M}} V_\mu^* d\xi_t(\mu) \quad \xi_t = \xi_0(\cdot \mid s_t, r_t, \dots)$$

Comparison with Bayesian upper bound**Analysis of UCRL2**

In this section we derive the following regret bound for UCRL2.

Theorem 10.4.1. *Jaksch et al. [2010] In an MDP with S states, A actions, and diameter D with probability of at least $1 - \delta$ the regret of UCRL2 after any T steps is bounded by*

$$\text{const} \cdot DS \sqrt{AT \log \left(\frac{T}{\delta} \right)}.$$

Proof. The main idea of the proof is that by Lemma 10.4.3 we have that

$$\tilde{g}_k^* \triangleq g_+^{\tilde{\pi}_k}(M_{t_k}) \geq g^* \geq g^{\tilde{\pi}_k}, \quad (10.4.15)$$

so that the regret in each step is upper bounded by the width of the confidence interval for $g^{\tilde{\pi}_k}$, that is, by $\tilde{g}_k^* - g^{\tilde{\pi}_k}$. In what follows we need to break down this confidence interval to the confidence intervals we have for rewards and transition probabilities.

In the following, we consider that the true MDP μ^* is always contained in the confidence regions M_t considered by the algorithm. Using Lemma 10.4.3 it is not difficult to show that with probability at least $1 - \frac{\delta}{12T^{5/4}}$ the regret accumulated due to $\mu^* \notin M_t$ at some step t is bounded by \sqrt{T} .

Further, note that the random fluctuation of the rewards can be easily bounded by Hoeffding (4.5.5), that is, if s_t and a_t denote the state and action at step t , we have

$$\sum_{t=1}^T r_t \geq \sum_t r(s_t, a_t) - \sqrt{\frac{5}{8} T \log \frac{8T}{\delta}}$$

with probability at least $1 - \frac{\delta}{12T^{5/4}}$.

Therefore, writing $v_k(s, a)$ for the number of times action a has been chosen in state s in episode k we have $\sum_t r(s_t, a_t) = \sum_k \sum_{s,a} v_k(s, a) r(s, a)$ so that by (10.4.15) we can bound the regret by

$$\sum_{t=1}^T (g^* - r_t) \leq \sum_k \sum_{s,a} v_k(s, a) (\tilde{g}_k^* - r(s, a)) + \sqrt{T} + \sqrt{\frac{5}{8} T \log \frac{8T}{\delta}} \quad (10.4.16)$$

with probability at least $1 - \frac{2\delta}{12T^{5/4}}$.

Thus, let us consider an arbitrary but fixed episode k , and consider the regret

$$\sum_{s,a} v_k(s, a) (\tilde{g}_k^* - r(s, a))$$

the algorithm accumulates in this episode. Let $\text{conf}_k^r(s, a)$ and $\text{conf}_k^p(s, a)$ be the width of the confidence intervals for rewards and transition probabilities in episode k . First, we simply have

$$\begin{aligned} \sum_{s,a} v_k(s, a) (\tilde{g}_k^* - r(s, a)) &\leq \sum_{s,a} v_k(s, a) (\tilde{g}_k^* - \tilde{r}_k(s, a)) \\ &+ \sum_{s,a} v_k(s, a) (\tilde{r}_k(s, a) - r(s, a)) \end{aligned} \quad (10.4.17)$$

where the second term is bounded by $|\tilde{r}_k(s, a) - \hat{r}_k(s, a)| + |\hat{r}_k(s, a) - r(s, a)| \leq 2\text{conf}_k^r(s, a)$ w.h.p. by Lemma 10.4.3, so that

$$\sum_{s,a} v_k(s, a) (\tilde{r}_k(s, a) - r(s, a)) \leq 2 \sum_{s,a} v_k(s, a) \cdot \text{conf}_k^r(s, a). \quad (10.4.18)$$

For the first term in (10.4.17) we use that after convergence of the value vector u_i we have by (10.4.13) and (10.4.14)

$$\tilde{g}_k^* - \tilde{r}_k(s, \tilde{\pi}_k(s)) = \sum_{s'} \tilde{P}_k(s'|s, \tilde{\pi}_k(s)) \cdot u_i(s') - u_i(s),$$

so that noting that $v_k(s, a) = 0$ for $a \neq \tilde{\pi}_k(s)$ and using vector/matrix notation we have

$$\begin{aligned} \sum_{s,a} v_k(s, a) (\tilde{g}_k^* - \tilde{r}_k(s, \tilde{\pi}_k(s))) &= \sum_{s,a} v_k(s, a) \left(\sum_{s'} \tilde{P}_k(s'|s, \tilde{\pi}_k(s)) \cdot u_i(s') - u_i(s) \right) \\ &= \mathbf{v}_k(\tilde{\mathbf{P}}_k - \mathbf{I})\mathbf{u} \\ &= \mathbf{v}_k(\tilde{\mathbf{P}}_k - \mathbf{P}_k + \mathbf{P}_k - \mathbf{I})\mathbf{w}_k \\ &= \mathbf{v}_k(\tilde{\mathbf{P}}_k - \mathbf{P}_k)\mathbf{w}_k + \mathbf{v}_k(\mathbf{P}_k - \mathbf{I})\mathbf{w}_k, \end{aligned} \quad (10.4.19)$$

where \mathbf{P}_k is the true transition matrix (in μ^*) of the optimistic policy $\tilde{\pi}_k$ in episode k , and \mathbf{w}_k is a renormalisation of the vector \mathbf{u} (with entries $u_i(s)$) where $w_k(s) := u_i(s) - \frac{1}{2}(\min_s u_i(s) + \max_s u_i(s))$, so that $\|\mathbf{w}_k\|_\infty \leq \frac{D}{2}$ by Lemma 10.4.6.

Since $\|\tilde{\mathbf{P}}_k - \mathbf{P}_k\|_1 \leq \|\tilde{\mathbf{P}}_k - \hat{\mathbf{P}}_k\|_1 + \|\hat{\mathbf{P}}_k - \mathbf{P}_k\|_1$, the first term of (10.4.19) is bounded as

$$\begin{aligned} \mathbf{v}_k(\tilde{\mathbf{P}}_k - \mathbf{P}_k)\mathbf{w}_k &\leq \|\mathbf{v}_k(\tilde{\mathbf{P}}_k - \mathbf{P}_k)\|_1 \cdot \|\mathbf{w}_k\|_\infty \\ &\leq 2 \sum_{s,a} v_k(s, a) \text{conf}_k^p(s, a) D. \end{aligned} \quad (10.4.20)$$

The second term can be rewritten as martingale difference sequence

$$\begin{aligned}\mathbf{v}_k(\mathbf{P}_k - \mathbf{I})\mathbf{w}_k &= \sum_{t=t_k}^{t_{k+1}-1} \left(P(\cdot|s_t, a) \mathbf{w}_k - w_k(s_t) \right) \\ &= \sum_{t=t_k}^{t_{k+1}-1} \left(P(\cdot|s_t, a) \mathbf{w}_k - w_k(s_{t+1}) \right) + w_k(s_{t_{k+1}}) - w_k(s_{t_k}),\end{aligned}$$

so that its sum over all episodes can be bounded by Azuma-Hoeffding inequality (5.3.4) and Lemma 10.4.4, that is,

$$\sum_k \mathbf{v}_k(\mathbf{P}_k - \mathbf{I})\mathbf{w}_k \leq D \sqrt{\frac{5}{2} T \log\left(\frac{8T}{\delta}\right)} + DSA \log_2\left(\frac{8T}{SA}\right) \quad (10.4.21)$$

with probability at least $1 - \frac{\delta}{12T^{5/4}}$.

Summing (10.4.18) and (10.4.20) over all episodes, by definition of the confidence intervals and Lemma 10.4.5 we have

$$\begin{aligned}&\sum_k \sum_{s,a} v_k(s,a) \text{conf}_k^r(s,a) + 2D \sum_k \sum_{s,a} v_k(s,a) \text{conf}_k^p(s,a) \\ &\leq \text{const} \cdot D \sqrt{S \log(AT/\delta)} \sum_k \sum_{(s,a)} \frac{v_k(s,a)}{\sqrt{N_k(s,a)}} \\ &\leq \text{const} \cdot D \sqrt{S \log(AT/\delta)} \sqrt{SAT}.\end{aligned} \quad (10.4.22)$$

Thus, combining terms (10.4.17)–(10.4.22) we obtain that

$$\sum_{s,a} v_k(s,a) (\tilde{g}_k^* - r(s,a)) \leq \text{const} \cdot D \sqrt{S \log(AT/\delta)} \sqrt{SAT} \quad (10.4.23)$$

with probability at least $1 - \frac{\delta}{12T^{5/4}}$.

Finally by (10.4.16) and (10.4.23) the regret of UCRL2 is upper bounded by $\text{const} \cdot D \sqrt{S \log(AT/\delta)} \sqrt{SAT}$ with probability at least $1 - 3 \sum_{T \geq 2} \frac{\delta}{12T^{5/4}} \geq 1 - \delta$. \square

The following is a corresponding lower bound on the regret that shows that the upper bound of Theorem 10.4.1 is optimal in T and A . It is still an open question what is the ‘right’ dependency on S and D .

Theorem 10.4.2. *Jaksch et al. [2010] For any algorithm and any natural numbers $T, S, A > 1$, and $D \geq \log_A S$ there is an MDP μ with S states, A actions, and diameter D , such that for any initial state $s \in \mathcal{S}$ the expected regret after T steps is*

$$\Omega(\sqrt{DSAT}).$$

Similar to the distribution dependent regret bound of Theorem 10.2.1 for UCB1, one can derive a logarithmic bound on the expected regret of UCRL2.

Theorem 10.4.3. *Jaksch et al. [2010] Let $\Delta \triangleq \rho^*(\mathcal{M}) - \max_\pi \{\rho(\mathcal{M}, \pi) : \rho(\mathcal{M}, \pi) < \rho^*(\mathcal{M})\}$ be the gap between the optimal gain and the second largest gain achievable in \mathcal{M} . Then the expected regret of UCRL2 is*

$$O\left(\frac{D^2 S^2 A \log(T)}{\Delta}\right).$$

Similar to UCB1, UCRL2 is not the first optimistic algorithm with theoretical guarantees. Thus, the *index policies* of Burnetas and Katehakis [1997] and Tewari and Bartlett [2008] choose actions optimistically by using confidence bounds for the estimates in the current state. However, the logarithmic regret bounds are derived only for *ergodic* MDPs in which each policy visits each state with probability 1.

The most well-known optimistic RL algorithm is R-Max [Brafman and Tennenholtz, 2003], that assumes in each not sufficiently visited state to receive the maximal possible reward. UCRL2 offers a refinement of this idea to motivate exploration. Sample complexity bounds as derived for R-Max can also be obtained for UCRL2, cf. [Jaksch et al., 2010].

In the discounted setting, the MBIE algorithm of Strehl and Littman [2005, 2008] is a precursor of UCRL2 that is based on the same ideas. The derived regret bounds are not easily comparable to Theorem 10.2.1, however they appear weaker in some sense, as regret in the discounted setting seems to be a less satisfactory concept. However, sample complexity bounds in the discounted setting for a UCRL2 variant have been given in [Lattimore and Hutter, 2014].

10.4.4 Bibliographical remarks

Different optimality criteria are treated in detail in Puterman [1994] Chapter 5.

Chapter 11

Conclusion

This book touched upon the basic principles of decision making under uncertainty in the context of reinforcement learning. While one of the main streams of thought is Bayesian decision theory, we also discussed the basics of approximate dynamic programming and stochastic approximation.

Consciously, however, we have avoided going into a number of topics related to reinforcement learning and decision theory, some of which would need a book of their own to properly address. Even though it was fun writing the book, we at some point had to decide to stop and consolidate the material we had, sometimes culling partially developed material in favour of a more concise volume.

Firstly, we haven't explicitly considered what models one can use for representing transition distributions, value functions or policies, beyond the simplest ones, as we felt that this would detract from the main body of the text. Textbooks for the latest fashion are always going to be abundant, and we hope that this book provides a sufficient basis to enable the use any current trends. There were also a large number of areas which have not been covered at all. In particular, while we touched upon the setting of two-player games and its connection to robust statistical decisions, we have not examined problems relevant to sequential decision making, such as Markov games and Bayesian games. In relation to this, while early in the book we discuss risk aversion and risk seeking, we have not discussed specific sequential decision making algorithms for such problems. Furthermore, even though we discuss the problem of preference elicitation, we do not discuss specific algorithms for it or the related problem of inverse reinforcement learning. A topic which went unmentioned, but which may become more important in the future, are hierarchical reinforcement learning methods and options, which allow constructing long-term actions (such as "go to the supermarket") from primitive actions (such as "open the door"). Finally, even though we mentioned the basic framework of regret minimisation, we have focused on the standard reinforcement learning problem, and ignored adversarial settings and problems with varying amounts of side information.

It is important to note that the book almost entirely elides social aspects of decision making. In practice, any algorithm that is going to be used to make autonomous decision is going to have a societal impact. In such cases, the algorithm designer must guard against negative externalities, such as hurting disadvantaged groups, violating privacy, or environmental damage. However, as a lot of these issues are context dependent, we urge the reader to consult recent work in economics, algorithmic fairness and differential privacy.

Appendix A

Symbols

\triangleq	definition
\wedge	logical and
\vee	logical or
\Rightarrow	implies
\Leftrightarrow	if and only if
\exists	there exists
\forall	for every
s.t.	such that

Table A.1: Logic symbols

$\{x_k\}$	a set indexed by k
$\{x \mid xRy\}$	the set of x satisfying relation xRy
\mathbb{N}	set of natural numbers
\mathbb{Z}	set of integers
\mathbb{R}	set of real numbers
Ω	the universe set (or sample space)
\emptyset	the empty set
Δ^n	the n -dimensional simplex
$\Delta(A)$	the collection of distributions over a set A
$\mathfrak{B}(A)$	the Borel σ -algebra induced by a set A
A^n	the product set $\prod_{i=1}^n A$
A^*	$\bigcup_{n=0}^{\infty} A^n$ the set of all sequences from set A
$x \in A$	x belongs to A
$A \subset B$	A is a (strict) subset of B
$A \subseteq B$	A is a (non-strict) subset of B
$B \setminus A$	set difference
$B \Delta A$	symmetric set difference
A^c	set complement
$A \cup B$	set union
$A \cap B$	set intersection

Table A.2: List of set theory symbols

\boldsymbol{x}^\top	the transpose of a vector \boldsymbol{x}
$ A $	the determinant of a matrix A
$\ x\ _p$	The p -norm of a vector $(\sum_i x_i ^p)^{1/p}$
$\ f\ _p$	The p -norm of a function $(\int f(x) ^p dx)^{1/p}$
$\ A\ _p$	The operator norm of a matrix $\max \{Ax \mid \ x\ _p = 1\}$
$\partial f(x)/\partial x_i$	Partial derivative with respect to x_i
∇f	Gradient vector of partial derivatives with respect to vector x

Table A.3: Analysis and linear algebra symbols

$\text{Beta}(\alpha, \beta)$ Beta distribution with parameters (α, β) . $\text{Geom}(\omega)$ Geometric distribution with parameter ω $\text{Wish}(n - 1, \nu)$ Wishart distribution with $n - 1$ degrees of freedom and ν shape parameters.

Table A.4: Miscellaneous statistics symbols

Appendix B

Probability concepts

This chapter is intended as a refresher of basic concepts in probability. This includes the definition of probability functions, expectations and moments. Perhaps unusually for an introductory text, we use the modern definition of probability as a *measure*, i.e. an additive function on sets.

Probability measures the likelihood of different events; where each event corresponds to a set in some universe of set. For that reason, we first remind the reader of elementary set theory and then proceed to describe how this relates to events.

B.1 Fundamental definitions

We start with ground set Ω that contains all objects we want to talk about. These objects are called the *elements* of Ω . Given a property Y of elements in Ω , one can define the set of all objects that satisfy this property. That is,

$$A \triangleq \{x \mid x \text{ have property } Y\}.$$

EXAMPLE 45.

$$B(c, r) \triangleq \{x \in \mathbb{R}^n \mid \|x - c\| \leq r\}$$

describes the set of points enclosed in an n -dimensional sphere of radius r with center $c \in \mathbb{R}^n$.

We use the following notations and definitions for sets. If an element x belongs to a set A , we write $x \in A$. Let the *sample space* Ω be a set such that $\omega \in \Omega$ always. We say that A is a *subset* of B or that B contains A , and write $A \subset B$, iff, $x \in B$ for any $x \in A$. Let $B \setminus A \triangleq \{x \mid x \in B \text{ and } x \notin A\}$ be the set difference. Let $A \Delta B \triangleq (B \setminus A) \cup (A \setminus B)$ be the symmetric set difference. The *complement* of any $A \subseteq \Omega$ is $A^c \triangleq \Omega \setminus A$. The *empty set* is $\emptyset = \Omega^c$. The *union* of n sets: A_1, \dots, A_n is $\bigcup_{i=1}^n A_i = A_1 \cup \dots \cup A_n$. The *intersection* of n sets A_1, \dots, A_n is $\bigcap_{i=1}^n A_i = A_1 \cap \dots \cap A_n$. A and B are *disjoint* if $A \cap B = \emptyset$. The *Cartesian product* or *product space* is defined as

$$\Omega_1 \times \dots \times \Omega_n = \{(s_1, \dots, s_n) \mid s_i \in \Omega_i, i = 1, \dots, n\} \quad (\text{B.1.1})$$

the set of all ordered n -tuples (s_1, \dots, s_n) .

B.1.1 Experiments and sample spaces

Conceptually, it might be easier to discuss concepts of probability if we think about this in terms of an experiment performed by a statistician. For example, such an experiment could be tossing a coin. The coin could come up heads, tails, balance exactly on the edge, get lost under the furniture, or simple disintegrate when it is tossed. The *sample space* of the experiment must contain *all possible* outcomes.

However, it is the statistician which determines what this set is. For example one statistician may only care whether the coin lands heads, or not (two outcomes). Another may care about how many times it bounces on the ground. Yet another may be interested in both the maximum height reached by the coin and how it lands. Thus, the sample space represents different aspects of the

experiment we are interested in. At the extreme, the sample space and corresponding outcomes may completely describe everything there is to know about the experiment.

Experiments

The set of possible experimental outcomes of an experiment is called the *sample space* Ω .

- Ω must contain all possible outcomes.
- After the experiment is performed, exactly one outcome ω in Ω is true.
- Each statistician i may consider a different Ω_i for the same experiment.

The following example considers the case where three different statisticians care about three different types of outcomes of an experiment where a drug is given to a patient. The first is interested in whether the patient recovers, the second in whether the drug has side-effects, while the third is interested in both.

EXAMPLE 46. Experiment: give medication to a patient.

- $\Omega_1 = \{\text{Recovery within a day, No recovery after a day}\}$.
- $\Omega_2 = \{\text{The medication has side-effects, No side-effect}\}$.
- $\Omega_3 = \text{all combinations of the above.}$

Clearly, the drug's effects are much more complex than the above simplified view. One could for example consider a very detailed patient state $\omega \in \Omega$ (which would e.g. describe every molecule in the patient's body)

Product spaces and repeated experiments

Sometimes we perform repeated experiments. Each experiment could be defined in a different outcome space, but many times we are specifically interested in repeated identical experiments. This occurs for example in situations where we give a treatment to patients suffering from a particular disease, and we measure the same outcomes (recovery, side-effects) in each one of them.

More formally, the set-up is as follows: We perform n experiments. The i -th experiment has sample space Ω_i . The sample space $\prod_{i=1}^n \Omega_i := \Omega_1 \times \dots \times \Omega_n$ can be thought of as a sample space of a *composite* experiment in which all n experiments are performed.

Identical experiment sample spaces In many cases, $\Omega_i = \Omega$ for all i , i.e. the sample space is identical for all individual experiments (e.g. n coin tosses). In this case we write $\Omega^n = \prod_{i=1}^n \Omega$.

B.2 Events, measure and probability

Probability is a type of function that is called a *measure*. In that sense it is similar to a function that weights, or measures things. Just like when weighing

two apples and adding the total gives you the same answer as weighing both apples together, so does the total probability of either of two mutually exclusive events equals the sum of their individual probabilities. However, sets are complex beasts and formally we wish to define exactly when we can measure them.

Many times the natural outcome space Ω that we wish to consider is extremely complex, but we only care about whether a specific *event* occurs or not. For example, when we toss a coin in the air, the natural outcome is the complete trajectory that the coin follows and its final resting position. However, we might only care about whether the coin lands heads or not. Then, the event of the coin landing “heads” is defined as all the trajectories that the coin follows which result in it landing heads. These trajectories form a subset $A \subset \Omega$.

Probabilities will always be defined on subsets of the outcome space. These subsets are termed events. The probability of events will simply be a function on sets, and more specifically a *measure*. The following gives some intuition and formal definitions about what this means.

B.2.1 Events and probability

Probability of a set

If A is a subset of Ω , the probability of A is a measure of the chances that the outcome of the experiment will be an element of A .

Which sets?

Ideally, we would like to be able to assign a probability to *every subset of Ω* . However, for technical reasons, this is not always possible.

EXAMPLE 47. Let X be uniformly distributed on $[0, 1]$. By definition, this means that the probability that X is in $[0, p]$ is equal to p for all $p \in [0, 1]$. However, even for this simple distribution, it might be difficult to define the probability of all events.

- What is the probability that X will be in $[0, 1/4]$?
- What is the probability that X will be in $[1/4, 1]$?
- What is the probability that X will be a rational number?

B.2.2 Measure theory primer

Imagine that you have an apartment Ω composed of three rooms, A, B, C . There are some coins on the floor and a 5-meter-long red carpet. We can measure various things in this apartment.

Area

- A: $4 \times 5 = 20m^2$.

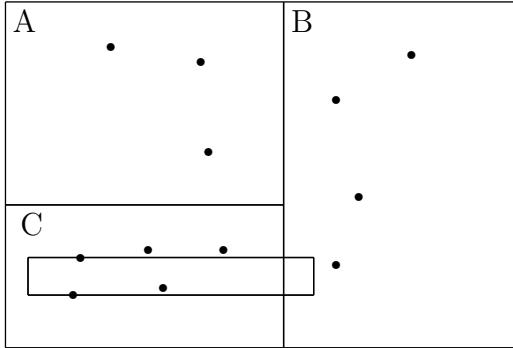


Figure B.1: A fashionable apartment

- B: $6 \times 4 = 24m^2$.
- C: $2 \times 5 = 10m^2$.

Coins on the floor

- A: 3.
- B: 4
- C: 5.

Length of red carpet

- A: 0m
- B: 0.5m
- C: 4.5m.

Measure the sets: $\mathcal{F} = \{\emptyset, A, B, C, A \cup B, A \cup C, B \cup C, A \cup B \cup C\}$. It is easy to see that the union of any sets in \mathcal{F} is also in \mathcal{F} . In other words, \mathcal{F} is closed under union. Furthermore, \mathcal{F} contains the whole space Ω .

Note that all those measures have an *additive property*.

B.2.3 Measure and probability

As previously mentioned, the probability of $A \subseteq \Omega$ is a measure of the chances that the outcome of the experiment will be an element of A . Here we give a precise definition of what we mean by measure and probability.

If we want to be able to perform probabilistic logic, we need to define some appropriate algebraic construction that relates events to each other. In particular, if we have a family of events \mathcal{F} , i.e. a collection of subsets of Ω , we want this to be closed under union and complement.

Definition B.2.1 (A field on Ω). A family \mathcal{F} of sets, such that for each $A \in \mathcal{F}$, one also has $A \subseteq \Omega$, is called a *field on Ω* if and only if

1. $\Omega \in \mathcal{F}$
2. if $A \in \mathcal{F}$, then $A^c \in \mathcal{F}$.
3. For any A_1, A_2, \dots, A_n such that $A_i \in \mathcal{F}$, it holds that: $\bigcup_{i=1}^n A_i \in \mathcal{F}$.

From the above definition, it is easy to see that $A_i \cap A_j$ is also in the field. Since many times our family may contain an infinite number of sets, we also want to extend the above to countably infinite unions.

Definition B.2.2 (σ -field on Ω). A family \mathcal{F} of sets, such that $\forall A \in \mathcal{F}, A \subseteq \Omega$, is called a σ -field on Ω if and only if

1. $\Omega \in \mathcal{F}$
2. if $A \in \mathcal{F}$, then $A^c \in \mathcal{F}$.
3. For any sequence A_1, A_2, \dots such that $A_i \in \mathcal{F}$, it holds that: $\bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$.

It is easy to verify that the \mathcal{F} given in the apartment example satisfies these properties. In general, for any finite Ω , it is easy to find a family \mathcal{F} containing all possible events in Ω . Things become trickier when Ω is infinite. Can we define an algebra \mathcal{F} that contains all events? In general no, but we can define an algebra on the so-called Borel sets of Ω , defined in B.2.3.

Definition B.2.3 (Measure). A measure λ on (Ω, \mathcal{F}) is a function $\lambda : \mathcal{F} \rightarrow \mathbb{R}^+$ such that

1. $\lambda(\emptyset) = 0$.
2. $\lambda(A) \geq 0$ for any $A \in \mathcal{F}$.
3. For any collection of subsets A_1, \dots, A_n with $A_i \in \mathcal{F}$ and $A_i \cap A_j = \emptyset$.

$$\lambda \left(\bigcup_{i=1}^{\infty} A_i \right) = \sum_{i=1}^{\infty} \lambda(A_i) \quad (\text{B.2.1})$$

It is easy to verify that the floor area, the number of coins, and the length of the red carpet are all measures. In fact, the area and length correspond to what is called a *Lebesgue measure*¹ and the number of coins to a *counting measure*.

Definition B.2.4 (Probability measure). A probability measure P on (Ω, \mathcal{F}) is a function $P : \mathcal{F} \rightarrow [0, 1]$ such that:

1. $P(\Omega) = 1$

¹See Section B.2.3 for a precise definition.

2. $P(\emptyset) = 0$
3. $P(A) \geq 0$ for any $A \in \mathcal{F}$.
4. If A_1, A_2, \dots are (pairwise) disjoint then

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i) \quad (\text{union})$$

(Ω, \mathcal{F}, P) is called a *probability space*.

So, probability is just a special type of measure.

The Lebesgue measure*

Definition B.2.5 (Outer measure). Let $(\Omega, \mathcal{F}, \lambda)$ be a measure space. The outer measure of a set $A \subseteq \Omega$ is:

$$\lambda^*(A) \triangleq \inf_{A \subseteq \bigcup_k B_k} \sum_k \lambda(B_k). \quad (\text{B.2.2})$$

In other words, it is the measure λ -measure of the smallest cover $\{B_k\}$ of A .

Definition B.2.6 (Inner measure). Let $(\Omega, \mathcal{F}, \lambda)$ be a measure space. The inner measure of a set $A \subseteq \Omega$ is:

$$\lambda_*(A) \triangleq \lambda(\Omega) - \lambda(\Omega \setminus A). \quad (\text{B.2.3})$$

Definition B.2.7 (Lebesgue measurable sets). A set A is (Lebesgue) measurable if the outer and inner measures are equal.

$$\lambda^*(A) = \lambda_*(A). \quad (\text{B.2.4})$$

The common value of the inner and outer measure is called the Lebesgue measure² $\bar{\lambda}(A) = \lambda^*(A)$.

The Borel σ -algebra*

When Ω is a finite collection $\{\omega_1, \dots, \omega_n\}$, there is a σ -algebra containing all possible events in Ω , denoted 2^Ω . This is called the *powerset*. However, in general this is not possible. For infinite sets equipped with a metric, we can instead define the *Borel σ -algebra* $\mathfrak{B}(\Omega)$, which is the smallest σ -algebra containing all *Borel σ -algebra* open sets of Ω .

B.3 Conditioning and independence

A probability measure can give us the probability of any set in the algebra. Each one of these sets can be seen as an *event*. For example, the set of all states where a patient has a fever constitutes the event that the patient has a fever. Thus, generally we identify events with subsets of Ω .

²It is easy to see that $\bar{\lambda}$ is a measure.

However, the basic probability on Ω does not tell us anything about what the probability of some event A , given the fact that some event B has occurred. Sometimes, these events are *mutually exclusive*, meaning that when B happens, A cannot be true; other times B implies A , and sometimes they are *independent*. To quantify exactly how knowledge of whether B has occurs can affect what we know about A , we need the notion of *conditional probability*.

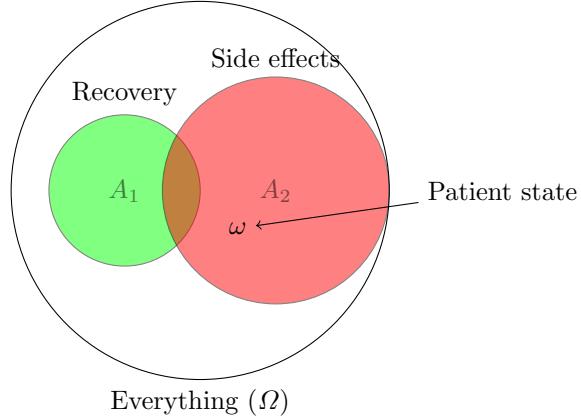


Figure B.2: Events as sets. The patient state $\omega \in \Omega$ after submitting to a treatment may belong to either of the two possible sets A_1, A_2 .

B.3.1 Mutually exclusive events

By events, we mean subsets of Ω . Thus, the probability of the event that a draw from Ω is in A is equal to the probability measure of A , $P(A)$. Some events are mutually exclusive, meaning that they can never happen at the same time. This is the same as saying that the corresponding sets have an empty intersection.

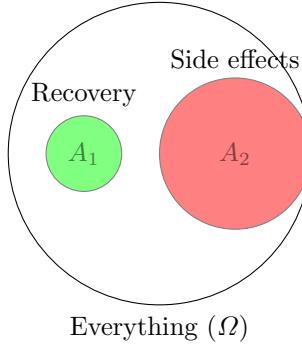


Figure B.3: Mutually exclusive events A_1, A_2 .

Definition B.3.1 (Mutually exclusive events). Two events A, B are mutually exclusive if and only if $A \cap B = \emptyset$.

By definition of the measure, $P(A \cup B) = P(A) + P(B)$ for any mutually exclusive events.

Lemma B.3.1 (Union bound). *For any events A, B , it holds that*

$$P(A \cup B) \leq P(A) + P(B). \quad (\text{B.3.1})$$

Proof. Let $C = A \cap B$. Then

$$\begin{aligned} P(A) + P(B) &= P(A \setminus C) + P(C) + P(B \setminus C) + P(C) \\ &\geq P(A \setminus C) + P(C) + P(B \setminus C) = P(A \cup B) \end{aligned}$$

□

The union bound is extremely important, and one of the basic proof methods in many applications of probability.

Finally, let us consider the general case of multiple disjoint events, shown in Figure B.4. When B is decomposed in a set of disjoint events $\{B_i\}$, we can

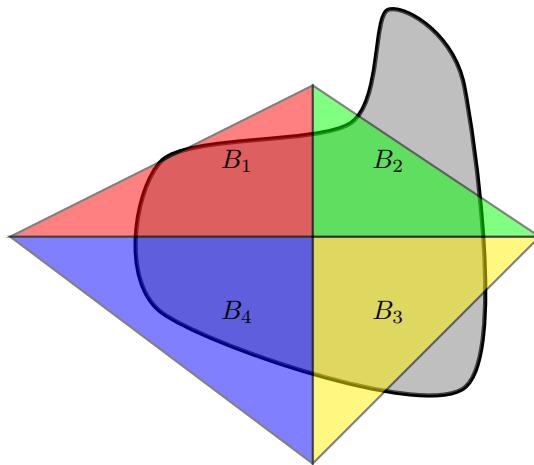


Figure B.4: Disjoint events and marginalisation

write:

$$P(B) = P\left(\bigcup_i B_i\right) = \sum_i P(B_i) \quad (\text{B.3.2})$$

$$P(A \cap B) = P\left(\bigcup_i (A \cap B_i)\right) = \sum_i P(A \cap B_i), \quad (\text{B.3.3})$$

for any other set A . An interesting special case occurs when $B = \Omega$, in which case $P(A) = P(A \cap \Omega)$, since $A \subset \Omega$ for any A in the algebra. This results in the *marginalisation* or *sum rule* of probability.

*marginalisation
sum rule*

$$P(A) = P\left(\bigcup_i (A \cap B_i)\right) = \sum_i P(A \cap B_i), \quad \bigcup_i B_i = \Omega. \quad (\text{B.3.4})$$

B.3.2 Independent events

Sometimes different events are independent, in the sense there is no interaction between their probabilities. This can be formalised as follows.

Definition B.3.2 (Independent events). Two events A, B are independent if $P(A \cap B) = P(A)P(B)$. The events in a family \mathcal{F} of events are independent if for any sequence A_1, A_2, \dots of events in \mathcal{F} ,

$$P\left(\bigcap_{i=1}^n A_i\right) = \prod_{i=1}^n P(A_i) \quad (\text{independence})$$

As a simple example, consider Figure B.5, where the universe is a rectangle of dimensions $(1, 1)$, two events A_1, A_2 are rectangles with A_1 having dimensions $(1, h)$ and A_2 having dimensions (w, h) . Let's take the probability distribution P which assigns probability $P(A)$ equal to the *area* of the set A . Then

$$P(A_1) = 1 \times h = h, \quad P(A_2) = w \times 1 = w.$$

Similarly, the intersecting rectangle has dimensions (w, h) . Consequently

$$P(A_1 \cup A_2) = w \times h = P(A_1) \times P(A_2)$$

and the two events are independent. Independent events are particularly im-

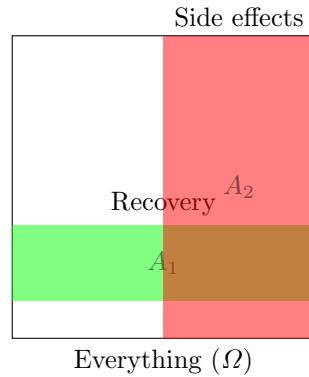


Figure B.5: Independent events A_1, A_2 .

portant in repeated experiments, where the outcomes of one experiment are independent of the outcome of another.

B.3.3 Conditional probability

Now that we have defined a distribution for all possible events, and we have also defined basic relationships between events, we'd also like to have a way of determining the probability of one event given that another has occurred. This is given by the notion of conditional probability.

Definition B.3.3 (Conditional probability). The conditional probability of A when B , s.t. $P(B) > 0$, is given is:

$$P(A | B) \triangleq \frac{P(A \cap B)}{P(B)}. \quad (\text{B.3.5})$$

Note that we can always write $P(A \cap B) = P(A | B)P(B)$ even if A, B are not independent.

Finally, we say that two events A, B are *conditionally independent* given C *conditionally independent* if

$$P(A \cap B | C) = P(A | C)P(B | C). \quad (\text{B.3.6})$$

This is an important notion when dealing with probabilistic graphical models.

B.3.4 Bayes' theorem

The following theorem trivially follows from the above discussion. However, versions of it shall be used repeatedly throughout the book. For this reason we present it here together with a detailed proof.

Theorem B.3.1 (Bayes' theorem). *Let A_1, A_2, \dots be a (possibly infinite) sequence of disjoint events such that $\bigcup_{i=1}^{\infty} A_i = \Omega$ and $P(A_i) > 0$ for all i . Let B be another event with $P(B) > 0$. Then*

$$P(A_i | B) = \frac{P(B | A_i)P(A_i)}{\sum_{j=1}^{\infty} P(B | A_j)P(A_j)} \quad (\text{B.3.7})$$

Proof. From (B.3.5), $P(A_i | B) = P(A_i \cap B)/P(B)$ and also $P(A_i \cap B) = P(B | A_i)P(A_i)$. Thus

$$P(A_i | B) = \frac{P(B | A_i)P(A_i)}{P(B)},$$

and we continue analyzing the denominator $P(B)$. First, due to $\bigcup_{i=1}^{\infty} A_i = \Omega$ we have $B = \bigcup_{j=1}^{\infty} (B \cap A_j)$. Since A_i are disjoint, so are $B \cap A_i$. Then from the union property of probability distributions we have

$$P(B) = P\left(\bigcup_{j=1}^{\infty} (B \cap A_j)\right) = \sum_{j=1}^{\infty} P(B \cap A_j) = \sum_{j=1}^{\infty} P(B | A_j)P(A_j),$$

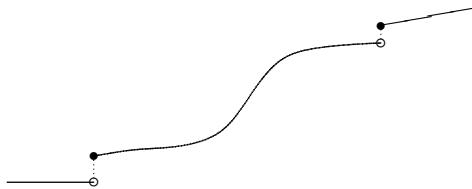
which finishes the proof. \square

B.4 Random variables

A random variable X is a special kind of random quantity, defined as a function of outcomes in Ω to some vector space. Unless otherwise stated, the mapping is on the real numbers \mathbb{R} . Thus, it also defines a mapping from a probability measure P on (Ω, \mathcal{F}) to a probability measure P_X on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$. More precisely, we define the following.

Definition B.4.1 (Measurable function). Let \mathcal{F} on Ω be a σ -field. A function $g : \Omega \rightarrow \mathbb{R}$ is said to be *measurable with respect to \mathcal{F}* , or \mathcal{F} -measurable, if, for any $x \in \mathbb{R}$,

$$\{s \in \Omega \mid g(s) \leq x\} \in \mathcal{F}.$$

Figure B.6: A distribution function F

Definition B.4.2 (Random variable). Let (Ω, \mathcal{F}, P) be a probability space. A random variable $X : \Omega \rightarrow \mathbb{R}$ is a real-valued, \mathcal{F} -measurable function.

The distribution of X

Every random variable X induces a probability measure P_X on \mathbb{R} . For any $B \subseteq \mathbb{R}$ we define

$$P_X(B) \triangleq \mathbb{P}(X \in B) = P(\{s \mid X(s) \in B\}). \quad (\text{B.4.1})$$

Thus, the probability that X is in B is equal to the P -measure of the points $s \in \Omega$ such that $X(s) \in B$ and also equal to the P_X -measure of B .

Here \mathbb{P} is used as a *short-hand* notation.

EXERCISE 36. Ω is the set of 52 playing cards. $X(s)$ is the value of each card (1, 10 for the ace and figures respectively). What is the probability of drawing a card s with $X(s) > 7$?

B.4.1 (Cumulative) Distribution functions

Definition B.4.3 ((Cumulative) Distribution function). The distribution function of a random variable X is the function $F : \mathbb{R} \rightarrow \mathbb{R}$:

$$F(t) = \mathbb{P}(X \leq t). \quad (\text{B.4.2})$$

Properties

- If $x \leq y$, then $F(x) \leq F(y)$.
- F is right-continuous.
- At the limit,

$$\lim_{t \rightarrow -\infty} F(t) = 0, \quad \lim_{t \rightarrow \infty} F(t) = 1.$$

B.4.2 Discrete and continuous random variables

On the real line, there are two types of distributions for a random variable. Here, once more, we employ the \mathbb{P} notation as a shorthand for the probability of general events involving random variables, so that we don't have to deal with the measure notation. The two following examples should give some intuition.

Discrete distributions

$X : \Omega \rightarrow \{x_1, \dots, x_n\}$ takes n discrete values (n can be infinite). The probability function of X is

$$f(x) \triangleq \mathbb{P}(X = x),$$

defined for $x \in \{x_1, \dots, x_n\}$. For any $B \subseteq \mathbb{R}$:

$$P_X(B) = \sum_{x_i \in B} f(x_i).$$

In addition, we write $\mathbb{P}(X \in B)$ to mean $P_X(B)$.

Continuous distributions

X has a continuous distribution if there exists a *probability density function* f s.t. $\forall B \subseteq \mathbb{R}$:

$$P_X(B) = \int_B f(x) dx.$$

B.4.3 Random vectors

We can generalise the above to random *vectors*. These can be seen as *vectors* of random variables. These are just random variables on some Cartesian product space, i.e. $X : \Omega \rightarrow \mathcal{V}$, with $\mathcal{V} = V_1 \times \dots \times V_m$. Once more, there are two special cases of distributions for the random vector $X = (X_1, \dots, X_m)$. The first is a vector of discrete random variables:

Discrete distributions

$$\mathbb{P}(X_1 = x_1, \dots, X_m = x_m) = f(x_1, \dots, x_m),$$

where f is *joint probability function*, with $x_i \in V_i$.

The second is a vector of continuous random variables.

Continuous distributions

For $B \subseteq \mathbb{R}^m$

$$\mathbb{P}\{(X_1, \dots, X_m) \in B\} = \int_B f(x_1, \dots, x_m) dx_1 \cdots dx_m$$

In general, it is possible that X has neither a continuous, nor a discrete distribution; for example if some V_i is discrete and some V_j are continuous. In that case it is convenient to use measure-theoretic notation, explained in the next section.

B.4.4 Measure-theoretic notation

The previously seen special cases of discrete and continuous variables can be handled with a unified notation if we take advantage of the fact that probability is only a particular type of measure. As a first step, we note that summation can also be seen as integration with respect to the counting measure and that Riemann integration is integration with respect to the Lebesgue measure.

Integral with respect to a measure μ

Introduce the common notation $\int \cdots d\mu(x)$, where μ is a measure. Let some real function $g : \Omega \rightarrow \mathbb{R}$. Then for any subset $B \subseteq \Omega$ we can write

- Discrete case: f is the probability function and we choose the *counting measure* for μ , so:

$$\sum_{x \in B} g(x)f(x) = \int_B g(x)f(x) d\mu(x)$$

Roughly speaking, the counting measure $\mu(\Omega)$ is equal to the number of elements in Ω .

- Continuous case: f is the probability density function and we choose the *Lebesgue measure* for μ , so:

$$\int_B g(x)f(x) dx = \int_B g(x)f(x) d\mu(x)$$

Roughly speaking, the Lebesgue measure $\mu(S)$ is equal to the volume of S .

In fact, since probability is a measure in itself, we do not need to complicate things by using f and μ at the same time! This allows us to use the following notation.

Lebesgue-Stiletjes notation

If P is a probability measure on (Ω, \mathcal{F}) and $B \subseteq \Omega$, and g is \mathcal{F} -measurable, we write the probability that $g(x)$ takes the value B can be written equivalently as:

$$\mathbb{P}(g \in B) = P_g(B) = \int_B g(x) dP(x) = \int_B g dP. \quad (\text{B.4.3})$$

Intuitively, dP is related to densities in the following way. If P is a measure on Ω and is absolutely continuous with respect to another measure μ , then $p \triangleq \frac{dP}{d\mu}$ is the (Radon-Nikodym) derivative of P with respect to μ . We write the integral as $\int g p d\mu$. If μ is the Lebesgue measure, then p coincides with the probability density function.

B.4.5 Marginal distributions and independence

Although this is a straightforward outcome of the set-theoretic definition of probability, we also define the marginal explicitly for random vectors.

Marginal distribution

The marginal distribution of X_1, \dots, X_k from a set of variables X_1, \dots, X_m , is

$$\mathbb{P}(X_1, \dots, X_k) \triangleq \int \mathbb{P}(X_1, \dots, X_k, X_{k+1} = x_{k+1}, \dots, X_m = x_m) d\mu(x_{k+1}, \dots, x_m). \quad (\text{B.4.4})$$

In the above, $\mathbb{P}(X_1, \dots, X_k)$ can be thought of as the probability measure for any events related to the random vector (X_1, \dots, X_k) . Thus, it defines a probability measure over $(\mathbb{R}^k, \mathcal{B}(\mathbb{R}^k))$. In fact, let $Y = (X_1, \dots, X_k)$ and $Z = (X_{k+1}, \dots, X_m)$ for simplicity. Then define $Q(A) \triangleq \mathbb{P}(Z \in A)$, with $A \subseteq \mathbb{R}^{m-k-1}$. Then the above can be re-written as:

$$\mathbb{P}(Y \in B) = \int_{\mathbb{R}^{m-k-1}} \mathbb{P}(Y \in B \mid Z = z) dQ(z).$$

Similarly, $\mathbb{P}(Y \mid Z = z)$ can be thought of as a function mapping from values of Z to probability measures. Let $P_z(B) \triangleq \mathbb{P}(Y \in B \mid Z = z)$ be this measure corresponding to a particular value of z . Then we can write

$$\mathbb{P}(Y \in B) = \int_{\mathbb{R}^{m-k-1}} \left(\int_B dP_z(y) \right) dQ(z).$$

Independence

If X_i is independent of X_j for all $i \neq j$:

$$\mathbb{P}(X_1, \dots, X_m) = \prod_{i=1}^M \mathbb{P}(X_i), \quad f(x_1, \dots, x_m) = \prod_{i=1}^M g_i(x_i) \quad (\text{B.4.5})$$

B.4.6 Moments

There are some simple properties of the random variable under consideration which are frequently of interest in statistics. Two of those properties are *expectation* and *variance*.

expectation

Expectation

Definition B.4.4. The expectation $\mathbb{E}(X)$ of any random variable $X : \Omega \rightarrow R$, where R is a vector space, with distribution P_X is defined by

$$\mathbb{E}(X) \triangleq \int_R t dP_X(t), \quad (\text{B.4.6})$$

as long as the integral exists.

Furthermore,

$$\mathbb{E}[g(X)] = \int g(t) dP_X(t),$$

for any function g .

variance

Definition B.4.5. The *variance* $\mathbb{V}(X)$ of any random variable $X : \Omega \rightarrow \mathbb{R}$ with distribution P_X is defined by

$$\begin{aligned} \mathbb{V}(X) &\triangleq \int_{-\infty}^{\infty} [t - \mathbb{E}(X)]^2 dP_X(t) \\ &= \mathbb{E}\left\{[X - \mathbb{E}(X)]^2\right\} \\ &= \mathbb{E}(X^2) - \mathbb{E}^2(X). \end{aligned} \quad (\text{B.4.7})$$

When $X : \Omega \rightarrow R$ with R an arbitrary vector space, the above becomes the *covariance matrix*:

$$\begin{aligned} \mathbb{V}(X) &\triangleq \int_{-\infty}^{\infty} [t - \mathbb{E}(X)][t - \mathbb{E}(X)]^\top dP_X(t) \\ &= \mathbb{E}\left\{[X - \mathbb{E}(X)][X - \mathbb{E}(X)]^\top\right\} \\ &= \mathbb{E}(XX^\top) - \mathbb{E}(X)\mathbb{E}(X)^\top. \end{aligned} \quad (\text{B.4.8})$$

B.5 Divergences

Divergences are a natural way to measure how different two distributions are.

KL-Divergence

Definition B.5.1. The *KL-Divergence* is a non-symmetric divergence.

$$D(P \parallel Q) \triangleq \int \frac{dP}{dQ} dP. \quad (\text{B.5.1})$$

Another useful distance is the L_1 distance

Definition B.5.2. The L_1 -distance between two measures is defined as:

$$\|P - Q\|_1 = \int \left| \frac{dP}{d\mu} - \frac{dQ}{d\mu} \right| d\mu, \quad (\text{B.5.2})$$

where μ is any measure dominating both P and Q .

B.6 Empirical distributions

When we have no model for a particular distribution, it is sometimes useful to construct the *empirical distribution*, which basically counts how many times we observe different outcomes.

Definition B.6.1. Let $x^n = (x_1, \dots, x_n)$ drawn from a product measure $x^n \sim P^n$ on the measurable space $(\mathcal{X}^n, \mathcal{F}_n)$. Let \mathfrak{S} be any σ -field on \mathcal{X} . Then empirical distribution of x^n is defined as

$$\hat{P}_n(B) \triangleq \frac{1}{n} \sum_{t=1}^n \mathbb{I}\{x_t \in B\}. \quad (\text{B.6.1})$$

The problem with the empirical distribution is that does not capture the uncertainty we have about what the real distribution is. For that reason, it should be used with care, even though it does converge to the true distribution in the limit. A clever way to construct a measure of uncertainty is to perform *sub-sampling*, that is to create k random samples of size $n' < n$ from the original sample. Each sample will correspond to a different random empirical distribution. Sub-sampling is performed *without replacement* (i.e. for each sample, each observation x_i is only used once). When sampling with replacement and $n' = n$, the method is called *bootstrapping*.

sub-sampling

without replacement

bootstrapping

B.7 Further reading

Much of this material is based on DeGroot [1970]. See Kolmogorov and Fomin [1999] for a really clear exposition of measure, starting from rectangle areas (developed from course notes in 1957). Also see Savage [1972] for a verbose, but interesting and rigorous introduction to subjective probability. A good recent text on elementary probability and statistical inference is Bertsekas and Tsitsiklis [2008].

B.8 Exercises

EXERCISE 37 (5). Show that for any sets A, B, D :

$$A \cap (B \cup D) = (A \cap B) \cup (A \cap D).$$

Show that

$$(A \cup B)^c = A^c \cap B^c, \quad \text{and} \quad (A \cap B)^c = A^c \cup B^c$$

EXERCISE 38 (10). Prove that any probability measure P has the following properties:

1. $P(A^c) = 1 - P(A)$.
2. If $A \subset B$ then $P(A) \leq P(B)$.
3. For any sequence of events A_1, \dots, A_n

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) \leq \sum_{i=1}^{\infty} P(A_i) \quad (\text{union bound})$$

Hint: Recall that If A_1, \dots, A_n are disjoint then $P(\bigcup_{i=1}^n A_i) = \sum_{i=1}^n P(A_i)$ and that $P(\emptyset) = 0$

Definition B.8.1. A random variable $X \in \{0, 1\}$ has Bernoulli distribution with parameter $p > [0, 1]$, written $X \sim \text{Bern}(p)$, if

$$p = \mathbb{P}(X = 1) = 1 - \mathbb{P}(X = 0).$$

The probability function of X can be written as

$$f(x \mid p) = \begin{cases} p^x (1-p)^{1-x}, & x \in \{0, 1\} \\ 0, & \text{otherwise.} \end{cases}$$

Definition B.8.2. A random variable $X \in \{0, 1\}$ has a binomial distribution with parameters $p > [0, 1]$, $n \in \mathbb{N}$ written $X \sim \text{Binom}(p, n)$, if the probability function of X is

$$f(x \mid n, p) = \begin{cases} \binom{n}{x} p^x (1-p)^{n-x}, & x \in \{0, 1, \dots, n\} \\ 0, & \text{otherwise.} \end{cases}$$

If X_1, \dots, X_n is a sequence of Bernoulli random variables with parameter p , then $\sum_{i=1}^n X_i$ has a binomial distribution with parameters n, p .

EXERCISE 39 (10). Let $X \sim \text{Bern}(p)$

1. Show that $\mathbb{E} X = p$
2. Show that $\mathbb{V} X = p(1-p)$
3. Find the value of p for which X has the greatest variance.

EXERCISE 40 (10). In a few sentences, describe your views on the usefulness of probability.

- Is it the only formalism that can describe both random events and uncertainty?
- Would it be useful to separate randomness from uncertainty?
- What would be desirable properties of an alternative concept?

Appendix C

Useful results

C.1 Functional Analysis

Definition C.1.1 (supremum). When we say that

$$M = \sup_{x \in A} f(x),$$

then: (i) $M \geq f(x)$ for any $x \in A$. In other words, M is an upper bound on $f(x)$. (ii) for any $M' < M$, there exists some $x' \in A$ s.t. $M' < f(x')$.

In other words, there exists no smaller upper bound than M . When the function f has a maximum, then the supremum is identical to the maximum.

Definition C.1.2 (infimum). When we say that

$$M = \inf_{x \in A} f(x),$$

then: (i) $M \leq f(x)$ for any $x \in A$. In other words, M is an lower bound on $f(x)$. (ii) for any $M' > M$, there exists some $x' \in A$ s.t. $M' > f(x')$.

Norms Let (S, Σ, μ) be a measure space. The L_p norm of a μ -measurable function f is defined as

$$\|f\|_p = \left(\int_S |f(x)|^p d\mu(x) \right)^{1/p}. \quad (\text{C.1.1})$$

Hölder inequality. Let (S, Σ, μ) be a measure space and let $1 \leq p, q \leq \infty$ with $1/p + 1/q = 1$ then for all μ -measurable f, g :

$$\|fg\|_1 \leq \|f\|_p \|g\|_q. \quad (\text{C.1.2})$$

The special case $p = q = 2$ results in the Cauchy-Schwarz inequality.

Lipschitz continuity We say that a function $f : X \rightarrow Y$ is Lipschitz, with respect to metrics d, ρ on X, Y respectively when

$$\rho(f(a) - f(b)) \leq d(a, b) \quad \forall a, b \in X. \quad (\text{C.1.3})$$

Special spaces. The n -dimensional Euclidean space is denoted by \mathbb{R}^n .

The n -dimensional simplex is denoted by Δ^n and it holds that for any $\mathbf{x} \in \Delta^n$, $\|\mathbf{x}\|_1 = 1$ and $x_k \geq 0$.

C.1.1 Series

Definition C.1.3 (The geometric series). The sum $\sum_{k=0}^n x^k$ is called the geometric series and has the property

$$\sum_{k=0}^n x^k = \frac{x^{n+1} - 1}{x - 1}. \quad (\text{C.1.4})$$

Taking derivatives with respect to x can result in other useful formulae.

C.1.2 Special functions

Definition C.1.4 (Gamma function). For a positive integer n ,

$$\Gamma(n) = (n - 1)! \quad (\text{C.1.5})$$

For a positive real numbers (or complex numbers with a positive real part), the gamma function is defined as

$$\Gamma(t) = \int_0^\infty x^{t-1} e^{-x} dx. \quad (\text{C.1.6})$$

Appendix D

Index

Index

- . , 100
- Adaptive hypothesis testing, 106
- Adaptive treatment allocation, 106
- approximate
 - policy iteration, 150
- average reward criterion, 179
- backwards induction, 110
- bandit problems, 107
 - stochastic, 107
- Bayes rule, 52
- Bayes' theorem, 21
- belief state, 109
- Beta distribution, 63
- bimomial coefficient, 62
- bootstrapping, 247
- Borel σ -algebra, 237
- branch and bound, 199
- classification, 51
- clinical trial, 106
- concave function, 27
- conditional probability, 240
- conditionally independent, 241
- covariance matrix, 246
- decision boundary, 51
- decision procedure
 - sequential, 86
- difference operator, 130
- discount factor, 107
- distribution
 - χ^2 , **66**
 - Bernoulli, **62**
 - Beta, **63**
 - binomial, **62**
 - exponential, **68**
 - Gamma, **67**
 - marginal, 88
 - normal, **66**
- divergences, 246
- empirical distribution, 247
- every-visit Monte-Carlo, 149
- expectation, 245
- experimental design, 106
- exploration vs exploitation, 11
- fairness, 52
- first visit
 - Monte-Carlo update, 149
- gamma function, 63
- Gaussian processes, 204
- Hoeffding inequality, 117
- inequality
 - Chebyshev, **78**
 - Hoeffding, **78**
 - Markov, **77**
- inf, *see* infimum
- infimum, 252
- Jensen's inequality, 27
- KL-Divergence, 246
- likelihood
 - conditional, 19
 - relative, 16
- linear programming, 133
- marginalisation, 239
- Markov decision process, 106, 110, **112**, 116, 135
- Markov process, 100
- martingale, 99
- matrix determinant, 72
- mixture of distributions, 38
- Monte Carlo
 - Policy evaluation, 148
- multinomial, 71
- multivariate-normal, 72

- observation distribution, 201
- policy, 107, 113
 ϵ -greedy, 143
 k -order Markov, 190
 blind, 190
 history-dependent, 113
 Markov, 113
 memoryless, 190
 optimal, 115
 stochastic, 162
- policy evaluation, 115
 backwards induction, 117
 Monte Carlo, 116
- policy iteration, 128
 modified, 130
 temporal-difference, 131
- policy optimisation
 backwards induction, 118
- powerset, 237
- preference, 22
- probability
 subjective, 16
- random vector, 243
- reset action, 148
- reward, 22
- reward distribution, 112, 201
- sample mean, 58
- series
 geometric, 90, **252**
- simulation, 148
- spectral radius, 123
- standard normal, 66
- statistic, 58
 sufficient, **59**
- stopping function, 86
- stopping set, 87
- student t -distribution, 70
- sub-sampling, 247
- sum rule, 239
- sup, *see* supremum
- supremum, 252
- temporal difference, 131
- temporal difference error, 132
- temporal differences, 130
- termination condition, 127
- trace, 73
- transition distribution, 112, 201
- unbounded procedures, 90
- union bound, 239
- utility, 24, 114
- Utility theory, 22
- value, 88
- value function
 optimal, 115
 state, 114
 state-action, 114
- value iteration, 126
- variable order Markov decision process, 202
- variance, 246
- Wald's theorem, 98
- wishart, 73
- without replacement, 247

Bibliography

- M. Alvarez, D. Luengo-Garcia, M. Titsias, and N. Lawrence. Efficient multi-output gaussian processes through variational inducing kernels. 2011.
- A. Antos, C. Szepesvari, and R. Munos. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1):89–129, 2008a.
- Andre Antos, Remi Munos, and Csaba Szepesvari. Fitted Q-iteration in continuous action-space MDPs. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge, MA, 2008b.
- Robert B. Ash and Catherine A. Doleans-Dade. *Probability & Measure Theory*. Academic Press, 2000.
- J. Asmuth, L. Li, M. L. Littman, A. Nouri, and D. Wingate. A Bayesian sampling approach to exploration in reinforcement learning. In *UAI 2009*, 2009.
- Jean-Yves Audibert and Sebastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *colt2009. Proceedings of the 22nd Annual Conference on Learning Theory*, pages 217–226, 2009.
- P. Auer, R. Ortner, and C. Szepesvari. Improved Rates for the Stochastic Continuum-Armed Bandit Problem. In *Learning Theory*, volume 4539 of *Lecture Notes in Computer Science*, page 454. Springer, 2007.
- Peter Auer and Ronald Ortner. UCB revisited: improved regret bounds for the stochastic multi-armed bandit problem. *Period. Math. Hungar.*, 61(1-2):55–65, 2010.
- Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2/3):235–256, 2002a.
- Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002b. doi: 10.1137/S0097539701398375. URL <http://dx.doi.org/10.1137/S0097539701398375>.
- Andrew G Barto. Adaptive critics and the basal ganglia. *Models of information processing in the basal ganglia*, page 215, 1995.

- Jonathan Baxter and Peter L. Bartlett. Reinforcement learning in POMDP's via direct gradient ascent. In *Proc. 17th International Conf. on Machine Learning*, pages 41–48. Morgan Kaufmann, San Francisco, CA, 2000. URL citeseer.nj.nec.com/baxter00reinforcement.html.
- Richard Ernest Bellman. A problem in the sequential design of experiments. *Sankhya*, 16:221–229, 1957.
- A. Bernstein. Adaptive state aggregation for reinforcement learning. Master's thesis, Technion – Israel Institute of Technology, 2007.
- Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.
- Dimitri P. Bertsekas and John N. Tsitsiklis. *Introduction to Probability: Dimitri P. Bertsekas and John N. Tsitsiklis*. Athena Scientific, 2008.
- J. A. Boyan. Technical update: Least-squares temporal difference learning. *Machine Learning*, 49(2):233–246, 2002.
- S. J. Bradtko and A. G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22(1):33–57, 1996.
- R. I. Brafman and M. Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *The Journal of Machine Learning Research*, 3:213–231, 2003. ISSN 1532-4435.
- Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012. doi: 10.1561/2200000024. URL <http://dx.doi.org/10.1561/2200000024>.
- Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. X-armed bandits. *Journal of Machine Learning Research*, 12:1655–1695, 2011.
- Apostolos N. Burnetas and Michael N. Katehakis. Optimal adaptive policies for Markov decision processes. *Math. Oper. Res.*, 22(1):222–255, 1997.
- George Casella, Stephen Fienberg, and Ingram Olkin, editors. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer, 1999.
- Herman Chernoff. Sequential design of experiments. *Annals of Mathematical Statistics*, 30(3):755–770, 1959.
- Herman Chernoff. Sequential models for clinical trials. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol.4*, pages 805–812. Univ. of Calif Press, 1966.
- Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. Technical Report 1610.07524, arXiv, 2016.
- Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. Technical Report 1701.08230, arXiv, 2017.

- K. Csilléry, M. G. B. Blum, O. E. Gaggiotti, O. François, et al. Approximate Bayesian computation (ABC) in practice. *Trends in ecology & evolution*, 25(7):410–418, 2010.
- Richard Dearden, Nir Friedman, and Stuart J. Russell. Bayesian Q-learning. In *AAAI/IAAI*, pages 761–768, 1998. URL citeseer.ist.psu.edu/dearden98bayesian.html.
- J. J. Deely and D. V. Lindley. Bayes empirical Bayes. *Journal of the American Statistical Association*, 76(376):833–841, 1981. ISSN 01621459. URL <http://www.jstor.org/stable/2287578>.
- Morris H. DeGroot. *Optimal Statistical Decisions*. John Wiley & Sons, 1970.
- M. P. Deisenroth, C. E. Rasmussen, and J. Peters. Gaussian process dynamic programming. *Neurocomputing*, 72(7-9):1508–1524, 2009.
- Christos Dimitrakakis. *Ensembles for Sequence Learning*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2006a.
- Christos Dimitrakakis. Nearly optimal exploration-exploitation decision thresholds. In *Int. Conf. on Artificial Neural Networks (ICANN)*, 2006b.
- Christos Dimitrakakis. Tree exploration for Bayesian RL exploration. In *Computational Intelligence for Modelling, Control and Automation, International Conference on*, pages 1029–1034, Wien, Austria, 2008. IEEE Computer Society. ISBN 978-0-7695-3514-2. doi: <http://doi.ieeecomputersociety.org/10.1109/CIMCA.2008.32>.
- Christos Dimitrakakis. Bayesian variable order Markov models. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9 of *JMLR : W&CP*, pages 161–168, Chia Laguna Resort, Sardinia, Italy, 2010a.
- Christos Dimitrakakis. Complexity of stochastic branch and bound methods for belief tree search in Bayesian reinforcement learning. In *2nd international conference on agents and artificial intelligence (ICAART 2010)*, pages 259–264, Valencia, Spain, 2010b. ISNTICC, Springer.
- Christos Dimitrakakis. Robust bayesian reinforcement learning through tight lower bounds. In *European Workshop on Reinforcement Learning (EWRL 2011)*, number 7188 in LNCS, pages 177–188, 2011.
- Christos Dimitrakakis. Monte-carlo utility estimates for bayesian reinforcement learning. In *IEEE 52nd Annual Conference on Decision and Control (CDC 2013)*, 2013. arXiv:1303.2506.
- Christos Dimitrakakis and Michail G. Lagoudakis. Algorithms and bounds for rollout sampling approximate policy iteration. In *EWRL*, pages 27–40, 2008a.
- Christos Dimitrakakis and Michail G. Lagoudakis. Rollout sampling approximate policy iteration. *Machine Learning*, 72(3):157–171, September 2008b. doi: 10.1007/s10994-008-5069-3. Presented at ECML’08.

- Christos Dimitrakakis and Nikolaos Tziortziotis. ABC reinforcement learning. In *ICML 2013*, volume 28(3) of *JMLR W & CP*, pages 684–692, 2013. See also arXiv:1303.6977.
- Christos Dimitrakakis and Nikolaos Tziortziotis. Usable ABC reinforcement learning. In *NIPS 2014 Workshop: ABC in Montreal*, 2014.
- Christos Dimitrakakis, Yang Liu, David Parkes, and Goran Radanovic. Subjective fairness: Fairness is in the eye of the beholder. Technical Report 1706.00119, arXiv, 2017.
- Michael O’Gordon Duff. *Optimal Learning Computational Procedures for Bayes-adaptive Markov Decision Processes*. PhD thesis, University of Massachusetts at Amherst, 2002.
- Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 214–226. ACM, 2012.
- Yaakov Engel, Shie Mannor, and Ron Meir. Bayes meets bellman: The gaussian process approach to temporal difference learning. In *ICML 2003*, 2003.
- Yaakov Engel, Shie Mannor, and Ron Meir. Reinforcement learning with gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pages 201–208. ACM, 2005.
- Eyal Even-Dar and Yishai Mansour. Approximate equivalence of markov decision processes. In *Learning Theory and Kernel Machines. COLT/Kernel 2003*, Lecture notes in Computer science, pages 581–594, Washington, DC, USA, 2003. Springer.
- Milton Friedman and Leonard J. Savage. The expected-utility hypothesis and the measurability of utility. *The Journal of Political Economy*, 60(6):463, 1952.
- Thomas Furmston and David Barber. Variational methods for reinforcement learning. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9 of *JMLR : W&CP*, pages 241–248, Chia Laguna Resort, Sardinia, Italy, 2010.
- J. Geweke. Using simulation methods for Bayesian econometric models: inference, development, and communication. *Econometric Reviews*, 18(1):1–73, 1999.
- Mohammad Ghavamzadeh and Yaakov Engel. Bayesian policy gradient algorithms. In *NIPS 2006*, 2006.
- C. J. Gittins. *Multi-armed Bandit Allocation Indices*. John Wiley & Sons, New Jersey, US, 1989.
- Robert Grande, Thomas Walsh, and Jonathan How. Sample efficient reinforcement learning with gaussian processes. In *International Conference on Machine Learning*, pages 1332–1340, 2014.

- Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.
- M. Hutter. Feature reinforcement learning: Part I: Unstructured MDPs. *Journal of Artificial General Intelligence*, 1:3–24, 2009.
- Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.
- Tobias Jung and Peter Stone. Gaussian processes for sample-efficient reinforcement learning with RMAX-like exploration. In *ECML/PKDD 2010*, pages 601–616, 2010.
- Sham Kakade. A natural policy gradient. *Advances in neural information processing systems*, 2:1531–1538, 2002.
- Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An optimal finite time analysis. In *ALT-2012*, 2012.
- Michael Kearns and Satinder Singh. Finite sample convergence rates for Q-learning and indirect algorithms. In *Advances in Neural Information Processing Systems*, volume 11, pages 996–1002. The MIT Press, 1999.
- Niki Kilbertus, Mateo Rojas-Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. Avoiding discrimination through causal reasoning. Technical Report 1706.02744, arXiv, 2017.
- Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. Technical Report 1609.05807, arXiv, 2016.
- Levente Kocsis and Csaba Szepesvári. Bandit based Monte-Carlo planning. In *Proceedings of ECML-2006*, 2006.
- AN Kolmogorov and SV Fomin. *Elements of the theory of functions and functional analysis*. Dover Publications, 1999.
- M. G. Lagoudakis and R. Parr. Least-squares policy iteration. *The Journal of Machine Learning Research*, 4:1107–1149, 2003.
- Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Adv. in Appl. Math.*, 6:4–22, 1985.
- Nam M. Laird and Thomas A. Louis. Empirical Bayes confidence intervals based on bootstrap samples. *Journal of the American Statistical Association*, 82(399):739–750, 1987.
- Tor Lattimore and Marcus Hutter. Near-optimal PAC bounds for discounted MDPs. *Theor. Comput. Sci.*, 558:125–143, 2014.
- T. Lwin and J. S. Maritz. Empirical Bayes approach to multiparameter estimation: with special reference to multinomial distribution. *Annals of the Institute of Statistical Mathematics*, 41(1):81–99, 1989.

- Shie Mannor and John N. Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5: 623–648, 2004.
- J. M. Marin, P. Pudlo, C. P. Robert, and R. J. Ryder. Approximate Bayesian computational methods. *Statistics and Computing*, pages 1–14, 2011.
- Thomas P Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001a.
- Thomas P. Minka. Bayesian linear regression. Technical report, Microsoft research, 2001b.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidje land, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
- R. Munos and C. Szepesvári. Finite-time bounds for fitted value iteration. *The Journal of Machine Learning Research*, 9:815–857, 2008.
- Ronald Ortner, Daniil Ryabko, Peter Auer, and Rémi Munos. Regret bounds for restless Markov bandits. *Theor. Comput. Sci.*, 558:62–76, 2014. doi: 10.1016/j.tcs.2014.09.026. URL <http://dx.doi.org/10.1016/j.tcs.2014.09.026>.
- Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. In *NIPS*, 2013.
- Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances in Neural Information Processing Systems*, pages 4026–4034, 2016.
- Brammert Ottens, Christos Dimitrakakis, and Boi Faltings. DUCT: An upper confidence bound approach to distributed constraint optimization problems. In *AAAI 2012*, 2012.
- Jan Peters and Stefan Schaal. Policy gradient methods for robotics. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2219–2225. IEEE, 2006.
- P. Poupart, N. Vlassis, J. Hoey, and K. Regan. An analytic solution to discrete Bayesian reinforcement learning. In *ICML 2006*, pages 697–704. ACM Press New York, NY, USA, 2006.
- Pascal Poupart and Nikos Vlassis. Model-based Bayesian reinforcement learning in partially observable domains. In *International Symposium on Artificial Intelligence and Mathematics (ISAIM)*, 2008.
- Marting L. Puterman. *Markov Decision Processes : Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New Jersey, US, 1994.
- Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. ISBN 13 978-0-262-18253-9.

- H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.
- Herbert Robbins. An empirical Bayes approach to statistics. In Jerzy Neyman, editor, *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. University of California Press, Berkeley, CA, 1955.
- Herbert Robbins. The empirical Bayes approach to statistical decision problems. *The Annals of Mathematical Statistics*, 35(1):1–20, 1964.
- Stephane Ross, Brahim Chaib-draa, and Joelle Pineau. Bayes-adaptive POMDPs. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, Cambridge, MA, 2008. MIT Press.
- D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, et al., editors, *Parallel Distributed Processing: Volume 1: Foundations*, pages 318–362. MIT Press, Cambridge, 1987.
- Leonard J. Savage. *The Foundations of Statistics*. Dover Publications, 1972.
- Wolfram Schultz, Peter Dayan, and P. Read Montague. A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599, 1997. ISSN 0036-8075. doi: 10.1126/science.275.5306.1593. URL <http://science.sciencemag.org/content/275/5306/1593>.
- S. Singh, T. Jaakkola, and M. I. Jordan. Reinforcement learning with soft state aggregation. *Advances in neural information processing systems*, pages 361–368, 1995.
- M. T. J. Spaan and N. Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research*, 24:195–220, 2005.
- A. L. Strehl and M. L. Littman. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008. ISSN 0022-0000.
- Alexander L. Strehl and Michael L. Littman. A theoretical analysis of model-based interval estimation. In *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005)*, pages 857–864. ACM, 2005.
- Alexander L Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L Littman. Pac model-free reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 881–888. ACM, 2006.
- Malcolm Strens. A Bayesian framework for reinforcement learning. In *ICML 2000*, pages 943–950, 2000.
- Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

- Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. Policy gradient methods for reinforcement learning with function approximation. In *NIPS 99*, 1999.
- Ole Tange. Gnu parallel-the command-line power tool. *The USENIX Magazine*, 36(1):42–47, 2011.
- Ambuj Tewari and Peter Bartlett. Optimistic linear programming gives logarithmic regret for irreducible MDPs. In *Advances in Neural Information Processing Systems 20 (NIPS 2007)*, pages 1505–1512. MIT Press, 2008.
- W. R. Thompson. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of two Samples. *Biometrika*, 25(3-4):285–294, 1933.
- T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. P. H. Stumpf. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202, 2009.
- Marc Toussaint, Stefan Harmelign, and Amos Storkey. Probabilistic inference for solving (PO)MDPs, 2006.
- Paul Tseng. Solving h-horizon, stationary markov decision problems in time proportional to log (h). *Operations Research Letters*, 9(5):287–297, 1990.
- John N Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Machine learning*, 16(3):185–202, 1994.
- Nikolaos Tziortziotis and Christos Dimitrakakis. Bayesian inference for least squares temporal difference regularization. In *ECML*, 2017.
- Nikolaos Tziortziotis, Christos Dimitrakakis, and Konstantinos Blekas. Cover tree Bayesian reinforcement learning. *Journal of Machine Learning Research (JMLR)*, 2014.
- J. Veness, K. S. Ng, M. Hutter, and D. Silver. A Monte Carlo AIXI approximation. Arxiv preprint arXiv:0909.0801, 2009.
- Nikos Vlassis, Michael L. Littman, and David Barber. On the computational complexity of stochastic controller optimization in POMDPs. *TOCT*, 4(4):12, 2012.
- Tao Wang, Daniel Lizotte, Michael Bowling, and Dale Schuurmans. Bayesian sparse sampling for on-line reward optimization. In *ICML '05*, pages 956–963, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5. doi: <http://doi.acm.org/10.1145/1102351.1102472>.
- Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.
- T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and M. J. Weinberger. Inequalities for the L_1 deviation of the empirical distribution. *Hewlett-Packard Labs, Tech. Rep*, 2003.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Yinyu Ye. The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, 36(4):593–603, 2011.

Henry H Yin and Barbara J Knowlton. The role of the basal ganglia in habit formation. *Nature Reviews Neuroscience*, 7(6):464, 2006.