

ADA511: Data science and data-driven engineering

Steffen Mæland
PierGianLuca Porta Mana

2023-06-17

Table of contents

Preface	5
I An invitation	6
1 Accept or discard?	7
2 Framework	9
2.1 What does the intro problem tell us?	9
2.2 Our focus: decision-making, inference, and data science	12
2.3 Our goal: optimality, not “success”	14
2.4 Decision Theory	15
3 Basic decision problems	18
3.1 Graphical representation and elements	18
3.2 Inference, utility, maximization	21
II Inference	23
4 Inference	24
4.1 What is inference?	24
4.1.1 From truth to probability	24
4.1.2 Hypothetical and counterfactual infer- ences	24
4.1.3 Known and unknown: by whom?	25
4.1.4 Future and past	26
4.2 Mathematical representation of inferences and their objects	26
5 Sentences	27
5.1 Data, outcomes, agents, decisions	27
5.1.1 Notation	29
5.2 Combining sentences	29
5.2.1 Basic sentences	29
5.2.2 Connectives	31

5.3	Distinguishing assumptions and outcomes . . .	32
5.4	Inferences: certain and uncertain	33
6	Truth inference	35
6.1	Building blocks	35
6.1.1	Analysis of the problem	35
6.1.2	Data, assumptions, desired conclusions	36
6.2	Background information and conditional . . .	37
6.3	Truth-inference rules	37
6.4	Logical AI agents and their limitations	38
7	Probability inference	40
7.1	When truth isn't known: probability	40
7.2	No new building blocks	43
7.3	Probability-inference rules	43
7.4	How the inference rules are used	45
7.4.1	Derived rules	47
7.5	Law of total probability or "extension of the conversation"	48
7.6	Bayes's theorem	48
7.7	consequences of not following the rules, . . .	48
7.8	Common points of certain and uncertain infer- ence	48
8	Data and information	50
8.1	Kinds of data	50
8.1.1	Binary	50
8.1.2	Nominal	50
8.1.3	Ordinal	50
8.1.4	Continuous	50
8.1.5	Complex data	50
8.1.6	"Soft" data	50
8.2	Data transformations	50
9	Allocation of uncertainty among possible data values: probability distributions	51
9.1	The difference between Statistics and Proba- bility Theory	51
9.2	What's "distributed"?	51
9.3	Distributions of probability	51
9.3.1	Representations	51
9.4	Summaries of distributions of probability . .	52
9.4.1	Location	52
9.4.2	Dispersion or range	52

9.4.3	Resolution	52
9.4.4	Behaviour of summaries under trans- formations of data and errors in data .	52
9.5	Outliers and out-of-population data	52
9.6	Marginal and conditional distributions of probability	52
9.7	Collecting and sampling data	52
9.7.1	“Representative” samples	52
9.7.2	Unavoidable sampling biases	53
9.8	Quirks and warnings about high-dimensional data	53
10	Making decisions	54
10.1	Decisions, possible situations, and consequences	54
10.2	Gains and losses: utilities	54
10.2.1	Factors that enter utility quantification	54
10.3	Making decisions under uncertainty: maxi- mization of expected utility	54
11	The most general inference problem	55

Preface

Science is built up with facts, as a house is with stones. But a collection of facts is no more a science than a heap of stones is a house. (H. Poincaré)

****WARNING: THIS IS A WORKING DRAFT. TEXT WILL CHANGE A LOT. MANY PASSAGES ARE JUST TEMPORARY, INCOHERENT, AND DISJOINTED.**

To be written.

- Difference between car mechanic and automotive engineer
- “Engineering based on data” is just how engineering and science in general have been in the past 400 years or so. Nothing new there.
- The amount of available data has changed. This may lead to a reduction – or in some cases an increase – in uncertainty, and therefore to different solutions.
- Luckily the fundamental theory to deal with large amount of data is exactly the same to deal with small amounts. So the foundations haven’t changed.

This course makes you acquainted with the foundations.

Part I

An invitation

1 Accept or discard?

Let's start with a question that could arise in a particular engineering problem:

A particular kind of electronic component is produced on an assembly line. At the end of the line, there is an automated inspection device that works as follows with every newly produced component coming out of the line.

The inspection device first makes some tests on the new component. The tests give an uncertain forecast of whether that component will fail within its first year of use, or after.

Then the device decides whether the component is accepted and packaged for sale, or discarded and thrown away.

When a new electronic component is sold, the manufacturer has a net gain of 1\$. If the component fails within a year of use, however, the manufacturer incur net *loss* of 10\$ (11\$ loss, minus the 1\$ gained at first), owing to warranty refunds and damage costs to be paid to the buyer. When a new electronic component is discarded, the manufacturer has 0\$ net gain.

For a specific new electronic component, just come out of the assembly line, the tests of the automated inspection device indicate that there is a 10% probability that the component will fail within its first year of use.

Should the inspection device accept or discard the new component?

First, try to give and motivate an answer.



This is not the real question of this exercise, however. In fact it doesn't matter if you don't get the correct answer; not even if you don't manage to get an answer at all.

The purpose here is for you to do some introspection about your own reasoning. Then examine and discuss these points:

- Which numerical elements in the problem seem to affect the answer?
- Can these numerical elements be clearly separated? How would you separate them?
- How would the answer change, if these numerical elements were changed? Feel free to change them, also in extreme ways, and see how the answer would change.
- Could we solve the problem if we didn't have the probabilities? Why?
- Could we solve the problem if we didn't know the various gains and losses? Why?
- Can this problem be somehow abstracted, and then transformed into another one with completely different details? For instance, consider translating along these lines:
 - inspection device → computer pilot of self-driving car
 - tests → camera image
 - fail within a year → pedestrian in front of car
 - accept/discard → keep on going/ break

2 Framework

2.1 What does the intro problem tell us?

Let's approach the "accept or discard?" problem of the previous chapter 1 in an intuitive way.

First let's say that we **accept** the component. What happens?

We must try to make sense of that 10% probability that the component fails within a year. Different people do this with different imagination tricks. We can imagine, for instance, that this situation is repeated 100 times. In 10 of these repetitions the accepted electronic component is sold and fails within a year after selling. In the remaining 90 repetitions, the component is sold and works fine for at least a year.

In each of the 10 imaginary repetitions in which the component fails early, the manufacturer loses 10\$. That's a total loss of $10 \cdot 10\$ = 100\$$. In each of the 90 imaginary repetitions in which the component doesn't fail early, the manufacturer gains 1\$. That's a total gain of 90\$. So over all 100 imaginary repetitions the manufacturer gains

$$10 \cdot (-10\$) + 90 \cdot 1\$ = -10\$,$$

that is, the manufacturer has not gained, but *lost* 10\$! That's an average of 0.1\$ *lost* per repetition.

Now let's say that we **discard** the component instead. What happens? In this case we don't need to invoke imaginary repetitions, but even if we do, it's clear that the manufacturer doesn't gain or lose anything – that is, the "gain" is 0\$ – in each and all of the repetitions.

The conclusion is that if in a situation like this we accept the component, then we'll lose 1\$ on average; whereas if we discard it, then on average we won't lose anything or gain anything.

We're jumping the gun here,
because we haven't learned the
method to solve this problem yet!

Obviously the best, or “least worst”, decision to make is to **discard** the component.

Exercises

1. Now that we have an idea of the general reasoning, check what happens with different values of the probability of failure and of the failure cost: is it still best to discard? For instance, try with
 - failure probability 10% and failure cost 5\$;
 - failure probability 5% and failure cost 10\$;
 - failure probability 10%, failure cost 1\$, non-failure gain 2\$.

Feel free to get wild and do plots.

2. Identify the failure probability at which accepting the component doesn't lead to any loss or any gain, so it doesn't matter whether we discard or accept. (You can solve this as you prefer: analytically with an equation, visually with a plot, by trial & error on several cases, or whatnot.)
3. Consider the special case with failure probability 0% and failure cost 10\$. This means no new component will ever fail. To decide in such a case we do not need imaginary repetitions; but **confirm** that we arrive at the same logical conclusion whether we reason through imaginary repetitions or not.
4. Consider this completely different problem:

A patient is examined by a brand-new medical diagnostics AI system.

The AI first performs some clinical tests on the patient. The tests give an uncertain forecast of whether the patient has a particular disease or not.

Then the AI decides whether the patient should be dismissed without treatment, or treated with a particular medicine.

If the patient is dismissed, then the life expectancy doesn't increase or decrease

if the disease is not present, but it decreases by 9 years if the disease is actually present. If the patient is treated, then the life expectancy decreases by 1 year if the disease is not present (owing to treatment side-effects), and also if the disease is present (because it cures the disease, giving 10 more years to the patient, minus 1 year for the side effects).

For this patient, the clinical tests indicate that there is a 10% probability that the patient has the disease.

Should the diagnostic AI dismiss or treat the patient? Find differences and similarities, even numerical, with the assembly-line problem.

From the solution of the problem and from the exploring exercises, we gather some instructive points:

- Is it enough if we simply know that the component is less likely to fail than not? in other words, if we simply know that the probability of failure is less than 50%?

Obviously not. We found that if the failure probability is 10% then it's best to discard; but if it's 5% then it's best to accept. In both cases the component was less likely to fail than not, but the decisions were different. Moreover, we found that the probability affected the loss if one made the non-optimal decision. Therefore:

Knowledge of exact probabilities is absolutely necessary for making the best decision

- Is it enough if we simply know that failure leads to a cost? that is, that its gain is less than the gain for non-failure?

Obviously not. The situation is similar to that with the probability. In the exercise we found that if the failure cost is 10\$ then it's best to discard; but if it's 5\$ then

it's best to accept. It's also best to accept if the failure cost is 10\$ but the non-failure gain is 2\$. Therefore:

Knowledge of the exact gains and losses is absolutely necessary for making the best decision

- Is this kind of decision situation only relevant to assembly lines and sales?

By all means not. We found a clinical situation that's exactly analogous: there's uncertainty, there are gains and losses (of time rather than money), and the best decision depends on both.

2.2 Our focus: decision-making, inference, and data science

Every data-driven engineering project is unique, with its unique difficulties and problems. But there are also problems common to all engineering projects.

In the scenarios we explored above, we found an extremely important problem-pattern. There is a decision or choice to make (and “not deciding” is not an option – or it's just another kind choice). Making a particular decision will lead to some consequences, some leading to a desired goal, others leading to something undesirable. The decision is difficult because its consequences are not known with certainty, given the information and data available in the problem. We may lack information and data about past or present details, about future events and responses, and so on. This is what we call a problem of **decision-making under uncertainty** or **under risk**¹, or simply a “decision problem” for short.

This problem-pattern appears literally everywhere. But our explored scenarios also suggest that this problem-pattern has a sort of systematic solution method.

In this course we're going to focus on decision problems and their systematic solution method. We'll learn a framework and some abstract notions that allow us to frame and analyse this kind of problem, and we'll learn a universal set of

¹We'll avoid the word “risk” because it has several different technical meanings in the literature, some even contradictory.

principles to solve it. This set of principles goes under the name of **Decision Theory**.

But what do decision-making under uncertainty and Decision Theory have to do with *data* and *data science*? The three are profoundly and tightly related on many different planes:

- We saw that *probability* values are essential in a decision problem. How do we find them? As you can imagine, *data* play an important part in their calculation. In our intro example, the failure probability must come from observations or experiments on similar electronic components.
- We saw that also the values of *gains and losses* are essential. *Data* play an important part in their calculation as well.
- *Data science* is based on the laws of *Decision Theory*. Here's an analogy: a rocket engineer relies on fundamental physical laws (balance of momentum, energy, and so on) for making a rocket work. Failure to account for those laws leads at best to sub-optimal solutions, at worst to disasters. As we shall see, the same is true for a data scientist and the rules of decision theory.
- *Machine-learning* algorithms, in particular, are realizations or approximations of the rules of *Decision Theory*. This is clear, for instance, considering that the main task of a machine-learning classifier is to decide among possible output labels or classes.
- The rules of *Decision Theory* are also the foundations upon which *artificial-intelligence* agents, which must make optimal inferences and decisions, are built.

These five planes will constitute the major parts of the present course.

@@ TODO add examples: algorithm giving outputs is a decision agent.

There are other important aspects in engineering problems, besides the one of making decisions under uncertainty. For

☛ For the extra curious

Decision theory in expert systems and artificial intelligence

instance the *discovery* or the *invention* of new technologies and solutions. These aspects can barely be planned or decided; but their fruits, once available, should be handled and used optimally – thus leading to a decision problem.

Artificial intelligence is proving to be a valuable aid in these more creative aspects too. This kind of use of AI is outside the scope of the present notes. Some aspects of this creativity-assisting use, however, do fall within the domain of the present notes. A pattern-searching algorithm, for example, can be optimized by means of the method we are going to study.

2.3 Our goal: optimality, not “success”

What should we demand from a systematic method for solving decision problems?

By definition, in a decision problem under uncertainty there is generally no method to *determine* the decision that surely leads to the desired consequence – if such a method existed, then the problem would not have any uncertainty! Therefore, if there is a method to deal with decision problems, its goal cannot be the determination of the *successful* decision. This also means that a priori we cannot blame an engineer for making an unsuccessful decision in a situation of uncertainty.

Imagine two persons, Henry and Tina, who must bet on “heads” or “tails” under the following conditions (but who otherwise don’t get any special thrill from betting):

- If the bet is “heads” and the coin lands “heads”, the person wins a *small* amount of money; but if it lands “tails”, they lose a *large* amount of money.
- If the bet is “tails” and the coin lands “tails”, the person *wins* a small amount of money; if it lands “heads”, they lose the same *small* amount of money.

Henry chooses the first bet, on “heads”. Tina chooses the second bet, on “tails”. The coin comes down “heads”. So Henry wins the small amount of money, while Tina loses the same small amount. What would we say about their decisions?

Henry’s decision was lucky, and yet *irrational*: he risked losing much more money than in the second bet, without any

possibility of at least winning more. Tina's decision was unlucky, and yet *rational*: the possibility and amount of winning was the same in the two bets, and she chose the bet with the least amount of loss. We expect that any person making Henry's decision in similar, future bets will eventually lose more money than any person making Tina's decision.

This example shows two points. First, "success" is generally not a good criterion to judge a decision under uncertainty; success can be the pure outcome of luck, not of smarts. Second, even if there is no method to determine which decision is successful, there is a method to determine which decision is rational or **optimal**, given the particular gains, losses, and uncertainties involved in the decision problem. We had a glimpse of this method in our introductory scenarios.

Let us emphasize, however, that we are not giving up on "success", or trading it for "optimality". Indeed we'll find that **Decision Theory automatically leads to the *successful* decision** in problems where uncertainty is not present or is irrelevant. It's a win-win. It's important to keep this point in mind:

Aiming to find the solutions that are *successful* can make us *fail* to find those that are optimal when the successful ones cannot be determined.
Aiming to find the solutions that are *optimal* makes us automatically find those that are *successful* when those can be determined.

We shall later witness this fact with our own eyes, and will take it up again in the discussion of some misleading techniques to evaluate machine-learning algorithms.

2.4 Decision Theory

So far we have mentioned that Decision Theory has the following features:

- ✓ it tells us what's optimal and, when possible, what's successful
- ✓ it takes into consideration decisions, consequences, costs and gains

- ✓ it is able to deal with uncertainties

What other kinds of features should we demand from it, in order to be applied to as many kinds of decision problems as possible, and to be relevant for data science?

If we find an optimal decision in regards to some outcome, it may still happen that the decision can be realized in several ways that are equivalent in regard to the outcome, but inequivalent in regard to time or resources. In the assembly-line scenario, for example, the decision **discard** could be carried out by burning, recycling, and so on. We thus face a decision within a decision. In general, a decision problem may involve several decision sub-problems, in turn involving decision sub-sub-problems, and so on.

In data science, a common engineering goal is to design and build an automated or AI-based device capable of making an optimal decision in a specific kind of uncertain situations. Think for instance of an aeronautic engineer designing an autopilot system, or a software company designing an image classifier.

Decision Theory turns out to meet these demands too, thanks to the following features:

- ✓ it is susceptible to recursive, sequential, and modular application
- ✓ it can be used not only for human decision-makers, but also for automated or AI devices

Decision Theory has a long history, going back to Leibniz in the 1600s and partly even to Aristotle in the –300s, and appearing in its present form around 1920–1960. What’s remarkable about it is that it is not only *a* framework, but *the* framework we must use. A logico-mathematical theorem shows that **any framework that does not break basic optimality and rationality criteria has to be equivalent to Decision Theory**. In other words, any “alternative” framework may use different technical terminology and rewrite mathematical operations in a different way, but it boils down to the same notions and operations of Decision

Theory. So if you wanted to invent and use another framework, then either (a) it would lead to some irrational or illogical consequences, or (b) it would lead to results identical to Decision Theory's. Many frameworks that you are probably familiar with, such as optimization theory or Boolean logic, are just specific applications or particular cases of Decision Theory.

Thus we list one more important characteristic of Decision Theory:

- ✓ it is **normative**

Normative contrasts with *descriptive*. The purpose of Decision Theory is not to describe, for example, how human decision-makers typically make decisions. Because human decision-makers typically make irrational, sub-optimal, or biased decisions. That's exactly what we want to avoid and improve!

🔗 For the extra curious

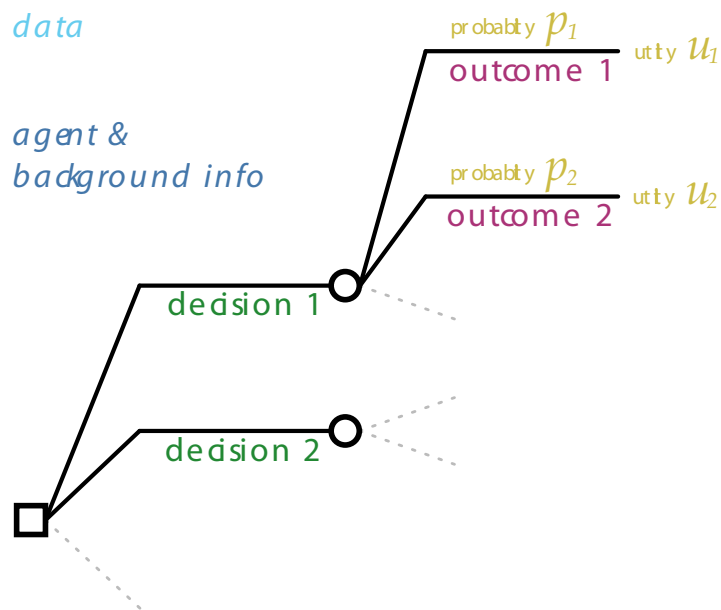
- *Judgment under uncertainty*
- *Heuristics and Biases*
- *Thinking, Fast and Slow*

3 Basic decision problems

Decision Theory analyses any decision-making problem in terms of nested or sequential *basic* or *minimal* decision problems. The assembly-line scenario of the introduction 1 is an example.

3.1 Graphical representation and elements

A basic decision problem can be represented by a diagram like this:



It has one *decision node*, usually represented by a square ■, from which the available decisions depart as lines. Each decision leads to an *uncertainty node*, usually represented by a circle ●, from which the possible outcomes depart as lines. Each outcome leads to a particular utility value. The uncertainty of each outcome is quantified by a probability.

A basic decision problem is analysed in terms of these elements:

- **Agent**, and **background** or **prior information**. The agent is the person or device that has to make the decision. An agent always possess (or has been programmed with) specific background information that is used and taken for granted in the decision-making process. This background information determines the probabilities and utilities of the outcomes, together with other available data and information. Since different agents typically have different background information, we shall somehow conflate agents and prior information.
- **Decisions**, also called **courses of actions**, available to the agent. They are assumed to be mutually exclusive and exhaustive; this can always be achieved by recombining them if necessary, as we'll discuss later.
- **Outcomes** of the possible decisions. Every decision can have a different set of outcomes, or some outcomes can appear for several or all decisions (in this case they are reported multiple times in the decision diagram). Note that even if an outcome can happen for two or more different decisions, its probabilities can still be different depending on the decision.
- **Probabilities** for each of the outcomes. Their values typically depend on the background information, the decision, and the additional data.
- **Utilities**: the gains or losses associated with each of the possible outcomes. Their values also depend on the background information, the decision, and the additional data.
- **Data** and other **additional information**, sometimes called **evidence**. They differ from the background information in that they can change with every decision instance made by the same agent, while the background information stays the same. In the assembly-line scenario, for example, the test results could be different for every new electric component.

We'll use the neutral pronouns *it/its* when referring to an agent, since an agent could be a person or a machine.

Note that it is not always the case that the *outcomes* are unknown and the *data* are known. As we'll discuss later, in some situations we reason in hypothetical or counterfactual ways, using hypothetical data and considering outcomes which have already occurred.

Reading

§ 1.1.4 in *Artificial Intelligence*

Exercise

- Identify the element above in the assembly-line decision problem of the introduction 1.
- Sketch the diagram of the assembly-line decision problem.


Some of the decision-problem elements listed above may themselves need to be analysed by a decision sub-problem. For instance, the utilities could depend on uncertain factors: thus we have a decision sub-problem to determine the optimal values to be used for the utilities of the main problem. This is an example of the modular character of decision theory.

We shall soon see how to mathematically represent these elements.

The elements above must be identified unambiguously in every decision problem. The analysis into these elements greatly helps in making the problem and its solution well-defined.

An advantage of decision theory is that its application *forces* us to make sense of an engineering problem. A useful procedure is to formulate the general problem in terms of the elements above, identifying them clearly. If the definition of any of the terms involves uncertainty of further decisions, then we analyse it in turn as a decision sub-problem, and so on.

Suppose someone (probably a politician) says:
“We must solve the energy crisis by reducing energy consumption or producing more energy”.
From a decision-making point of view, this

 Remember: What matters is to be able to identify these elements in a concrete problem, understanding their role. Their technical names don't matter.

person has effectively said *nothing whatsoever*. By definition the “energy crisis” is the problem that energy production doesn’t meet demand. So this person has only said “we would like the problem to be solved”, without specifying any solution. A decision-theory approach to this problem requires us to specify which concrete courses of action should be taken for reducing consumption or increasing productions, and what their probable outcomes, costs, and gains would be.

3.2 Inference, utility, maximization

The solution of a basic decision-making problem can be roughly divided into three main stages: inference, utility assessment, and expected-utility maximization.

📦 Inference is the stage where the probabilities of the possible outcomes are calculated. Its rules are given by the **Probability Calculus**. Inference is independent from decision: in some situations we may simply wish to assess whether some hypotheses, conjectures, or outcomes are more or less plausible than others, without making any decision. This kind of assessment can be very important in problems of communication and storage, and it is specially considered by **Information Theory**.


The calculation of probabilities can be the part that demands most thinking, time, and computational resources in an inference or decision problem. It is also the part that typically makes most use of data – and where data can be most easily misused. Roughly half of this course will be devoted in understanding the laws of inference, their applications, uses, and misuses.

📦 Utility assesment is the stage where the gains or losses of the possible outcomes are calculated. Often this stage requires further inferences and further decision-making sub-problems. The theory underlying utility assessment

🔗 For the extra curious

See MacKay’s options-vs-costs rational analysis in [Sustainable Energy – without the hot air](#)

is still much underdeveloped, compared to probability theory.

 **Expected-utility maximization** is the final stage where the probabilities and gains or costs of the possible outcomes are combined, in order to determine the optimal decision.

Part II

Inference

4 Inference

We now put momentarily aside the full decision-making problem (but keeping it in the back of our minds), and focus on the problem of **inference**. Drawing inferences is very often a goal or need in itself, with no underlying decision process.

4.1 What is inference?

A first meaning of *drawing an inference* could be to assess whether an outcome or hypothesis is **true** or **false**, on the basis of some data and other information. This definition is too limited, however, especially in problems of engineering and data science; we need to make it more general in two respects. Some general aspects of inference should also be emphasized.

4.1.1 From truth to probability

First, it is often impossible to determine the truth or falsity of a hypothesis with certainty. This is why decision-making *under uncertainty* was our starting point. So by *drawing an inference* we mean **to assess the plausibility** or **credibility** or **probability** rather than the truth or falsity. Note that truth-determination is still included as a special case as **fully certain** and **impossible** cases of plausibility.

4.1.2 Hypothetical and counterfactual inferences

Second, it is not always the case that we want to assess the truth or plausibility of something *unknown* on the basis of something *known*. An agent must often make **hypothetical** or **counterfactual** reasoning.

Consider for instance the assembly-line scenario. To assess the outcome of accepting an electronic component, we momentarily imagine that the component is actually accepted and sold, and then find out what the consequences of this imagined situation would be. This is an example of *hypothetical reasoning*. Hypothetical reasoning is sometimes assisted by real experiments performed in restricted and controlled conditions. Such supporting experiments are not always viable, however, and often they do not exactly reflect the hypothetical situation.

Consider also the case where we know that a particular electronic component, already sold, has failed within a year. We can assess what would have been the plausibility of this failing we *had employed* a different kind of transistors in its construction. From this assessment we may find out whether it would be better to make changes in the assembly line. This is an example of *counterfactual reasoning*.

In either case we are determining the plausibility of something known given something unknown, or a combination of the two. Both these kinds of reasoning are staples of scientific research. We will see that they continuously enter in the calculation of probabilities.

4.1.3 Known and unknown: by whom?

What's known to an agent can be unknown to another, and vice versa. This means that **every inference is always relative to an agent's background knowledge**. This fact is especially important in data science and artificial intelligence:

Consider an engineer testing the programming of a particular AI system. The engineer inputs specially-chosen data and checks the AI's decisions about specially-chosen outcomes. In this scenario we have two ongoing inferences, drawn by two agents:

- The AI is one agent. This agent knows the inputs, but doesn't know the true outcomes. This agent's inference is about the outcomes.
- The engineer is the other agent. This agent knows inputs and true outcomes – both are “data” for this agent

– but doesn’t know the AI’s final decisions. This agent’s inference is about the AI’s decisions.

4.1.4 Future and past

An inference may concern not only unknown future outcomes – in which case we’ll often call it a *forecast* – but also unknown *past* outcomes. For instance, imagine that a particular device has stopped working, and an engineer needs to assess the plausibility of a mechanical failure or of an electronic failure. In this case the inference is about something unknown to the engineer (and maybe to anyone else), but that has already happened. Astrophysicists and archaeologists draw this kind of inferences all the time.

4.2 Mathematical representation of inferences and their objects

We shall soon explore the rules that govern inference. But first we must address two questions:

- Inference *about what?* and *given what?* So far we’ve spoken about “outcomes”, “data”, “hypotheses”, “background knowledge”, and similar concepts. They are not identical concepts; can we view them all as instances of a more unified notion?
- How to mathematically *represent* this unified notion and those concepts? A mathematical representation is necessary for systematically approaching inference and decision problems of any complexity, and also for implementation in computer algorithms and artificial-intelligence agents.

5 Sentences

5.1 Data, outcomes, agents, decisions

Is there a flexible and general way of representing data, outcomes, background information, hypotheses, assumptions, and even decisions?

When speaking of “data”, what comes to mind to many people is basically numbers or collections of numbers. So numbers could perhaps be used to representing all the concepts above. This option turns out to be too restricted, however.

I give you this number: 8, saying that it is “data”. But what is it about? You, as an agent, can hardly call this number a piece of information, because you have no clue what to do with it. Instead, if I tell you: “[The number of official planets in the solar system is 8](#)”, then we can say that I’ve given you data. So “data” is not just numbers: a number is not “data” unless there’s some verbal, non-numeric context accompanying it – even if this context is understood only implicitly. Sure, we could represent this meta-data information as numbers too; but this move would only shift the problem one level up: we would need auxiliary verbal context explaining what the meta-data numbers are about.

Data can, moreover, be completely non-numeric. A clinician saying “The patient has fully recovered from the disease” (we imagine to know who’s the patient and what was the disease) is giving us a piece of information that we could further use, for instance, to make prognoses about other, similar patients. The clinician’s statement surely is “data”, but essentially non-numeric data. In some situations we can represent it as “1”, while “0” would represent “not recovered”. Yet the opposite convention could also be used, showing that these numbers have intrinsically nothing to do with the clinician’s data.

But the examples above actually reveal the answer to our needs. In the examples we expressed the data by means of

sentences. Clearly any piece of information, outcome, hypothesis, decision can be expressed by a sentence. We shall therefore use **sentences**, also called **propositions** or **statements**,¹ to represent and communicate background information, data, outcomes, hypotheses, knowns, unknowns, and any other things on which or from which inferences can be drawn. In some cases we can of course summarize a sentence by a number, as a shorthand, when the full meaning of the sentence is understood.

But what is a sentence, more exactly? The everyday meaning of this word will work for us, even though there is still a lot of research in logic and artificial intelligence on how to define and use sentences. We shall adopt this useful definition:

A “sentence” is a verbal message for which we can determine whether it is **true** or **false**, at least in principle and in such a way that all interested receivers of the message would agree.

For the extra curious
[Propositions](#)

For instance, in most engineering contexts the phrase “This valve will operate for at least two months” is a sentence; whereas the phrase “Apples are much tastier than pears” is not, because it’s a matter of personal taste – there’s no objective criterion to determine its truth or falsity. However, the phrase “Rita finds apples tastier than pears” could be a sentence (its truth is found by asking Rita).

Note that a sentence can contain numbers, and even *pictures* and *graphs*: this possibility is not excluded from the definition above.

The use of sentences in our framework has important practical consequences:

- **Clarity, analysis, goal-orientation.** A data engineer must acquire information and convey information. “Acquiring information” does not consist simply in making measurements or counting something: the engineer must understand *what* is being measured and *why*. If data is gathered from third parties, the engineer must ask what exactly the data mean and how they were

¹These three terms are not always equivalent in formal logic, but here we’ll use them as synonyms.

acquired. In designing and engineering a solution, it is important to understand what information or outcomes the end user exactly wants. A data engineer will often ask “wait, what do you mean by that?”. This question is not just an unofficial parenthesis in the official data-transfer workflow between the engineer and someone else. It is an integral part of that workflow: it means that the data has not been completely transferred yet.

- **Artificial Intelligence.** Sentences are the central components of knowledge representation and inference in artificial-intelligence agents.

Reading

§ 7.1 in *Artificial Intelligence*.

5.1.1 Notation

In these notes we’ll denote sentences by sans-serif italic letters: A, B, a, b, \dots For example,

$$O := \text{‘The power output is 100 W’}$$

means that the symbol O stands for the sentence above. Often we shall simply write sentences in abbreviated form, when their full meaning is understood from the context. For instance, we could abbreviate the sentence above to “ $O = 100 \text{ W}$ ”, where O is a physical variable denoting the power output; or even just to “100 W”.

We’ll next see how more complex sentences are built from simpler ones. No matter whether complex or simple, any sentence can be represented by symbols like the ones above.

5.2 Combining sentences

5.2.1 Basic sentences

In analysing the assumptions and outcomes of a decision problem it is convenient to find a collection of **basic sentences**²

²A more technical term is *atomic sentences*.

out of which all other sentences of interest can be constructed. Often these basic sentences represent elementary pieces of information in the problem.

Consider for instance the following sentence which could appear in our earlier assembly-line problem:

“The electronic component is still whole after the shock test and the subsequent heating test. The voltage reported in the power test is either 90 mV or 110 mV.”

In our example the statement above represents data, that is, it’s the description of a factual situation. But keep in mind that in a different problem – say, one where it need to be assessed whether the component is still whole – the same statement could represent a hypothesis, that is, one possible state of affairs among other possible ones.

In the statement above we can identify at least four basic sentences, which we denote by convenient symbols:

s := ‘The component is whole after the shock test’
 h := ‘The component is whole after the heating test’
 v_{90} := ‘The power-test voltage reading is 90 mV’
 v_{110} := ‘The power-test voltage reading is 110 mV’

The decision problem may actually require more basic sentences than just these. For instance, it might become necessary to consider basic sentences with other values for the reported voltage, such as

v_{110} := ‘The power-test voltage reading is 100 mV’
 v_{80} := ‘The power-test voltage reading is 80 mV’

and so on.

There may also be other basic sentences coming from data or hypotheses that aren’t stated explicitly because obvious. Yet they may have to be spelled out. An example in our decision problem is the sentence

‘The electronic component cannot be broken after the shock test and whole after the subsequent heating test’

which must necessarily be true for physical reasons.

5.2.2 Connectives

How do we construct the initial data sentence, and more complex sentences out of simpler ones in general?

We consider one way or operation to change a sentence into another related to it, and two ways or operations to combine two or more sentences together. These operations are called **connectives**. Our natural language offer many more operations to combine sentences, but these three turn out to be all we need in virtually all engineering problems. The three connectives and their symbols are:

Not: \neg for example,

$\neg s =$ 'The component is broken after the shock test'

And: \wedge for example,

$s \wedge h =$ 'The component is whole after the shock and heating tests'

Or: \vee for example,

$v_{90} \vee v_{110} =$ 'The power-test voltage reading is 90 mV, or 110 mV, or both'

The connectives can be applied multiple times, to form increasingly complex sentences.

❗ Note some important subtleties of the connectives:

- There is no strict correspondence between the words “not”, “and”, “or” in natural language and the three connectives. For instance the **and** connective could correspond to the words “but” or “whereas”, or just to a comma “,”.
- **Not** doesn't mean some kind of complementary quality, but only the negation. For instance, \neg 'The chair is black' does not mean 'The chair is white', although in some situations these two sentences could amount to the same thing.

- It's best to always declare explicitly what the **not** of a sentence concretely means. In our example, we take \neg 'The component is whole' to mean 'The component is broken'. But in other examples the negation of "being whole" could comprise several different conditions. A good guideline is to always state the **not** of a sentence in *positive* terms.
- **Or** does not exclude that both the sentences it connects can be true. So in our example $v_{90} \vee v_{110}$ does not exclude, a priori, that the reported voltage could be both 90 mV and 110 mV. (There is a connective for that: "exclusive-or", but it can be constructed out of the three we already have.)

From the last remark we see that the sentence

'The power-test voltage reading is 90 mV or 110 mV'

does *not* correspond to $v_{90} \vee v_{110}$. It is implicitly clear that a voltage reading cannot yield two different values at the same time. Convince yourself that the correct way to write that sentence is this:

$$(v_{90} \vee v_{110}) \wedge \neg(v_{90} \wedge v_{110})$$

Finally, the full data statement of our present example can be written in symbols as follows:

$$s \wedge h \wedge (v_{90} \vee v_{110}) \wedge \neg(v_{90} \wedge v_{110})$$

5.3 Distinguishing assumptions and outcomes

Now we know how to represent arbitrarily complex sentences and express them in symbols. Let's introduce a way to clearly distinguish sentences that constitute the *assumptions* from those that constitute the *outcome* in a basic decision problem. In our assembly-line example, suppose that assumptions are the statement discussed in the previous section. The outcome to be inferred is expressed by this sentence:

$$f := \text{'The component will fail within a year'}$$

To distinguish assumptions and outcomes we can simply use a vertical bar,³ like “ \mid ” :

- on its *left* side we write the sentence representing the *outcome*,
- on its *right* side we write the sentences that make up our *assumptions*, **and**-ed together:

$$outcome \mid assumptions$$

So in our example we write:

$$f \mid s \wedge h \wedge (v_{90} \vee v_{110}) \wedge \neg(v_{90} \wedge v_{110})$$

The collection of assumptions on the right side of the bar “ \mid ” is called the **conditional**. The expression above is read

“ $c_{2,90}$ *given* $t_{1,50} \wedge t_{2,50} \wedge \dots$ ”

or

“ $c_{2,90}$ *conditional on* $t_{1,50} \wedge t_{2,50} \wedge \dots$ ”.

We are now equipped with all the notions and symbolic notation to deal with our first concrete goal: drawing uncertain inferences.

5.4 Inferences: certain and uncertain

In a decision problem we have a set of different outcomes that we want to assess given the same assumptions:

$$outcome\ 1 \mid assumptions$$

$$outcome\ 2 \mid assumptions$$

$$outcome\ 3 \mid assumptions$$

...

The first goal in a decision problem is to assess these outcomes. What do we mean by “assess”? We cannot demand

³Notation in formal logic uses the symbols “ \models ” or “ \vdash ”, and writes assumptions on the *left*, outcomes on the *right*. We use the notation used in probability logic.

that the truth or falsity of the outcomes be determined with certainty.

6 Truth inference

6.1 Building blocks

Consider the following trivial problem. An inspector examines an electronic component out of a production line. The information available to the inspector is the following:

- The component can either come from the production line in Oslo, or from the one in Rome.
- If the component is defective, it cannot come from Oslo.
- The component is found to be defective.

The question is: from which production line does the component come from?

The answer is obvious: from the Rome line. But how could we draw this obvious and sure inference? Which rules did we follow? Did we make any hidden assumptions, or use information that wasn't explicitly mentioned?

Logic is the huge field that formalizes and makes rigorous the rules that a rational person or an artificial intelligence should use in drawing sure inferences. We'll get a glimpse of it here, as a trampoline for jumping towards the more general inferences that we need in data-driven engineering problems.

6.1.1 Analysis of the problem

Let's write down the basic sentences that constitute our data and the inferences we want to draw. We identify three basic sentences, which we can represent by these symbols:

- o := 'The component comes from the Oslo line'
- r := 'The component comes from the Rome line'

- $d :=$ ‘The component is defective’

Obviously the inspector possesses even more information which is implicitly understood. It’s clear, for instance, that the component cannot come from both Oslo and Rome. Let’s denote this information with

- $I :=$ (a long collection of sentences explaining all other implicitly understood information).

With the sentences above we can express more complex details and hypotheses appearing in the inspector’s problem, in particular:

- $o \vee r =$ ‘The component comes from either the Oslo line or the Rome line’
- $\neg(o \wedge r) =$ ‘The component cannot come from both the Oslo and the Rome lines’
- $\$ \neg o$ ‘The component does not come from the Oslo line’ $\$$

6.1.2 Data, assumptions, desired conclusions

The inspector knows for certain the following facts:

- $o \vee r$, ‘The component comes from either the Oslo line or the Rome line’
- $\neg(o \wedge r)$, ‘The component cannot come from both the Oslo and the Rome lines’
- d , ‘The component is defective’
- I , all remaining implicit information

We **and** them all together:

$$d \wedge (o \vee r) \wedge \neg(o \wedge r) \wedge I .$$

The inspector knows, moreover, this hypothetical consequence:

- $\neg o | d \wedge (o \vee r) \wedge \neg(o \wedge r) \wedge I$, if the component is defective, it cannot come from the Oslo production line.
-

6.2 Background information and conditional

6.3 Truth-inference rules

Deduction systems in formal logic give us a set of rules for making correct inferences, that is, for correctly determining whether the conclusions of interest are true or false. These rules are represented in a [wide variety of ways](#), as steps leading from one conclusion to another one. The picture here on the margin, for instance, shows how a proof of our inference would look like, using the so-called sequent calculus.

$$\frac{\frac{D \wedge \neg b \vdash \neg r}{D \vdash b \vee \neg r} \quad \frac{D \wedge r \vdash b}{D \wedge D \vdash b \vee b}}{D \wedge D \vdash b} \quad \frac{}{D \vdash b}$$

Figure 6.1: The bottom formula is our conclusion; the formulae above it represent steps in the proof. Each line denotes the application of an inference rule. The two formulae with no line above are our two assumptions.

We can compactly encode all inference rules in the following way. First, represent **true** by the number 1, and **false** by 0. Second, symbolically write that conclusion C is **true**, given assumptions A , as follows:

$$T(C \mid A) = 1 .$$

or with 0 if it's **false**.

The rules of truth inference are then encoded by the following equations, which must always hold for any sentences A, B, C , no matter whether they are basic or complex:

Rule for “not”:

$$T(\neg A \mid B) + T(A \mid B) = 1 \quad (6.1)$$

Rule for “and”:

$$T(A \wedge B \mid C) = T(A \mid B \wedge C) \cdot T(B \mid C) = T(B \mid A \wedge C) \cdot T(A \mid C) \quad (6.2)$$

Rule for “or”:

$$T(A \vee B \mid C) = T(A \mid C) + T(B \mid C) - T(A \wedge B \mid C) \quad (6.3)$$

Rule of self-consistency:

$$T(A \mid A \wedge C) = 1 \quad (6.4)$$

Let's see how the inference rule (**?@eq-example-rule**), for example, is encoded in these equations. The rule starts with saying that $a \wedge b$ is **true** according to D . This means that $T(a \wedge b \mid D) = 1$. But, by rule (6.2), we must then have $T(b \mid a \wedge D) \cdot T(a \mid D) = 1$. This can only happen if both $T(b \mid a \wedge D)$ and $T(a \mid D)$ are equal to 1. So we can conclude that $T(a \mid D) = 1$, which is exactly the conclusion under the line in rule (**?@eq-example-rule**).

Exercise

Try to prove our initial inference

$$\frac{(b \vee r) \wedge \neg(b \wedge r) \mid D \quad \neg r \mid D}{b \mid D}$$

using the basic rules (6.1, 6.2, 6.3, 6.4). Remember that you can use each rule as many times as you like, and that there is not only one way of constructing a proof.

6.4 Logical AI agents and their limitations

The basic rules above are also the rules that a logical artificial-intelligent agent should follow.

Reading

[Ch. 7 in *Artificial Intelligence*](#)

Many – if not most – inference problems that a data engineer must face are, however, of the *uncertain* kind: it is not possible to surely infer the truth of some data, and the truth of some initial data may not be known either. In the next chapter we shall see how to generalize the logic rules to uncertain situations.

For the extra curious

Our cursory visit of formal logic only showed a microscopic part of this vast field. The study of logic rules

continues still today, with many exciting developments and applications. Feel free take a look at *Logic in Computer Science*, *Mathematical Logic for Computer Science*, *Natural Deduction Systems in Logic*

7 Probability inference

7.1 When truth isn't known: probability

In most real-life and engineering situations we don't know the truth or falsity of sentences and hypotheses that interest us. But this doesn't mean that nothing can be said or done in such situations.

When we cross a busy city street we look left and right to check whether any cars are approaching. We typically don't look up to check whether something is falling from the sky. Yet, couldn't it be **false** that cars are approaching? and couldn't it be **true** that *some object is falling from the sky*? Of course both events are possible. Then why do we look left and right, but not up?

The main reason¹ is that we *believe strongly* that cars might be approaching, *believe very weakly* that some object might be falling from the sky. In other words, we consider the first occurrence to be very *probable*; the second, extremely improbable.

We shall take the notion of **probability** as intuitively understood (just as we did with the notion of truth). Terms equivalent for “probability” are *degree of belief*, *plausibility*, *credibility*.

❗ In technical discourse, *likelihood* means something different and is *not* a synonym of “probability”, as we'll explain later.

Probabilities are quantified between 0 and 1, or equivalently between 0% and 100%. Assigning to a sentence a probability 1 is the same as saying that it is **true**; and a probability

¹We shall see later that one more factor enters the explanation.

0, that it is **false**. A probability of 0.5 represents a belief completely symmetric with respect to truth and falsity.

It is important to emphasize and agree on some facts about probabilities:

- **Probabilities are assigned to *sentences*.** Consider an engineer working on a problem of electric-power distribution in a specific geographical region. At a given moment the engineer may believe with 75% probability that the measured average power output in the next hour will be 100 MW. The 75% probability is assigned not to the quantity “100 MW”, but to the *sentence*

‘The measured average power output in the next hour will be 100 MW’

This difference is extremely important. Consider the alternative sentence

‘The average power output in the next hour will be *set* to 100 MW’

the quantity is the same, but the meaning is very different. The probability can therefore be very different (if the engineer is the person deciding the output, the probability is 100%). The probability depends not only on a number, but on what it’s being done with that number – measuring, setting, third-party reporting, and so on. Often we still write simply ‘ $O = 100\text{ W}$ ’ or even just ‘100 W’, provided that the full sentence behind the shorthand is understood.

- **Probabilities are agent- and context-dependent.** A coin is tossed, comes down heads, and is quickly hidden from view. Alice sees that it landed heads-up. Bob instead doesn’t manage to see the outcome and has no clue. Alice considers the sentence ‘Coin came down heads’ to be **true**, that is, to have 100% probability. Bob considers the same sentence to have 50% probability.

Note how Alice and Bob assign two different probabilities to the same sentence; yet both assignments are completely rational. If Bob assigned 100% to ‘heads’, we would suspect that he had seen the outcome after all; if he assigned 0% to ‘heads’, we would consider that groundless and silly. We would be baffled if Alice assigned 50% to ‘heads’, because she saw the outcome was

actually heads; we would hypothesize that she feels unsure about what she saw.

An omniscient agent would know the truth or falsity of every sentence, and assign only probabilities 0 or 1. Some authors speak of “*actual* (but unknown) probabilities”; if there were “actual” probabilities, they would be all 0 or 1, and it would be pointless to speak about probabilities at all – every inference would be a truth inference.

- **Probabilities are not frequencies.** The fraction of defective mechanical components to total components produced per year in some factory is a quantity that can be physically measured and would be agreed upon by every agent. It is a *frequency*, not a degree of belief or probability. It is important to understand the difference between them, to avoid making sub-optimal decisions; we shall say more about this difference later. Frequencies can be unknown to some agents, probabilities cannot be unknown (but can be difficult to calculate). Be careful when you read authors speaking of an “unknown probability”; either they actually mean “unknown frequency”, or a probability that has to be calculated (it’s “unknown” in the same sense that the value of $1 - 0.7 \cdot 0.2 / (1 - 0.3)$ is unknown to you right now).
- **Probabilities are not physical properties.** Whether a tossed coin lands heads up or tails up is fully determined by the initial conditions (position, orientation, momentum, rotational momentum) of the toss and the boundary conditions (air velocity and pressure) during the flight. The same is true for all macroscopic engineering phenomena (even quantum phenomena have never been proved to be non-deterministic, and there are [deterministic and experimentally consistent](#) mathematical representations of quantum theory). So we cannot measure a probability using some physical apparatus; and the mechanisms underlying any engineering problem boil down to physical laws, not to probabilities.

Reading

Dynamical Bias in the Coin Toss

These facts are not just a matter of principle. They have important practical consequences. A data engineer who is not attentive to the source of the data (measured? set? reported, and so maybe less trustworthy?), or who does not carefully assess the context of a probability, or who mixes it up with something else, or who does not take advantage (when possible) of the physics involved in the engineering problem, will design a system with sub-optimal performance² – or even cause deaths.

7.2 No new building blocks

In discussing [truth-inference](#) we introduced notations such as $T(a \mid b \wedge D)$, which stands for the truth-value 0 or 1 of sentence a in the context of data D and supposing (even if only hypothetically) sentence b to be true. We can simply extend this notation to probability-values, using a P instead of T :

$$P(a \mid b \wedge D) \in [0, 1]$$

represents the probability or degree of belief in sentence a in the context of data D and supposing also sentence b to be true. Keep in mind that both a and b could be complex sentences (for instance $a = (\neg c \vee d) \wedge e$). Note that truth-values are included as the special cases 1 or 0:

$$P(a \mid b \wedge D) = 0 \text{ or } 1 \iff T(a \mid b \wedge D) = 0 \text{ or } 1$$

7.3 Probability-inference rules

Extending our truth-inference notation to probability-inference notation has been straightforward. But how do we draw inferences when probabilities are involved?

Consider the inference about my umbrella in a more uncertain situation:

²This fact can be mathematically proven.

$$\frac{P(\text{'My umbrella is either blue or red'} \mid D) = 1 \quad P(\text{'My umbrella is not red'} \mid D) = 0.5}{P(\text{'My umbrella is blue'} \mid D) = ?}$$

or more compactly, using the symbols we introduced earlier,

$$\frac{P[(b \vee r) \wedge \neg(b \wedge r) \mid D] = 1 \quad P(\neg r \mid D) = 0.5}{P(b \mid D) = ?}$$

This says, above the line, that: according to our data D my umbrella is either blue or red (and can't be both), with full certainty; and according to our data we have no preferential beliefs on whether my umbrella is not red. What should then be the probability of my umbrella being blue, according to our data?

Intuitively that probability should be 50%: $P(b \mid D) = 0.5$. But which rules did we follow in arriving at this probability? More generally, which rules should we follow in assigning new probabilities from given ones?

The amazing result is that *the rules for truth-inference, formulae (6.1, 6.3, 6.2, 6.4), extend also to probability-inference*. The only difference is that they now hold for all values in the range $[0, 1]$, rather than only values 0 and 1.

This important result was taken more or less for granted at least since Laplace in the 1700s. But was formally proven for the first time in the 1940s by R. T. Cox; the proof has been refined since then. What kind of proof is it? It shows that if we don't follow the rules we arrive at illogical conclusions; we'll show some examples later.

Here are the fundamental rules of probability inference. In these rules, all probabilities can have values in the range $P() \in [0, 1]$, and the symbols a, b, D represent sentences of any complexity:

It is amazing that **ALL** inference is nothing else but a repeated application of these four rules – billions of times or more, in some inferences. All machine-learning algorithms are just applications or approximations of these rules. Methods that you may have heard about in statistics are just specific applications of these rules. Truth inferences are also special applications of these rules. Most of this course is, at



THE FUNDAMENTAL LAWS OF INFERENCE



“Not” \neg rule

$$P(\neg a \mid D) + P(a \mid D) = 1$$

“And” \wedge rule

$$P(a \wedge b \mid D) = P(a \mid b \wedge D) \cdot P(b \mid D) = P(b \mid a \wedge D) \cdot P(a \mid D)$$

“Or” \vee rule

$$P(a \vee b \mid D) = P(a \mid D) + P(b \mid D) - P(a \wedge b \mid D)$$

Self-consistency rule

$$P(a \mid a \wedge D) = 1$$

bottom, just a study of how to apply these rules in particular kinds of problems.



Reading

- *Probability, Frequency and Reasonable Expectation*
- Ch. 2 of *Bayesian Logical Data Analysis for the Physical Sciences*
- §§ 1.0–1.2 of *Data Analysis*
- Feel free to skim through §§ 2.0–2.4 of *Probability Theory*

7.4 How the inference rules are used

The fundamental rules represent, first of all, constraints of logical consistency among probabilities. If we have probabili-

ties $P(a | D) = 0.7$, $P(b | a \wedge D) = 0.1$, $P(a \wedge b | D) = 0.2$, then there's an inconsistency somewhere, because these values violate the and-rule: $0.2 \neq 0.1 \cdot 0.7$. In this case we must find the inconsistency and solve it. Since probabilities are quantified by real numbers, however, it's possible and acceptable to have slight discrepancies owing to numerical round-off errors.

The rules also imply more general constraints. For example we must *always* have

$$\begin{aligned} P(a \wedge b | D) &\leq \min\{P(a | D), P(b | D)\} \\ P(a \vee b | D) &\geq \max\{P(a | D), P(b | D)\} \end{aligned}$$

Exercise

Try to prove the two constraints above

The main use of the rules in concrete applications is for calculating new probabilities from given ones. The calculated probabilities will be automatically consistent. For each equation shown in the rules we can calculate one probability given the remaining ones in the equation, with some special cases when values of 0 or 1 appear.

For example, if we have $P(a \wedge b | D) = 0.2$ and $P(a | D) = 0.7$, from the and-rule we can find $P(b | a \wedge D)$:

$$\begin{aligned} \underbrace{P(a \wedge b | D)}_{0.2} &= P(b | a \wedge D) \cdot \underbrace{P(a | D)}_{0.7} \\ \implies P(b | a \wedge D) &= \frac{P(a \wedge b | D)}{P(a | D)} = \frac{0.2}{0.7} \approx 0.2857 \end{aligned}$$

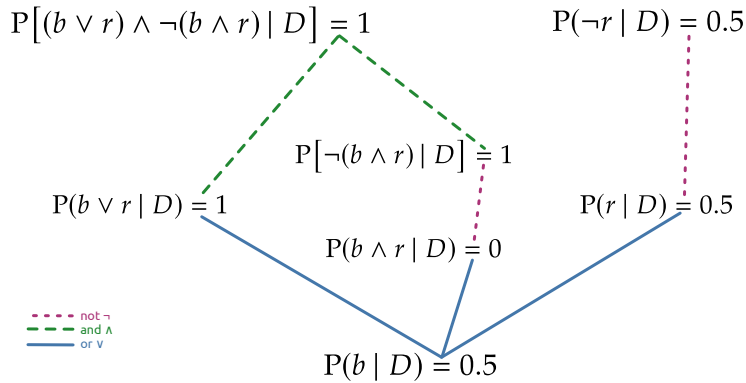
Let us now solve the umbrella inference from the previous section. Starting from

$$P[(b \vee r) \wedge \neg(b \wedge r) | D] = 1, \quad P(\neg r | D) = 0.5$$

we arrive at

$$P(b | D) = 0.5$$

by following from top to bottom the steps depicted here:



@@ example medical diagnosis

7.4.1 Derived rules

The rules above are in principle all we need to use. But from them it is possible to derive some additional shortcut rules that are automatically consistent with the fundamental ones.

First, it is possible to show that all rules you may know from Boolean algebra are a consequence of the fundamental rules. For example, we can always make the following convenient replacements anywhere in a probability expression:

$$\begin{aligned}
 A \wedge A &= A \vee A = A & \neg\neg A &= A \\
 A \wedge B &= B \wedge A & A \vee B &= B \vee A \\
 \neg(A \wedge B) &= \neg A \vee \neg B & \neg(A \vee B) &= \neg A \wedge \neg B \\
 A \wedge (B \vee C) &= (A \wedge B) \vee (A \wedge C) \\
 A \vee (B \wedge C) &= (A \vee B) \wedge (A \vee C)
 \end{aligned}$$

Two other derived rules are used extremely often, so we treat them separately.

7.5 Law of total probability or “extension of the conversation”

7.6 Bayes’s theorem

7.7 consequences of not following the rules,

@@ §12.2.3 of AI

- Exercise: *Monty-Hall problem & variations*
- Exercise: *clinical test & diagnosis*

7.8 Common points of certain and uncertain inference

No premises? No conclusions!

! Differences in terminology

- Some texts speak of the probability of a “random³ variable”, or more precisely of the probability that a random variable takes on a particular value. As you notice, we have just expressed that idea by means of a *sentence*. The viewpoint and terminology of random variables is a special case of that of sentences. As already discussed, in concrete applications it is important to know how a variable “takes on” a value: for example it could be directly measured, indirectly reported, or purposely set. Thinking in terms of sentences, rather than of random variables, allows us to account for these important differences.
- Some texts speak of the probability of an “event”. For all purposes an “event” is just what’s expressed in a sentence.

It’s a question for sociology of science why some people keep on using less flexible points of view or terminolo-

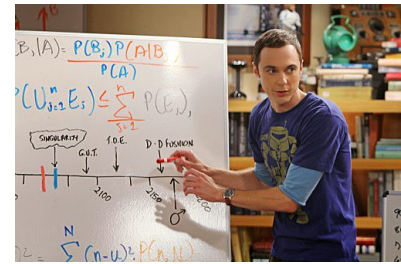


Figure 7.1: Bayes’s theorem guest-starring in *The Big Bang Theory*

gies. Probably they just memorize them as students and then a fossilization process sets in.

³What does "random" mean? Good luck finding an understandable and non-circular definition in texts that use that word. In these notes, if the word "random" is ever used, it means "unpredictable" or "unsystematic".

8 Data and information

8.1 Kinds of data

8.1.1 Binary

8.1.2 Nominal

8.1.3 Ordinal

8.1.4 Continuous

- unbounded
- bounded
- censored

8.1.5 Complex data

2D, 3D, images, graphs, etc.

8.1.6 “Soft” data

- orders of magnitude
- physical bounds

8.2 Data transformations

- log
- probit
- logit

9 Allocation of uncertainty among possible data values: probability distributions

9.1 The difference between Statistics and Probability Theory

Statistics is the study of collective properties of collections of data. It does not imply that there is any uncertainty.

Probability theory is the quantification and propagation of uncertainty. It does not imply that we have collections of data.

9.2 What's “distributed”?

Difference between distribution of probability and distribution of (a collection of) data.

9.3 Distributions of probability

9.3.1 Representations

- Density function
- Histogram
- Scatter plot

Behaviour of representations under transformations of data.

9.4 Summaries of distributions of probability

9.4.1 Location

Median, mean

9.4.2 Dispersion or range

Quantiles & quartiles, interquartile range, median absolute deviation, standard deviation, half-range

9.4.3 Resolution

Differential entropy

9.4.4 Behaviour of summaries under transformations of data and errors in data

9.5 Outliers and out-of-population data

(Warnings against tail-cutting and similar nonsense-practices)

9.6 Marginal and conditional distributions of probability

9.7 Collecting and sampling data

9.7.1 “Representative” samples

Size of minimal representative sample = $(2^{\text{entropy}})/\text{precision}$

- *Exercise: data with 14 binary variates, 10000 samples*

9.7.2 Unavoidable sampling biases

In high dimensions, all datasets are outliers.

Data splits and cross-validation cannot correct sampling biases

9.8 Quirks and warnings about high-dimensional data

10 Making decisions

10.1 Decisions, possible situations, and consequences

10.2 Gains and losses: utilities

10.2.1 Factors that enter utility quantification

Utilities can rarely be assigned a priori.

10.3 Making decisions under uncertainty: maximization of expected utility

11 The most general inference problem