dedication (optional)

Summary

Write your summary here...

Preface

Write your preface here...

Table of Contents

List of Tables

List of Figures

Abbreviations

PII	=	Personally Identifiable Information
PPDP	=	Privacy-Preserving Data Publishing
QID	=	Quasi Identifier
IMDB	=	Internet Movie Database
PINQ	=	Privacy Integrated Queries
JADE	=	Java Agent framework for Distance learning Environments
AMS	=	Agent Management System
DF	=	Directory Facilitator

Chapter 1

Introduction

In modern world

1.1 Objective

1.2 Scope

1.3 Research Question/Goal

Can we do distributed machine learning while still provide a differential privacy guarantee?

1.4 Thesis Structure

Here there will a short summary of each chapter

Chapter 2

Background and Motivation

In this section we will first explain some basic concepts and expressions that are used in the privacy context such anonymization operations and Personally Identifiable Information(PII). Then we will have a look at some classic examples of failure to preserve privacy when data publishing and how these attacks motivated us to choose our topic for this thesis.

2.1 Concepts and Expressions

In the most basic form of privacy-preserving data publishing (PPDP), the data holder has a table of attributes from the following four categories: Explicit Identifier, Quasi Identifier, Sensitive Attributes, and Non-Sensitive Attributes [10]. A summary of each category can be found in Table 2.1.

Attribute name	**Definition**	**Example**
Explicit Identifier	Explicitly identifies record owners	Government identity number (e.g SSN)
Quasi Identifier(QID)	Potentially identifies record owners	Birth date and gender
Sensitive Attributes	Sensitive information about a person	Income, disability status
Non-Sensitive Attributes	All other attributes	Favorite band

Table 2.1: Table of basic categories of database attributes

From these categories, it would be easy to think that Personally Identifiable Information (PII) would only be found in the first attribute. As we will see in the next section, this is not the case. Recent privacy laws have defined PII in a much broader way. They account for the possibility of deductive disclosure and do not lay down a list of attributes that constitutes as PII. For example, the European Parliament made a set of directives known as the

Data Protection Directive, in which personal data is defined as: any information relating to an [] natural person [] who can be identified, directly or indirectly, in particular by reference [] to one or more factors specific to his physical, physiological, mental, economic, cultural, or social identity[9].

In order to remove any PII from a dataset, it needs to go through a process called anonymization. This constitutes a series of modifications/manipulations of with the ultimate end goal of protecting the privacy of the dataset's participants. Fung et al.[10] operates with a number of five basic operations which might be applied for this purpose. These operations are shortly described in Table 2.2.

Anonymization Operation	**Definition**
Generalization	Replaces the value with more general value, such as a mean value
Suppression	Replaces the value with a special value, indicating that the replaced values are not disclosed
Anatomization	De-associates the relationship between the quasi-identifier and sensitive information
Permutation	Partitions a set of data records into groups and shuffles their sensitive values
Perturbation	Replace the original value with a synthetic value that keep the statistical characteristics

Table 2.2: Table of anonymization operations (adapted from [10])

2.2 Privacy Breaches

In the recent years there have been many failures in privacy preserving data publishing. Many companies have been faced with a PR disaster after releasing data about their customers thinking them being anonymized, only to have people de-anonymize their data and breaching the privacy of the datasets' participants. In this section we will have a look at some of these privacy failures.

2.2.1 Netflix Prize Competition

Netflix, the world's largest online movie streaming website, decided in 2006 to crowdsource a new movie suggestion algorithm and offered a cash prize of 1 million dollar for the most efficient algorithm. To help the research, they released 100 million supposedly anonymized movie ratings from their own database. In order to protect the privacy of their users, Netflix removed all user level information; such as name, username, age, geographic location, browser used, etc. They also deliberately perturbed "some of the rating data for some customers[...] in one or more of the following ways: deleting ratings; inserting alternative ratings and dates, and modifying random dates"[3]. The released data records included an anonymized user ID, movie name, date of rating, and the user's rate on a scale from 1 to 5.

Two researchers from the University of Texas,Narayanan and Shmatikov[20], demonstrated that an adversary who knows only a little bit about an individual subscriber can easily identify this subscriber's record in the dataset. Using the publicly available dataset from the Internet Movie Database (IMDB) as the source of background knowledge, they matched certain subscribers with their Netflix records, and uncovered their apparent political preferences and other potentially sensitive information. The paper also offered a formal mathematical treatment of how a small amount of auxiliary knowledge about an individual can be used to do a fairly reliable re-identification. In the case of the Netflix dataset, the authors [20] found that with only 8 movie ratings, 99% of the records could be uniquely identified. Furthermore, they proved that the de-anonymization algorithm they employed is robust to discrepancies in the rating and dates.

2.2.2 Group Insurance Commission

In 1997, Latanya Sweeney wrote a paper on how she had identified the medical records of Massachussets governor William Weld based on publicly available information from the database of Group Insurance Commission. She achieved this analyzing data from a public voter list, and linked it with patient-specific medical data through a combination of birth date, zip code, and gender[26]. As these columns were similar in both databases, their combination could be used to identify medical records that belong to either one person, or a small group of people. Sweeney hypothesized that 87% of the US population could be identified by having the combination of the three aforementioned records. It's worth noting here that this theory is not conclusive. A paper by Daniel Barth-Jones suggests that the re-identification of Weld may have been a fluke due to his public figure, and that ordinary people risk of identification is much lower[2].

2.2.3 New York Taxi dataset

The New York City Taxi and Limousine Commission released a dataset in 2013 containing details about every taxi ride that year, including pickup and drop-off times, location, fare, as well as anonymized (hashed) versions of the taxi's license and medallion numbers. Vijay Pandurangan, a researcher for Google, wrote a blog-post where he showed how he exploited a vulnerability in the hashing-function to re-identify the drivers. He then showed how this could be potentially used to calculate any driver's personal income[21].

Another researcher, called Anthony Tockar, wrote an article during his internship at Neustar Research where he proved that the dataset also contained an inherent privacy risk to the passengers which had been riding New York Taxis. Even though there was no information in the dataset on who had been riding the taxis, Tockar showed that by using auxiliary information such as timestamped pictures, he could stalk celebrities and figure out to where they were driving, and how much they tipped the driver. He also used map data from Google Maps to create a map of drop-off locations for people that had exited a late night visit from gentleman's club and taken a cab home. He then used websites like Spokeo and Facebook to find the cab customer's ethnicity, relationship status, court records, and even a profile picture[27].

2.3 Attack Vectors

2.3.1 Linkage Attacks [Rename]

In each of the examples in the previous section, the privacy breach was achieved through an attack model called linkage attacks. These types of attacks are characterized that they create a decision rule which link at least one data entry in the anonymized dataset with public information which contain individual identifiers, given that the probability of these two matching exceeds a selected confidence threshold.

In the literature[4, 10], they broadly classify the attack models into two categories: Record linkage and attribute linkage. In both these types of attack, we need to assume that the attacker knows the QID of the victim.

Record Linkage[rename]

In the case of attribute linkage, some quasi-identifier value QID identifies a small number of records in the original dataset, which is called a group. If the victim's QID is the same, he or she is then vulnerable to being linked to this much smaller number of records in the group. With the help of some additional information, there is then a chance that the attacker could uniquely identify the victim's records in the group. This is what happened to governor William Weld as mentioned in section2.2.2. Sweeney linked medical data with a voter list, which both included the QID= <Zip,Birth date,Sex >. She then employed the background knowledge that governor Weld was admitted to the hospital at the certain date, which allowed her to uniquely identify him from the small group of people that shared the same QID as him.

k-anonymity Sweeney[26] proposed a notion called k-anonymity in order to try and prevent record linkage through QID. She defined that a table T with a quasi-identifier QI_T would satisfy k-anonymity if and only if each sequence of values $T[QI_T]$ appears with at least k occurrences in $T[QI_T]$. From that definition it appears that k-anonymity is designed to prevent record linkage through hiding the record of the victim in a big group of records with the same QID. This method has a weakness however, as an attacker can still infer a victim's sensitive attribute, such as having the attribute hasDisease=true, if most records in a group have similar values on those sensitive values.

Attribute Linkage[rename]

The aforementioned weakness is an example of an attribute linkage attack. An attacker might not be able to precisely identify the victim through a record, but can still infer his or her sensitive values from the published data. The attacker does this based on the set of sensitive values associated to the group the victim belongs to.

To prevent this type of attack, Machanavajjhala et al[16] proposed an idea based on diminishing the correlation between the QID attributes and the sensitive values, which they called l-diversity. The method requires each group with similar QID to have l distinct values for the sensitive attributes.

2.3.2 Background Information

Needs a segway into the next section. Maybe Dalenius desideratum, something that sets up Differential privacy

2.4 Challenges in Data Privacy

This section will summarize the previous section, and elaborate on our motivation for doing our project.

2.5 Related Work

Boutet et al. worked on a privacy-preserving distributed collaborative filtering which relied on user profile obfuscation and randomized response.[5].

Boutsis et al developed a participatory sensing system for smartphones which assumed that the data was distributed locally. Their system was called LOCATE, which handled ensured the privacy of the users, but they did not provide a differential privacy guarantee. [6]

Chaudhuri and Monteloni designed a logistic regression algorithm which guaranteed differential privacy in 2009, but their algorithm was designed to run on a single centralized database[7].

Han et al. investigated the problem of preserving differential privacy in distributed constrained optimization. By creating an algorithm based on stochastic gradient descent, which preserved privacy by adding noise to the public coordination signals (i.e gradients). [14]

Ji et al. [15] recently proposed a distributed solution using logistic regression, which learned from both private and publicly available medical datasets. Their solution differ from our own as they employ a globally synchronized structure, whereas our own solution works asynchronously. Their approach also requires a global aggregation of gradients, compared to our own which employ distributed ensemble learning,

Chapter 3

Basic Theory

In a world where massive amounts of sensitive personal data are being collected, attacks on the individual's privacy are becoming more and more of a threat. One type of attack is the identification of an individual's personal information from massive data sets, such as people's movie ratings from the Netflix data set[20], and the medical records of a former governor of Massachusetts[2]. These types of privacy breaches may lead to the unwanted discovery of a person's embarrassing information, and could also lead to the theft of an individual's private data or identity.

Many different approaches have been tried by data custodians to privatize the data they hold, such as removing any columns containing Personally Identifiable Information (PII), anonymizing the data by providing k-anonymity protection[26], or perform group based anonymization through l-diversity[16]. All of these methods mentioned have been proved to be susceptible in some way or form to attacks [11]. Motivated by these shortcomings, a researcher at Microsoft came up with a data theoretical framework called differential privacy, which operates off a solid mathematical foundation and have strong theoretical guarantees on the privacy and utility of the released data.

3.1 Differential Privacy

The term ”differential privacy” was defined by Dwork as a description of a promise, made by a data holder 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.” [8] In an ideal situation, databases which implement differential privacy mechanisms can make confidential data widely available for accurate data analysis, without resorting to data usage agreements, data protection plans, or restricted views. Nevertheless, the Fundamental Law of Information Recovery states that overly accurate answers to too many questions will destroy privacy in a spectacular way [8], meaning that data utility will eventually be consumed.

3.1.1 Definition of Differential Privacy

The classic example for explaining a security breach is the case of Mr White: Suppose you have access to a database that allows you to compute the income of all residents in a specified area. If you knew that Mr White was going to move, simply querying the database before and after his relocation would allow you to deduce his income.

Definition 1: a mechanism M is a random function that takes a dataset D as input, and outputs a random variable M(D).

Definition 2: the distance of two datasets, d(D_1, D_2), denotes the minimum number of sample changes that are required to change D_1 into D_2.

Formally, differential privacy is defined as follows: A randomized function f gives ϵ-differential privacy if for all data sets D_1 and D_2 differing on at most one row, and all $S \subseteq Range(M)$,

$$Pr[M(D_1) \in S] \leq \mathrm{e}^{(\epsilon)} \times Pr[M(D_2) \in S] \tag{3.1}$$

What this means is that the risk to an individual's privacy should not be substantially increase as a result of participating in a statistical database, as the risk is bounded by the parameter ϵ. Therefore an attacker should not be able to learn anything about any participant that they would not have learned if the participant had opted out of participating. To achieve this standard, privacy preserving data analysis platforms such as PINQ[17], Airavat[23] and Fuzz[13] have all implemented features such as privacy budgeting and noise mechanism, which will be explained in the following sections.

Must write about privacy budget and how that factors in the queries

3.1.2 Privacy Budget

The quotient $\frac{Pr[M(D_1) \in S]}{Pr[M(D_2) \in S]}$ is called the knowledge gain ratio, which measures the extent to which an attacker can ascertain the difference between the two datasets[1]. Differential privacy requires that this ratio is limited to e^{ϵ}. This is because as the ratio grows larger, an attacker can determine with greater probability that the query result was obtained from one dataset over the other. Sarathy and Muralidhar[24] argues that the knowledge gain ratio r, should satisfy the following requirement:

$$r \leq 1 + \epsilon \tag{3.2}$$

What this mean, is that when the ϵ is very small and close to zero, $e^{\epsilon} \approx 1 + \epsilon$, we have achieved ϵ-differential privacy. When making a practical implementation, the ϵ value represents the privacy budget of the dataset. The budget value is set by the data analyst, but as it was shown in [24] that the attacker's knowledge gain rises exponentially with the rising number of queries. Setting a high value for ϵ will have a measurable impact on data privacy.

Privacy budgeting was introduced to limit the amount of information a data analyst can obtain about any individual with data records in the dataset. The data analysis platform will track every query to ensure that both individual queries and aggregation queries do not exceed the given budget. This privacy standard forbids further queries to the database once the budget has been consumed.

3.1.3 Noise Mechanisms

This is guaranteed by applying noise to the result of an query to the dataset, by using the function M.There are many different mechanisms for applying this noise, but the two most common are the Laplace mechanism and the Exponential mechanism.

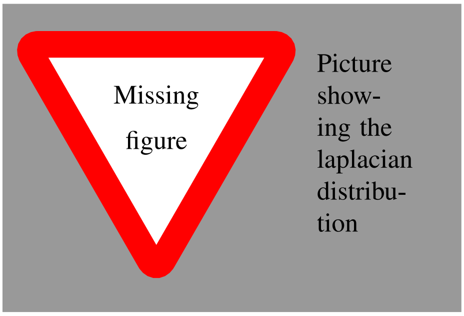
Laplace Mechanism

The Laplace mechanism involves adding random noise which follows the Laplace statistical distribution. The most common question that needs to be answered before doing research with differentially private data, is how we should define our Laplace random variable, i.e how much noise needs to be added. The Laplace distribution centered around zero has only one parameter, its scale b, and this is proportional to its standard deviation.

$$Lap(x|b) = \frac{1}{2b} exp(\frac{|x|}{b}) \tag{3.3}$$

The scale is naturally dependent on the privacy parameter ϵ, and also on the risk of the most different individual having their private information leaked from the data. This risk is called the sensitivity of the query, and is defined mathematically as:

$$\Delta f = \underset{D_1,D_2}{max} ||f(D_1) - f(D_2)||_1 \tag{3.4}$$



Not sure if we need the pic

This equation states that the maximum difference in the values that the query f may take is on a pair of databases that differ on only one row. Dwork proved that adding a random Laplace variable, $(\Delta f/\epsilon)$, to a query you could guarantee ϵ-differential privacy[8].

Exponential Mechanism

The Exponential mechanism proposed by McSherry and Talwar [18] is a method for selecting one element from a set, and is commonly used if a non-numeric value query is used. An example would be: ”What is the most common eye color in this room?”. Here it would not make sense to perturb the answer by adding noise drawn from the Laplace distribution. The idea of the mechanism is to select the output from all the possible answers at random, with the probability of selecting a particular output being higher for those outputs that are ”closer” to the true output.

More formally, let A be the range of of possible outputs for the query function f. Also, let $u_f(D, a)$ be a utility function that measures how good an output $a \in A$ is as an answer to the query function f given that the input dataset is D (Note that higher values of u_f represents better outputs). The sensitivity function will then be defined as the maximum possible change in the utility function's value u_f due to the addition or removal of one person's data from the input, i.e:

Definition 4: the sensitivity of score function u_f is defined as

$$S(u_f) = \underset{d(D_1,D_2)=1,a\in A}{max} ||u_f(D_1, a) - u_f(D_2, a)|| \tag{3.5}$$

3.2 Logistic Regression

Logistic regression is a popular probability model developed by D.R Cox in 1958. The most basic form is used to predict a binary response based on a set of features.

$$h_\theta(x) = \theta^T x = \sum_{i=0}^{n} \theta_i x_i,$$

The logistic regression model is

$$p(y = 1|\mathbf{x}, \mathbf{w}) = \sigma(y\mathbf{w}^T\mathbf{x}) = \frac{1}{1 + exp(y\mathbf{w}^T\mathbf{x})} \tag{3.6}$$

(1) It can be used for binary classification or for predicting the certainty of a binary outcome. See Cox and Snell (1970) for the use of this model in statistics. This note focuses only on computational issues related to maximum-likelihood or more generally maximum a-posteriori (MAP) estimation. A common prior to use with MAP is: p(w) N (0, 1 I) (2) Using ¿ 0 gives a regularized estimate of w which often has superior generalization performance, especially when the dimensionality is high (Nigam et al., 1999). Given a data set (X, y) = [(x1, y1), ...,(xN , yN)], we want to find the parameter vector w which maximizes:

$$log(1 + exp(y_i\mathbf{w}^T\mathbf{x}_i)) - \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} \tag{3.7}$$

The gradient of this objective is

$$g = \Delta_w l(\mathbf{w}) = \sum_i (1 - \sigma(y_i\mathbf{w}^T\mathbf{x}_i))y_i\mathbf{x}_i - \lambda\mathbf{w} \tag{3.8}$$

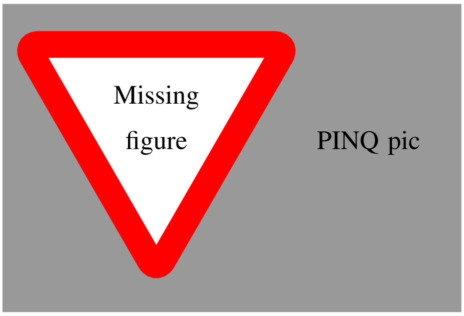
3.3 Programming Frameworks Used [Rename]

3.3.1 PINQ

Privacy Integrated Queries (PINQ) is a platform for computing on privacy-sensitive datasets, while providing guarantees of differential privacy for the underlying records. It is powered by the LINQ declarative query language and it allows analysts to run SQL-like queries on datasets while protecting individual privacy.

PINQ was developed as a prototype platform at Microsoft Research by lead researcher Frank McSherry[17], and is placed as a thin layer between the query engine and the analyst

Consider adding the PINQ picture from Microsoft here



It does not manage data or execute queries, but instead it supplies differentially private implementations of common transformations and aggregations written in LINQ.

When a query is run through PINQ, it first evaluates the privacy guarantee to ensure that the privacy cost is within the range of the privacy budget. If the query is valid, the cost of the query is detracted from the privacy budget and passes the query to the database engine for execution. When the results in returned from LINQ, the last step of the process is to add the proper amount of noise based on the epsilon value

Is this right?

Write how we adapted PINQ to Java 8 here.

3.3.2 JADE

The Java Agent framework for Distance learning Environments(JADE) is a middleware which facilitates the development of multi-agent systems. An application based on JADE is made of a set of components called Agents, where each one have an unique name. Agents execute tasks and interact by exchanging messages between each other. Agents execute on top of a Platform that provides them with basic services such as message delivery. A platform is composed of one or more Containers, where the Containers can be executed on different hosts thus achieving a distributed platform. The Main Container is a special container which exists in the platform, as it has two special properties. 1: It must be the first container to start in the platform, and all other containers must register to it. 2: Two special agents are included; the Agent Management System (AMS) which represents the single authority in the platform, and is an agent tasked with platform management actions such as starting and killing other agents. The other special agent is the Directory Facilitator (DF), which provides a directory which announces which agents are available on the platform. This acts like a yellow pages service where agents can publish the services they provide and find other agents providing services they need.

Write how we adapted the use of JADE here

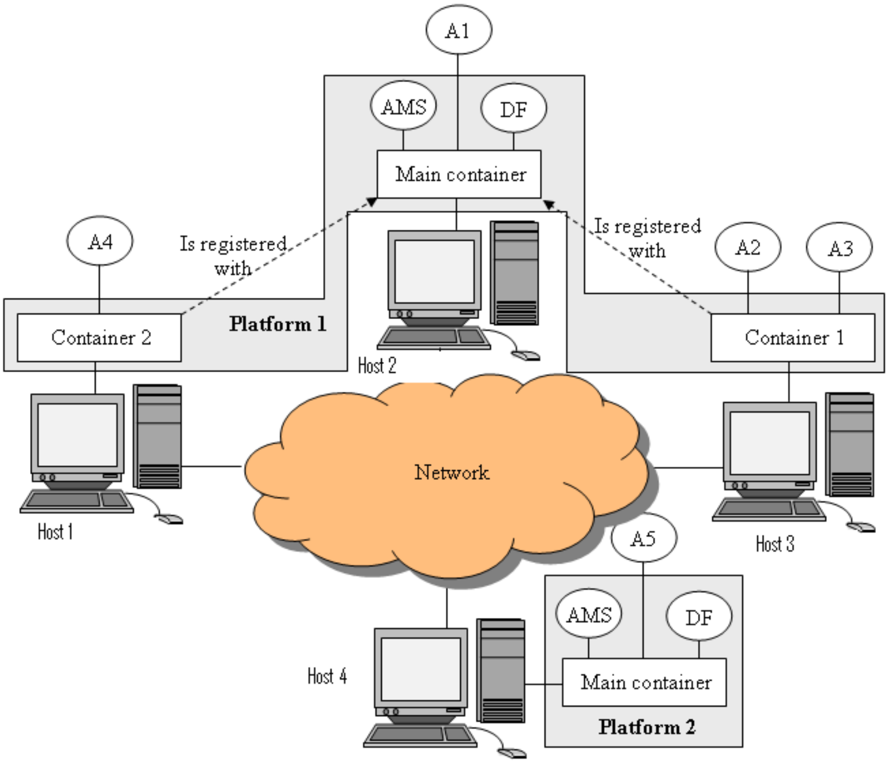


Figure 3.1: JADE Architechture

Consider remaking this figure so we don't have to cite it

3.4 Homomorphic Encryption

Homomorphic encryption is an encryption scheme which allows computations to be carried out on ciphertext, meaning plaintext that have been encrypted using a cipher. The result of the computations is also encrypted, and can be deciphered back to plaintext using a key K. This has long been considered as crypthography's holy grail [19], as this would allowing operating on encrypted text without knowing the decryption key. For example, given ciphertexts $C_1 = Enc_K(Data1)$ and $C_2 = Enc_K(Data2)$, an additively homomorphic encryption scheme would allow to combine C_1 and C_2 to obtain $Enc_K(Data1 + Data2)$. More concretely this means that if you encrypt your data using such an encryption scheme, you can transfer your data to an untrusted server which can perform some arbitrary computations on that data without being able to decrypt the data itself.

Up until recently, all published homomorphic encryption schemes only supported one basic operation, most commonly addition. These schemes could only be called partially homomorphic, as they did not provide any extensive functionality. The notion of a fully homomorphic encryption schemes was first proposed by Rivest, Adleman, and Dertouzos in 1978 [22], but it wasn't until 2009 that Craig Gentry published a doctoral thesis where he proved that he had constructed a fully homomorphic scheme[12]. Gentry's solution was based on ”ideal lattices” as well as a method to double-encrypt the data in such a way that the errors could be handled ”behind the scenes”. By periodically unlocking the inner layer of encryption underneath an outer layer of scrambling, the computer could hide and recover from errors without ever analyzing the secret data.

The downside of Gentry's two-layered approach is that it requires a massive computational effort. Bruce Schneier, a leading American cryptographer, pointed out ”Gentry estimates that performing a Google search with encrypted keywords – a perfectly reasonable simple application of this algorithm – would increase the amount of computing time by about a trillion. Moore's law calculates that it would be 40 years before that homomorphic search would be as efficient as a search today, and I think he's being optimistic with even this most simple of examples[25].”

Chapter 4

Experiment

Introduce the experiment

4.0.1 Dataset

This section will introduce the dataset(s) used. What features it contains, what we try to learn/classify, and why we chose to use it.

4.1 Algorithm

This section explain the logistic regression algorithm, how it is commonly used, and what modifications are needed when used in a distributed setting. Explanation on how it is used in a differentially private manner is explained in the architecture section.

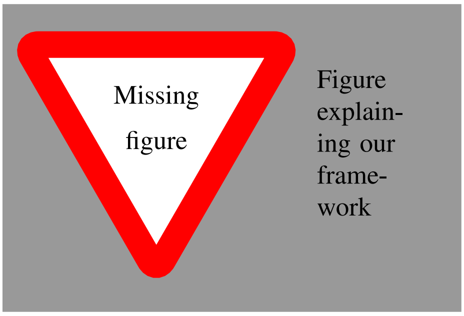
4.1.1 Distribution

Introduce notion of distributed machine learning.
Why did we choose to perform distributed learning?
How does it fit with the notion of differential privacy?
Can we guarantee differential privacy while doing it distributed
Record-based differential privacy. Does it work?

This section should maybe go somewhere else, but where?

4.2 Architecture

In this section we will describe the architecture of Archipelago, our distributed machine learning system. Using notions borrowed from PINQ, we designed a distributed system using the JADE framework. Consists of explanation of peers, NoisyQueryable, DataLoader



4.2.1 Communication

Peer

How is each peer set up, and what behaviors do they implement? How do they update then propagate the model being learned. How do they know when to stop?

messaging

How do the peers communicate with each other? What does a message look like? What is the PeerGraph? What controls the messages and determines where they should go?

4.2.2 Learning

How is a logistic model implemented in our framework? How is a model created and passed around the network? How does the ensemble learning choose the best model? What kind of performance metrics are used?

4.2.3 Privacy

How does our framework guarantee differential privacy? What concepts have been borrowed from PINQ? What is NoisyQueryable?

4.2.4 Experiment

How are the experiments set up? Explain the testing scheme.

Chapter 5

Analysis

What are the results of our experiments?

What did we learn from the basic structure of creating our framework?

What difficulties did we encounter?

What can we take away from our experiments?

What should have been done better?

Chapter 6

Conclusion

Notes

Bibliography

[1] Abowd, J. M., Vilhuber, L., 2008. How protective are synthetic data? In: Privacy in Statistical Databases. Springer, pp. 239–246.

[2] Barth-Jones, D. C., 2012. The're-identification'of governor william weld's medical information: a critical re-examination of health data identification risks and privacy protections, then and now. Then and Now (June 4, 2012).

[3] Bell, R. M., Koren, Y., Dec. 2007. Lessons from the netflix prize challenge. SIGKDD Explor. Newsl. 9 (2), 75–79.
URL `http://doi.acm.org/10.1145/1345448.1345465`

[4] Bonchi, F., Ferrari, E., 2010. Privacy-aware Knowledge Discovery: Novel Applications and New Techniques. CRC Press.

[5] Boutet, A., Kermarrec, A.-M., Frey, D., Guerraoui, R., Jegou, A., 2013. Privacy-Preserving Distributed Collaborative Filtering.
URL `https://hal.inria.fr/hal-00799209/file/RR-8253.pdf`

[6] Boutsis, I., Kalogeraki, V., Mar. 2013. Privacy preservation for participatory sensing data. In: 2013 IEEE International Conference on Pervasive Computing and Communications (PerCom). IEEE, pp. 103–113.
URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6526720`

[7] Chaudhuri, K., Monteleoni, C., 2009. Privacy-preserving logistic regression. In: Advances in Neural Information Processing Systems. pp. 289–296.
URL `http://papers.nips.cc/paper/3486-privacy-preserving-logistic-reg`

[8] Dwork, C., Roth, A., 2013. The algorithmic foundations of differential privacy. Theoretical Computer Science 9 (3-4), 211–407.

[9] European Parliament and Council of the European Union, Apr. 2006. Directive 2006/24/EC.
URL `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:105:0054:0063:EN:PDF`

[10] Fung, B. C., Wang, K., Fu, A. W.-C., Yu, P. S., 2010. Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques, 1st Edition. Chapman & Hall/CRC.

[11] Ganta, S. R., Kasiviswanathan, S. P., Smith, A., 2008. Composition attacks and auxiliary information in data privacy. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 265–273.

[12] Gentry, C., 2009. Fully homomorphic encryption using ideal lattices. In: Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing. STOC '09. ACM, pp. 169–178.
URL http://doi.acm.org/10.1145/1536414.1536440

[13] Haeberlen, A., Pierce, B. C., Narayan, A., 2011. Differential privacy under fire. In: Proceedings of the 20th USENIX Conference on Security. SEC'11. USENIX Association, Berkeley, CA, USA, pp. 33–48.
URL http://dl.acm.org/citation.cfm?id=2028067.2028100

[14] Han, S., Topcu, U., Pappas, G. J., Nov. 2014. Differentially Private Distributed Constrained Optimization.
URL http://arxiv.org/abs/1411.4105

[15] Ji, Z., Jiang, X., Wang, S., Xiong, L., Ohno-Machado, L., Jan. 2014. Differentially private distributed logistic regression using private and public data. BMC medical genomics 7 Suppl 1 (Suppl 1), S14.
URL http://www.biomedcentral.com/1755-8794/7/S1/S14

[16] Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M., 2007. l-diversity: Privacy beyond k-anonymity. ACM Transactions on Knowledge Discovery from Data (TKDD) 1 (1), 3.

[17] McSherry, F., June 2009. Privacy integrated queries. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data (SIGMOD). Association for Computing Machinery, Inc., for more information, visit the project page: http://research.microsoft.com/PINQ.
URL http://research.microsoft.com/apps/pubs/default.aspx?id=80218

[18] McSherry, F., Talwar, K., Oct 2007. Mechanism design via differential privacy. In: Foundations of Computer Science, 2007. FOCS '07. 48th Annual IEEE Symposium on. pp. 94–103.

[19] Micciancio, D., Mar. 2010. Technical Perspective: A First Glimpse of Cryptography's Holy Grail.
URL http://cacm.acm.org/magazines/2010/3/76275-technical-perspective-a-first-glimpse-of-cryptographys-holy-g
fulltext

[20] Narayanan, A., Shmatikov, V., 2008. Robust de-anonymization of large sparse datasets. In: Security and Privacy, 2008. SP 2008. IEEE Symposium on. IEEE, pp. 111–125.

[21] Pandurangan, V., Jun. 2014. On taxis and rainbows: Lessons from nycs improperly anonymized taxi logs. Visited: 2016-03-06.
URL `https://medium.com/@vijayp/of-taxis-and-rainbows-f6bc289679a1`

[22] Rivest, R. L., Adleman, L., Dertouzos, M. L., 1978. On data banks and privacy homomorphisms.

[23] Roy, I., Setty, S. T. V., Kilzer, A., Shmatikov, V., Witchel, E., 2010. Airavat: Security and privacy for mapreduce. In: Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation. NSDI'10. USENIX Association, Berkeley, CA, USA, pp. 20–20.
URL `http://dl.acm.org/citation.cfm?id=1855711.1855731`

[24] Sarathy, R., Muralidhar, K., Apr. 2011. Evaluating laplace noise addition to satisfy differential privacy for numeric data. Trans. Data Privacy 4 (1), 1–17.
URL `http://dl.acm.org/citation.cfm?id=2019312.2019313`

[25] Schneier, B., Jul. 2009. Networks (2nd ed.).
URL `https://www.schneier.com/blog/archives/2009/07/homomorphic_enc.html`

[26] Sweeney, L., 2002. k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10 (05), 557–570.

[27] Tockar, A., Sep. 2014. Riding with the stars: Passenger privacy in the nyc taxicab dataset. Visited:2016-03-06.
URL `http://research.neustar.biz/2014/09/15/riding-with-the-stars-passenger-privacy-in-the-nyc-taxicab-dataset/`

Appendix

Write your appendix here...