NTNU – Trondheim
Norwegian University of
Science and Technology

Reclaiming Data Ownership: Differential Privacy in a Decentralized Setting

Alexander Benjamin
Asplund
Peter F Hartvigsen

Master of Science in Computer Science
Submission date: July 2015
Supervisor: Anders Kofod-Petersen, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Abstract

In the field of privacy-preserving data mining the common practice has been to gather data from the users, centralize it in a single database, and employ various anonymization techniques to protect the personally identifiable information contained within the data. Both theoretical analyses and real-world examples of data breaches have proven that these methods have severe shortcomings in protecting an individual's privacy. A breakthrough may have been achieved in 2006 when a method called differential privacy was proposed as a mathematical guarantee for the privacy of each record in a data set. Since then, an avenue of research has been to make this concept work in a distributed setting.

In this thesis we propose a decentralized framework that allows users to perform classification after aggregating their locally trained models in a privacy-preserving manner. We describe a series of experiments on the tuning of each major parameter involved, and show the effects of these on the privacy-utility trade-off. We also compare our classification performance to other cases in the literature and show how we achieve competitive performance.

Based on our results, we have produced a set of criteria for applying differential privacy to a machine learning application, as well as two business sectors where we see potential for a successful system. We hope that our research will pave the way for distributed applications where users maintain control of their own data, and use it for learning without giving up their privacy.

Sammendrag

Innenfor forskningsfeltet kalt *privacy-preserving data mining* har det lenge vært vanlig praksis å samle data fra en mengde brukere, sentralisere den i en stor database, og så anvende ulike anonymiseringsteknikker for å beskytte de sensitive personopplysningene iboende i dataen. Både teoretiske analyser og virkelige hendelser med datainnbrudd har bevist at disse metodene har alvorlige svakheter i hvordan de beskytter individers personopplysninger. Et mulig gjennombrudd ble oppnådd i 2006 da en metode kalt *differential privacy* ble framlagt med en matematisk garanti for å beskytte hver rad i et datasett. Siden da har det vært et av fokusene i forskning å få dette konseptet til å virke i en distribuert omgivelse.

I denne masteroppgaven har vi foreslått et desentralisert rammeverk som lar brukerne gjennomføre klassifisering etter å ha aggregert sine lokalt trente modeller, på en måte som beskytter deres personopplysninger. Vi beskriver en rekke eksperimenter hvor vi justerte hvert enkelt parameter, og viser hvilken innvirkning dette har på avveiningen mellom nytten av resultatet og personvernsnivået. Vi har også sammenlignet vår klassifiseringsytelse med andre resultater presentert i forskningslitteraturen og viser at vi har oppnådd konkurransedyktige resultater.

Basert på våre resultater, har vi produsert et sett med kriterier for hvordan man skal anvende *differential privacy* i en maskinlæringsapplikasjon. I tillegg har vi vist til to sektorer i forretningslivet hvor vi ser potensiale for å skape et vellykket system. Vi håper at vår forskning vil virke hjelpe muliggjøre distribuerte applikasjoner hvor brukerne beholder kontroll over sine data, og kan bruke den for læring uten å gi fra seg sine personopplysninger.

Preface

This masters thesis is the culmination of our work, carried out during the final semester of our Master of Computer Science degree at the Norwegian University of Science and Technology, NTNU. We would like to thank our supervisor, Professor Anders Kofod-Petersen, for his supervision and guidance throughout the year.

Trondheim, July 14,2015 - Alexander Asplund and Peter Frøystad

Table of Contents

List of Tables

List of Figures

Abbreviations

AMS	=	Agent Management System
DF	=	Directory Facilitator
DP	=	Differential Privacy
DPA	=	European Data Protection Authorities
IMDB	=	Internet Movie Database
JADE	=	Java Agent framework for Distance learning Environments
PII	=	Personally Identifiable Information
PINQ	=	Privacy Integrated Queries
PPDM	=	Privacy-Preserving Data Mining
PPDP	=	Privacy-Preserving Data Publishing
QID	=	Quasi Identifier
RQ	=	Research Question
SGD	=	Stochastic Gradient Descent

Symbols

ϵ	=	Privacy parameter that represents the privacy budget
ϵ_A	=	Parameter which represents the privacy level of our aggregation mechanism
λ	=	Regularization parameter
α	=	Learning rate parameter
θ	=	Parameter vector learned in logistic regression
γ	=	Parameter used in adaptive learning rate
η	=	Noise vector used to guarantee differential privacy
A	=	Represents our aggregation mechanism
D	=	Represents a dataset
M	=	Represents a general privacy mechanism, like the Laplacian or the Exponential.

Chapter 1

Introduction

All over the world people are interacting with technology more than ever; when using their cell phone, shopping online, visiting a doctor who uses electronic records, and in countless other acts. This usage generates a massive amount of information, leading to data being more deeply integrated into our daily lives than ever before. Sintef published a report in 2013 which stated that: ”A full 90% of all the data in the world has been generated over the last two years [Dragland, 2013].” With this massive influx of information, new fields of both academic study and commercial interest have appeared to find out how to best analyze this data.

The terms ”big data” and ”analytics” have been widely used as common designations for this emerging field of technology. The communal definition for describing big data stems from a 2001 research report[Laney, 2001], in which analyst Doug Laney defined the problem as a three-dimensional challenge: ”Big data is high-volume, -velocity and -variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making.” The first part of his challenge, commonly known as the 3 Vs of big data, deals with the necessary qualifications for data to be called ”big data”, while part two and three is the how and why.

The wide variety of the potential applications of big data analytics have also raised essential questions about whether our social and ethical norms are sufficient to protect privacy in a world which has entered ”the era of big data”. Both in the European Union and in the United States there have been efforts made to create new laws for handling data privacy. The Council to the President, an advisory group to the US President, concluded in their 2014 report [U.S Government, 2014] that preserving privacy values would be their number one recommendation when designing a new policy framework for big data. Furthermore, they advised that more than 70 million USD should be made available to federal research in privacy-enhancing technologies.

1.1 Problem Statement

The objective of this study is to contribute to the aforementioned field of study, more specifically in the area of Privacy-Preserving Data Mining (PPDM). In our work we explore the utility of employing a privacy-preserving technique called differential privacy. Relying on previous research on differential privacy, homomorphic encryption, and peer to peer communication, we would like to create an framework that allows for distributed, scalable machine learning while preserving the privacy of the participants.

RQ1: How big is the loss of accuracy in a distributed, differentially private system, compared to a centrally trained model?
While there have been research on both distributed and differentially private machine learning systems, there have been very little research done on a combination of both. Results from research on differential privacy indicate that there often is a trade-off between privacy and a loss of accuracy. We want to study this trade-off in our distributed approach and analyze which factors comes into play and how they can be handled in a way that leads to an optimal result.

RQ2: How can the variance in accuracy between participants be minimized?
Our system architecture is based on a notion of independent peers which collaborate to create aggregated logistic regression models which is used for classification. Due to there not being one single centralized classifier, there will most likely be a variance in the accuracy of the classifiers each peer hold. We want to explore options on how to reduce this variance, so that we can reduce the likelihood of one peer having a well-performing classifier while another produces poor classification results.

RQ3: Can we validate and enhance earlier research in distributed differentially private machine learning?
Our own research does not exist in a vacuum, so we intend to study and employ techniques reported in the literature, and try to validate and enhance their results.

1.2 Thesis Structure

Chapter 2: Background and Motivation

This chapter introduces some concepts in privacy-preserving data publishing, and provides a thorough background on our motivation for doing this project.

Chapter 4: Basic Theory

This chapter presents the basic theory necessary to gain an understanding of our project. Important concepts such as differential privacy and multi-party logistic regression is presented in an condensed and straightforward manner.

Chapter 4: Related Work

As our work is but a part of a bigger research effort, we use this chapter to give an overview of works similar to our own. Most of these papers we are presenting have helped shape our own project in some way, either through theory or as inspiration.

Chapter 5: Design of Experiments

Here we present the architecture and execution of our experimental procedure. We detail the usage of data sets and how they were preprocessed, as well as how we tuned the parameters involved. Toward the end of the chapter, we explain the algorithms we employ.

Chapter 6: Experiment Planning and Results

This chapter list the set of experiments we have performed in a clear and concise manner. The first part of the chapter explain each experiment in detail, and provides the parameter setup we used for each run. The last part lists the results we achieved.

Chapter 7: Analysis

All of the scientific results we have gained through our experimentation are analyzed in this chapter. The main part of the analysis will be used to explain the impact each parameter can have on the final results of our classifier.

Chapter 8: Toward a Real-World Application

This chapter is intended to be discussion on the real-world utility of our distributed privacy framework. We start by proposing a set of suitability criteria for such an application, and then explore two potential business cases where we consider our framework to be suitable for future implementation.

Chapter 9: Reflections and Conclusion

This final chapter provides a reflection on the research we have explored, as well as the some of the challenges we have encountered. We then suggest a route for future work on extending and improving our framework, and wraps up with a final conclusion.

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 as 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 [Fung et al., 2010]. 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”‘[European Parliament, 2006].

In order to remove any PII from a data set, it needs to go through a process called anonymization. This constitutes a series of manipulations with the ultimate end goal of protecting the privacy of the data set's participants.Fung et al. [2010] operates with a number of five basic operations which might be applied for this purpose. These operations are briefly 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 Fung et al. [2010])

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 that they had been properly anonymized, only to have people de-anonymize their data and breaching the privacy of the data sets' 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”[Bell and Koren, 2007]. The

released data records included an anonymized user ID, movie name, date of rating, and the user rating of that movie on a scale from 1 to 5.

Two researchers from the University of Texas,Narayanan and Shmatikov [2008], demonstrated that an adversary who knows only a little bit about an individual subscriber can easily identify this subscriber's record in the data set. Using the publicly available data set 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 data set, the authors [Narayanan and Shmatikov, 2008] 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[Sweeney, 2002]. 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[Barth-Jones, 2012].

2.2.3 New York Taxi data set

The New York City Taxi and Limousine Commission released a data set 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[Pandurangan, 2014].

Another researcher, called Anthony Tockar, wrote an article during his internship at Neustar Research where he proved that the data set also contained an inherent privacy risk to the passengers which had been riding New York Taxis. Even though there was no information in the data set 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 customers' ethnicity, relationship status, court records, and even a profile picture[Tockar, 2014].

2.3 Privacy Breach Through Linkage Attacks

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 data set with public information which contain individual identifiers, given that the probability of these two matching exceeds a selected confidence threshold.

In the literature[Bonchi and Ferrari, 2010; Fung et al., 2010], 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.

2.3.1 Record linkage

In the case of attribute linkage, some quasi-identifier value QID identifies a small number of records in the original data set, which are 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 Section 2.2.2. 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.

Sweeney [2002] 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.

2.3.2 Attribute linkage

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. [2007] 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.

2.4 Challenges in Data Privacy

Several studies have been performed to assess which privacy risks exists in fields such as mobile applications citations, health care data, and in social networks, and all of them found deficiencies in either the collection or handling of individuals' data. A study run by the European Data Protection Authorities (DPA) found that out of 1211 mobile applications surveyed, 59% caused concern with respect to pre-installation privacy communications, and that 31% requested permissions exceeding what the surveyors would expect based on their understanding of the applications functionalityDPA [2014].

The law might not necessarily be enough to sufficiently prevent the misuse of personally sensitive information, such as patient's health care data. A study performed by Yale's center for bioethics concluded that: "Law likely cannot catch up with burgeoning data collection, data aggregation, and data mining activities, nor with technological advance, let alone adequately anticipate it." Yet the author also argued that technological progress would lead to "Better alternatives to identification and de-identification; means of tracking data; [...] improved data security; and returning benefit to data originators"Kaplan [2014].

2.5 Motivation

During the fall of 2014, we performed a systematic mapping review on the subject of real-time machine learning on data streams. In this report we investigated a massive amount of research literature written in this field, and found that many state-of-the-art solutions are focusing on making their solutions both distributed and scalable[Asplund and Frøystad, 2014]. Combining this with our strong advocacy for the need for stronger privacy solutions, we came up with a research goal of creating our own distributed machine learner which could provide a strong privacy guarantee.

We hope to show that a competitive solution can be created in a distributed learning setting, which also can provide a privacy guarantee for the people who supply the data required for learning. If we are successful, our research can open an avenue of practical solutions where the paradigm in data mining shifts from collecting data in massive centralized databases, to a distributed approach where the data producers also become data owners.

2.5.1 Data security

This project is motivated by the aforementioned challenges and breaches of data privacy, and wish to contribute to the development of privacy-preserving technology. 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.

2.5.2 Data ownership

Addtionally, we are strongly motivated by the idea that there should be a reversal in data ownership. Currently, companies offering services to users collect the data stream generated by a user and store it centrally in a data center owned by the company. The user

has to trust that these data centers will not be breached or leaked. Furthermore, the user has to trust that the company policies or ethical standards will not change in the future and that the company or their data will not be bought by an independent third party. If data streams were instead collected in some user-controlled repository, risk of breaches would be reduced and the user would maintain full access control and monitoring. Tim Berners-Lee voiced his support for this idea at the IP EXPO in 2014: ”I would like us to build a world in which I have control of my data. I can sell it to you and we can negotiate a price, but more importantly I will have legal ownership of all the data about me”[Curtis, 2014]. He also brought up another compelling reason to ensure that users retain the data they produce:

> ”In general if you put together all that data, from my wearable, my house, from other companies like the credit card company and the banks, from all the social networks, I can give my computer a good view of my life, and I can use that. That information is more valuable to me than it is to the cloud” [Hern, 2014]

A user will have multiple applications that gather information from their daily life, such as exercise, social and office applications. While each of these data streams on their own can be useful for the companies that collect them, they can have even more powerful uses when put together to give a more complete context. Instead of each company pulling user data to their data centers, users could push data stream to their personal storage, and offer it to a company for a negotiable price. The user then has control of who accesses the data and how, while also allowing for data analysis across completely separate applications.

2.5.3 Future legal requirements

The European Parliament is working towards new legislation that will create a set of common data protection rules for all EU member states[European Parliament, 2013]. This legislation offers right to erasure and right to portability. The matter of portability is a step in the direction of the ideas of data ownership discussed in Section 2.5.2. Perhaps most significantly, the regulation requires that all companies operating from the EU or having customers in the EU will be required to comply with. Companies that do not comply can risk being imposed periodic data protection audits or fines up to 100 million or 5% of annual worldwide turnover.

The European Parliament is not alone in looking new laws to regulate data privacy. The Council to the President, an advisory group to the US President, concluded in their 2014 report [U.S Government, 2014] that preserving privacy values would be their number one recommendation when designing a new policy framework for big data. Furthermore, the so called ”Privacy Bill of Rights” outlined by the Obama administration in 2012 is moving forward, and a new discussion draft was published in 2015[U.S Government, 2015]. Among the requirements put forward in this bill is transparency about how data is used, the degree of control a person has over how their data is used.

Chapter 3

Basic Theory

Common to all the attack vectors described in Section 2.3 is that the attacker rely on background knowledge, often also called auxiliary information, to perform their linkage attacks. Protecting a database against this threat has long been a major challenge in database design. Already back in 1977 Tore Dalenius [1977] defined a desideratum for data privacy which states:

> Access to the published data should not enable the adversary to learn anything extra about target victim compared to no access to the database, even with the presence of any adversarys background knowledge obtained from other sources.

This privacy goal was rejected by Cynthia Dwork, who showed the general impossibility of Dalenius' goal due to the existence of auxiliary information. Instead she chose to formulate a probabilistic privacy goal, which places an upper bound on how much the risk of privacy breach can increase by participating in a database.

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.” [Dwork and Roth, 2013] 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 [Dwork and Roth, 2013], 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. The distance of two data sets, $d(D_1, D_2)$, 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 M gives ϵ-differential privacy if for all data sets D_1 and D_2 where $d(D_1, D_2) = 1$, and all $S \subseteq Range(M)$,

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

That is, the presence or absence of a particular record should not affect the probability of any given output of $M(D)$ by more than some multiplicative factor.

Informally, the presence or absence of a single record in a database should not have a noticeable impact on the output of any queries sent to it. Though the existence of the database itself might allow attackers to learn information about a person, opting out of the database will not significantly help reduce the risk of information disclosure. Conversely, participating in the database does not significantly increase the risk of disclosure either, thus fulfilling Dworks promise quoted in the beginning of Section 3.1.

Privacy preserving data analysis platforms such as PINQ[McSherry, 2009], Airavat[Roy et al., 2010] and Fuzz[Haeberlen et al., 2011] have all implemented features such as privacy budgeting and noise mechanisms to compute useful queries while fulfilling Equation 3.1.

3.1.2 Privacy budget

The quotient $\frac{Pr[M(D_1) \in S]}{Pr[M(D_2) \in S]}$ measures the extent to which an attacker can ascertain the difference between the two data sets[Abowd and Vilhuber, 2008]. Sarathy and Muralidhar [2011] calls this ratio the "knowledge gain ratio". 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 data set over the other.

Privacy budgeting was introduced to limit the amount of information a data analyst can obtain about any individual with data records in the data set. 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.

Defining and depleting a privacy budget is possible due to the sequential composition property of ϵ-differentially private mechanisms, as shown by McSherry [2009]. Given N mechanisms M_i that offer $\frac{\epsilon}{N}$-differential privacy, applying each mechanism M_i in sequence offers ϵ-differentially privacy.

3.1.3 Noise mechanisms

Given a target function f to compute on a database D, it is necessary to design a randomized function M which fulfills Equation 3.1 while yielding a useful approximation to the true f. This randomized function M can be created by adding noise to the computation of f. 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 Laplace distribution centered around zero has only one parameter, its scale b, and this is proportional to its standard deviation.

$$Lap(x|b) = \frac{1}{2b} exp(-\frac{|x|}{b}) \tag{3.2}$$

When using the Laplace mechanism it is necessary to choose a suitable value for the parameter b. Increasing values of b results in increased noise variance. The scale of b is naturally dependent on the privacy parameter ϵ, and also on the effect the presence or absence of a single record can have on the output of function f. This risk is called the sensitivity of the function, and is defined mathematically as:

$$\Delta f = \max_{D_1, D_2} ||f(D_1) - f(D_2)||_1 \tag{3.3}$$

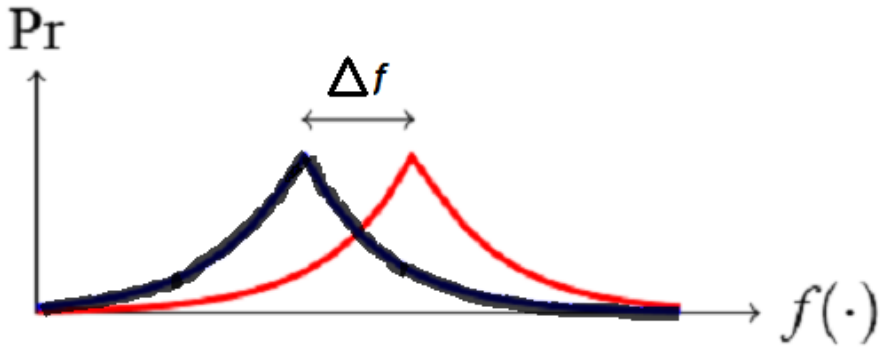

Figure 3.1: Probability distributions of the Laplace mechanism for two neighboring databases. (Adapted from [Hsu et al., 2014])

This equation states that the sensitivity Δf is the maximum difference in the values that the function f may take on any pair of databases that differ on only one row. Dwork proved that adding a noise drawn from $Lap(\Delta f/\epsilon)$ to a query, ϵ-differential privacy is guaranteed[Dwork and Roth, 2013].

Exponential mechanism

The exponential mechanism proposed by McSherry and Talwar [2007] 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 exponential 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 data set 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

$$S(u_f) = \max_{d(D_1,D_2)=1, a\in A} ||u_f(D_1, a) - u_f(D_2, a)|| \tag{3.4}$$

‘

3.2 Multi-Party Logistic Regression With Differential Privacy

3.2.1 Logistic regression

The logistic regression model is

$$p(y = 1|\mathbf{x}, \theta) = \frac{1}{1 + exp(-\theta^T \mathbf{x})} \tag{3.5}$$

where θ is the parameter vector we wish to learn. It can be used for predicting the probability of a binary outcome or binary classification by setting a classification threshold. Given a training set we choose the θ with the largest likelihood, the maximum likelihood estimator (MLE)[Elkan, 2014]. The likelihood of parameter θ is

$$\prod_{i=1}^{m} p(y_i|x_i, \theta) \tag{3.6}$$

For convenience, the point of maximum log-likelihood is used instead. Since the log function is monotonically increasing, the maximum likelihood estimator and the maximum log conditional likelihood estimator is the same, assuming labels y_i are independent of each other when conditioned on their respective feature vectors x_i. We then select

$$\theta_{MLE} = \arg\max_{\theta} [\sum_{i=1}^{m} \log p(y_i|x_i, \theta)] - \lambda \|\theta\|_2^2 \tag{3.7}$$

where the term $\lambda\|\theta\|_2^2$ is a regularization term to restrict the magnitude of θ and avoid overfitting the training data. Using $\lambda > 0$ gives a regularized estimate of w which often has superior generalization performance, especially when the dimensionality is high (Nigam et al., 1999).

3.2.2 Stochastic gradient descent

We find the point of maximum log-likelihood by mini-batch stochastic gradient descent (SGD) of the regularized objective. In normal batch gradient descent, the gradient is computed by computing an error sum over the full data set for each gradient step. This can be very time consuming for larger data sets. In single instance SGD, each gradient descent step is calculated using only a single training instance. This gives noisy gradient steps, but has the advantage of being able to converge without doing a full pass through the data set. If convergence is not achieved after a single pass, SGD can do multiple passes over the data set if needed. This makes SGD more adaptable to varying data set sizes than full batch gradient descent.

Mini-batch SGD is a trade-off between the basic batch gradient descent and single instance SGD[Cotter et al., 2011]. The data set is divided into $|D|/b$ batches of size b, and a gradient step is taken for each batch. Given mini-batch SGD, the rule for updating each dimension j of θ given a single batch of size b becomes

Update rule

$$\theta_j^{t+1} = \theta_j^t + \alpha[\sum_{i=1}^{b}(y_i - p(y_i|x_i,\theta))x_{ij} - 2\lambda\|\theta_j^t\|] \tag{3.8}$$

As in regular SGD, multiple passes over the data set are performed as necessary. A single, full pass over the data set is called an epoch. As stated by Cotter et al. [2011], an adaptive learning rate that decreases sufficiently over time is necessary for existing theoretical proofs of SGD convergence. We use the adaptive learning rate used in the SGD implementation by Bottou [2011], which relies on results in [Xu, 2011].

$$\alpha = \frac{\gamma}{1 + \gamma\lambda t} \tag{3.9}$$

where λ is the regularization constant, t is the current epoch and γ is a parameter used to tune the adaptive learning rate. Note that this means that we do not need to search for suitable values of the learning rate α - instead, we must identify a suitable γ.

Convergence criteria

While we have defined a maximum number of epochs for the stochastic gradient descent, convergence to a near-optimal parameter vector might occur well before this limit is reached. To save time, we used a convergence criteria to stop optimization when the improvement in the objective function is sufficiently small. The optimization terminates when the change in the objective function after a gradient step is less than 0.01.

3.2.3 Sensitivity of logistic regression aggregation mechanism

In order to build logistic regression models in a privacy-preserving manner, it is necessary to determine the sensitivity of the output model. Chaudhuri and Monteleoni [2009] showed that the sensitivity of logistic regression is at most

$$\frac{2}{n\lambda} \tag{3.10}$$

where n is the size of the training set and λ is the regularization parameter used in model training.

This solves only the case of training a privacy-preserving logistic regression model on local data. Our research goal involves any number of peers cooperating to build useful models without compromising the privacy of their local data. Pathak et al. [2010] proposed an approach where locally trained logistic regression classifiers are aggregated by averaging. Secrecy is achieved by using an homomorphic encryption method to compute the aggregate classifier, ensuring that local data is not shared while allowing a differentially private model to be published. This encrypted computation method is presented in more detail in Section 3.6. It is important to note that the approach of Pathak et al. [2010] assumes that the participants are honest-but-curious. This assumption means that participants will follow the established protocol, but will read any information that is somehow available to it. Their method is not robust against malicious sabotage.

When aggregating K locally trained models, their approach computes the final model

$$\boldsymbol{\theta} = \frac{1}{K}\sum_{j=1}^{K}\boldsymbol{\theta_j} + \boldsymbol{\eta} \tag{3.11}$$

where $\boldsymbol{\eta}$ is a noise vector that guarantees ϵ-differential privacy. This noise vector is drawn from the Laplace distribution with parameter $\frac{2}{n_j\epsilon\lambda}$, where n_j is the size of the smallest data set used in training of the K models. This means that they use a bound on output sensitivity of

$$\Delta\hat{\boldsymbol{\theta}^{\boldsymbol{s}}} = \frac{2}{n_j\lambda} \tag{3.12}$$

which is the same as in Equation 3.10, except that the lowest n_j is used. Since the lowest n_j corresponds to the highest noise variance, this gives protection to all the participants regardless of the size of their data set.

It is important to note that this does not offer full protection to participants. It only offers differential privacy guarantee for individual records in their data set. This means that aggregate information about a participants data set will be incorporated in the learned models. Sufficient knowledge about the other $d_{i\neq k}$ data sets would allow a third party to learn information about the user. For example, an individual might not want insurance companies to know about the averages of features in their biometric records. The current system would not help with that concern - it only protects against specific knowledge about individual biometric records.

As stated by Dwork, groups of records can be protected Dwork [2006]. Dwork points out that the end goal of differential privacy is allow learning aggregate information in a way that privacy. By this she means that it is inherent in the task at hand that aggregate information must be preserved somehow. While this point stands, we suggest an approach in Section 9.6 that could allow full protection of even the aggregate information of users in our setting, while allowing model training on their data. In this project, we only protect the individual records held by each user.

3.3 Ensemble Learning

In ensemble learning, the predictions of individually trained models are combined to form a final prediction[Opitz and Maclin, 1999]. In this project we will be using a variant of ensemble learning called bootstrap aggregating or bagging, as presented by Breiman [1996]. Breiman proved that when changes in the training set have a significant effect on the trained model, bagging can give better performance than training a single model on the learning set. Bootstrap aggregating involves creating new learning sets by sampling from the original set with replacement, and training a model on each new set produced. These models are all added to the ensemble, which then makes predictions by taking a majority vote.

We did not strictly use bagging according to its formal definition, as the bootstrap step was not used. In our approach models are instead trained on disjoint subsets of the training set, which are then published after being aggregated according to Equation 3.11. There can be many such models published, so they are added to ensembles and prediction is done in the same fashion as in bagging.

3.4 Cross-validation

We initially divided the data sets into a training set and a testing set, the latter being intended to evaluate the performance and properties of our approach. We needed to explore many different combinations and variations of our experiment during, but the test set should only be used as a final step. If the test set is used for repeated validation of different parameters, we would risk overfitting it and getting unrealistic test results.

One way to do reliable accuracy estimation is with cross-validation, which makes more efficient use of training data than creating a separate holdout set and has less bias than as shown by Kohavi [1995]. Cross-validation involves partitioning the training set into K disjoint sets. Then, for each $t \in [1, K]$ partition t is used as the test set, and the remaining partitions are combined to form the training set. Accuracy is reported as number of correctly classified instances divided by the total number of instances over all K partitions.

Kohavi recommends 10-fold stratified cross-validation. Stratified cross-validation involves ensuring that each fold has the same class distribution as the original data set. Since the data sets we tested with have thousands of records and close to uniform class distribution, we concluded that stratified folds was not necessary. Our experiments were evaluated with 10-fold cross validation with each fold being a random, disjoint subset of the training set.

3.5 Programming Frameworks Used

3.5.1 JADE

To minimize the risk of errors we wanted to implement the experiment in such a way that it was easy to reason about the behavior of the components and identify mistakes. Since the core of our experiment involves peers communicating and cooperating to create predictive models, we decided an agent-based model was suitable.

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 has 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.

Note that while all containers in a single platform must register with the Main container in that platform, multiple Jade platforms can be instantiated separately and communicate with each other, allowing for scalability of Jade deployments.

Details of our implementation can be found in Section 5.2.

3.6 Homomorphic Encryption

Homomorphic encryption is an encryption scheme which allows computations to be carried out on ciphertext, meaning plaintext that has been encrypted using an algorithm and a public key. The result of the computations is also encrypted, and can be deciphered back to plaintext using a private key. This has long been considered crypthography's holy grail [Micciancio, 2010], as this would allowing operating on encrypted text without knowing the decryption key. For example, given ciphertexts $C_1 = Enc(Data1)$ and $C_2 = Enc(Data2)$, 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 et al. [1978], but it wasn't

realized until 2009 when Craig Gentry published a doctoral thesis where he proved that he had constructed a fully homomorphic schemeGentry [2009]. 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[Schneier, 2009].”

Chapter 4

Related Work

In this chapter we will present some of the existing work related to our thesis and research goals. The papers referenced in this chapter are naturally also related to this thesis' constituent papers. We have chosen to mention papers that are related to one or more of the major theoretical framework we have employed. This chapter will therefore be structured in three sections: the first introduces both the main theoretical contributions to the concept of differential privacy, and a survey of the research field. The second section explores some centralized approaches to machine learning with differential privacy guarantees. The third and last section explore and compare works that have employed a distributed approach similar to our own framework.

4.1 Differential Privacy

As mentioned in Chapter 3, differential privacy was defined by Cynthia Dwork in her seminal work published in 2006 [Dwork, 2006]. This paper lay the mathematical foundations for the privacy guarantee we employ in our work. Dwork later expounded on her work in the book Algorithmic Foundations of Differential Privacy[Dwork and Roth, 2013], which we used as a main piece of reference when gathering knowledge in the early phases of this thesis project.

Other defining works include the two papers written by Frank McSherry. His work on mechanism design [McSherry and Talwar, 2007] allowed for the expansion of Dwork's Laplacian mechanism design, by providing the theoretical analysis that other mechanisms could satisfy the same guarantee (see section 3.1.3 for more). His paper [McSherry, 2009] on the design and implementation of PINQ were hugely influential in the early phases of our project, as we could use the paper and the publicly available code for reference when implementing our own design.

Although our own work does not include an extensive literary survey on differential privacy, we have leaned extensively on the recent survey performed by Ji et al. [2014b]. This work review the current differential privacy research on various forms of machine learning problems, both supervised and unsupervised. Their conclusion that adding noise

to the target function a single time was much preferable to adding noise multiple times during the training process, helped give us direction during the early phases of our project.

4.2 Centralized Approaches

In the years following the release of Dwork's seminal paper there have been a steadily increasing amount of publications, reaching a peak in 2013 with 141 papers published and indexed by the Scopus scientific database. Some of these works have focused on the same area as us, namely classification using logistic regression, but instead opted to focus on a centralized approach. Mentioned below are the work which have either influenced our own, or perfomed similar experiments.

Chaudhuri and Monteloni designed a logistic regression algorithm which guaranteed differential privacy in 2009. They also provided a mathematical proof for the upper bound of the sensitivity of logistic regression (see Section 3.2.3), which formed the basis of our own solution. The same authors provided a follow-up paper[Chaudhuri et al., 2011], in which they further developed a method called object perturbation to add noise to the regularized objective function. Their results which showed that objective perturbation is generally superior to output perturbation has proved very useful to the field of differential privacy.

Zhang et al. [2012] further improved upon Chaudhuri's work by creating a new functional mechanism for objective perturbation, which they tested on a set of census data by employing both linear and logistic regression.

4.3 Distributed Approaches

As our research goal states, we wish to create a framework to test the feasibility of employing a distributed, differentially private learner. One of the first works in this field was performed by Pathak et al. [2010], who proposed a privacy-preserving protocol for composing a differentially private aggregate classifier. Their protocol trained classifiers locally in different parties, and the parties would then interact with an curator through a homomorphic encryption scheme to create a perturbed aggregate classifier. We took inspiration from their protocol when we created our own ensemble classifier, extending the work of Pathak et al. in several ways. We've taken steps to ensure better scalability by adding better group forming for each of the peers, and we've added an publishing step to the aggregation mechanism which allows for the creation of an ensemble classifier in each peer. Lastly we've also performed more extensive experiments to validate the employed method, as Pathak et al. seemed to have focused more on the theoretical side of the experimentation.

Since the work of Pathak et al was presented in 2010 there have been some research published on how to create private distributed learners. One such example is the work of Boutet et al. [2013], who presented a privacy-preserving distributed collaborative filtering scheme which relied on user profile obfuscation and randomized response. Another interesting paper is the work of Zhang et al. [2014], which investigate mechanisms to sanitize location data used in recommendation system with the help of differential privacy.

Rajkumar and Agarwal [2012] presented an alternative to Pathak's method in 2012. It works in a multiparty setting by using a stochastic gradient descent based procedure to directly optimize the overall multiparty objective rather than Pathak's method of combining classifiers learned from optimizing local objectives. Their algorithm achieves a slightly weaker form of differential privacy than that of Pathak et al., but is more robust to the number of parties and the relative fractions of data owned by the different parties.

Ji et al. [2014a] recently proposed a distributed solution using logistic regression, which learned from both private and publicly available medical data sets. Their solution differ from our own as they employ a globally synchronized structure, whereas our own solution works asynchronously. They also design a mechanism which first uses public data sets to compute the gradient without any form of noise addition, and then perform a distributed logistic regression step with differential privacy.

As part of an ongoing research project Eigner et al. [2014] published a paper presenting PrivaDA, a novel design architecture for distributed differential privacy which supports a variety of perturbation mechanisms. The system leverages recent advances in secure multiparty computations and claims to generate noise in a fully distributed manner while maintaining the optimal utility. As this research is ongoing, they only demonstrate the viability of their approach through theoretical analysis, but their performance reviews seem to indicate promising future work. From what we can entail from this paper, their research seems to be very similar to our own but much more extensive in nature. They have not yet posted any detailed results from performing any machine learning experiments, so we can not compare output at this time. It could definitely be interesting for future work to create some form of collaboration with this team.

Chapter 5

Design of Experiments

5.1 Overview

As stated in Section 1.1, we wanted to create and test an architecture that facilitates fully decentralized machine learning in a way that maintains the privacy of the participants.

We consider a setting with N peers that each have a local data set. These data sets are assumed to be independently sampled for each peer, but may be sampled from the same distribution. When the system initializes, each peer trains a logistic regression model on its local data set. The data set and the trained model is private and should only be known by its owner.

While the output of each mechanism described in Section 3.2.3 is an average of the input models, produced in way that guarantees differential privacy, the computation itself must be done in a centralized manner. Doing this securely is achieved by using the protocol detailed by Pathak et al., which uses homomorphic encryption to compute the aggregate model without allowing any of the participants to know the original private model of another participant[Pathak et al., 2010]. Since this protocol requires some central computation, one of the peers is chosen at random to be the curator, responsible for acting as the central party described in the solution by Pathak et al. The other peers in the group will submit the necessary information to this curator, including their private model, in an encrypted fashion. Once the peer acting as curator has received a model from all participants, it computes the average model, adds sufficient noise to guarantee ϵ-differential privacy and publishes the final result. The scope of this publish step can vary. In our experiments we have tested one version that publishes a model to all available peers and one that only publishes the model to the peers in the group that helped create it.

Each peer holds a privacy budget, as discussed in Section 3.1.2 , that limits how many times it can be involved in a mechanism application. If the budget of a peer is depleted, it is no longer a candidate for the randomly formed aggregation groups. When the number of peers available have decreased to an amount where there are not enough peers to form a group with the size specified, the experiment terminates.

5.1.1 Limitations of current implementation

Certain parts of the system outlined in the previous section are not implemented in this project, and were replaced by black-box substitutes that simulate the required behavior. We have only done this for components that are already described and tested in other work.

The protocol created by Pathak et al. for computing aggregates securely was not implemented. In our implementation models are sent unencrypted to the peer acting as curator. While this part would need to be replaced with a full implementation of the approach by Pathak et al., the output returned by the curator is exactly the same, using the computation in Equation 3.11.

Finally, the selection of random groups is done in a non-scalable manner. A centralized actor that has full knowledge of all participating peers randomly selects groups of these peers and sends a message to each peer with the list of participants. This should be replaced by a decentralized method for the system to be scalable. How we intend to do this is discussed further in Section 9.6 on Future Work.

5.2 Architecture

We designed a distributed system using the JADE framework. The core component in this system is a PeerAgent, which represents a participant in the distributed learning setting. This agent contains what would be the local data of a person using some application. In the remaining sections, whenever we say ”peer” we are refering to the PeerAgent described here, holding a local data set and with means of communcating with other PeerAgent instances.

To form aggregate models it is necessary to select groups of peers to create each model. In our experiment, this is implemented with a singleton agent we named the GroupAgent. This agent draws random subset of size g from the set of all peers. The size and number of groups formed is given by parameters selected at the beginning of the experiment, which are described in detail in Section 5.4.

As stated in Section 5.1, the experiment terminates when there no longer exist a sufficient amount of active peers to form a group. That is, when the number of peers with sufficient budget is less than the group size parameter, the experiment is stopped. This behavior was implemented into an agent we named CompletionAgent. This agent listens to messages from the curators in the different groups. Once it has received message of the creation of the expected number of aggregated models, it initiates the final step of the experiment. This includes testing performance metrics and preparing the JADE environment for the next experiment.

5.3 Dataset and Preprocessing Steps

This section will introduce the data sets we have used for analysis. Each section introduces a data set, and details what features it contains, what we try to learn and classify, and why we chose to use it.

Each data set was divided into a training set partitioned among the peers in the system, and a testing set not used until the very end of our project. The test sets were used to verify the results observed throughout cross-validated experimentation. How the test set was selected is specified later for each data set. A feature with constant value 1.0 was appended to all data records before fitting models, to act as the intercept or bias term. When necessary we scaled the features of the data sets to 0-1 range. This is due to the proof in [Chaudhuri and Monteleoni, 2009] which states the assumption $|X_i| \leq 1$. The scaling was done by the formula

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{5.1}$$

X_{min} and X_{max} were calculated using only the training set of each data set, to avoid leaking information about the test set into the training set. The test sets were then scaled in the same manner using the same X_{min} and X_{max} as calculated on the training set.

This is a potential source of unrealistic information leakage. In a real implementation, globally determining scaling on the training set would be difficult. Instead, it might be necessary to perform this scaling locally at each peer. This means that the scaling constants X_{min} and X_{max} would be at least somewhat different for each peer, and would only be calculated on their particular partition of the training set. We performed the scaling globally on the training set because we believe any effect from this to be small. In particular, we believe it does not detract from the conclusions we make that are the most relevant to our research question, which relates to usefulness of model aggregation and any options we identify to reduce standard deviation across the peer population.

5.3.1 Spambase

The Spambase data set Hopkins et al. [1999] was used as a baseline training set. This data set is publicly available from the UCI machine learning directory, and contains 57 input attributes of continuous format which serves as input features for spam detection and 1 target attribute in discrete format which represents the class.

We chose this data set as it is a popular data set to analyze the performance of binary classifiers, so that we could compare the results of other logistic regression classifiers against our own. While this data set might not seem like the ideal choice for testing a differentially private classifier due to its lack of grouped personal information, we argue that it still fits well for the purpose of demonstration. In a spam-classifying system based on our distributed model, a logistic regression model can be built by training it locally in each user's personal mail folder and then aggregated into an ensemble. That way you can build a diverse spam-classifier without the users having to give up their personal email to a centralized database.

We randomly partitioned this data set into a training set and a test set, with 80% of the records in the training set and the remaining 20% in the test set.

5.3.2 Adult

As we needed a larger data set to enable us to scale the amounts of peers in the experiments, we chose to use the Adult data set from the UCI machine learning repository. It consists of personal information records extracted from the US census database and the task is to predict whether a given person has an annual income over or under $50,000. The original Adult data set has six continuous and eight categorical features. We downloaded the data set pre-processed by Platt [1999], where the continuous features are discretized into quintiles, and each quintile is represented by a binary feature. Each categorical feature is converted to as many binary features as its cardinality. The data set contains 32,561 training and 16,281 test instances each with 14 features.

This data set is very popular to use in various machine learning experiments, ranking second on the download list for the UCI machine learning repository. This data set has also been used before in research in privacy research, most notably in the work by Pathak et al. [2010]. This has given us some baselines to compare our own results against, which can be found in Chapter 7

5.3.3 SUSY

The SUSY dataset [Whiteson, 2014] was produced as a benchmark classification task where the goal is to distinguish between a signal process which produces supersymmetric particles and a background process which does not. This data set is available from the UCI machine learning repository, where the data set was preprocessed to extract the 18 high level features. It contains exactly 5,000,000 data records, where 10% is designated as a test set. We did not use this particular test set. We down-sampled the data set to 100,000 records, and generated a training set from 80% of this subset and a test set from 20% of this subset.

We employed this data set as it had a sufficient number of records, and this allowed us to run experiments where the peers had the necessary amount of data available. Since it was already preprocessed, it was easy to put to use, which also was a big motivating factor when we decided to employ it. The classification task is however not of much value for our project, so we mostly used this data set to confirm the results we saw from the other two data sets.

5.4 Parameter Description

The number of peers, P, specifies how many different peers participate in the experiment, and necessarily the number of partitions of the training data sets. The training set is divided into P parts of equal size.

Aggregate models are created from local models at each peer through an aggregation process that is performed one or more times with subsets of peers. The parameter g specifies how many peers will participate in a single model aggregation. Since each peer has a unique subset of data, this parameter determines how many partitions of the training set contribute to the published aggregate models. These data partitions are not directly used in

training published models, but rather indirectly through the aggregation of models trained locally on each partition.

The privacy parameter ϵ determines the level of privacy required for each peer and its set of data. Note that this parameter does not apply to the original training set as a whole; each peer has its own private database, which is protected by ϵ-differential privacy.

Each peer will get n number of records. This parameter will obviously have a significant effect on how well the local model each peer creates is able to generalize to new data instances.

Each peer trains a local logistic regression classifier on its data partition. This requires selection of an adaptive learning rate parameter α, a regularization constant λ and a maximum number of epochs of stochastic gradient descent. We tuned α by running 3-fold cross-validation where each peer fits its local model to identify the best γ by powers of 10 in the range $[10^{-2}, 10^{2}]$. When using SGD with adaptive learning rate, this range proved sufficient to find classifiers that were as good as our best baseline throughout experimentation. 3-fold cross-validation was chosen because of time constraints on the project. Each experiment in its entirety is tested with 10-fold cross-validation, so it was necessary to reduce local model training time in order to run in a reasonably short time on a single computer.

In normal data mining applications the regularization λ would be tuned in this manner as well, but in our case the sensitivity of the aggregation mechanism depends on λ, as seen in Equation 3.12. This means that the peers will have to communicate to either agree on a regularization level or to determine the smallest regularization constant to identify the worst case noise level. In our experiments we chose a global regularization level, which was used by all peers. We identified the best λ by testing a coarse grid of powers of 2 in the range $[2^{-8}, 2^{8}]$ whenever we tested with a new data set or the number of records owned by each peer changed. If several values of λ presented similar prediction performance results, we chose the highest value. This is motivated by both a desire to have each model generalize as well as possible to future data instances, and the fact that lower values of regularization increases the noise added in model aggregation.

When we wanted to test a new number of records per peer R or the privacy level ϵ, we re-tuned λ to ensure that it had the optimal value. This is necessary, since a lower number of records can mean that higher regularization is necessary. Additionally, a lower value of ϵ increases the variance of noise added to the aggregated models. This effect can be mitigated by increasing regularization, though the model will be unable to fit well to data if the regularization is too strong.

The manner in which we choose λ should be considered a bit of an optimistic approach. In reality, global cross-validation to determine best regularization is not practical and would create challenges with the privacy guarantees. Since we mostly experimented with less than 100 peers, we believe our global cross-validation is reasonable close to an approach where peers cooperate to pick the strongest possible λ with acceptable performance on average.

Finally, the parameter ϵ can be divided across several applications of the aggregation mechanism, as described in Section 3.1.2. This was achieved with a per-aggregation parameter ϵ_A. Each data partition can participate in the aggregation mechanism n times, where $n \times \epsilon_A \leq \epsilon$.

5.5 Validation

The test sets were created as described in Section 5.3. These test sets were set aside and could not be used when tuning and evaluating our solution. In order to explore the effects of the various parameters we used cross-validation with number of folds $n = 10$. For a given combination of parameters, performance metrics were measured as their average across ten repetitions. In repetition i, data fold i was used as validation set and the remaining $n - 1$ data folds were combined to form the test set.

Following the approach outlined in Section 5.4, we established suitable ranges of logistic regression hyperparameters, before proceeding to testing with different levels of privacy, peer numbers, group sizes and record numbers. Once we felt confident that we had established a set of experiments that could answer our research questions, we started performing the actual experiments. All numbers reported in Chapter 7 are mean measurements on the test set across ten full executions of the experiment, with the training set being randomized before each iteration.

5.6 Experiment Execution

For each experiment, we selected a single parameter for exploration from the set of parameters specified in Section 5.4. We wanted to explore the results of the system as the parameter took a range of values. To start, we specified a range of fixed, scalar values for each of the parameters. For example, one of the experiments we performed was intended to test the effect of varying number of peers participating in the system. One possible parameter combination to test this could then be

$$P = [10, 20, 30, 40, 50], g = 10, R = 500, \epsilon = 0.1, \epsilon_A = 0.1, \lambda = 1.0$$

We then perform a full simulation with p peers is repeated and evaluated 10 times, either by cross-validation or test set, for each p in the specified parameter set P.

5.6.1 The execution of a single experiment

As previously stated, each set of parameters was tested 10 times. For each of these iterations, there is a set of data instances used to train logistic regression models, and a set of data instances used to quantify the predictive performance of the peers after the number of aggregations available given a particular combination of P, g, ϵ and ϵ_A have occurred. In the following paragraphs we provide more details on the steps we used to perform our experiments given a set of training and testing instances.

Instantiation

First, the data is shuffled. Shuffling is performed since the manner of which the data is distributed among the peers, can turn out to have a significant effect on the quality of their local model. If each peer was given thousands of records these partitions would likely be fairly uniform, and shuffling the training data would be less important. However, since we wanted to experiment with data quantities below 100 instances per peer, we expected

significant variance in the trained models. The data is partitioned into P partitions of equal size. P instances of PeerAgent are created, and one partition of training instances is given to each peer.

Fitting local classifiers

Once the environment is set up, all agents are registered and the peers have been given their data partition, the peers fit a logistic regression model to the data they have been given. The model fitting is done by mini-batch SGD as described in Section 3.2.2. Each peer runs 3-fold cross-validation over a single epoch to determine the optimal α, and picks the α which has the lowest average prediction error on average. When α has been determined, the logistic regression model is fitted to the peers local data, with the chosen α and the λ specified for that particular experiment. The fitting is done by running mini-batch SGD for a maximum of 100 epochs.

Application of aggregation mechanism

After all peers have fitted their local classifiers, the aggregation phase begins. The steps of this phase is shown in Algorithm 2. The GroupAgent has a list of the peers that still permit their locally trained classifiers to take part in producing a published, perturbed model. While there still are enough peers to form a set of size g, the GroupAgent randomly selects a subset of these peers. Among this subset, it randomly picks one peer to act as the curator, responsible for performing the steps that result in an aggregated model that can be published with a differential privacy guarantee. It then sends a message to each of the selected peers.

Once a PeerAgent is notified that it has been selected to be the curator of a group, it starts waiting for messages from the other peers in the set. The other $g - 1$ peers that are selected to be contributors will send their local classifiers to the curator as soon as they are notified by the GroupAgent. When the peer acting as curator has received all the $g - 1$ models, it applies the aggregation mechanism given in Algorithm 1. It includes its own local classifier in this step, which produces a final aggregate model from the g other models. This aggregation mechanism computes the same output that follows from the approach of Pathak et al. [2010], but does not include the homomorphic encryption steps that ensure secrecy.

As presented in Section 3.2.3, the sensitivity of logistic regression depends on the sizes of the data sets used to train the models. Specifically, the mechanism needs to know the size of the smallest training set in order to guarantee differential privacy. It is important to note that the method we are testing assumes honest-but-curious participants, as assumed

by Pathak et al. [2010].

Input: ϵ - privacy parameter;
$\boldsymbol{\theta}$ - set of models trained by participating peers;
N - set of peer training set sizes;
λ - regularization level used when training each model in $\boldsymbol{\theta}$;
Output: Perturbed aggregate of the models in M
$n_{min} \leftarrow min(N)$;
$\eta \leftarrow Laplace(0; \frac{2}{n_{min}\epsilon\lambda})$;
$model_{agg} \leftarrow \frac{1}{K}\sum_{j=1}^{|\boldsymbol{\theta}|}\theta_j + \eta$;
return $model_{agg}$

Algorithm 1: ϵ-differentially private aggregation mechanism

Input: P - the set of peers;
ϵ - privacy parameter;
ϵ_A - privacy level of a mechanism application;
A - the ϵ_A-differentially private aggregation mechanism;
g - number of peers in a single mechanism application
for *peer* $\in$ *P* **do**
  $budget_{peer} \leftarrow \epsilon$;
end
while $|P| \geq g$ **do**
  $group \leftarrow randomSample(P, g)$;
  $model_{agg} \leftarrow A(group)$;
  for *peer* $\in$ *group* **do**
    $budget_{peer} \leftarrow budget_{peer} - \epsilon_A$;
    if $budget_{peer} < \epsilon_A$ **then**
      $P \leftarrow P \setminus peer$;
    end
  end
  $publish(P, model_{agg})$
end

Algorithm 2: Our application of the aggregation mechanism

Propagation of published models

Originally in our system, aggregated models were only propagated to the peers that had participated in creating that model, as can be seen in Figure 5.1. What resulted from this, especially when ϵ was set to a low level such as 0.1 or lower, was that the high amount of noise made the classifiers have a big standard deviation on their mean classification rate. This meant that the classifiers could be very accurate in some peers, classifying up towards 90% accuracy, while also being significantly worse in other peers.

We theorized that we could improve the ensemble classifier in each peer if we could propagate the aggregated models to all the peers in the network, instead of just those who had participated in making them. Our hypotheses was that this would lead to more stable classifiers with lower standard deviation, due to a smoothing effect in having more models in the ensemble classifier in each peer. This is basically the same idea as boot-

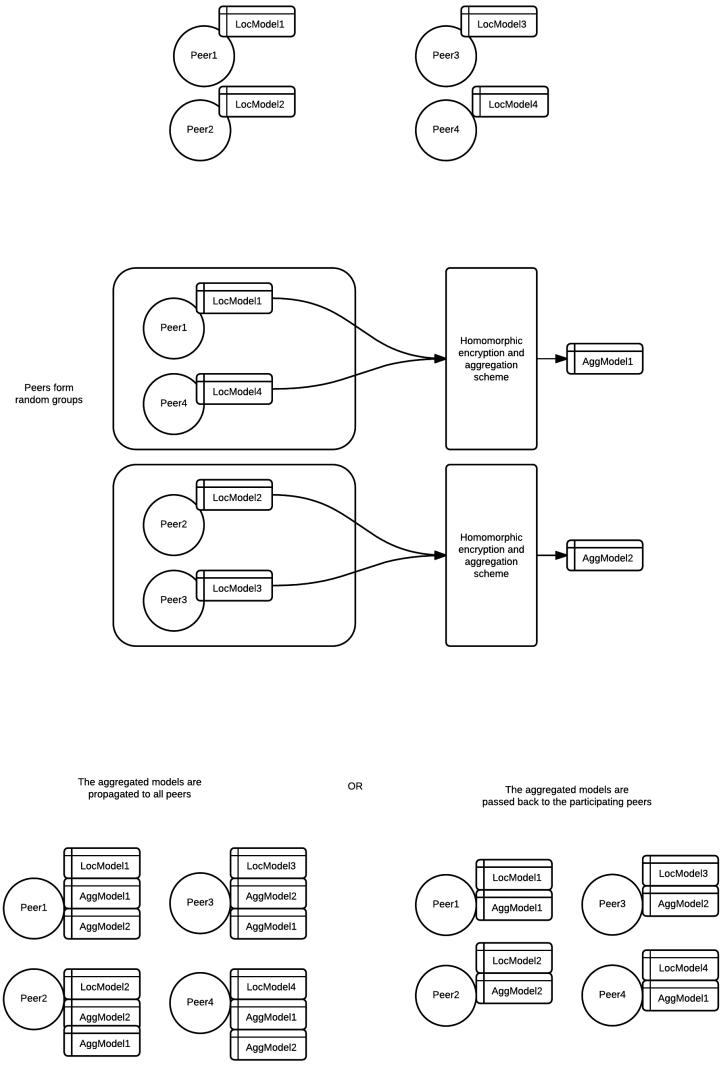

Figure 5.1: One iteration of model aggregation

strap aggregating, or bagging, which has been proven to lead to improvements in unstable procedures[Breiman, 1996].

For this reason, we decided to run experiments to compare the different possible methods of model publication. In all cases, the published models will have been perturbed with Laplacian noise to give ϵ-differential privacy. In the group publication setting, only the peers that join together to produce a perturbed model will receive the final result. In the full publication setting, all peers active in the network will receive all perturbed models.

Note that there is no selection or pruning of the ensemble classifier owned by each peer. If a peer receives a model, it will blindly add it to the ensemble. This means each peers ensemble model will grow much faster in the full publishing setting, and they will all contain essentially the same models, the only exception being the unperturbed model produced by the peer locally. We anticipated that this would lead to a reduction in ensemble model accuracy variance.

5.6.2 Reset of the experimental environment

Between each execution of the experiment the JADE environment must be prepared for the next experiment. In our initial implementation we had problems with dynamically restarting the environment, as it was too slow a method. For this reason, the experiment environment is reset by simply removing all PeerAgent instances and the GroupAgent. We still had problems with removing the CompletionAgent quickly enough to allow the experiments to execute without pause, so instead of removing this agent, we simply reset its state and reconfigure it with the next experimental setup.

Once the CompletionAgent is reset and the other agents are removed, the environment is prepared for the next experiment. If all iterations of testing have completed for the particular combination of parameters, the experiment moves on to the next set of parameters. If not, the same parameter combination is tested once more.

Chapter 6

Experiment Planning and Results

6.1 Experiment Plans

In this section we list the set of experiments we have performed in order to answer the research questions as stated in Section 1.1. In the first section, we list all the experiments and the parameter configuration we used in each experiment. The results of these experiments will be provided in section 6.2. When more than one peer was involved in the execution of an experiment, we measured and report the mean error rate of all peers.

6.1.1 Measuring error rates

The experiments in this section are intended to show the error rates that results from differentially private model aggregation compared to locally trained models. This experiment is concerned with our goal of validating previous work and how error rates differ in the centralized and the distributed, differentially private setting.

Experiment Name	**Peers**	**Data per peer**	ϵ
Centralized logistic regression	1	3000	N/A
Disjoint logistic regression	10	300	N/A
Aggregated model	10	300	1.0
Ensemble model	10	300	1.0
Aggregated model	10	300	0.1
Ensemble model	10	300	0.1

Table 6.1: Measuring error rates

In each of the experiments in Table 6.1, we first chose an optimal regularization λ in the range $[2^{-8}, 2^{8}]$ by cross validation. We then measured the classification error with the chosen λ on the test set. In all experiments, the aggregation group size was set to be equal to the number of peers, and ϵ_A was set to be equal to ϵ. This means that there could be produced at most one aggregated model, and it would be available to all the participants.

The *Centralized logistic regression* experiment was intended to establish the best achievable performance with our implementation of logistic regression trained by SGD. It corresponds to the traditional non-private and centralized training of classification models. Note that the results of these experiments may not be state-of-the-art for each data set, since we have not performed advanced feature extraction and selection. This is acceptable, since the intention of these experiments was to establish a baseline that we can compare with when producing models that are formed in a private and decentralized manner. We were mainly interested in observable difference in performance metrics when we compared our centralized solution and distributed, differentially private solution.

The *Disjoint logistic regression* experiment considers a situation where the participating peers have a subset of the data and locally train one model each. Each peer fits a model and makes predictions independently.

The *Aggregated model* experiment lets the peers create an aggregate model using Pathak et al. [2010] approach, and the peers use this model only when labeling data. This means that the locally trained model is only used to produce the aggregate model and never for classification.

In the *Ensemble model* experiment the peers also produce an aggregate model, but when classifying data their local model and the aggregated model classify in an ensemble. The *Aggregated model* and *Ensemble model* experiments were run twice, with two different values of ϵ which represent a significant difference in privacy level.

6.1.2 Confirming expected effects of differential privacy

In this project we have implemented training of a logistic regression, the aggregation mechanism guaranteeing differential privacy and the communication scheme that forms groups of peers to create aggregate models. When tuning the parameters of the implementation, we expected certain changes in the measured performance based on the theoretical and experimental results of previous work. The experiments in this section are intended to be validation of both previous work and of our implementation. In particular, we expected certain effects when changing the privacy parameter ϵ and the regularization parameter λ. By confirming the expected behavior, we could increase confidence in the correctness of our implementation, while also visualizing the dynamics of differential privacy.

Changes in ϵ

The variance of the noise added when producing aggregated models increases with the parameter ϵ. To confirm this behavior, we ran experiments with all parameters fixed except for ϵ. Each peer was given 368 data records. The regularization level was chosen based on the results in Section 6.1.1.

Experiment Name	**Peers**	**Group size**	$\boldsymbol{\lambda}$	$\boldsymbol{\epsilon}$
Spam, effect of ϵ	10	10	2^{-2}	$[2^{-8}, 2^{8}]$

Table 6.2: Effects of privacy level. Adult.

All the peers in this experiment collaborate to produce one aggregated model, and the

full privacy budget is expended in the single aggregation. When the peers are tasked to label the test data, they use their local model and the aggregated model in an ensemble.

Changes in λ

As stated by Equation 3.12, increasing the regularization parameter λ will decrease the variance of noise added when aggregating. For this reason we expected that higher values of regularization should help counter the perturbing effect of lower values of ϵ. However, as the regularization value grows, the predictive performance should degrade as the models become unable to fit to the data. To confirm these effects, we tested wide ranges of regularization strength with different levels of privacy. Each peer was given 368 data records.

Experiment Name	**Peers**	**Group size**	$\boldsymbol{\lambda}$	$\boldsymbol{\epsilon}$
Spam, observing λ, no privacy	10	10	$[2^{-8}, 2^{3}]$	2^{10}
Spam, observing λ, common privacy	10	10	$[2^{-8}, 2^{3}]$	0.1
Spam, observing λ, stronger privacy	10	10	$[2^{-8}, 2^{3}]$	0.01

Table 6.3: Effect of regularization strength

The values of ϵ for the common privacy level in Table 6.3 is chosen based on Dwork [2008], which suggests that 0.1 is a common value.

6.1.3 Changes in data availability

As we are testing in a decentralized setting were data quantities might be low, we wanted to compare how the local, the aggregated models, and both of them in an ensemble respond to changes in data availability. The experiment seen in Table 6.4 is concerned with exploring the way data availability affects error rate in the distributed setting. Since all other parameters except for the amount of data per peer is fixed, the overall quantity of data in the system changes for each parameter combination in this experiment. The regularization level was chosen based on the results in Section 6.1.1.

Experiment Name	**Peers**	**Data per peer**	$\boldsymbol{\lambda}$	**Type**
Spambase, data availability, disjoint	10	$10 - 360$	2^{-2}	Local
Spambase, data availability, aggregated	10	$10 - 360$	2^{-2}	Aggregated
Spambase, data availability, ensemble	10	$10 - 360$	2^{-2}	Ensemble

Table 6.4: Effect of data availability

6.1.4 Changes in number of participants

The experiment shown in Table 6.5 is an attempt to partially answer to our research question about the possible loss of accuracy in our distributed setting from the participation perspective. As we expected the error rate might be affected by the number of participating peers, we designed this experiment to give an idea of how peer prediction quality

changes when more peers are present. Except for the number of peers, all parameters were kept fixed, including the amount of data per peer. This means that as the number of peers increase, there is more data available in the system. The regularization level was chosen based on the results in Section 6.1.1.

Experiment Name	Peers	Group size	Data per peer	λ	ϵ
Adult, increasing participants	$5 - 50$	5	500	2^2	1.0

Table 6.5: Effect of number of peers

6.1.5 Peer error rate variance

With this experiment we wanted to explore the variance in the quality of models held by each peer, and see how it changes when aggregate models are introduced. In order to increase the chances that we can observe variance among peers, the amount of data owned by each peer is set at a low level of 250. The regularization level was chosen based on the results in Section 6.1.1.

This experiment was concerned with reduction of peer error rate variance and to some extent validation of previous work. Concerning the latter, we hoped to show whether or not there was benefit in differentially private model aggregation compared to each peer using a local model. Concerning the former, we wished to see if publishing aggregated models and using them in ensembles to make predictions could help reduce peer error rate variance.

Experiment Name	Peers	Data	λ	ϵ	Type
Adult, peer variance, only local	10	250	2^2	1.0	Local
Adult, peer variance, aggregated	10	250	2^2	1.0	Aggregated
Adult, peer variance, ensemble of both	10	250	2^2	1.0	Ensemble

Table 6.6: Observing peer error rate variance

6.1.6 Effect of aggregation group size and model propagation

We believed that the number of peers participating in creating aggregated models could affect the quality of the produced models. In the experiment seen in Table 6.7 we have a fixed number of peers and a fairly high level of ϵ. This is because we want to observe the value of aggregating models, which should be more apparent in a situation with lower perturbation. Since participation in a single model aggregation fully expends the privacy budget of a peer and the number of peers is fixed, a higher number of aggregate models are produced when the group size is smaller. Smaller groups result in each peer having a larger ensemble with each model being based on less data. Larger groups result in each peer having a smaller ensemble with each model being based on more data.

This experiment relates to two of our research questions - reduction of accuracy loss and peer accuracy variance in our distributed, differentially private setting. We hoped to identify group configurations that minimized both of these performance metrics.

We run this experiment with two different approaches for publishing the aggregated models, as discussed in Section 5.6.1. The regularization level was chosen based on the results in Section 6.1.1.

Experiment Name	Peers	Group size	ϵ	λ	Publication
Changing group sizes	30	$[1, 5, 10, 15, 20, 25, 30]$	1.0	2^2	Party
Changing group sizes	30	$[1, 5, 10, 15, 20, 25, 30]$	1.0	2^2	All

Table 6.7: Effect of aggregation group size

6.1.7 Value of budgeting privacy

As discussed in Section 3.1.2, it is possible to spread the usage of the privacy guarantee in a budgeted fashion. Table 6.8 shows the experiment where we wanted to explore the potential benefit of performing repeated aggregations, at the cost of lower ϵ in each aggregation.

This relates to our first research question, with considers loss in accuracy, in two ways. Firstly, creating ten aggregated models by spreading the budget and then using them in an ensemble might on its own have interesting effects on quality of predictions. Secondly, it might be necessary for peers in a distributed setting to participate in aggregations several times. This is especially true if aggregated models cannot be shared globally for privacy or practical reasons.

The regularization level was chosen based on the results in Section 6.1.1.

Experiment Name	Peers	ϵ	ϵ_A	λ
Budgeting privacy	10	0.1	$[\frac{0.1}{16}, 0.1]$	2^2

Table 6.8: Effect of budgeting privacy

6.2 Results

This section provides the results of the experiments listed in Section 6.1, which are analyzed further in Chapter 7.

6.2.1 Measuring error rate

Experiment Name	**Mean error**	**Std. dev.**	λ
Centralized logistic regression	0.162	0.0020	2^3
Disjoint logistic regression	0.172	0.0014	2^2
Aggregated model, $\epsilon = 1.0$	0.154	0.0015	2^{-1}
Ensemble model, $\epsilon = 1.0$	0.162	0.0013	2^1
Aggregated model, $\epsilon = 0.1$	0.160	0.0030	2^0
Ensemble model, $\epsilon = 0.1$	0.164	0.0013	2^2

Table 6.9: Measuring error rate: Adult

Experiment Name	**Mean error**	**Std. dev.**	λ
Centralized logistic regression	0.104	0.0059	2^{-8}
Disjoint logistic regression	0.159	0.0026	2^{-5}
Aggregated model, $\epsilon = 1.0$	0.163	0.0103	2^{-3}
Ensemble model, $\epsilon = 1.0$	0.150	0.0045	2^{-5}
Aggregated model, $\epsilon = 0.1$	0.182	0.0256	2^{-1}
Ensemble model, $\epsilon = 0.1$	0.157	0.0117	2^{-2}

Table 6.10: Measuring error rate: Spambase

6.2.2 Confirming expected effects of differential privacy

Changes in ϵ

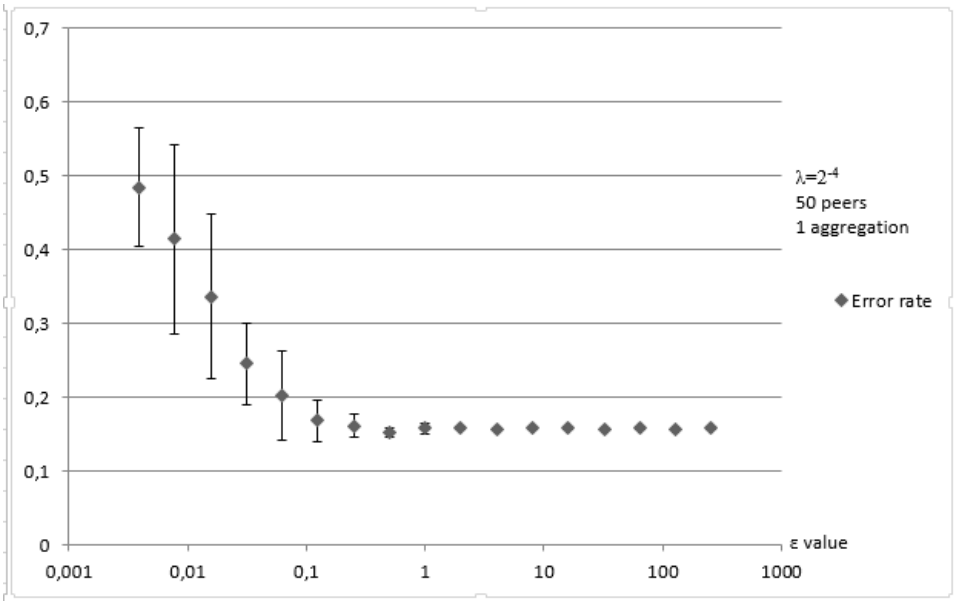

Figure 6.1: Effect of privacy level (Spambase)

Changes in λ

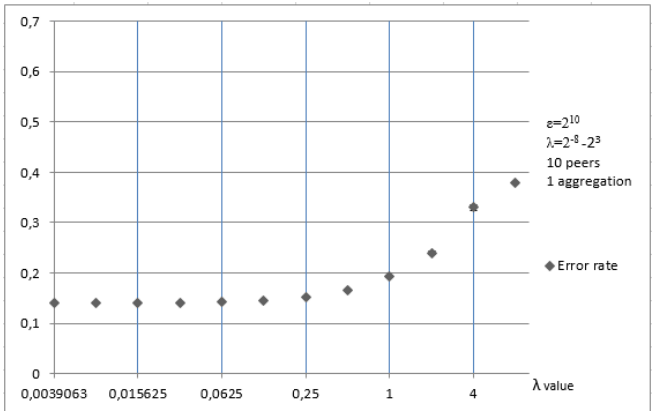

Figure 6.2: Effect of regularization, no privacy (Spambase)

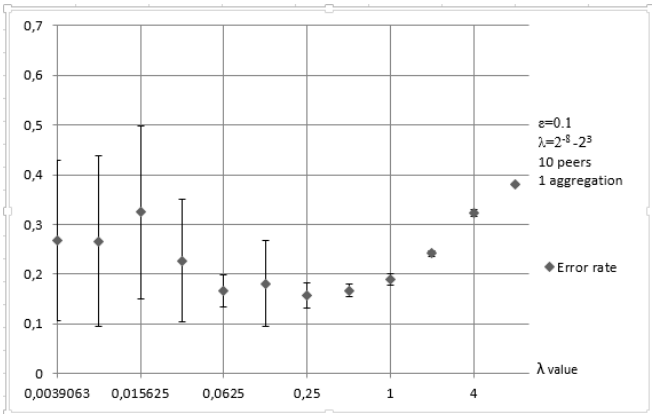

Figure 6.3: Effect of regularization, common privacy (Spambase)

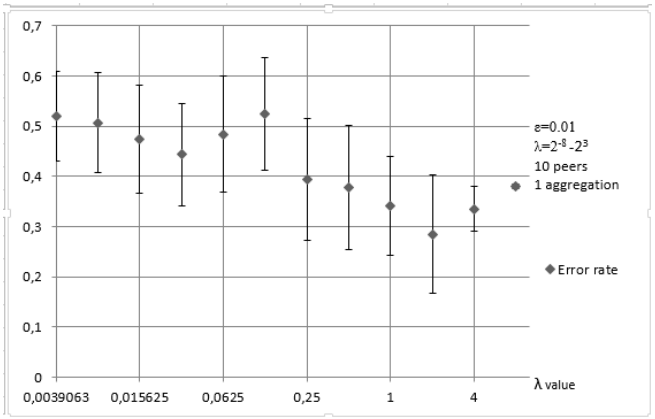

Figure 6.4: Effect of regularization, high privacy (Spambase)

6.2.3 Changes in data availability

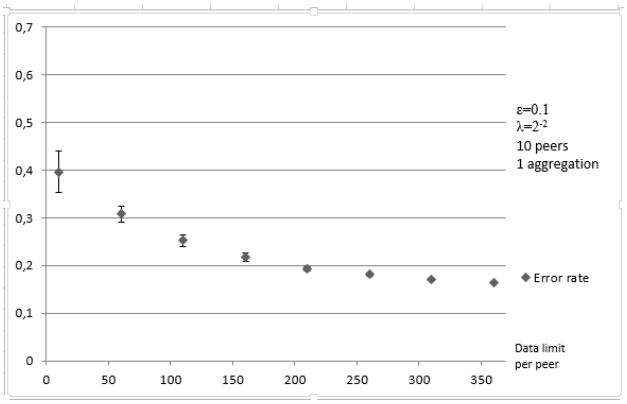

Figure 6.5: Spambase, data availability, disjoint

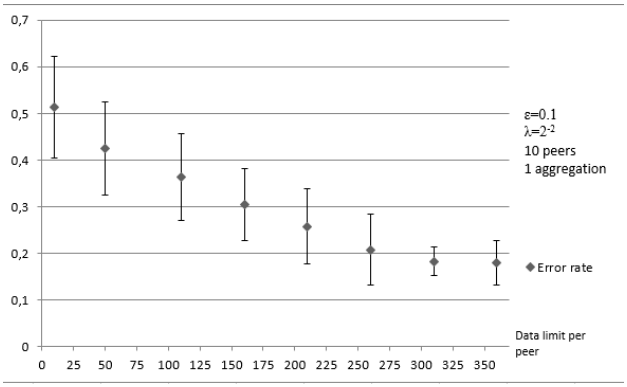

Figure 6.6: Spambase, data availability, aggregated

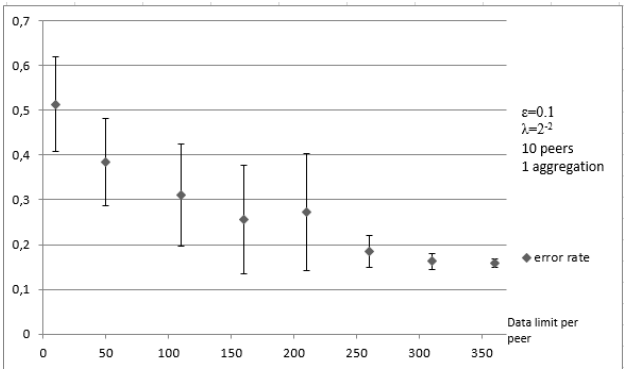

Figure 6.7: Spambase, data availability, ensemble

6.2.4 Changes in number of participants

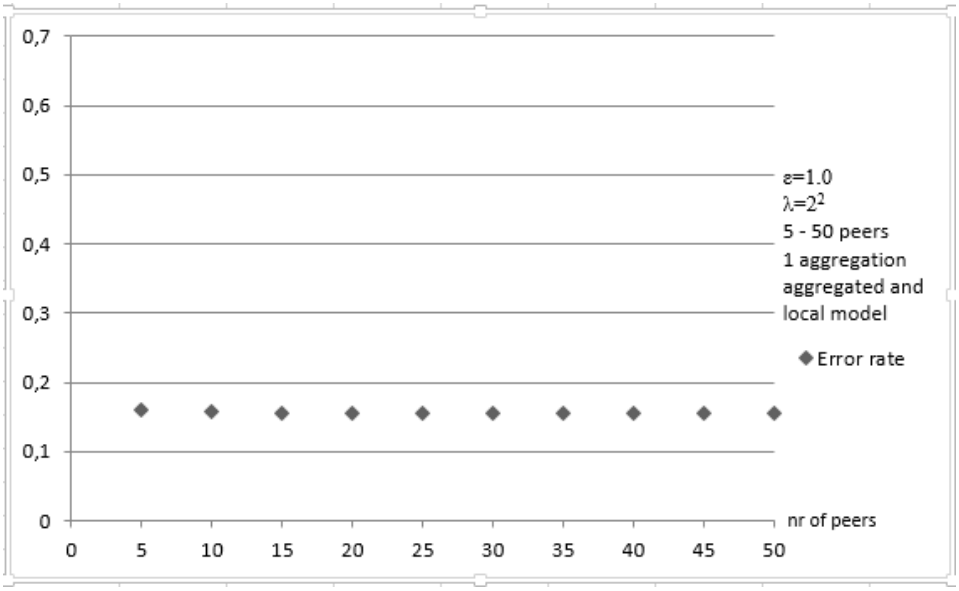

Figure 6.8: Effect of peer numbers. Adult.

6.2.5 Peer error rate variance

Model	Mean error	Peer std. dev.
Local, no privacy	0,175	0.006
Aggregated, d. privacy	0,158	0.000
Ensemble with both	0.166	0.002

Table 6.11: Variance among peers. Adult.

6.2.6 Effect of aggregation group size and model propagation

Group size	Mean error	Error std. dev.	Peer error std. dev.
1	0.174	0.0048	0.0068
5	0.168	0.0061	0.0054
10	0.171	0.0064	0.0078
15	0.172	0.0048	0.0071
20	0.170	0.0052	0.0057
25	0.168	0.0051	0.0052
30	0.168	0.0040	0.0031

Table 6.12: Effect of aggregation group size. Party-publishing. Adult.

Group size	Mean error	Error std. dev.	Peer error std. dev.
1	0.168	0.0009	0.0003
5	0.165	0.0013	0.0007
10	0.165	0.0012	0.0008
15	0.165	0.0008	0.0026
20	0.165	0.0005	0.0026
25	0.165	0.0007	0.0027
30	0.165	0.0006	0.0027

Table 6.13: Effect of aggregation group size. All-publishing. Adult.

6.2.7 Value of budgeting privacy

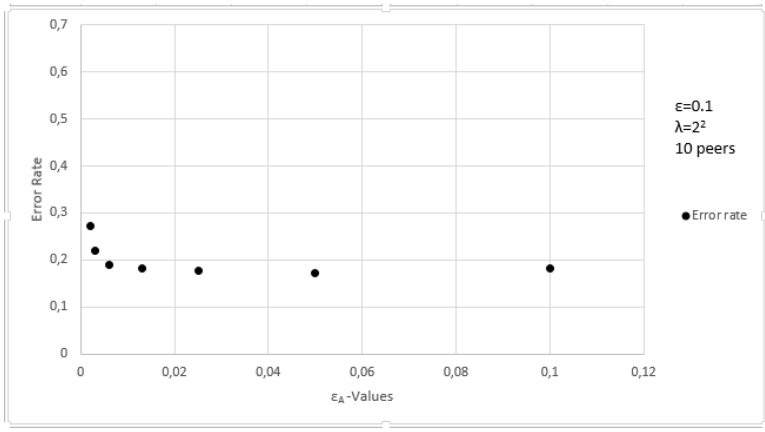

Figure 6.9: Effect of Privacy Budgeting. (Adult)

Chapter 7

Analysis

In this chapter we will give our analysis of the experimental results seen in Section 6.2.

7.1 Comparing Measured Error Rate

Experiment and author	Error Rate
Optimal Logistic regression [Caruana and Niculescu-Mizil, 2006]	0.114
Multiparty DP LogReg [Pathak et al., 2010]	0.24

Table 7.1: Table with baseline results from the Adult Dataset

Experiment and author	Error Rate
LOGICBOOST[Sharma and Arora, 2013]	0.1024
LogReg-TRIRLS[Kumar et al., 2012]	0.1389

Table 7.2: Table with baseline results from the Spambase Dataset

These two tables form the baseline for the analysis of our classification framework. In Table 7.2 we have included the results of two classifiers reported in the literature [Sharma and Arora, 2013; Kumar et al., 2012] for the Spambase data set. These are both centralized approaches to logistic regression, where both had as a research goal of trying to find an ideal classifier for spam detection. In similar fashion, we provide two baseline classification results in Table 7.1 for the Adult data set. Here we chose to provide baselines for two approaches, where the first [Caruana and Niculescu-Mizil, 2006] is an optimal result using logistic regression in a centralized manner. The second entry is the result of the differentially private system of Pathak et al. [2010], on which we've stated that we wish to improve in RQ3.

7.1.1 Adult data set

We first consider our error rate measurements on the Adult data set, seen in Table 6.9. We measured a mean error rate of 0.162 for *Centralized logistic regression*. This is not as good as the baseline result of 0.114 reported by [Caruana and Niculescu-Mizil, 2006], but still is well below the positive class rate of 0.24. In the *Disjoint logistic regression* experiment, where each peer has a smaller subset of the data and disjointly train models and classify data, we measured an error rate of 0.172. This is expected, since there is significiantly less data available to fit the models.

For *Aggregated model,* $\epsilon = 1.0$, we measured an error rate of 0.154, which is lower than our centralized result. This is very interesting, as it indicates that an aggregated model formed from models trained on data subsets of size 300 can yield lower error rates than a model trained in a centralized model on a data set of 3000. This same behavior was observed for the Susy data set, which is listed in the Appendix. It is not clear if this is a common occurence or if it is only applicable to these particular data sets. It is not observed for the Spambase data set, where the error rate jumps from 0.104 to 0.163 when comparing the centralized and the distributed, aggregation setting. A possible explanation for why we don't see the positive effect for the Spambase data set might be the scale of the regularization used. For Susy and Adult, the optimal λ was found to be 2^{-2} and 2^{-1} respectively, while it was 2^{-3} for Spambase. A lower λ increases variance of noise, so the aggregated models are more perturbed for Spambase, even though ϵ is the same for all data sets. Lastly, it is notable that the error rate on the Adult data set increased only slightly from 0.154 to 0.160 for the aggregated model when ϵ changed from 1.0 to 0.1. This indicates that a strong privacy guarantee is feasible.

For *Ensemble model,* $\epsilon = 1.0$ error rate increases slightly to 0.162, which is slightly worse than *Aggregate model*, but still not worse than the centralized model. For this particular data set, it is not clear whether it is worthwhile to include the locally trained model in an ensemble when classifying.

We should point out that Pathak et al. [2010] report the error rate of their optimal, centralized solution applied to the Adult set as 0.24. This is strange, since it is very close to the class distribution of the Adult data set. While they show that their differentially private solution approaches their centralized solution, they give no compelling reason to believe that the error rate of 0.24 is a non-trivial performance. A very similar error rate could be achieved by only predicting the negative class.

7.1.2 Spambase data set

We next consider the error rate measurements on the Spambase data set, seen in Table 6.10. We measured a mean error rate of 0.104 for *Centralized logistic regression*, which is very close to the results of [Sharma and Arora, 2013] at 0.1024.

Contrary to the results seen for the Adult data set, neither the *Aggregated model* or the *Ensemble model* is able to achieve error rates comparable to the *Centralized logistic regression* test. As discussed in the previous section, this might be due to the lower regularization required for the Spambase data set.

While the positive effects of aggregation is not seen for the Spambase data set with the configuration we have tested, it appears to support the *Ensemble model* approach. While

the error rate is 0.163 for *Aggregated model,* $\epsilon = 1.0$, it is 0.150 for *Ensemble model,* $\epsilon = 1.0$. The same pattern is seen with $\epsilon = 0.1$, where the error rate goes from 0.182 for the *Aggregate model* setting to 0.157 for the *Ensemble model* setting. We also note that a positive effect is also seen with *Ensemble model* for the Susy data set, when $\epsilon = 0.1$. It appears that using the aggregated model and the locally trained model in an ensemble can help reduce the error rate, but that this is not always the case.

7.1.3 Validating model quality

We have restricted our analysis to the error rates of our solution, though confusion matrices and ROC-curves were produced for all experiments. The confusion matrices were used to verify that the classifiers were discriminating between the two classes. While error rates are presented as mean value among all peers, we give only the confusion matrix of a single, randomly selected peer. The intention is that the error rates we report reflect the model quality on average for all participants in the system, while the confusion matrices are intended to support the claim that the reported error rates correspond to non-trivial classifiers. Confusion matrices for the Spambase data set is included in the Appendix.

		Actual	
		1	0
Predicted	1	2393	1146
	0	1453	11289

Table 7.3: Confusion Matrix: Adult. Local model only.

		Actual	
		1	0
Predicted	1	2245	888
	0	1601	11547

Table 7.4: Confusion Matrix: Adult. Aggregated, DP model only.

		Actual	
		1	0
Predicted	1	2102	895
	0	1744	11540

Table 7.5: Confusion Matrix: Adult. Ensemble of Aggregated and Local.

An important point to discuss in this context is validation at the peer level. How would a peer know if its local model or ensemble is good enough? We have validated with cross validation and tested with a held out test set, but this luxury is not available in a real setting. One option would be to design a scheme to perform cross validation at the aggregation group level.

Ideally, peers could cooperate to form a sort of test set when the system is live by sending incoming data instances to each other. This would allow the system to respond more quickly to changes in data distribution, as deterioration in prediction accuracy can be measured in a near-global fashion as soon as the changes occur. This approach is problematic, since peers may want to keep all incoming data private. The other alternative would be to have peers measure their performance on their respective incoming data only. If peers have very different incoming data distributions, this would give a less biased estimate over time, but building a good estimate could take much longer, depending on the data rate.

7.2 Importance of Epsilon

As explained in section 3.1, differential privacy works by disguising an individual's data in a data set by adding noise to their records. The amount of noise added is determined by the privacy parameter ϵ, and grows exponentially the closer the parameter gets to zero.

Figure 6.1 shows the effect of the privacy parameter ϵ in our experiment. We wanted to test the effect of varying the ϵ-value in the range from 2^{-10} to 2^{9}, especially to find out how the classifier would perform when faced with data with high amount of noise added to it. The positive class rate in the UCI Spambase data set is 0.4, so any error rate at this level is no better than a random classifier. The plot shows how sensitive output is to the values of ϵ.

Our experiments indicate that an ideal range for the ϵ-value seems to be between 0.1 and 1.0 for all datasets. If the value falls below this range, both the error rate and the error variance quickly grows and the classifier becomes unusable. Some of the earlier research by Dwork [2008] theorized that 0.01 might be suitable in analysis with a need for an extra strong privacy guarantee. In our case that is not a feasible limit, as the error rate at this privacy level is growing towards the class distribution. Other research performed in the literature Pathak et al. [2010]; Kellaris and Papadopoulos [2013]; Hsu et al. [2014] seem to agree that 0.1 is a natural lower limit.

Looking at Table 6.9, we see that the difference between the our optimal result from Adult at 0.157 with ϵ=1.0 and the high privacy result of 0.163 with ϵ=0.1, is not particularly big. The results of running a big range of epsilon over the Spambase dataset (Figure 6.1), also indicate that there is not much improvement in increasing the ϵ beyond 1.0. This is interesting, as it suggest there might exist a natural bound on the trade-off between privacy and utility. If a theorem could prove this bound of where there will be no further increase in utility by decreasing the privacy, this would serve as a big breakthrough for the field of differential privacy. To the best of our knowledge however, no such theorem has yet been found. Lee and Clifton [2011] have tried providing such a bound, but their proof is limited to a very specific scenario on a unreasonably simple example database. Nevertheless, their effort suggest that future research might bring more conclusive proof. In our case, we can only report what we have discovered after performing carefully monitored experiments.

7.3 The Importance of Data

As can be told from Figure 6.5, 6.7 and 6.5, the amount of data available for each peer is important. The more data a peer have, the greater the chance is that it will make a decent local classifier.

Figure 6.7 and 6.6 indicate how the classification performance improves as the amount of data records available to each peer grows. When each peer only have a small amount of data to create their local classifier, the classifier tends to have display terrible performance. As peers gain more data, both the performance and the variance of the classifier improves.

The reason for this improvement is two-fold. 1: A bigger sample size for the logistic regression model generally leads to better performance Peduzzi et al. [1996]. 2: The sensitivity of logistic regression (see equation 3.10) is bounded by the size of the training set. What this means is that the more data a peer have available, the less amount of noise is needed to obfuscate their logistic model.

The observation we've made is therefore thoroughly grounded in theory, and is important to highlight when discussing our main research question. Until a certain amount of data has been gathered, a system based on our distributed architecture will display very poor performance. What is interesting however, is that the amount of data needed seems to be a relatively low number. In the paper written by Pathak et al. [2010], they report that each party was given at least 3256 data records. In our experiments we found that a much smaller amount of data could still result in decent classifiers.

7.4 Importance of Regularization

7.4.1 Regularization in a low privacy setting

Figure 6.2 shows the normal effect regularization has on accuracy for the Spambase data set, by setting ϵ so high that noise is essentially nonexistent. As the regularization parameter λ grows large, the model becomes less able to fit the training data, eventually resulting in models predicting only the negative class, which constitutes 60% of the data set. This happens because the high regularization forces the parameter vector to the zero vector, resulting in uniform class probability for all samples.

On this particular data set it appears that a logistic regression model has low risk of overfitting, since the cross-validated error does not increase when the level of regularization is very low. Ignoring the effects of privacy mechanisms, this would mean that selecting some regularization parameter in the lower half of the range should offer reasonable performance. Choosing a level at the high end of this range could still be preferable, motivated by a desire to attain a model that is highly generalizable.

7.4.2 Regularization in a high privacy setting

When noise with significant variance is added to the model creation process, tuning λ will adjust noise variance as well. Equation 3.12 states that the noise variance is inversely proportional to λ. The choice of regularization then must balance the model flexibility at lower levels of λ with the decreased noise at higher levels of lambda. When epsilon is

reduced, and therefore noise variance increased, the choice of regression becomes more clearly restricted than in the noise-free situation. Based on figure Figure 6.1, we picked $\epsilon = 0.1$ to attempt to visualize this effect. At this level of privacy we see a clear decrease in performance, while the models still are making better than random predictions. Figure 6.3 shows mean error rate when ϵ is set to this level of privacy. The effect of noise is now clearly visible far into the range of regularization, only returning to comparable error rate when $\lambda = 2^{-2}$.

A further decrease in ϵ should shift the optimal regularization level even higher, eventually meeting with the point. This situation is shown in Figure 6.4, where the experiment moves into lower variance just as regularization is getting too high to support good model fit. The result is that there is no good value for λ. A situation like this could be remedied by either decreasing the level of privacy, or increasing the minimum amount of data available at the peers, both of which would decrease noise variance as given by Equation 3.12. This could be something to keep in mind if putting a differential privacy method into practice, whenever the sensitivity depends on data set size - postponing model training to a later date when more data is available can be beneficial for both model generalization and minimizing the impact of privacy.

7.4.3 Thoughts and guidelines on regularization

The fact that regularization level affects the noise variance has another interesting consequence when tackling more than one data set. Typically, different data sets tolerate varying levels of regularization. For the Spambase data set, Figure 6.2 indicates that λ should be at most 2^{-2} or 2^{-3}, while the Appendix figures for the Adult data set with same per-peer record count show near optimal regularization values as high as 2^2. Such a significant difference in regularization can make or break a model, due to the effect it has on the noise variance.

The method we are testing have a couple of notable aspects concerning selection of regularization strength. Firstly, it is necessary to take care to validate models in a way that takes into account the increased variance in trained models due to noise.

Secondly, the fact that regularization has such a significant effect on the noise variance means that truly knowing the error rate of the final model would require aggregation and noise addition. Indeed, this issue arises in both the original approach by Pathak et al. [2010] and our method. Since noise is not added until the aggregate models are produced, a realistic measurement of performance can't be performed until after the aggregation step. This means that either the privacy budget would be expended in the first iteration of cross validation, or the noise level for each aggregation would have to be reduced. In our experiment we have determined good values of λ by performing grid searches over ranges of values, but doing a grid search in a decentralized setting while maintaining privacy guarantees is non-trivial. It would be necessary to create a new protocol which includes some method of λ selection.

We believe the most practical way to do this would be by letting each peer choose the strongest possible regularization that is within some bounded distance from the optimal value found in cross-validation on their local data. The protocol by Pathak et al. [2010] briefly described in Section 3.2.3 also includes a step to secretly communicate the minimum data set size among all the participants. The peers could additionally communi-

cate their chosen λ and the curator could then use the minimal regularization level when choosing the level of noise to add.

7.5 Analysis of Peer Participation Effects

The previous paragraphs considered effects of noise variance on the ability of peers to make predictions. In this section we consider another important aspect that determines the quality of aggregated models - the number of peers that participate to create them.

Figure 6.8 shows what happens when the aggregation group size stays fixed, but different numbers of peers are available. A higher number of peers mean that more models can be created. In this experiment the aggregation group size was five, which means that just one aggregated model was produced at the low end of the range, while 10 models were produced at the high end. If there is any utility of aggregating models, it is not visible in the mean error rate for the Adult data set with this amount of records. We suspected that an effect would be stronger when the peers have less data, since aggregation should allow the peers to benefit from each other data. We tested with different amounts of available data as low as 75 instances per peer, but we made the same observations in these experiments. The charts showing other levels of data availability are included in the Appendix.

One possible explanation for the small changes in error rate might be the uniformity of data distribution. In our current experimental setup, the data is partitioned randomly between the peers, which means that the peers will have same number of samples from the same data distribution. In reality, users might have very different numbers of samples with different distributions. This is discussed further in Section 9.6 on Future Work. On the other hand, these experiments don't show any negative effects of predicting with an ensemble of aggregated models. While we expect that there are situations where ensemble prediction could be detrimental, such issues could be mitigated by actively removing less useful models from the ensemble as it grows.

7.6 Analysis of Model Variance

While it is an important goal to minimize error rates, the variance in performance across peers might be just as important. It could be better that all participants achieve sufficient predictive performance rather than having some peers with fantastic performance while others make useless predictions.

We see from the results in Figure 6.11 that variance is smaller when the aggregated models are used, but the difference is not very big, as the variance is small in both cases. We believe the weakness of the effect might be for the same reason as the small amount of change in error rate seen in Section 7.5. The data is distributed uniformly to the peers, which means that they will get fairly similar models as long as they have sample data to get close to what a model trained on the full data set would look like. Given a data set partitioned into less uniform subsets, the model variance might increase and the beneficial effect of aggregations could become more apparent. However, this would require first attaining and preprocessing a suitable dataset. It would also require additional changes

such as implementing minimum data set sizes for participation, to avoid excessive noise being added in aggregation. Due to limited time, this is left for future work.

7.7 Analysis of Aggregation Groups Size and Model Propagation

The results of this experiment are seen in Table 6.12 and 6.13. In these experiments we measured how the mean error rate changes as the size of the aggregation groups vary. We ran two such experiments, one where aggregated models were shared globally with all peers and one where they were shared only with the party that participated in producing each model. Each peer will include in its ensemble any model shared with it.

Firstly, note that an aggregation group size of one corresponds to each peer simply fitting a model and publishing it with noise added. It seems that there is at least some benefit of producing aggregates, as a group size of one has the highest error rate. There also appears to be some positive trend for higher group sizes in the Party-publising case. Unfortunately, since the different is quite small, a certain conclusion cannot be drawn for this particular data set.

While the difference in error rates is too small to be significant between the two approaches of publishing models, there is a significant effect on the standard deviation of the error between iterations of the experiment. Though the scale of standard deviation is very small for both cases, this indicates that globally sharing models might be preferable. Repeating these experiments on data sets less uniformly distributed among peers could increase scale of the variance, increasing the importance of sharing models globally. Of course, globally publishing all models might not be feasible for practical reasons or acceptable for privacy reasons. Instead, spreading the aggregated models as widely as possible might achieve a similar effect.

Next, we consider the standard deviation, or error variance, measured among the peers. This metric indicates how consistently the network achieves good predictive models. Firstly, we note that publishing to all peers yielded much lower standard deviation for most group sizes. Again, the scale of variance is fairly small, but we believe this effect could be much more considerable with other data sets, especially if data is not uniformly distributed. Interestingly, the variance in peer error rate is smallest when aggregation group sizes is small. This means it might be necessary to make a trade-off between mean error rate and variance between peers.

This effect might be caused by the fact that the peers include their locally fitted models in their ensemble when classifying data. As seen from the results in Section 6.2.5, the variance in error rate among peers is zero for aggregated models. This is as expected, since all peers share the exact same set of models. The local models on the other hand result only from the local data of each peer. Since all models in the ensembles are weighted equally, the effect of the local model decreases as a higher number of aggregate models are published. If this is the case of lower peer error rate variance at lower group sizes, a similar decrease in variance might be achieved by decreasing the weight of the local model. On the other hand, this might be detrimental for those peers that succeeded in training high quality models. In an online learning setting, models weights could be learned over time

as new labeled data arrives.

7.8 Analysis of Privacy Budgeting

Next we consider the results of the privacy budgeting experiment, seen in Figure 6.9. For lower values of per aggregation ϵ_A, more aggregated models are produced, but they are more perturbed. For most levels of ϵ_A, including the maximum level of 0.1, the error rate remained fairly fixed just below 0.2. Once the lowest values of ϵ_A is reached, at $\frac{\epsilon}{16}$ and $\frac{\epsilon}{32}$ the error rate rises sharply.

Firstly, we note that there appears to be no improvement in the error rate when aggregating multiple, more perturbed models and using them in an ensemble. Furthermore, when ϵ_A is too low, the error rate increases. This indicates that it might make more sense to fully expend the privacy budget on a single aggregation, in a setting where the aggregated model can be shared globally. This maximizes the chance that the aggregated model is not too perturbed to be a good fit to data, while there is nothing to lose from only being able to produce a single model.

In a setting where models cannot be shared globally, either for practical or privacy reasons, the results are perhaps even more interesting. Since the error rate stays approximately the same in the range $[\frac{\epsilon}{8}, \epsilon]$, it is possible to choose a lower ϵ_A and allow each peer to take part in $\frac{\epsilon}{\epsilon_A}$ model aggregations. The error rate could be very similar, but it would let the information held in each data partition to be spread to a wider set of peers.

The same effects can be observed in the budgeting experiment on the Spambase data set, though the error rate increases at much higher values of ϵ_A, likely due to the lower regularization required for Spambase.

Chapter 8

Toward a Real-World Application

Due to the relative novelty of differential privacy as a concept, we've found the research and discussion of the real-world applicability of a DP-based system lacking in content. We therefore want to contribute to this field by adding some of our own experiences and thoughts. This chapter will therefore discuss the potential applications of a system based on our distributed machine learner, and some of the prerequisite criteria we've found to need fulfillment before designing an application in the first place.

8.1 Criteria for an Application

What we've discovered during our pre-studies and experimentation have resulted in the following suitability criteria for a potential application.

1. The interest in privacy is strong, and the risk of re-identification is significant.
2. The information to be analyzed must be stored in some form of a structured database.
3. The amount of data must be significant.
4. The analysis of the data can tolerate some distortion in the information from the database.
5. The analysis of the data do not involve study of outliers, or other individuals.

These criteria are not meant to provide an all-compassing set of rules that when followed will mean instant success for a given application, but rather as a definition of characteristics that can lead to good application opportunities if considered.

8.1.1 Strong privacy interest

As a first criterion, a solid interest in privacy is necessary due to implementation of a differential privacy mechanism will lead to a trade-off between utility and privacy. As we've

have shown in our analysis the effects of this trade-off can be minimized, but stronger privacy guarantees will inevitably lead to decaying accuracy due to the noise addition. The underlying assumption in this criterion is therefore that the risk of re-identification is significant, but manageable to secure by applying a differential privacy mechanism. The potential for future re-identification should be a factor in determining the strength of the privacy risk, but it should not prevent consideration of other factors, such as the potential public utility one can gain from data analysis.

8.1.2 Structured data

The mechanisms that were originally designed to support differential privacy were created to address the threats to privacy from the sharing of information contained in a centralized database. We've shown through our experimentation that there are an potential applicability in designing a distributed machine learning system, but there are still a criterion that the data must be structured and labeled before any form for learning can be applied. Future experimentation can potentially prove that this criterion can be relaxed. This can then open avenues for exploring the usage of other branches of machine learning, such as deep learning and unsupervised learning. For the moment however, structured data is an important prerequisite.

8.1.3 Data size

As we showed in Section 7.3, the amount of data available for learning can have massive impact on the accuracy of the resulting classifier. This is a direct result of equation 3.12, which specifies the upper bound on the sensitivity for distributed logistic regression, i.e the noise needed to hide a record in the data set. Since a higher value of n_j, i.e a larger amount of data in the smallest data set, will lead to a smaller amount of noise added to each aggregated model and generally better accuracy. The exact amount of data needed to create a decent classifier is difficult to exactly quantify, but our results indicate that a smaller number of records than that initially tested by Pathak et al. [2010] is truly needed to construct a decent classifier. . The value of the privacy parameter ϵ will also play a factor in the amount of data needed, as a stronger guarantee of privacy requires a larger amount of data (see Figure 6.1).

Any application that bases itself on our distributed framework will therefore need a sizable amount of data available in each peer to create the initial models. This can pose a problem if the application is released without any form of pre-performed training, as the users will not gain any value from the application before the necessary amount of data is available, and therefore they likely will be less inclined to provide the data needed. This problem can be compared to the classical cold start problem prevalent in recommender systems. In a typical collaborative filtering setting the system will match a user's rating against other users' and find the people with the highest similarity score. The cold start problem appear when recommendations are required for items that no user has yet rated [Schein et al., 2002], which often is the case in newly designed systems without the necessary user mass.

8.1.4 Tolerance for data distortion

The fourth criterion is that the usage of the data analysis must tolerate some distortion. Since our mechanism for supporting differential privacy function by inserting Laplacian noise, the resulting classifier will be a distorted version of the ”true” classifier. The users of a system based on our framework must therefore be willing to accept a potential loss of utility due to distortion in return for a privacy guarantee for their data.

Certain studies of large data sets might not be able to accommodate even small drop in accuracy in their system. Examples of this would be studies of health care data from hospitals or precise scientific research, where just a small source of error might render the results inoperable. Another example is the study of census data, where a researcher noted that the introduction of noise often resulted in demographic research errors [Yakowitz, 2011].

This is not to say that research on health data or census data is completely infeasible, as both our own research on the Adult data set (US Census data) and the research of Ji et al. [2014a] on health data has shown promising results. What it means is that certain considerations need to be taken when performing these studies, such as simplifying the problem, finding workarounds for dealing with wider margin of errors, and more.

8.1.5 Study type

The fifth criterion for adopting differential privacy is that it precludes the study of outliers. The goal of such a study is inconsistent with the differential privacy goal of preventing the identification of the presence or absence of a record in a database. An application based on our architecture should therefore instead focus on finding similarities between users, and how to best leverage the aggregation mechanism we employ.

8.2 Potential Future Applications

8.2.1 Wearable health sensor data analysis

A growing worldwide market is the sale and usage of wearable sensors, such as environmental sensors, motion sensors, and health sensors. A IHS report [Nissil et al., 2014] from 2014 estimates that the market for sensors in wearables will expand to 135 million units in 2019, up from 50 million in 2013. These wearables will evolve from being just a single purpose device such as a pedometer and grow into more multipurpose devices such as a smartwatch, which will consist of several sensors which can monitor varying areas of use.

The wearable devices are implementing fitness and health monitoring by using a mixture of sensors, such as motion, pulse, hydration and skin temperature sensors. All of these wearables will therefore generate a massive amount of data about the person who are using them. This data can be considered as highly sensitive information, as it can unveil a lot about their user’s health, and the manufacturers of these devices knows this. Dana Liebelson, a reporter for Huffington Post, queried several US-based fitness device companies about their privacy. One of the replies she got, was that ”the company does not sell information collected from the device that can identify individual users”, but that they

were considering marketing aggregate information that cannot be linked back to an individual[Liebelson, 2014]. As we saw in section 2.2 and 2.3, many of the popular methods for aggregating and anonymizing a data set carries an inherent risk of a privacy breach.

A study [Raij et al., 2011] have been performed to try to measure the privacy concerns of people using wearable sensors, and found that activity trackers that monitor heart rate, steps, and pulse for instance, was usually seen as inoffensive to the users privacy at the start of the study. The researchers then had some of the testers wear sensors, and could from the resulting data infer periods of heightened stress, as well as derive certain context and behaviors that could trigger the aforementioned stress. The participants were then given a similar questionnaire to the initial one, and many then reported a heightened sense of concern.

This is where we see a potential application for our distributed framework. Although there would be some initial problems due to our learner at the moment requiring labeled data to create a classifier, there can be potential in a health application where users keep control of their own data. According to IBM research, areas in which enhanced data and analytics yield the greatest results include: pinpointing patients who are the greatest consumers of health resources or at the greatest risk for adverse outcomes; providing these individuals with the information they need to make informed decisions and more effectively manage their own health as well as more easily adopt and track healthier behaviors[IBM, 2013]. While pinpointing a single user might not be feasible due to the differential privacy guarantee, there could be merit in an application where the user could be anonymously classified if he or she have a potential health risk such as diabetes. This classification would be based on the model created by their local data. The user could then be sent information on how to handle their health more efficiently, and potentially also be given an exercise regime that could be monitored through their wearables.

Evaluating this application against the suitability criteria we suggested in Section 8.1, we can see that a health application would have little trouble fulfilling the first three criteria. While the public attitude towards their privacy can vary from the completely unconcerned to a small proportion of the public that has strong views on privacy, research would suggest that people have increasing expectations on security and choice about access to their records [Singleton et al., 2008]. The two next criteria that deal with data structure and size should be relatively easy to fulfill, as a wearable health sensor would quickly register a plethora of data records for use in analysis. Looking at the estimates for market expansion there is definitely potential to gain a critical user mass due to the increasing interest in such devices.

The two next criteria will most likely prove a bigger challenge to overcome, as earlier research on privacy-preserving health data analysis have found that a high-dimensional data generally makes the usage of differential privacy inappropriate due to degrading utility [Gardner et al., 2013; Mohammed et al., 2013]. One would therefore need to be careful when deciding the goal of study, and start out with a well-thought out plan of data collection, analysis, and visualization.

8.2.2 Private sharing of business data

A potentially interesting and lucrative market can be found in facilitating the sharing of data between businesses in a private manner. Our motivating example is found in the busi-

ness of oil market analysis, where competing firms gather a lot of data about oil price, rig placements, supply ship availability, and more. They use this data to create analytical models which help them in their work, and they also sell this information to external clients. Often the firms would like to collaborate their models or their data with their competitors. This could for example be done to validate that they are seeing the same market trends, but due to the sensitive nature of their data and their fear of losing a competitive edge, they cannot do this in a practical way.

It is in a situation like this that our distributed learner could be applied, and allow the sharing of data between competitors as our system could provide a privacy guarantee to all of the participants. The participants would never lose control of their data, as all they would need to install a program that allows them to connect as a peer in our network: No third party would ever need access to their data. Since the goal of the application is to privately collaborate and find common trends in data from different parties, this would fit perfectly in with the 4th and 5th suitability criteria.

The challenge would come mainly from trying to employ a common data structure, as competing firms will most likely have their own way of handling the data they collect. An application would need to be built with this in mind, as data would need either a common format or a good amount of pre-processing. The latter could be solved by providing an add-on feature to the application that would allow the businesses to easily pre-process the data on their own schedule before making them available for analysis.

Chapter 9

Reflections and Conclusion

9.1 Reflection on Implementation Challenges and Solution

While JADE was useful in offering an intuitive abstraction for modeling peers, it was challenging to use dynamically. In particular, resetting experiments proved difficult. Since creating a JADE environment creates a new process which also opens a particular TCP port. If a JADE environment was created using the same port, the new instance would fail to start up. Since creation and tear-down of the JADE platform is not instantaneous, it was necessary. We also had issues with re-registering new instances of some of the agents we used, since the unregistering is an asynchronous call. All in all, it might have been simpler and less error prone to create a multi-threaded application using a tool that required less investment, like OpenMP. On the other hand, such an implementation would not lend itself as easily to a production prototype - our JADE implementation could with some extensions be tested on in a setting with many devices.

9.2 Reflections on Privacy and Utility

Although the promise of differential privacy (see Section 3.1) seems to be an ideal guarantee for each of the data subjects in a data set, it has received mixed reviews from legal scholars and computer scientist on its usefulness for resolving the privacy-utility tradeoff. Narayanan and Shmatikov praised differential privacy as ”a major step in the right direction[Narayanan and Shmatikov, 2010]”. Sarathy and Muralidhar on the other hand, contend that differentially private mechanisms are impracticable in computationally intensive context, as well offering either very little privacy or very little utility[Sarathy and Muralidhar, 2011]. Xiao et al. also criticized that the mechanisms often placed undue burdens on data researchers due to decreased utility, especially with large data sets used in research on populations[Xiao et al., 2011]. UCLA law professor Paul Ohm presented a legal review of current anonymization techniques, including differential privacy, and prac-

tically dismissed it as a solution from a legal standpoint. His critique was based on several factors; Such as the mechanism not being very intuitive, having limited usefulness in high noise situations, and potentially bug-prone security functions. He concluded that "utility and privacy are, at bottom, two goals at war with one another. In order to be useful, anonymized data must be imperfectly anonymous[Ohm, 2010]."

All of the aforementioned critique of differential privacy can be considered valid criticism in certain situations, as we have experienced the difficulty of balancing the utility-privacy trade-off ourselves during our project. Our experiments have shown that there are several parameters outside of just the privacy parameter ϵ that need to be fine-tuned, such as regularization, learning rate, and group size. Small changes in the composition of these three can lead to wildly varying results for our classifier, but at least these parameters can be tuned only with the intention of maximizing the utility of the resulting classifier. When tuning the ϵ parameter on the other hand, one always have to keep in mind the trade-off. This tuning is made harder due to the exponential nature of the privacy guarantee, as well as its not well-defined bounds. As Dwork herself has stated[Dwork, 2008]:

> The choice of ϵ is essentially a social question [...]That said, we tend to think of ϵ as, say, 0.01, 0.1, or in some cases, ln2 or ln3. If the probability that some bad event will occur is very small, it might be tolerable to increase it by such factors as 2 or 3, while if the probability is already felt to be close to unacceptable,then an increase by a factor of $e^{0.01} \approx 1.01$ might be tolerable, while an increase of e, or even only $e^{0.1}$, would be intolerable.

Our own experiments have indicated that the ideal range for ϵ lies between 0.1 and 1.0, which fits reasonably well with Dwork's description. At lower values than 0.1, the error rate quickly rise as well as the error variance between the peers, meaning that an ϵ value of 0.01 returns too much noise to be of any utility. On the Spambase data set for example, the error rate increased from 0.157 to 0.336 when ϵ changed from 0.1 to 0.01. A similar change from 1.0 to 0.1 however, only decreased the result by one percentage point, meaning the error rate went from 0.157 to 0.150.

We will therefore conclude that although the trade-off between privacy and utility is by no means solved by our current efforts, we have shown that it is not such a contradictory problem as some people might claim. By performing careful pre-processing and extensive parameter tuning, one can achieve useful results from classification even though the models are obscured by differential privacy.

9.3 Reflection on the Practical Applicability of Differential Privacy

As we can tell from the previous section, differential privacy has been met with both enthusiasm and criticism. In this section we will reflect on some of the strongest criticism of differential privacy, such as what was presented in the paper by Sarathy and Muralidhar [2011], as well as Ohm [2010]. Both of these papers try to discredit the applicability of differential privacy by presenting mostly legal arguments, mainly targeting their criticism on a single mechanism: the output perturbation. These authors seems to be assume that

this is the only way of providing differential privacy, which would only be true if all the literature on differential privacy consisted only of Dwork's original paper from 2006.

In the paper by Sarathy and Muralidhar [2011], the opening example is of an internist at a hospital, which query the database with differential privacy protection on the output. They then claim: "even knowing the distribution of noise that is randomly added to each cell, the internist has no hope of interpreting the response. The true values could be almost anything." While this example is technically correct, it is flawed by the fact that no such system would be allowed to exist in the first place, especially only with the basic output perturbation mechanism.

As we pointed out in Chapter 8, a system need to consider certain criteria before development, and design it accordingly. In a healthcare based information system, certain considerations would certainly be necessary; One would definitely need a lax enough privacy guarantee that the output is not overtly perturbed, and the data would need to be aggregated in a manner that would decrease the sensitivity as much as possible to require the least amount of noise.

That is not to say that there are not valid criticism in Sarathy's paper: differential privacy can only make probabilistic guarantees, and if your legal standard is stricter than that, it might not be the best choice. Their rhetoric on the other hand, with claims such as: "differential privacy is either not practicable or not novel" seems to be unfounded at best when you consider their representation of differential privacy is basically a straw-man argument. If legal scholars such as Sarathy and Ohm intends to level criticism from a legal standpoint, it would be advisable to form an objective criticism after considering the entirety of the research field, instead of selectively picking one aspect.

In their effort to discredit differential privacy, the authors have ignored both how academic and business research work to address these problems that they have raised. Research are being still being performed on new differential privacy mechanisms with optimal utility [Eigner et al., 2014], and there has also been research on creating an economic model for choosing the epsilon value when designing a data study [Hsu et al., 2014]. On the business side, Google have released a framework called RAPPOR, based on a concept called randomized response [Erlingsson et al., 2014]. Additionally, a study by Chin and Klinefelter [2012] infers that Facebook appears to be using differential privacy-supporting technologies in its targeted advertising system without apparent loss of utility.

Based on these developments, as well as our own findings in this area, we find it safe to conclude that differential privacy has great potential for real-world applicability. Despite of all the critique and the difficulty it brings, we have still found it entirely rewarding to perform research on the usage of differential privacy. As a solution, it is not an optimal fix-all concept which can be applied to every form of machine learning problem and provide privacy for the participants. There is still research to be done on how to make it work on various forms of data, and also on how to make it in an online setting. All of these steps will hopefully lead to a solution that can one day work in a way so that people will have control of their own data, make it available for research and still be guaranteed that their data and their privacy is safe.

9.4 Conclusion and Final Remarks

RQ1: How big is the loss of accuracy in a distributed, differentially private system, compared to a centrally trained model?
We have found that under ideal conditions and perfectly tuned parameters, the loss of accuracy can be minimized to a negligible difference compared to a centralized and noise-less solution. We have however noted in our reflection that the research community have not yet reached consensus on appropriate standards for the privacy parameter ϵ, but our results indicate that a range between 0.1 and 1.0 seems to yield the best results. Further testing and standardization by the community is ultimately needed before deciding in which situations different values will be appropriate.

RQ2: How can the variance in accuracy between participants be minimized?
Since our tests involved uniform distributions of data among the participants, we observed very low variance in accuracy even in the predictions of locally trained models. While it is clear that variance will be zero if all peers use the same, aggregated models, the gain is very small. Due to this, we have to conclude that in a setting with uniform data distribution among participants, there is no compelling reason to use the aggregates if the goal is to reduce variance. We believe that experiments in future work conducted with non-uniform data distribution might lead to a different conclusion.

Note that we did in some cases see better error rate for the aggregated model than the locals, so when considering both the error rate and its variance the aggregated model can be the best choice despite the insignificant difference in variance.

RQ3: Can we validate and enhance earlier research in distributed differentially private machine learning?
We have validated the research performed by Pathak et al. [2010] by using their proposed solution as a baseline, and then extending it with some design of our own. We've ran similar tests on the same dataset as they did, and found that our framework classified with a mean error of 0.165 compared to their best case of 0.24. When we ran the experiment with the same configuration as them, ergo without our upgrades, we achieved a mean error of 0.185. This indicates that they might have an non-optimal implementation, or that some details of their configuration might be missing or misleading.

9.5 Threats to Validity

9.5.1 Platform

A potential threat to the validity of our work, is how we performed the setup of the Jade platform. Since we wanted to perform our experiments on over a range of parameters, we needed to find a way to reset the platform after a successive experiment and re-run it with a new set of configurations. We solved this by having a jade agent called CompletionAgent be responsible for waiting for every peer to message indicating their completion, which would trigger the CompletionAgent to deregister all the peers from the MainContainer and then reset the whole environment. The environment would then be set up again with new parameters.

What we see as a potential source for concern in this process is the possibility for error during the deregistration. During the implementation of this process we encountered some problems in making it work, as the CompletionAgent seemed to take an unreasonable amount of time in completing its purpose. Although we found a solution to this problems, there is still a risk that peers do not deregister as they should and carry through into the next iteration of testing. This could lead to false information being injected into our experiment, which would skew our results.

We have however minimized this risk by continuously developing unit test to verify new code additions, as well as using JADE's native GUI to supervise the behavior of the peers while running. We therefore conclude that the risk is negligible.

9.5.2 Homomorphic encryption

As mentioned in Section 3.6, homomorphic encryption is still in an early stage of development and therefore cannot be called a well-proven technology. Our method for aggregating models from various peers is based on a homomorphic encryption scheme developed by Pathak et al. but due to time constraints we could not actually implement it and instead had to opt for simulating the results of applying this scheme. We therefore do not have real-world results that can validate the applicability of this scheme, nor do we know if applying this scheme would lead to increased run-time and resource consumption. While this remains an interesting area for future research, as it stands now it remains a possible threat to the validity of the results we've achieved and therefore also the conclusion we have drawn from them.

9.5.3 Validation scheme

As we mentioned in Section 3.4, we chose not to implement stratified cross validation as recommended by Kohavi [1995]. At the time the decision was made, the data sets we employed had close to uniform class distribution, which would most likely not gain any form of improvement from employing stratification. As the experiment progressed, we added the Adult data set to our experimental procedure, which has a class distribution of ca 75/25%. Instead of re-doing all of our previous experiments, we instead chose to keep on validating with the normal form of cross-validation.

This choice might have introduced a small source of error into our results from the Adult data set, as Kohavi [1995] notes that stratification is generally better scheme in these situations, both in terms of bias and variance. We defend our choice by pointing out that Kohavi's paper report that the biggest improvement in stratification comes when using a low amount of folds, and only shows minuscule improvement in the case of 10-fold which is what we employ. The advantage of stratification is also most apparent in data sets with many categories, whereas all our data sets are binary. We therefore conclude that the variance of the results on the Adult data set might be slightly worse than the optimal solution, but the risk of this is so negligible compared to other sources of variance, such as the noise and regularization parameters.

9.6 Future Work

In this section we propose some future areas of research which could lead to significant improvements for our framework. The three first suggestions are topics we really wish we could have achieved during this thesis project, but there just wasn't enough time and resources available. The last three suggestions are much more comprehensive, and can possibly be considered a whole new project or research area.

Further develop and test the propagation of aggregated models: We experienced that when we shared the aggregated models globally in our network, we could decrease the standard deviation in our classification error, as well as sometimes improving the classifier. Further research should go in expanding this behavior, as you could potentially propagate models only to peers in geographic and/or demographic vicinity. This could possibly lead to more specific models, which could give better classification rate to a specialized subset of peers.

Test on datasets with uneven distribution: Another important area of research would be to further test the applicability of peers sharing data to create better aggregated models. Our original research questions were designed to explore the validity of our proposed method of doing differentially private machine learning, and our current research has been limited to testing on a small amount of data sets which are publicly available. In the future more research are needed on data sets with an uneven underlying distribution, which could potentially provide results highlighting the usefulness of sharing information between peers. An ideal data set would be one where each peer only holds data which makes up only a part of the solution.

Implement the Newscast algorithm for selecting peers: The Newscast algorithm is a gossip protocol which facilitates a robust spread of information[Jelasity and Van Steen, 2002]. The core of the protocol involves periodic and pairvise interaction between processes. Implementing this algorithm would allow our system to scale better when a big number of peers are added to the network. The biggest bottleneck of our system at the moment is the peer sampling during the group forming, as it requires a single agent to act as a manager for how groups are formed. The basic idea of the Newscast algorithm is that each node, or peer in our situation, has a partial view of the system. All nodes exchange their views periodically, which allows them to keep an up-to-date view locally and spread their information throughout the network. Further research into this algorithm would allow us to customize this algorithm so that peers in our network could form groups based on their partial views of the network.

Full data protection for the data of each peer: This would involve dividing the epsilon by the biggest data set size, as formalized by Dwork and Roth [2013]. This is an ever tighter privacy guarantee, but it would mean that the results would contain too much noise. The reason for this is that the increased noise variance needed to protect full data partitions given the sensitivity bound of Pathak et al. [2010] would ensure that essentially no information is released to the aggregated model. We suspect that it would be possible to modify Equation 3.12 in a way to include the number of participants in the aggregation such that sensitivity decreases as more peers take part. This guarantee would need to be formalized mathematically. If we succeeded in formalizing such a sensitivity bound, the

increase in senstivity resulting from full data protection could be countered by increasing the number of participants.

System that works in an online setting: It could potentially improve the system, as you would have new data coming in which could replace old data with spent budgets. It could also be potentially a big trade-off as you won't have the same data history as you would have in a system without differential privacy. Dwork has written about this in her book, so we can take inspiration from there.

Security mechanisms for stopping sabotage: In our current system we have assumed that the peers will be honest-but-curious when sharing their data, meaning that we have no way of detecting dishonest peers. In a real world system there would need to be safeguards against people which intend to either destroy the validity of the classifier created by feeding misinformation into the system, or people who tries to intercept and expose the data from other peers. Potential research areas for finding solutions could be intrusion detection in distributed systems, fraud detection, trust networks and reputation systems, and also further research into encryption.

Bibliography

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

Asplund, A., Frøystad, P., Dec. 2014. Learning from data streams: A systematic mapping study. Tech. rep., Norwegian University of Science and Technology, Department of Computer and Information Science.

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).

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

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

Bottou, L., 2011. Stochastic gradient descent (v.2).
URL http://leon.bottou.org/projects/sgd

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

Breiman, L., 1996. Bagging predictors. Machine learning 24 (2), 123–140.

Caruana, R., Niculescu-Mizil, A., 2006. An empirical comparison of supervised learning algorithms. In: Proceedings of the 23rd international conference on Machine learning. ACM, pp. 161–168.

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-regression

Chaudhuri, K., Monteleoni, C., Sarwate, A. D., Feb. 2011. Differentially Private Empirical Risk Minimization. The Journal of Machine Learning Research 12, 1069–1109.
URL http://dl.acm.org/citation.cfm?id=1953048.2021036

Chin, A., Klinefelter, A., 2012. Differential privacy as a response to the reidentification threat: The facebook advertiser case study. North Carolina Law Review 90 (5).

Cotter, A., Shamir, O., Srebro, N., Sridharan, K., 2011. Better mini-batch algorithms via accelerated gradient methods. In: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K. (Eds.), Advances in Neural Information Processing Systems 24. Curran Associates, Inc., pp. 1647–1655.
URL http://papers.nips.cc/paper/4432-better-mini-batch-algorithms-via-accelerated-gradient-methods.pdf

Curtis, S., Oct. 2014. Sir Tim Berners-Lee calls for new model for privacy on the web.
URL http://www.telegraph.co.uk/technology/internet/11148584/Tim-Berners-Lee-calls-for-new-model-for-privacy-on-the-web.html

Dalenius, T., 1977. Towards a methodology for statistical disclosure control. Statistik Tidskrift 15 (429-444), 2–1.

DPA, E. D. P. A., Sep. 2014. European results of the 2014 global privacy enforcement network sweep.
URL http://dataprotection.ie/docimages/GPEN_Summary_Global_Results_2014.pdf

Dragland, Å., 2013. Big data–for better or worse. SINTEF, retrieved on July 22.

Dwork, C., 2006. Differential privacy. In: in ICALP. Springer, pp. 1–12.

Dwork, C., 2008. Differential privacy: A survey of results. In: Theory and Applications of Models of Computation. Springer, pp. 1–19.

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

Eigner, F., Kate, A., Maffei, M., Pampaloni, F., Pryvalov, I., 2014. Differentially private data aggregation with optimal utility. In: Proceedings of the 30th Annual Computer Security Applications Conference. ACSAC '14. ACM, New York, NY, USA, pp. 316–325.
URL http://doi.acm.org/10.1145/2664243.2664263

Elkan, C., 2014. Maximum likelihood, logistic regression, and stochastic gradient training.
URL http://cseweb.ucsd.edu/~elkan/250B/logreg.pdf

Erlingsson, Ú., Pihur, V., Korolova, A., 2014. Rappor: Randomized aggregatable privacy-preserving ordinal response. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. ACM, pp. 1054–1067.

European Parliament, Mar. 2013. Q&A on EU data protection reform.
URL http://www.europarl.europa.eu/news/en/news-room/content/20130502BKG07917/html/QA-on-EU-data-protection-reform

European Parliament, C. o. t. E. U., Apr. 2006. Directive 2006/24/ec.
URL http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:105:0054:0063:EN:PDF

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.

Gardner, J., Xiong, L., Xiao, Y., Gao, J., Post, A. R., Jiang, X., Ohno-Machado, L., 2013. Share: system design and case studies for statistical health information release. Journal of the American Medical Informatics Association 20 (1), 109–116.

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

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

Hern, A., Oct. 2014. Sir Tim Berners-Lee speaks out on data ownership.
URL http://www.theguardian.com/technology/2014/oct/08/sir-tim-berners-lee-speaks-out-on-data-ownership

Hopkins, M., Reeber, E., Suermondt, J., 1999. UCI machine learning repository, spambase data set.
URL https://archive.ics.uci.edu/ml/datasets/Spambase

Hsu, J., Gaboardi, M., Haeberlen, A., Khanna, S., Narayan, A., Pierce, B. C., Roth, A., 2014. Differential privacy: An economic method for choosing epsilon. CoRR abs/1402.3329.
URL http://arxiv.org/abs/1402.3329

IBM, 2013. White paper:data-driven healthcare organizations use big data analytics for big gains. Tech. rep.
URL http://www-03.ibm.com/industries/ca/en/healthcare/documents/Data_driven_healthcare_organizations_use_big_data_analytics_for_big_gains.pdf

Jelasity, M., Van Steen, M., 2002. Large-scale newscast computing on the internet. Tech. rep., Citeseer.

Ji, Z., Jiang, X., Wang, S., Xiong, L., Ohno-Machado, L., Jan. 2014a. 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

Ji, Z., Lipton, Z. C., Elkan, C., 2014b. Differential privacy and machine learning: a survey and review. arXiv preprint arXiv:1412.7584.

Kaplan, B., 2014. Patient health data privacy. Yale University Institute for Social and Policy Studies Working Paper, 14–028.

Kellaris, G., Papadopoulos, S., Mar. 2013. Practical differential privacy via grouping and smoothing. Proc. VLDB Endow. 6 (5), 301–312.
URL http://dx.doi.org/10.14778/2535573.2488337

Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2. IJCAI'95. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 1137–1143.
URL http://dl.acm.org/citation.cfm?id=1643031.1643047

Kumar, R. K., Poonkuzhali, G., Sudhakar, P., 2012. Comparative study on email spam classifier using data mining techniques. In: Proceedings of the International MultiConference of Engineers and Computer Scientists. Vol. 1. pp. 14–16.

Laney, D., 2001. 3d data management: Controlling data volume, velocity and variety. META Group Research Note 6.

Lee, J., Clifton, C., 2011. How much is enough? choosing ε for differential privacy. In: Proceedings of the 14th International Conference on Information Security. ISC'11. Springer-Verlag, Berlin, Heidelberg, pp. 325–340.
URL http://dl.acm.org/citation.cfm?id=2051002.2051032

Liebelson, D., Jan. 2014. Are fitbit, nike, and garmin planning to sell your personal fitness data?
URL http://www.motherjones.com/politics/2014/01/are-fitbit-nike-and-garmin-selling-your-personal-fitness-data

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.

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

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.

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-grail/fulltext

Mohammed, N., Jiang, X., Chen, R., Fung, B. C., Ohno-Machado, L., 2013. Privacy-preserving heterogeneous health data sharing. Journal of the American Medical Informatics Association 20 (3), 462–469.

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.

Narayanan, A., Shmatikov, V., 2010. Myths and fallacies of personally identifiable information. Communications of the ACM 53 (6), 24–26.

Nissil, S., Bouchaud, J., Boustany, M., Oct. 2014. White paper: Mems & sensors for wearables report. Tech. rep., IHS Technology.
URL https://technology.ihs.com/496122/mems-sensors-for-wearables-2014

Ohm, P., 2010. Broken promises of privacy: Responding to the surprising failure of anonymization. UCLA Law Review 57, 1701.

Opitz, D., Maclin, R., 1999. Popular ensemble methods: An empirical study. Journal of Artificial Intelligence Research 11, 169–198.

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

Pathak, M., Rane, S., Raj, B., 2010. Multiparty Differential Privacy via Aggregation of Locally Trained Classifiers. In: Advances in Neural Information Processing Systems. pp. 1876–1884.
URL http://papers.nips.cc/paper/4034-multiparty-differential-privacy-via-aggregation-of-locally-trained-classifiers

Peduzzi, P., Concato, J., Kemper, E., Holford, T. R., Feinstein, A. R., 1996. A simulation study of the number of events per variable in logistic regression analysis. Journal of clinical epidemiology 49 (12), 1373–1379.

Platt, J. C., 1999. Fast training of support vector machines using sequential minimal optimization. In: Advances in kernel methods. MIT Press, pp. 185–208.

Raij, A., Ghosh, A., Kumar, S., Srivastava, M., 2011. Privacy risks emerging from the adoption of innocuous wearable sensors in the mobile environment. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, pp. 11–20.

Rajkumar, A., Agarwal, S., 2012. A differentially private stochastic gradient descent algorithm for multiparty classification. In: International Conference on Artificial Intelligence and Statistics. pp. 933–941.

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

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

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

Schein, A. I., Popescul, A., Ungar, L. H., Pennock, D. M., 2002. Methods and metrics for cold-start recommendations. In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp. 253–260.

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

Sharma, S., Arora, A., 2013. Adaptive approach for spam detection. IJCSI International Journal of Computer Science Issues 10 (4), 23–26.

Singleton, P., Lea, N., Tapuria, A., Kalra, D., 2008. Public and professional attitudes to privacy of healthcare data: a survey of the literature.

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

Tockar, A., Sep. 2014. Riding with the stars: Passenger privacy in the nyc taxicab dataset. Visited:2015-03-04.
URL http://research.neustar.biz/2014/09/15/riding-with-the-stars-passenger-privacy-in-the-nyc-taxicab-dataset/

U.S Government, 2014. Big data: Seizing opportunities, preserving values.

U.S Government, Feb. 2015. Administration Discussion Draft: Consumer Privacy Bill of Rights Act of 2015.
URL https://www.whitehouse.gov/sites/default/files/omb/legislative/letters/cpbr-act-of-2015-discussion-draft.pdf

Whiteson, D., 2014. Uci machine learning repository, susy data set.
URL https://archive.ics.uci.edu/ml/datasets/SUSY

Xiao, X., Wang, G., Gehrke, J., 2011. Differential privacy via wavelet transforms. Knowledge and Data Engineering, IEEE Transactions on 23 (8), 1200–1214.

Xu, W., 2011. Towards optimal one pass large scale learning with averaged stochastic gradient descent. CoRR abs/1107.2490.
URL http://arxiv.org/abs/1107.2490

Yakowitz, J., 2011. Tragedy of the data commons. Harv. JL & Tech. 25, 1.

Zhang, J., Zhang, Z., Xiao, X., Yang, Y., Winslett, M., 2012. Functional mechanism: Regression analysis under differential privacy. Vol. 5. pp. 1364–1375, cited By 10.
URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84873178476&partnerID=40&md5=952ebdb74978490cfd94284aba62bbb1

Zhang, J. D., Ghinita, G., Chow, C. Y., 2014. Differentially private location recommendations in geosocial networks. In: Proceedings of the 2014 IEEE 15th International Conference on Mobile Data Management - Volume 01. MDM '14. IEEE Computer Society, Washington, DC, USA, pp. 59–68.
URL http://dx.doi.org/10.1109/MDM.2014.13

Appendix

9.7 Additional Results

9.7.1 Measuring error rate

Experiment Name	**Mean error**	**Std. dev.**	λ
Centralized logistic regression	0.104	0.0059	2^{-8}
Disjoint logistic regression	0.159	0.0026	2^{-5}
Aggregated model, $\epsilon = 1.0$	0.163	0.0103	2^{-3}
Ensemble model, $\epsilon = 1.0$	0.150	0.0045	2^{-5}
Aggregated model, $\epsilon = 0.1$	0.182	0.0256	2^{-1}
Ensemble model, $\epsilon = 0.1$	0.157	0.0117	2^{-2}

Table 9.1: Measuring accuracy: Spambase

Experiment Name	**Mean error**	**Std. dev.**	λ
Centralized logistic regression	0.306	0.0809	2^{2}
Disjoint logistic regression	0.279	0.0035	2^{-3}
Aggregated model, $\epsilon = 1.0$	0.271	0.0056	2^{-2}
Ensemble model, $\epsilon = 1.0$	0.274	0.0034	2^{-3}
Aggregated model, $\epsilon = 0.1$	0.291	0.0112	2^{0}
Ensemble model, $\epsilon = 0.1$	0.285	0.0087	2^{0}

Table 9.2: Measuring accuracy: Susy

		Actual	
		1	0
Predicted	1	273	49
	0	81	518

Table 9.3: Confusion Matrix: Spambase. Local model only.

		Actual	
		1	0
Predicted	1	268	38
	0	86	529

Table 9.4: Confusion Matrix: Spambase. Aggregated, DP model only.

		Actual	
		1	0
Predicted	1	257	35
	0	97	5d32

Table 9.5: Confusion Matrix: Spambase. Ensemble of Aggregated and Local.

9.7.2 Changes in number of participants

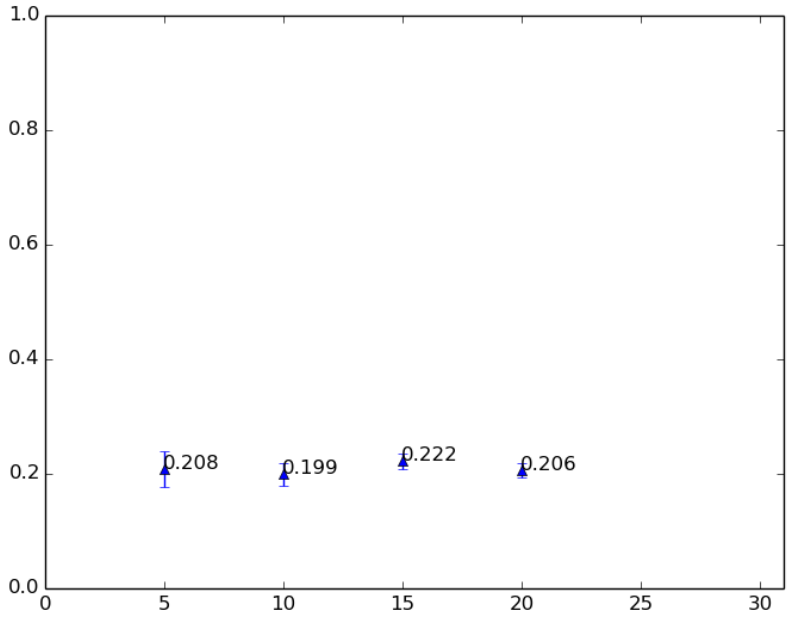

Figure 9.1: Effect of peer numbers. Spambase.

9.7.3 Peer error rate variance

Model	Mean error	Peer std. dev.
Local, no privacy	0,159	0.016
Aggregated, d. privacy	0,163	0.000
Ensemble with both	0.150	0.006

Table 9.6: Variance among peers. Spambase.

Note that the results seen in Table 9.6 differ in the experimental setup described Section 6.1.5 in that each peer had 300 records instead of 250.

9.7.4 Effect of aggregation group size and model propagation

Group size	Mean error	Error std. dev.	Peer error std. dev.
1	0.233	0.0184	0.0435
5	0.217	0.0185	0.0399
10	0.223	0.0179	0.0368
15	0.221	0.0245	0.0290
20	0.205	0.0323	0.0257

Table 9.7: Effect of aggregation group size. Party-publishing. Spambase.

Group size	Mean error	Error std. dev.	Peer error std. dev.
1	0.215	0.0076	0.0036
5	0.211	0.0135	0.0125
10	0.197	0.0116	0.0191
15	0.200	0.0090	0.0225
20	0.195	0.0110	0.0202

Table 9.8: Effect of aggregation group size. All-publishing. Spambase.

9.7.5 Value of budgeting privacy

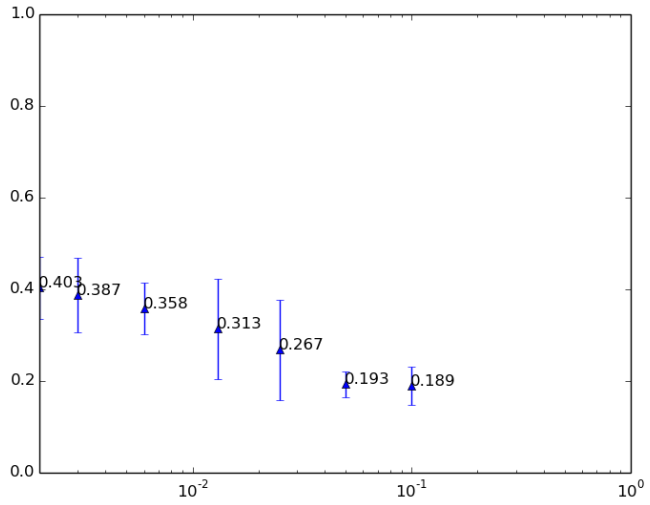

Figure 9.2: Effect of Privacy Budgeting. Spambase.