Privacy Preserving Distributed Data Processing and Game Theoretic Methods for Data Utility and Privacy Tradeoffs

A Dissertation Presented in Fulfillment of the Requirements for the Degree of MRes

Nan Wu

Department of Computing
Faculty of Science
Macquarie University, NSW 2109, Australia

Submitted June 2019

Declaration

I certify that the work in this thesis entitled Privacy Preserving Distributed Data processing and Game Theoretic methods for Data Utility and Privacy tradeoffs has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree to any other university or institution other than Macquarie University. I also certify that the thesis is an original piece of research and it has been written by me. Any help and assistance that I have received in my research work and the preparation of the thesis itself have been appropriately acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signed: ....

Date: 13 June 2019 ....

Abstract

In this thesis, we focus on differential privacy and the trade-off between privacy level and accuracy of shared databases. For differential privacy, the Laplace mechanism is commonly utilised, for which the choice of value for a privacy parameter ϵ, also known as the privacy budget, plays a key role in the trade-off between privacy level and accuracy. We aim to build a game-theoretical model of choosing this privacy budget ϵ, while optimising the utility of the shared databases to which the differential privacy mechanism is applied. In this thesis, we consider differentially private queries applied in the context of two different models. The first model is an information-theory based query system using a discrete Laplace mechanism. The utility and leakage are quantified by information theory with min-entropy between original information, real answer and reported answer in the system. The second model is based on data analysis with privacy-aware machine learning using differentially-private gradient queries. We quantify the quality of the trained model by fitness cost, which is a function of differential-privacy parameters and the size of the distributed datasets, to capture the trade-off between privacy and utility by machine learning. Then, game theory is used to analyse the utility-leakage tradeoffs for both of these two models respectively.

Contents

List of Publications

This thesis has resulted in the following publications.

1. N. Wu, F. Farokhi, D. B. Smith and M. A. Kaafar, "The Value of Collaboration in Machine Learning with Differentially-Private Gradient Queries" (Accepted by IEEE Security & Privacy 2020)
2. N. Wu, F. Farokhi, D. B. Smith and M. A. Kaafar, "Data Market and Game theory" (under preparation to be submitted to Privacy Enhancing Technologies Symposium)

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Data analysis methods using machine learning (ML) can unlock valuable insights for improving revenue or quality-of-service from, potentially proprietary, private datasets. Having large high-quality datasets improves the quality of the trained ML models in terms of the accuracy of predictions on , potentially untested data. The subsequent improvements in quality can motivate multiple data owners to share and merge their datasets in order to create larger training datasets. For instance, financial institutes may wish to merge their transaction or lending datasets to improve the quality of trained ML models for fraud detection or computing interest rates. However, government regulations (e.g., the roll-out of the General Data Protection Regulation in EU, the California Consumer Privacy Act or the development of the Data Sharing and Release Bill in Australia) increasingly prohibit sharing customer's data without consent [8]. Our work here is motivated by the need to conciliate the tension between quality improvement of trained ML models and the privacy concerns for data sharing. Game theory is the main method implemented in this thesis to tradeoff conflicts between accuracy and privacy level in privacy-aware data analysis.

Game theory has been widely used in wireless networks, such as cognitive radio, sensor networks, and mobile social networks [42]. It studies how rational players choose from a range of available strategies to obtain an optimised system resource allocation. The interaction among independent and self-interested players is analysed, especially in Equilibrium analysis and relative system performance [25]. In this thesis, we are interested in the tradeoff between accuracy level and privacy level in data analysis with differential privacy mechanism implemented, which is directly related to the value choice of privacy budget ϵ. So, it can be foreseen that game theory has the advantage in analysing the contrasting objectives for utility and privacy level from independent database curators.

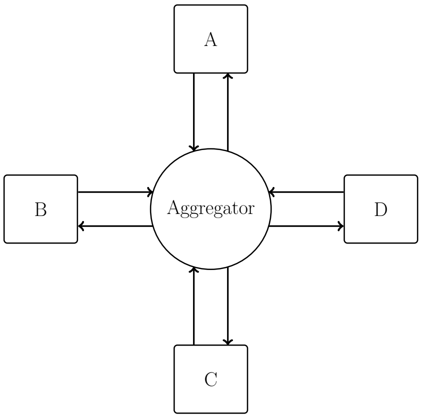


Figure 1.1: System model of multiple users sharing information at an Aggregator

To our knowledge, the existing methods for tradeoff between utility and privacy level are basically based on probability and other mathematical studies in privacy quantifications. Most of these existing methods take an economic view to look at the problem in order to obtain the value range for privacy budget ϵ [3,4,33]. Also, the system models used in these works assume an individual who holds a database with sensitive information and an attacker trying to reveal this. The cost functions for both these two parties are represented by ϵ through mathematical methods to estimate the probability of preserving or revealing the sensitive information in the database. Then, an upper limit is set for the total cost, which is a combination of the individual's cost and attacker's cost, so as to obtain the appropriate value of ϵ. However, these methods are not generic to fit in various data types and

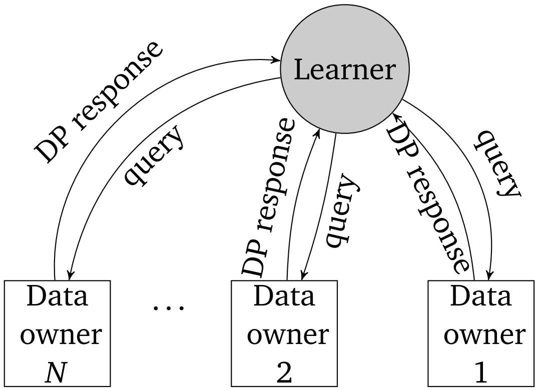


Figure 1.2: The communication structure between the learner and the distributed data owners for submitting queries and providing differentially-private (DP) responses.

queries, and the tradeoff is between only two parties. Instead, by introducing a game theoretical method, the players in the system can be enabled to make their choice of ϵ separately. At the same time, the players in the game are not limited to just an individual and an attacker, but also extend to multiple users who are rational and may be semi-hostile to each other. The extended system model is shown in Figure 1.1, where A, B, C, D are database owners that share information at an Aggregator. Then, by carefully designing the game rule and relative risk-utility function, the game can reach an equilibrium outcome, where each player achieves their best response value of privacy budget to every other player.

In system models of existing works, the datasets used are simplified and limited to the value for one attribute. And the responded answer, after the randomised mechanism is applied, is chosen from a finite number of available values. So, in this thesis, we significantly extend the state-of-art to a more realistic situation with multiple attributes in the dataset and where the answer is chosen from a continuous range of values. After this, a complete model for a system of privacy-aware machine learning is introduced.

We investigate a machine learning setup in which a learner wants to train a model based on multiple datasets from different data owners. In general machine learning with multiple datasets, the learner trains its model with gradients from all datasets. For the purpose of preserving privacy for data contributors, the learner can only submit queries to data owners and they respond by providing differentially-private (DP) responses as illustrated in Figure 1.2. In this thesis, the learner submits a gradient query to each data owner. Upon receiving DP responses from data owners to the gradient queries, the learner adjusts the parameters in the ML model in the direction of the average of the DP gradients. Therefore, the quality of the DP responses (in terms of the magnitude of the additive DP noise) from the data owners to the gradient queries determines the performance of the ML training algorithm.

An important parameter in the ML training algorithm is the step size, the amount by which the model parameters are adjusted in each iteration. If the fitness cost of the ML meets the assumptions of smoothness, strong convexity, and Lipschitz-continuity of the gradient, we can prove that, by selecting the step sizes to be inversely proportional with the iteration number and inversely proportional with the maximum number of iterations squared (see Algorithm 1 in Section 3.1), the difference between the fitness of the trained ML model using DP gradient queries and the fitness of the trained ML model in the absence of any privacy concerns becomes small. In fact, the magnitude of the difference becomes inversely proportional to the size of the training datasets squared and the privacy budgets of the data owners squared; see Theorem 2 in Section 3.2. Several ML models and fitness costs, such as linear and logistic regression, satisfy the above-mentioned assumptions. This enables us to predict the outcome of collaboration among privacy-aware data owners and the learner in terms of the fitness cost of the ML training model. However, if the fitness function does not meet these assumptions, we must select the step size to be inversely proportional to the square root of the iteration number. This way, the step size fades away much slower and the effect of the DP noise is more pronounced on the iterates of the learning algorithm. Therefore, we must add an averaging layer on top of the algorithm to reduce the negative impact of the DP noise; see Algorithm 2 in Section 3.1. This is based on the developments of [50] with appropriate changes in the averaging step to suit the ML problem with DP gradient queries. In this case, we can prove that the difference between the fitness of the trained ML model using DP gradient queries and the fitness of the trained ML model in the absence of any privacy concerns is inversely proportional to the size of the training datasets (no longer squared) and the privacy budget (no longer squared); see Theorem 3 in Section 3.2.

For experimental verification of the theoretical results, two financial datasets are used in this thesis. First, we use a regression model on a dataset containing information on loans made on Lending Club, a peer-to-peer lending platform [32], to automate the process of setting interest rates of loans. Second, we train a support vector machine for detecting fraudulent transactions based on a dataset containing transactions made by European credit card-holders in September 2013 [39]. We use the experiments to validate theoretical predictions and to gain important insights into the outcome of collaborations among privacy-aware data owners. For instance, even if the learner has access to one large dataset with relaxed privacy constraints, the performance of the trained ML model can be very bad if two small conservative datasets (i.e., very small privacy budgets) also contribute to the learning. Therefore, it is best to exclude smaller conservative datasets from collaboration. This is a *counter-intuitive observation as it clearly indicates that more data is not always good*, if it is obfuscated by conservative data owners. Larger, but conservative, datasets are sometimes worth including in the training as they do not degrade performance heavily with their conservative privacy budgets, yet improve the performance of the trained ML model because of their size.

In this thesis, we propose information theoretical methods to quantify utility and leakage in privacy-aware data analysis for two models. And in each model, we capture the trade-off between privacy and accuracy level. And we build a game-theoretical model of choosing the value for privacy budget ϵ, while optimising the utility of the shared databases to which the differential privacy mechanism is applied.

1.2 Thesis Contributions

This thesis makes the following contributions:

- We evaluate the existing works describing utility-leakage tradeoffs between a two parties model of an individual and an attacker, and introduce game-theoretic methods to solve accuracy-leakage level tradeoffs from differential privacy.

- We develop DP gradient descent algorithms for training ML models on distributed private datasets owned by different entities; see Algorithms 1 and 2 in Section 3.1.

- We prove that the quality of the trained ML model using DP gradient descent algorithm scales inversely with privacy budgets squared, and the size of the distributed datasets squared, which can establish a trade-off between privacy and utility in privacy-preserving ML; We develop a theory that enables to predict the outcome of a potential collaboration among privacy-aware data owners (or data custodians) in terms of the fitness cost of the ML training model prior to executing potentially computationally-expensive ML algorithms on distributed privately-owned datasets; see Theorems 2 and 3 in Section 3.2.

- We validate our theoretical analysis by evaluating our differentially private ML algorithms using distributed financial datasets belonging to multiple institutes/banks for determining interest rates of loans using regression, and for detecting credit card fraud using support vector machine classifier; We further validate the predictions of the analysis with the actual performance of the proposed privacy-aware learning algorithms applied to the distributed financial datasets; see Section 3.3.

- Our experimental results indicate that, in the case of three banks collaborating to train a support vector machine classifier to detect credit card fraud, within only 100 iterations, the fitness of the trained model using DP gradient queries is in average within 90% of the fitness of the trained model in the absence of privacy concern if the privacy budget is equal to 1 and each bank has access to a dataset of 30,000 records of credit card transactions and their validity. We observe similar performance results for training a regression model over interest rates of loans with the privacy budget of 10 and datasets of 350,000 records each.

- We use the concept of differential entropy to quantify the leakage that occurs in a trained machine learning model using an $\epsilon-$ differentially privacy gradient descent algorithm. By carefully study the interactions between each player in the system and how the chosen actions effect the payoff function of other players, we develop a compensation game and test Equilibrium outcomes.

Chapter 2

Literature Review

2.1 Privacy preserving machine learning with distributed datasets

ML using Secure Multi-Party Computation and Encryption. Secure multi-party computation provide avenues for securing the iterations of distributed ML algorithms across multiple data owners. In the past, secure multi-party computation has been used in various ML models, such as decision trees [36], regression [13], association rules [53], and clustering [30, 54]. Training ML models using encrypted data was discussed in [5, 9, 20, 29, 34]. In [19], efficient conversion of models for use of encrypted input data was discussed. The use of secure multi-party computation reduces the computational efficiency of ML algorithms by adding a non-trivial computational and communication performance overhead.

ML with Differential Privacy. A natural way for alleviating privacy concerns is to deploy privacy-enabled ML using differential privacy (DP) [11, 49, 55, 58]. In [11], a privacy-preserving regularized logistic regression algorithm is provided for learning from private databases by bounding the sensitivity of regularized logistic regression, and perturbing the learned classifier with noise proportional to the sensitivity. This technique is proved to be DP and simulations are used to investigate the trade-off between privacy and learning utility. In [55], a large class of optimization-based DP machine learning algorithms are developed by appropriately perturbing the objective function of the ML training algorithm. The mechanism is applied to linear and logistic regression models and shown to provide high accuracy. In the mentioned studies, privacy-preserving ML, however, often relies on an entire dataset, constructed by merging smaller datasets, being stored in one location. The ML model is then either trained on the aggregated dataset, and is systematically obfuscated using additive noise to guarantee differential privacy, or trained on an obfuscated centrally-located data. Such methods do not address the underlying problem that the smaller datasets are owned by multiple entities with restrictions on sharing sensitive data.

Distributed/Collaborative Privacy-Preserving ML. ML based on distributed private datasets has been recently investigated in, e.g., [21,27,28,45,56]. Note that this problem is intimately related to distributed optimization using differentially-private oracles, as such ML problems can be cast as distributed optimization problems in which distributed training datasets are represented within cost functions or constraints of the entities. Using stochastic gradient descent with additive Gaussian/Laplace noise to ensure DP is also common in the literature; (e.g., [1,41,51,57]). In [51], noisy gradients are used to train a deep neural network. The scale of the required additive noise for DP is reduced in [1] by employing the idea of moment accountant, instead of standard composition rules. Stochastic gradient descent is also utilized in [41] for recurrent neural network language models. Generalizations for obfuscating individual and group-level trends by DP additive noise are presented in [57]. Because iterative methods rely on multiple rounds of inquiries of private datasets, for instance, by submitting multiple gradient queries, the privacy budget must be inversely scaled by the total number of iterations to ensure that a reasonable privacy guarantee can be achieved (alternatively, privacy guarantees get weaker as the number of iterations grows because of the composition rule of differential privacy). Hence, if the parameters of the optimization algorithm are not carefully chosen, bounds on the performance of the ML training algorithm deteriorates with an increasing total number of iterations; e.g., see [24]. In [28, 56], the privacy budget was kept constant and therefore by communicating more, as the number of the iterations grows, the privacy guarantee weakens. However, in those studies, if the privacy budget had been scaled inversely proportional to the total number of iterations, privacy guarantees would be maintained over the entire horizon but performance would deteriorate with increasing total number of iterations, as in [24].

All these studies, however, do not address the issues of convergence of the learning algorithm, selection of appropriate step size in the stochastic gradient descent, and forecasting of the quality of the trained ML model based on the privacy budget prior to running extensive potentially computationally-expensive experiments. These missing steps are some of the important contributions of this thesis.

2.2 Game-theoretic methods for network security problem

In this section, existing results of utilising game-theoretic approaches to solve security problems in network are reviewed.

Game theory is a mathematical analysis tool for interactive players in a game [43]. In game theory, the basic elements used to describe a game are shown in Table 2.1. All players participate in the game by choose from a set of available strategies.

Table 2.1: Description for the basic elements in a game

Players	The interactive decision makers.
Actions	In each move of a player, an action is taken. The player is assumed to know the possible actions of each other.
Payoff	After every player has taken actions, the received return is payoff. A payoff could be either positive or negative.
Strategies	A player's strategy is the plan of an action that based on the knowledge of the action history. The strategies can be pure or mixed strategies.

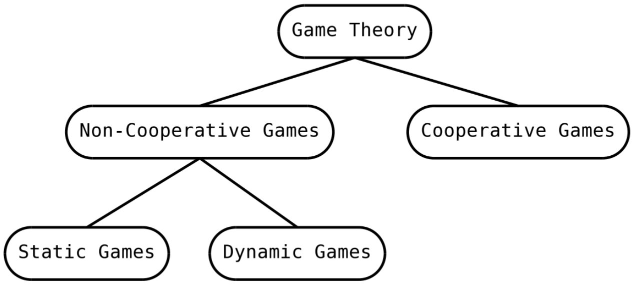


Figure 2.1: Basic classification of game theory [48]

Then, based on the utility function, the player evaluates the resulting outcome for each strategy. In non-cooperative game, players are rational and will seek for maximum utility by choosing the optimal strategy. In a game with multiple players, the action of one player will directly affect the utility of other players. Thus, the games can have a pure strategy where the decision is deterministic or mixed strategies where the decision follows a probability distribution. In this thesis, the games all use pure strategies. An Equilibrium in a game is a combination of the players' strategies so that each player's strategy is the best response to the strategies of the other players [18]. Such strategy leads to a maximum payoff given other players' strategies.

In [23], Hamilton et al. built a link between game theory and information warfare. Their investigation is initialised from tactical analysis in information warfare, which consists of the search technique and the evaluation function. It is stated that the player in a game uses the evaluation function to monitor the performance of strategies while the search technique is used in choosing from different moves. An important concept of mini-max is mentioned as a mostly used technique.

While there are significant advantages in using game theory in areas of information warfare, there are challenges in reality applications. In another work from Hamilton et al., the challenges in applying game theory to the domain of information warfare are discussed [22]. According to potential conflicts in straightforward implementation of common search techniques for player and opponent, the fundamental issues are classified into seven different scenarios. After detailed evaluations for each issue, the authors drew a conclusion that game theoretic techniques can be modified to fit in information warfare.

As investigated in [48] by Roy et al., it is demonstrated that there exist game-theoretic solutions facing with the challenges proposed in [22]. They confirmed that game theoretic approaches are promising to solve changing security threats in the cyber system. In this work, they evaluated the problem in areas of game theory. It is narrowed down to different types of games as shown in Figure 2.1. As stated by the authors, the existing research on security games focuses on non-cooperative games. They considered static games and dynamic games and discussed existing works in detailed in this survey. For static games, it is classified into two broad categories which are complete and imperfect information and incomplete and imperfect information as shown in Figure 2.2.

For complete and imperfect information, there is a work by Carin et al. [10]. In this work, the authors constructed an attack/protect economic model, and used QuERIES' approach to estimate the probabilities and costs, and then gave quantifications for both models. The work concentrates on protecting the critical intellectual property. The authors fitted the scenario into static game and proposed a Markov Decision Process to calculate theoretical results and computational algorithms. As the performance of QuERIES in small-scale simulations is positive, it is proved that the methodology has the ability to improve risk assessments with rational and capable attacks.

For incomplete imperfect information as shown in Figure 2.3, Liu et al. focused on pairs of attacking/defending nodes in ad hoc network [37] . The game used in their work is a two-player static Bayesian game between one potential attacking node and one defending node. For the attacking node, it uses two pure strategies: attack and not attack. While for the defending node, it also has two pure strategies: monitor

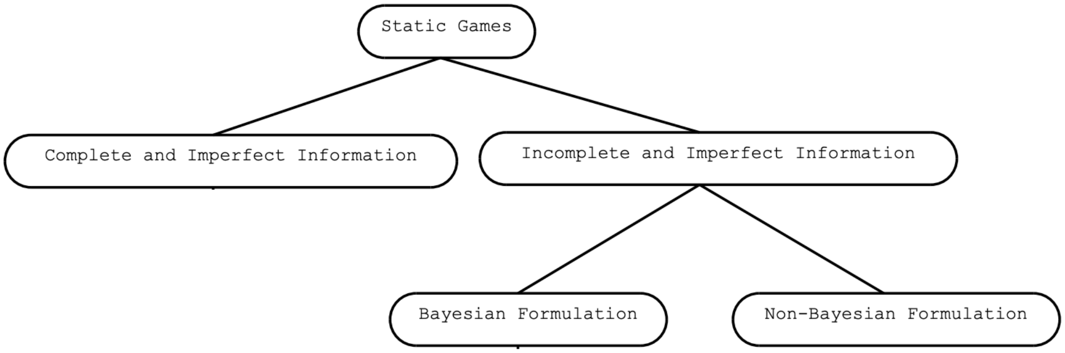

Static Games
Complete and Imperfect Information
Incomplete and Imperfect Information
Bayesian Formulation
Non-Bayesian Formulation

Figure 2.2: Classification of static games [48]

and not monitor. Costs and beliefs are assumed to be the common knowledge in the game. The authors stressed that in realistic models, the dynamic game is more practical for the defender to update its information of the opponent. The work investigated the Bayesian Nash Equilibria of the game and provided simulation results from intrusion detection system.

In [40], Manshaeu et al. highlighted the application of game theory in solving security and privacy problems in the network. It is noted that in this work, they collected and re-ordered existing various security or privacy problems with their types of game approach and main results in computer networks. They considered the situation when network nodes need to disclose some private information and to tradeoff between security and trust in the network. In this paper, they referred to a work by Raya et al in [46] and built game-theoretic models for privacy-preserving systems. Then, the authors proved that the strategy is a perfect Bayesian equilibrium of the game. They analysed that privacy loss of individual players is minimised while the trust-privacy tradeoff is optimised. This paper links privacy and game theory, and gives an inspiration to apply game-theoretic methods to network security problems. And it points out a way to apply game-theoretic approaches in determining ϵ under differential privacy in the network.

It should be noted that the players in a game may not always be able to get complete information of payoffs and strategies choices from opponent, such that it

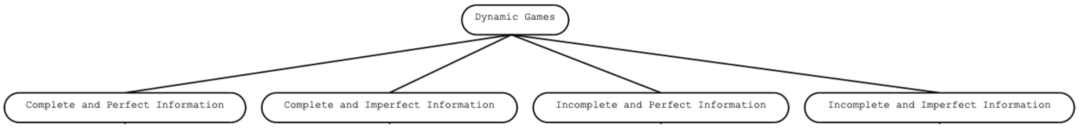

Dynamic Games
Complete and Perfect Information
Complete and Imperfect Information
Incomplete and Perfect Information
Incomplete and Imperfect Information

Figure 2.3: Classification of dynamic games [48]

is rare for agents to be fully rational [40]. There is a problem in accuracy degree a player can obtain. So the application of incomplete and imperfect information theoretical game is a key direction to solve this problem. In addition, agents require more accuracy estimation of security game parameters, which needs distributed machine learning to provide services of detection, analysis and decisions comparing in network security.

In this thesis, the relationship between the agents of the network is supposed to be independent and rational, and they make decision competitively to each other. So, non-cooperative game is chosen to tradeoff risks and utility between the individuals. All players in the game will choose its strategy wisely so as to maximise its payoff. In [35], it is said that Nash Equilibriums could be considered as a kind of optimal strategies for the network users. A Non-cooperative Nash Equilibrium occurs when no single player in the game can improve its utility through a unilateral deviation. And for a two player zero sum game, the Nash equilibrium is a saddle-point equilibrium with a single objective function minimised by one player and maximised by the other. The goal is to find a way to design the game and the tradeoff function in an appropriate way, and to optimise the entire utility in the system.

2.3 Differential privacy

Payoff function plays an important role in game theory. It is vital to consider what quantification to choose to represent the utility and cost in the payoff function.

Differential privacy provides a measurement of privacy loss, normally denoted as ϵ. The algorithm of differential privacy mechanism is parameterised such that the privacy loss can be bounded by any desired value ϵ. The notion of differential privacy is proposed by Dwork in [16]. The fundamental idea of differential privacy is that no matter whether an individual is present or absent in the database, the answer from a randomised query should not be affected [14]. It has been defined in [16] that a randomised mechanism M is ϵ-differentially private if for all of data sets $\mathscr{D}_1$ and $\mathscr{D}_2$ differing on no more than one row, and for any $S \subseteq Range(M)$,

$$Pr[M(\mathscr{D}_1) \in S] \leq exp(\epsilon) \times Pr[M(\mathscr{D}_2) \in S]. \tag{2.1}$$

In this equation (2.1), ϵ factor is a key element in limiting how much the probabilities difference is between the same answer received from two neighbouring databases that are differing on one entry. Intuitively, privacy budget ϵ is considered to be directly related to the accuracy of the database, because it determines the magnitude of the added noise. Hence, ϵ can be regarded as the degree of privacy level in some cases.

Laplace random mechanism is widely used in differential privacy. For the given query function $f(.)$ and randomised mechanism $M(.)$, the output from this channel is $M(\mathscr{X}) = f(\mathscr{X}) + \mathscr{Y}$, where $\mathscr{Y}$ is drawn i.i.d from $Lap(\frac{\triangle f}{\epsilon})$, and $\triangle f = max_{\mathscr{D}_1,\mathscr{D}_2}|f(\mathscr{D}_1) - f(\mathscr{D}_2)|$ is the global sensitivity of f. In circumstances where the reported answer has to be one of the several possible values, the probability distribution will be a discrete Laplace probability. Depends on the query, the reported answer may be either one value from a finite number of possible values or a certain integer number results from a counting query. Discrete Laplace mechanism suits such scenarios and its probability distribution function could be derived from continuous Laplace function. The probability of the reported answer to be k with ϵ differentially privacy is $P(\mathscr{Y} = k) = \frac{\frac{1}{2\sigma}e^{-|k|/\sigma}}{\sum_{j=-\infty}^{\infty}\frac{1}{2\sigma}e^{-|j|/\sigma}}$. On simplification, it becomes $P(\mathscr{Y} = k) = \frac{1-p}{1+p}p^{|k|}$, where $p = e^{-1/\sigma}$.

Smaller ϵ stands for higher privacy level and results in lower utility. However, for the same value of ϵ, if any of the following element changes, such as the size of the attributes, the size of dataset, or the number of possible values for one attributes, the probability of identifying an individual in a dataset will be consequently different. For example, when two datasets using the same value of privacy budget ϵ for differential privacy but only differing in data size, the larger dataset leaks more information than the smaller one. So, ϵ is related to the privacy level but is not able to give an absolute measurement of privacy in all situations. As a result, there requires a more generic metric to quantify privacy level.

2.4 Utility-Leakage tradeoffs in Attacker-Individual two-party models

In this section, we will evaluate attacker-individual two-party models and analyse the utility-leakage tradeoffs from two different aspects. The first one is by using information theoretical method to quantify utility and leakage in data analysis. The second one takes an economics view to analysis the probability of adversarial posterior belief and form a cost function to balance the controversy between the attacker and the individual.

2.4.1 Information theoretic methods for Quantifications

For some models with the output of the randomised query being specific values, discrete Laplace mechanism is used to fit this kinds of scenarios. There are several existing works on quantifying the utility and leakage and evaluating the tradeoffs from different perspectives. An information flow chart is used in [3] to demonstrate how the randomised function channel works. The dataset $\mathscr{X}$ is the input to the channel K. It first answers the query f and gives a real answer $\mathscr{Y}$. Then this real answer is randomised through a H randomisation mechanism and outputs a reported answer $\mathscr{Z}$. Hence, The utility of this oblivious mechanism is defined as

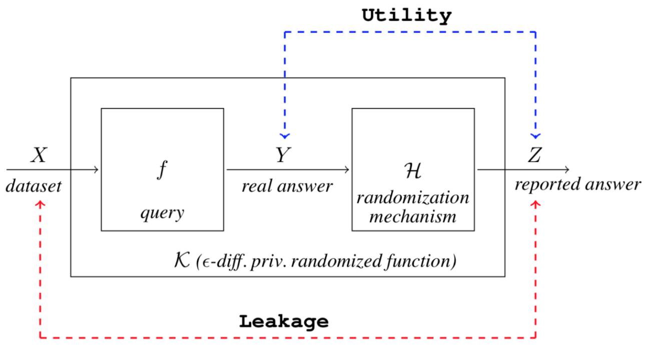


Figure 2.4: Information flow of leakage and utility for oblivious mechnisms [3]

the difference between the reported answer $\mathscr{Z}$ and the real answer $\mathscr{Y}$. The dataset leakage $L(\mathscr{X},\mathscr{Z})$ is the difference between $\mathscr{X}$ and $\mathscr{Z}$, and used to quantify the amount of information about the whole dataset leaked to the opponents. From the perspective of information theory, the utility and leakage in this system could be measured by looking into the similarity or entropy between the input database, real answer and randomised reported answer. The flow chart is shown as Figure 2.4.

A metric used for privacy leakage is based on mutual information study [2]. This privacy metric is based on evaluations of similarity between the information from the original and perturbed records. Based on quantitative information flow, the process of the original datasets being produced into the differentially private datasets is demonstrated. Differential privacy and mutual information is first compared by Alvim et al. in [4]. They quantified the information leakage based on the Rényi min-entropy information theory, and optimised the proposed randomisation mechanism for an increased utility, while preserving $\epsilon-$ differential privacy. $\mathscr{X}, \mathscr{Y}$denote the two random variables with carriers datasets $\mathscr{X}, \mathscr{Y}$, whose probability distributions are $p_X(.)$ and $p_Y(.)$, respectively. The Rénti entropy of order $\alpha(\alpha > 0, \alpha \neq 1)$ of a random variable $\mathscr{X}$ is defined as $H_\alpha(\mathscr{X}) = \frac{1}{1-\alpha} log2 \sum_{x\in\mathscr{X}} p(x)^\alpha$. For the case of $\alpha = \infty$, the above equation is derived as $H_\infty(\mathscr{X}) = -log2 \sum_{y\in\mathscr{Y}} p(y) max_{x\in\mathscr{X}} p(x|y)$. Based on this, the min-entropy leakage is defined as $I_\infty = H_\infty(\mathscr{X}) - H_\infty(\mathscr{X}|\mathscr{Y})$. This concept can be related to an attacker's model of the probability that the attacker's guess of the real answer is the same as the actual ones.

2.4.1.1 Quantification of Leakage

In [4], the correlation $L(\mathscr{X},\mathscr{Z})$ is used to measure the probability distribution of the information that the opponent can learn about the database by observing the reported answer $\mathscr{Z}$. It is thus qualified as $L(\mathscr{X},\mathscr{Z}) = I_\infty(\mathscr{X};\mathscr{Z})$ through min-entropy methods. Then, the paper shows that differential privacy will leads to a tight bound on both min-entropy leakage and utility. Because the bounds are tight in certain conditions, it will promisingly be applied in the thesis to quantify privacy level wisely. The bounds for min-entropy leakage $L(\mathscr{X};\mathscr{Z})$ is as follows,

$$I_\infty(\mathscr{X};\mathscr{Z}) \leq ulog2\frac{ve^\epsilon}{v-1+e^\epsilon}, \tag{2.2}$$

where u is the number of individuals and v is the number of possible values for a response. The results show that the min-entropy leakage of a randomised mechanism K is tightly bounded.

2.4.1.2 Quantification of Utility

The idea of the utility comes from the probability of a successful guess for the real answer from the reported one. That is for each report answer z, the user remaps the guess to a value $y' \in \mathscr{Y}$ with a remapping function $\rho(z) : \mathscr{Z} \rightarrow \mathscr{Y}$. The expectation utility for between is given as $U(\mathscr{Y},\mathscr{Z}) = \sum_{y,z} p(y,z)g(y,\rho(z))$, where $p(y,z)$ is the mapping function from $\mathscr{Y}$ to $\mathscr{Z}$ and $g(.)$ is gain function.

The binary gain function is a common gain measuring the difference between the remapping from $\mathscr{Z}$ back to $\mathscr{Y}'$ and $\mathscr{Y}$. The gain is 1 if the guess is exact the real answer and is 0 for all other guess. For conditions when the actual distance between $\mathscr{Y}'$ and $\mathscr{Y}$ is sensitive, loss functions which measure the distance between real answer and reported answer is used. In [4], binary gain is used. By substituting g with binary gain, the utility function is $U(\mathscr{Y},\mathscr{Z}) = \sum_{y,z} p(y,z)\delta_y(\rho(z))$.

However, for generic situations, the utility function is as following:

$$U(\mathscr{Y},\mathscr{Z}) = \begin{cases} \sum\limits_{y,z} p(y,z)\|y-z\|, & z \text{ follows discrete Laplace distribution,} \\ \int_{y,z} f(y,z)dist(y,z), & z \text{ follows continuous Laplace distribution.} \end{cases}$$

where $dist(y,z)$ is the loss function measuring the distance between y' and $y \in \mathscr{Y}$.

Similar to the Leakage analysis, the utility with the above quantification with binary gain is proved to have a tight bound in [4], which is provided in the following,

$$U(\mathscr{X},\mathscr{Y}) \leq \frac{(e^{\epsilon})^n(1-e^{\epsilon})}{(e^{\epsilon})^n(1-e^{\epsilon})+c(1-(e^{\epsilon})^n)}, \tag{2.3}$$

where n is the maximum distance from y in $\mathscr{Y}$, and c is a natural number used for one edge of border range. The authors of [3] then constructed an optimal randomisation mechanism and increased the whole utility for one database under uniform prior distribution with truncated geometric mechanism. The upper bounds for the leakage and utility are observed as a monotonic function of privacy budget ϵ.

According to a paper by Kalantari et al., a robust privacy-utility tradeoffs is developed for an arbitrary set of finite-alphabet source distribution. In [31], privacy level is quantified by using differential privacy, and utility is quantified by expectation of Hamming distortion maximised over the set of distributions. Considering the uncertainty of the true distributions of the source set, utility is modelled as the maximum Hamming distortion over the entire source set. As Hamming distortion is a metric with adding noise for privacy and it could help determine whether the original data has been changed [17]. Kalantari et al. categorised source distributions into three possible classes with optimised different differential privacy mechanism respectively. Then, they demonstrated how the worst-case guarantee of differentially private information loss compares to average-case guarantee of mutual information leakage, and the context awareness improves the utility of differential privacy mechanisms. In addition, the upper bounds for mutual information leakage and Hamming distortion differential privacy leakage are compared.

In [3, 17, 31], the quantifications of data utility and leakage are defined for discrete queries. In this thesis, we extend to continuous queries and multiple parties model, so as to make the solution more generic.

2.4.2 An Economics view for Utility Cost quantifications

There are existing methods of choosing ϵ for differential privacy from the perspective of economics views.

Everyone in the network contributes its information and benefits from accessing others' shared information. So, privacy level could be considered as a cost. Lee and Clifton revealed that though the privacy parameter ϵ in differential privacy is used to quantify the information loss of sensitive data in a database, it is not a sheer measurement for most realistic cases. In [33], they argued that the privacy guarantees of privacy differentially mechanism datasets are various for the same value of ϵ. So, instead of arbitrarily choosing the value of privacy parameter ϵ, Lee and Clifton propose a method by investigating the probability of an adversary correctly guess whether an individual is absence or presence in a database. They built an adversary model with an attacker who has full access to all records in the universe U but not aware of which individual is messing in the database $\mathscr{X}'$. Then, based on this assumption, they introduced a definition of adversary posterior belief in an attack model. They proposed a notion for adversary posterior belief $\beta(\omega)$ as $\beta(\omega) = P(\mathscr{X}' = \omega|\gamma) = \frac{P(M(\omega)=\gamma)}{\sum_{\psi\in\Psi} P(M(\psi)=\gamma)}$. This is the probability of the adversary's posterior belief on the possible word to be ω with the query response $\gamma = M(\mathscr{X}')$. This is used by the attacker to find which possible word has the largest possibility to be the real answer. It could be found out that the lowering of ϵ reduces the utility of the answer. So Lee and Clifton used this feature to find the proper value of ϵ in this adversary model.

To get an upper bound on the adversary probability of a real answer, they made assumptions of the worst case and the bound is as follows,

$$\beta(\omega) \leq \frac{1}{1+(n-1)e^{-\frac{\epsilon\triangle v}{\triangle f}}}. \tag{2.4}$$

Then, they determined the right value of ϵ by maintaining the upper bound on this adversary's posterior belief below a given threshold. ϵ controls how much an adversary's belief on a certain word can change. After observing the output of a privacy mechanism, this belief is updated as a Bayesian agent. In this work [33], the authors could determine the range of ϵ value under certain conditions with constraints.

Based on [33], another economic model is proposed by Hsu et al. In [26], which enables users of differential privacy to choose ϵ in a more principled approach. Instead of considering only the attackers' perspective, they proposed a two-party model with an individual and an analyst. The Individual is likely to contribute its information with a payment, while the analyst's problem is how accuracy the shared information is. A cost function is introduced under differential privacy for an individual who might want to contribute its information and a real-valued accuracy function for an analyst curious about the individual's information. Then, they combine the two views to derive the expected cost to the individual and so as to determine ϵ. This is the first comprehensive two party model. The authors then compared the true cost of privacy with a non-privacy study. As differential privacy requires additional noise to protect sensitive information while give adequate accuracy, privacy studies is expected to cost more than non-privacy studies. So when the budget is the same, the privacy study is better than non-privacy studies as the latter has no guarantees on privacy.

The authors also compared the necessity of the complex model to the earlier attack models in [15]. In the previous work by Dwork et al., it had only one notion

ϵ for privacy measurement. However, in realistic cases, the individuals need to consider more complex conditions including whether to participate in the privacy study and how much cost it could pay for such a certain accuracy. So the model proposed by Hsu et al. is more practical and detailed to face with real events.

The model will be extended to an interactive multiple users model. Each individual is regarded as a semi-opponent by the others. Also, we will focus on the cost function and utility function used in the existing works. The methods used to measure leakage and utility level will be studied and modified in our extended models.

Chapter 3

Privacy preserving Federated Learning

In this Chapter, we apply machine learning in distributed private data owned by multiple data owners, entities with access to non-overlapping training datasets. We use noisy, differentially-private gradients to minimize the fitness cost of the machine learning model using stochastic gradient descent. We quantify the quality of the trained model, using the fitness cost, as a function of privacy budget and size of the distributed datasets to capture the trade-off between privacy and utility in machine learning. This way, we can predict the outcome of collaboration among privacy-aware data owners prior to executing potentially computationally-expensive machine learning algorithms. Then, the prediction of relative training loss could be used as a loss function in forming the utility function in 2.4.1.2.

3.1 Federated Learning with DP Gradient Queries

3.1.1 Setup

Consider a group of $N \in \mathbb{N}$ private agents or data owners $\mathcal{N} := \{1, \dots, N\}$ that are connected to a node responsible for training a ML model, identified as a learning agent, over an undirected communication graph as in Figure 1.2. Each agent has access to a set of private training data $\mathcal{D}_i := \{(x_i, y_i)\}_{i=1}^{n_i} \subseteq \mathbb{X} \times \mathbb{Y} \subseteq \mathbb{R}^{p_x} \times \mathbb{R}^{p_y}$, where x_i and y_i, respectively, denote inputs and outputs. Each data owner, for instance, could be a private bank/financial institution. In this case, the private datasets can represent information about loan applicants (such as salary, employment status, and credit rating[1]) as inputs and historically approved interest rates per annum by the bank (in percentage points) as outputs.

Assumption 1. *Private datasets are mutually exclusive, i.e.,* $\mathscr{D}_i \cap \mathscr{D}_j = \emptyset$ *for all* $i, j \in \mathscr{N}$.

Assumption 1 states that two identical records, equal in every possible aspect, cannot be in two or more datasets. This is a realistic assumption in many real-life applications, such as financial and energy data. For instance, across multiple banks and financial-service providers, transaction records (e.g. for purchasing goods) are unique by the virtue of timestamps, amounts, and the uniqueness of purchases for an individual. In energy systems, one household cannot transact (for purchasing power) with two or more energy retailers and thus its consumption pattern can only be stored by one retailer. The reasons behind this assumption are two-fold. First, to guarantee ϵ-differential privacy, we need to ensure that the records are not repeated so that an adversary cannot reduce the noise levels by averaging the reports containing information about repeated entries and thus exceeding ϵ (due to the composition rule for differential privacy). If the datasets had common entries, there would need to be a privacy-preserving mechanism for identifying those common entries without potential information leakage with respect to non-common entries, which is a daunting task. The mutually exclusive or non-overlapping nature of the datasets also results in statistical independence of additive privacy-preserving noise. This independence is extremely useful in computing the magnitude of the additive noise for forecasting the performance of privacy-aware learning algorithms.

The learning agent is interested in extracting a meaningful relationship between the inputs and outputs using ML model $\mathfrak{M} : \mathbb{X} \times \mathbb{R}^{p_\theta} \rightarrow \mathbb{Y}$ and the available training

[1] Categorical attributes, such as gender, can always be translated into numerical ones according to a rule.

datasets $\mathscr{D}_i$, $\forall i \in \mathscr{N}$, by solving the optimization problem in

$$\theta^* \in \operatorname*{arg\,min}_{\theta\in\Theta} \left[g_1(\theta) + \frac{1}{n}\sum_{j\in\mathscr{N}} \sum_{\{x,y\}\in\mathscr{D}_j} g_2(\mathfrak{M}(x;\theta), y) \right], \tag{3.1}$$

where $g_2(\mathfrak{M}(x;\theta), y)$ is a loss function capturing the "closeness" of the outcome of the trained ML model $\mathfrak{M}(x;\theta)$ to the actual output y, $g_1(\theta)$ is a regularizing term, $n := \sum_{\ell\in\mathscr{N}} n_\ell$, and $\Theta := \{\theta \in \mathbb{R}^{p_\theta} \,|\, \|\theta\|_\infty \leq \theta_{\max}\}$. Note that a large enough $\theta_{\max}$ can always be selected such that the search over Θ does not add any conservatism (in comparison to the unconstrained case), if desired. We use $f(\theta)$ to denote the cost function of (3.1) for the sake of the brevity of the presentation, i.e.,

$$f(\theta) := g_1(\theta) + \frac{1}{n} \sum_{\{x,y\}\in\bigcup_{j\in\mathscr{N}}\mathscr{D}_j} g_2(\mathfrak{M}(x;\theta), y). \tag{3.2}$$

Remark 1 (Generality of Optimization-Based ML). *In an automated loan assessment example, a bank maybe interested in employing a linear regression model to estimate the interest rate of the loans based on attributes of customers (thus developing an "AI platform" for loan assessment and delivery). A linear regression model, as the name suggests, considers a linear relationship between input x and output y in the form of $y = \mathfrak{M}(x;\theta) := x^\top\theta$, where $\theta \in \mathbb{R}^{p_\theta}$ is the parameter of the ML model. We can train the regression model by solving the optimization problem* (3.1) *with $g_2(\mathfrak{M}(x;\theta), y) = \|y - \mathfrak{M}(x;\theta)\|_2^2$, and $g_1(\theta) = 0$. In addition to linear (or non-linear) regression discussed earlier, which clearly is of the form in* (3.1)*, several other ML algorithms follow this formulation. Another example is linear support vector machines (L-SVM). In this problem, it is desired to obtain a separating hyper plane of the form $\{x \in \mathbb{R}^{p_x} : \theta^\top[x^\top\ 1]^\top = 0\}$ with its corresponding classification rule $\mathrm{sign}(\mathfrak{M}(x;\theta))$ with $\mathfrak{M}(x;\theta) := \theta^\top[x^\top\ 1]^\top$ to group the training data into two sets (corresponding to $y = +1$ and $y = -1$). This problem can be cast as* (3.1) *with $g_1(\theta) := (1/2)\theta^\top\theta$ and $g_2(\mathfrak{M}(x;\theta), y) := \max(0, 1 - \mathfrak{M}(x;\theta)y)$. We can easily see that the extension to non-linear SVM can also be cast as an optimization-based ML problem. Another example* *is artificial neural network (ANN). In this case,* $\mathfrak{M}(x;\theta)$ *describes the input-output behaviour of the ANN with* θ *capturing parameters, such as internal thresholds. This problem can be cast as* (3.1) *with* $g_1(\theta) := 0$ *and* $g_2(\mathfrak{M}(x;\theta), y) := \|y - \mathfrak{M}(x;\theta))\|_2$.

If the data owners could come to an agreement to share private data (and it was not illegal to disclose customers' private information without their consent), the learning agent could train the ML model by solving the optimization problem (3.1) directly. In practice, however, data owners may not be able to share their private data. In this case, the learning agent can submit queries $\mathfrak{Q}_i(\mathcal{D}_i;k) \in \mathcal{Q}$ to agent $i \in \mathcal{N}$ for $k \in \mathcal{T} := \{1,\dots,T\}$, where T denotes the number of communication rounds (i.e., the number of queries) agreed upon by all the data owners prior to the exchange of information, index k identifies the current communication round, and $\mathcal{Q}$ denotes the output space of the query. Agent $i \in \mathcal{N}$ can then provide a differentially-private response $\overline{\mathfrak{Q}}_i(\mathcal{D}_i;k) \in \mathcal{Q}$ to the query $\mathfrak{Q}_i(\mathcal{D}_i;k) \in \mathcal{Q}$.

Definition 1 (Differential Privacy)**.** *The response policy of data owner* $\ell \in \mathcal{N}$ *is* ϵ_ℓ*-differentially private over the horizon* T *if*

$$\mathbb{P}\Big\{(\overline{\mathfrak{Q}}_\ell(\mathcal{D}_\ell;k))_{k=1}^T \in \mathcal{Y}\Big\} \leq \exp(\epsilon_\ell)\mathbb{P}\Big\{(\overline{\mathfrak{Q}}_\ell(\mathcal{D}'_\ell;k))_{k=1}^T \in \mathcal{Y}\Big\},$$

where $\mathcal{Y}$ *any Borel-measurable subset of* $\mathcal{Q}^T$ *is the range for all outcomes of privacy mechanism, and* $\mathcal{D}_\ell$ *and* $\mathcal{D}'_\ell$ *are two adjacent datasets differing at most in one entry, i.e.,* $|\mathcal{D}_\ell \setminus \mathcal{D}'_\ell| = |\mathcal{D}'_\ell \setminus \mathcal{D}_\ell| \leq 1$.

The learning agent then processes all the received responses to the queries in order to generate its ML model: $\hat{\theta} := \varsigma((\overline{\mathfrak{Q}}_j(\mathcal{D}_j;k))_{k\in\mathcal{T},j\in\mathcal{N}})$, where $\varsigma : \prod_{k\in\mathcal{T}} \mathcal{Q}^T \to \mathbb{R}^{p_\theta}$ is a mapping used by the learning agent for fusing all the available information.

In the next subsection, we present an algorithm for generating queries, and then use the provided differentially-private responses for computing a trained ML model.

3.1.2 Algorithm

In the absence of privacy concerns, one strategy for training the ML model by the learning agent is to provide unfettered access to the original private data of the data owners in $\mathscr{N}$. In this case, the learning agent can follow the projected (sub)gradient descent iterations in

$$\theta[k+1] = \Pi_{\Theta}[\theta[k] - \rho_k \xi_f(\theta[k])], \tag{3.3}$$

where $\rho_k > 0$ is the step-size at iteration k, $\xi_f(\theta[k])$ is a sub-gradient, an element of sub-differentials $\partial_\theta f(\theta[k])$, of the cost function f with respect to the variable θ evaluated at $\theta[k]$ [52], and $\Pi_\Theta[\cdot]$ denotes projection operator into the set Θ defined as $\Pi_\Theta[a] := \arg\min_{b\in\Theta} \|a-b\|_2$. For continuously differentiable functions, the gradient is the only sub-gradient. The use of sub-gradients, instead of gradient in this thesis, is motivated by the possible choice of non-differentiable loss functions in ML, e.g., the cost function of the L-SVM.

We assume that g_1 and g_2 are convex functions of θ. This implies that f is also a convex function of θ. The existence of sub-differentials is guaranteed for convex functions [52]. We define $\bar{g}_2^{x,y}(\theta) = g_2(\mathfrak{M}(x;\theta), y)$. The update law in (3.1) can be rewritten as $\theta[k+1] = \Pi_\Theta[\theta[k] - \rho_k \xi_{g_1}(\theta[k]) - \frac{\rho_k}{n}\sum_{\ell\in\mathscr{N}_j\setminus\{j\}} n_\ell \mathfrak{Q}_\ell(\mathscr{D}_\ell;k)]$, where ξ_{g_1} is a sub-gradient of g_1, $\xi_{\bar{g}_2^{x,y}}$ is a sub-gradient of $\bar{g}_2^{x,y}$, and $\mathfrak{Q}_\ell(\mathscr{D}_\ell;k)$ is a query that can be submitted by the learning agent to data owner $\ell \in \mathscr{N}$ in order to provide the aggregate sub-gradient: $\mathfrak{Q}_\ell(\mathscr{D}_\ell;k) = \frac{1}{n_\ell}\sum_{\{x,y\}\in\mathscr{D}_\ell} \xi_{\bar{g}_2^{x,y}}(\theta[k])$. Responding to the query $\mathfrak{Q}_\ell(\mathscr{D}_\ell;k)$ clearly intrudes on the privacy of the individuals in dataset $\mathscr{D}_\ell$. Therefore, data owner ℓ only responds in a differentially-private manner by reporting the noisy aggregate:

$$\overline{\mathfrak{Q}}_\ell(\mathscr{D}_\ell;k) = \mathfrak{Q}_\ell(\mathscr{D}_\ell;k) + w_\ell[k], \tag{3.4}$$

where $w_\ell[k]$ is an additive noise to establish differential privacy with privacy budget

Algorithm 1 ML training algorithm with distributed private datasets using DP gradients for strongly-convex smooth fitness cost.

```
Require: $T$
Ensure: $(\theta[k])_{k=1}^{T}$
Initialize $\theta[1]$
for $k = 1, \ldots, T-1$ do
    Learner submits query $\mathfrak{Q}_\ell(\mathscr{D}_\ell; k)$ to data owners in $\mathcal{N}$
    Data owners return DP responses $\overline{\mathfrak{Q}}_\ell(\mathscr{D}_\ell; k)$
    Learner follows the update rule
        $$\theta[k+1] = \theta[k] - \frac{\rho}{T^2 k}\Big(\xi_{g_1}(\theta[k]) + \sum_{\ell \in \mathcal{N}} \frac{n_\ell}{n} \overline{\mathfrak{Q}}_\ell(\mathscr{D}_\ell; k)\Big),$$
end for
```

ϵ_ℓ over the horizon T; see Definition 1. As stated before, here, the horizon T is the total number of iterations of the projected sub-gradient algorithm. Note that each neighbour responds to one query in each iteration.

We assume that $\Xi := \max_{(x,y)\in\mathbb{X}\times\mathbb{Y}} \left\|\xi_{\bar{g}_2^{x,y}}(\theta[k])\right\|_1 < \infty$. This implies the gradients or the sub-gradients of fitness function have a bounded magnitude.

Theorem 1. *The policy of data owner ℓ in* (3.4) *for responding to the queries is ϵ_ℓ-differentially private over horizon $\{1, \ldots, T\}$ if $w_\ell[k]$ are i.i.d.*[2] *noises with the density function $p(w) = (\frac{1}{2b})^{p_\theta} \exp(-\frac{\|w\|_1}{b})$ with scale $b = 2\Xi T/(n_\ell \epsilon_\ell)$.*

Proof. See Appendix A.1. □

Theorem 1 states that i.i.d. Laplace additive noise can ensure DP gradients. Each response in (3.4), for a given k, using the additive noise density in Theorem 1 is (ϵ_ℓ/T)-differentially private. Therefore, over the whole horizon $\{1, \ldots, T\}$, all the responses meet the definition of ϵ_ℓ-differential privacy. This follows from the composition of T differentially-private mechanisms [16]. In [28, 56], each response is constructed to ensure ϵ-differential privacy, which implies that the

[2] independently and identically distributed

overall algorithm is ϵT-differentially private, thus reducing the privacy guarantee with increasing the number of the iterations.

In the presence of the additive noise, the iterates of the learner follow the stochastic map

$$\theta[k+1] = \Pi_{\Theta}[\theta[k] - \rho_k(\xi_f(\theta[k]) + w[k])], \tag{3.5}$$

where $w[k] := \frac{1}{n}\sum_{\ell\in\mathcal{N}} n_\ell w_\ell[k]$.

Algorithm 1 summarizes our proposed ML algorithm with distributed private datasets using DP gradients. In Chapter 3.2, we observe that the performance of Algorithm 1 can only be assessed under the assumptions of differentiability, smoothness, and strong convexity of the fitness cost. These assumptions are satisfied for several ML models and fitness costs, such as regression. To avoid these assumptions and to also reduce the effect of the additive noise, we can define the averaging variable

$$\bar{\theta}[k+1] = \frac{k-1}{1/\sqrt{T}+k}\bar{\theta}[k] + \frac{1/\sqrt{T}+1}{1/\sqrt{T}+k}\theta[k]. \tag{3.6}$$

Algorithm 2 summarizes the proposed ML algorithm with distributed private datasets using DP sub-gradients with the additional averaging step as per equation (3.6). Now, we are ready to analyze the performance our privacy-preserving ML training algorithms.

3.2 Predicting the Performance of ML on Distributed Private Data

For Algorithm 1, we can prove the following convergence result under the assumptions of differentiability, smoothness, and strong convexity of the ML fitness function.

Theorem 2. *Assume that f is a L-strongly convex continuously-differentiable function*

Algorithm 2 ML algorithm with distributed private datasets using DP sub-gradients.

```
Require: $T$, $c_1$
Ensure: $(\theta[k])_{k=1}^{T}$
Initialize $\theta[1]$ within $\Theta$
for $k = 1, \dots, T-1$ do
    Learner submits query $\mathfrak{Q}_\ell(\mathscr{D}_\ell; k)$ to data owners in $\mathcal{N}$
    Data owners return DP responses $\overline{\mathfrak{Q}}_\ell(\mathscr{D}_\ell; k)$
    Learner follows the update rule
        $\theta[k+1] = \Pi_\Theta\left[\theta[k] - \frac{c_1}{\sqrt{k}}\left(\xi_{g_1}(\theta[k]) + \sum_{\ell \in \mathcal{N}} \frac{n_\ell}{n} \overline{\mathfrak{Q}}_\ell(\mathscr{D}_\ell; k)\right)\right],$
    Learner follows the averaging rule
        $\bar{\theta}[k+1] = \frac{k-1}{1/\sqrt{T}+k}\bar{\theta}[k] + \frac{1/\sqrt{T}+1}{1/\sqrt{T}+k}\theta[k].$
end for
```

with λ-Lipschitz gradient and $\theta_{\max} = \infty$ (i.e., there is no constraint). For any $\varepsilon > 0$, there exists a large enough T such that the iterates of Algorithm 1 satisfy

$$\min_{1 \le k \le T} \mathbb{E}\{f(\theta[k])\} - f(\theta^*) \le \frac{8\Xi^2 \rho}{Ln^2}\left(\sum_{\ell \in \mathcal{N}} \frac{1}{\epsilon_\ell^2}\right) + \varepsilon, \tag{3.7}$$

and

$$\min_{1 \le k \le T} \mathbb{E}\{\|\theta[k] - \theta^*\|_2^2\} \le \frac{32\Xi^2 \rho}{L^2 n^2}\left(\sum_{\ell \in \mathcal{N}} \frac{1}{\epsilon_\ell^2}\right) + \frac{\varepsilon}{4L}. \tag{3.8}$$

Proof. See Appendix A.2. □

Theorem 2 establishes the convergence of Algorithm 1 for smooth strongly convex functions. This quantifies the *trade-off between privacy and utility* by capturing the closeness to the trained ML model with and without taking into account the privacy constraints of the data owners. In fact, the inequalities in (3.7) and (3.8) enable us to predict the outcome of a potential collaboration among privacy-aware data owners (or data custodians) in terms of the fitness cost of the ML training model prior to executing potentially computationally-expensive ML algorithms on distributed

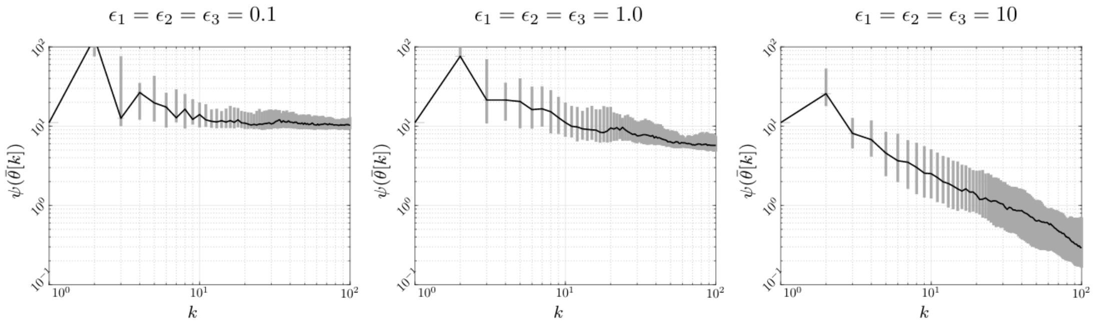


Figure 3.1: Statistics of relative fitness of the stochastic gradient method in Algorithm 2 for learning lending interest rates versus the iteration number for $T = 100$ with various choices of privacy budgets. The boxes, i.e., the vertical lines at each iterations, illustrate the range of 25% to 75% percentiles for extracted from a hundred runs of the algorithm and the black lines show the median relative fitness.

privately-owned datasets

To relax the conditions required for convergence of the ML training, we can use Algorithm 2. In this case, we do not even need the fitness function to be differentiable because the algorithm uses sub-gradients, rather than gradients. For the noisy projected sub-gradient decent algorithm in Algorithm 2, the following result can be proved.

Theorem 3. *For any T, there exists large enough constants*[3] $c_1, c_2 > 0$ *such that the iterates of Algorithm 2 satisfy*

$$\mathbb{E}\{f(\bar{\theta}[T])\} - f(\theta^*) \leq \frac{c_2 \Xi}{n} \sqrt{\sum_{\ell \in \mathcal{N}} \frac{1}{\epsilon_\ell^2}}, \tag{3.9}$$

Further, if g_1 is a L-strongly convex function,

$$\mathbb{E}\left\{\left\|\bar{\theta}[T] - \theta^*\right\|_2^2\right\} \leq \frac{4c_2 \Xi}{Ln} \sqrt{\sum_{\ell \in \mathcal{N}} \frac{1}{\epsilon_\ell^2}}. \tag{3.10}$$

Proof. See Appendix A.3. □

[3] Note that the constants in the statement of the theorem can be functions of T and, therefore, the bounds in (3.9) and (3.10) are useful for comparing the variations in the performance of the sub-gradient descent algorithm for various privacy budgets and sizes of the datasets as long as T is fixed.

The upper bounds on the performance of the training Algorithms 1 and 2 in Theorems 2 and 3 are increasing functions of $(1/n^2)\sum_{\ell\in\mathcal{N}}1/(\epsilon_\ell)^2$ and $(1/n)[\sum_{\ell\in\mathcal{N}}1/(\epsilon_\ell)^2]^{1/2}$, respectively. By increasing ϵ_ℓ, i.e., relaxing the privacy guarantees of data owners, the performance of the ML training algorithm improves, as expected because of having access to better quality gradient oracles.

Remark 2 (Comparison with Central Bounds). *Under the assumption that all the data owners have equal privacy budgets $\epsilon_i = \epsilon$, $\forall i$, the bound in* (3.7) *scales as ϵ^{-2} and the bound in* (3.9) *scales as ϵ^{-1}. These bounds are in line with the lower and the upper bounds in [7] for strongly convex and general convex loss functions. The same outcome also holds if $N = 1$ and $\epsilon_1 = \epsilon$, which is the case of centralized privacy-preserving learning.*

Finally, we note that these results provide bounds on the distance between the non-private ML model and the privacy-preserving ML models learned in a distributed manner as a function of the privacy budgets and the size of the datasets. Issues, such as non-independent and non-identical datasets, influence the performance of the non-private model and thus also indirectly influence the performance of the privacy-preserving models. In the next section, although the datasets are not restricted be i.i.d. (e.g., the number of fraudulent transactions in the credit card fraud detection is low and arguably contains activities that have originated from same/similar fraudsters), the theoretical bounds tightly match the experimental results.

3.3 Experimental Validation of the ML Performance

In this section, we examine the results of this chapter, specifically the performance of Algorithm 2, on two financial datasets on lending and credit card fraud. Particularly, we use the relative fitness of the iterates in Algorithm 2 to illustrate its performance. The relative fitness of θ is given by$\psi(\theta) := \frac{f(\theta)}{f(\theta^*)} - 1$.

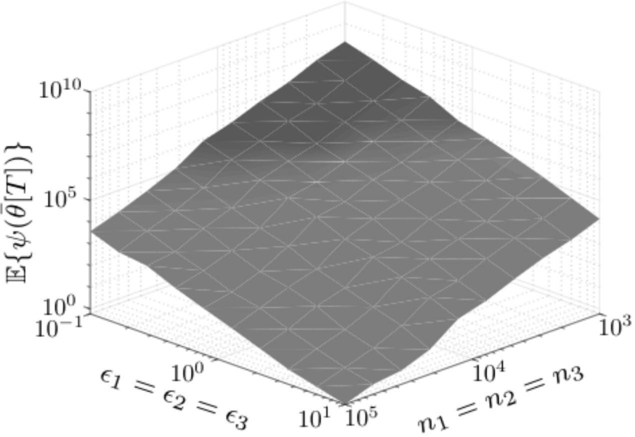


Figure 3.2: Relative fitness of the stochastic gradient method in Algorithm 2 for learning lending interest rates after $T = 100$ iterations versus the size of the datasets and the privacy budgets.

This measure shows how good θ is in comparison to the optimal ML model θ^* in terms of the training cost in (3.1). We opt for studying the relative fitness, scaled by $f(\theta^*)$ as opposed as the absolute fitness $f(\theta) - f(\theta^*)$, because we consider datasets with different sizes for two distinct ML learning models and thus we want to factor out the effects of the variations of $f(\theta^*)$. Finally, note that, by construction, $\psi(\theta) \geq 0$. Further, the lower the value of $\psi(\theta)$, the better θ performs in comparison to θ^*.

3.3.1 Lending Dataset

First, we use a lending dataset with a linear regression model to demonstrate the value of the methodology and to validate the theoretical results.

3.3.1.1 Dataset Description

The dataset contains information regarding nearly 890,000 loans made on a peer-to-peer lending platform, called the Lending Club, which is available on Kaggle [32]. The inputs contain loan attributes, such as total loan size, and borrower information, such as number of credit lines, state of residence, and age. The outputs are the

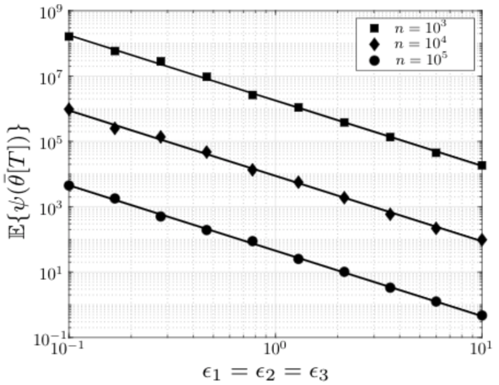


Figure 3.3: Relative fitness of the stochastic gradient method in Algorithm 2 for learning lending interest rates after $T = 100$ iterations versus the privacy budgets. The solid line illustrate the bound in Theorem 2.

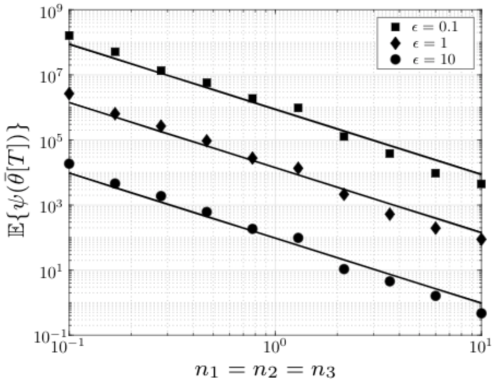


Figure 3.4: Relative fitness of the stochastic gradient method in Algorithm 2 for learning lending interest rates after $T = 100$ iterations versus the size of the datasets. The solid line illustrate the bound in Theorem 2.

interest rates of the loans per annum. We encode categorical attributes, such as state of residence and loan grade assigned by the Loan Club, with integer numbers. We also remove unique identifier attributes, such as id and member id, as well as irrelevant attributes, such as the uniform resource locator (URL) for the Loan Club page with listing data. Finally, we perform feature selection using the Principal Component Analysis (PCA) to select the top ten important features. This step massively improves the numerical stability of the algorithm.

For the PCA, we only use the last ten-thousand entries of the dataset to ensure that the feature selection does not violate the distributed nature of the algorithm.

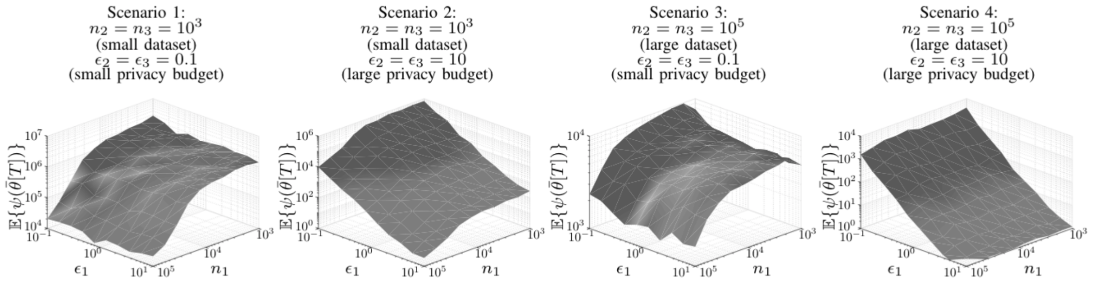


Figure 3.5: Relative fitness of the stochastic gradient method in Algorithm 2 for learning lending interest rates after $T = 100$ iterations versus the size of the dataset and the privacy budget of the first data owner for four distinct scenarios of collaboration.

Note that, if we were to use the entire dataset for the PCA, the data should have been available at one location for processing which is contradictory to the assumptions of the chapter regarding the distributed nature of the dataset and the privacy requirements of the data owners. After performing the PCA, the eigenvectors corresponding to the most important features are communicated to the distributed datasets. The first n_1 entries of the Lending Club are assumed to be the private data of the first data owner. The entries between $n_1 + 1$ to $n_1 + n_2$ belong to the second data owner and the entries between $n_1 + n_2 + 1$ to $n_1 + n_2 + n_3$ are with the third data owner. We may use any other approach for splitting the Lending Club dataset among the private data owners as long as the distributed datasets are not overlapping.

The data owners then balance their datasets using the-said eigenvectors. The eigenvectors, here, serve as a common dictionary between the data owners for communication and training.

3.3.1.2 Experiment Setup

The experiments demonstrate the outcome of collaborations among $N = 3$ financial institute s, e.g., banks, for training a ML model to automate the process of assigning interest rates to loan applications based on the attributes of the borrower and the

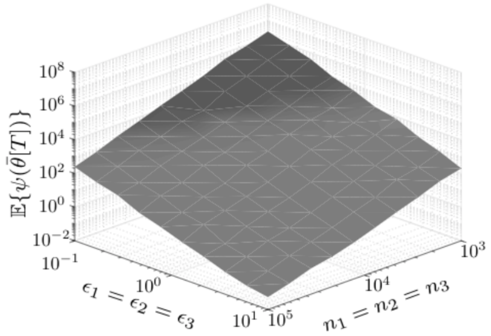


Figure 3.6: Relative fitness of the stochastic gradient method in Algorithm 2 for fraud detection after $T = 100$ iterations versus the size of the datasets and the privacy budgets.

loan. Each institute has access to a private dataset of n_i historical loan applications and approved interest rates. The value of ϵ_i for each institute essentially determines eagerness for collaboration and openness to sharing private proprietary datasets. For a linear regression model, we consider a linear ML model relating the inputs and the outputs as in $y = \mathfrak{M}(x;\theta) := \theta^\top x$ with $\theta \in \mathbb{R}^{p_\theta}$ denoting the parameters of the ML model. We train the model by solving the optimization problem (3.1) with $g_2(\mathfrak{M}(x;\theta), y) = \|y - \mathfrak{M}(x;\theta)\|_2^2$, and $g_1(\theta) = 0$.

3.3.1.3 Results

First, we demonstrate the behaviour (e.g., convergence) of the iterates of the stochastic gradient descent procedure in Algorithm 2. Consider the case where $n_1 = n_2 = n_3 = 250,000$. Figure 3.4 shows the statistics of the relative fitness of the stochastic gradient method in Algorithm 2 for a ML model determining lending interest rates, $\psi(\bar{\theta}[k])$, versus the iteration number k for $T = 100$ for three choices of privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. The algorithm is stochastic because the data owners provide differentially-private responses to the gradient queries, obfuscated with Laplace noise in Theorem 1. Thus each run of the algorithm follows a different relative fitness trend. The boxes, i.e., the vertical lines at each iterations, illustrate the range of 25% to 75% percentiles of the relative fitness extracted from one-

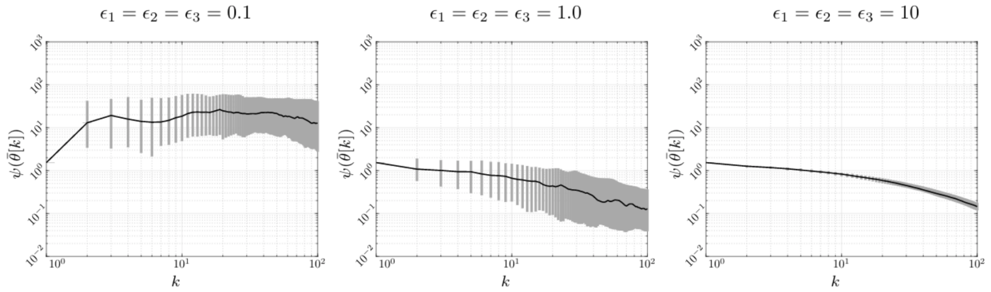


Figure 3.7: Statistics of relative fitness of the stochastic gradient method in Algorithm 2 for fraud detection versus the iteration number for $T = 100$ with various choices of privacy budgets. The boxes, i.e., the vertical lines at each iterations, illustrate the range of 25% to 75% percentiles for extracted from a hundred runs of the algorithm and the black lines show the median relative fitness.

hundred runs of the algorithm. The black lines show the median relative fitness versus the iteration number. The effect of the privacy budgets on the quality of the iterates at the end of T iterations is evident, as expected from Theorem 3. As ϵ increases, i.e., the data owners become more willing to share data, the performance of the trained ML model improves.

After establishing the desired transient behaviour of the algorithm, we can investigate the effect of the size of the datasets and the privacy budgets on the performance of the trained ML model, i.e., the ML model after all the iterations have passed. Figure 3.2 shows the expectation (i.e., the statistical mean) of the relative fitness of the stochastic gradient method in Algorithm 2 for the trained ML model after $T = 100$ iterations versus the size of the datasets $n_1 = n_2 = n_3$ and the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. As predicted by Theorem 3, the fitness improves as the size of the datasets $n_1 = n_2 = n_3$ and/or the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$ increase. To quantify the tightness of the upper-bound in Theorem 3 for Algorithm 2, we isolate the effects of the size of the datasets and the privacy budgets on the relative fitness. Figure 3.3 illustrates the expectation of the relative fitness of the stochastic gradient method in Algorithm 2 after $T = 100$ iterations versus the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. In this figure, the markers (i.e., ■, ♦, and ●) are

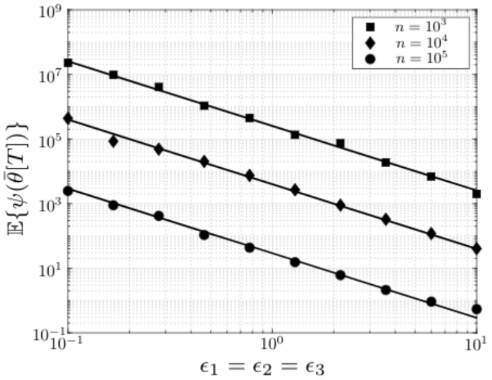


Figure 3.8: Relative fitness of the stochastic gradient method in Algorithm 2 for fraud detection after $T = 100$ iterations versus the privacy budgets. The solid line illustrate the bound in Theorem 2.

from the experiments and the solid lines are fitted to the experimental data. We can see that the slope of the linear lines in the log-log scale in Figure 3.3 is -2. This shows that $\psi(\bar{\theta}[k]) \propto \epsilon_i^{-2}$. Hence, our bound in Theorem 3 is not tight as it states that $\psi(\bar{\theta}[k])$ is upper bounded by a function of the form $1/\epsilon_i$. This is because Theorem 3 does not use the fact that the cost function for the regression is strongly convex and has Lipschitz gradients. These assumptions are utilized in Theorem 2 and the bounds in this theorem are in fact tight, as Theorem 2 states that $\psi(\bar{\theta}[k])$ is upper bounded by a function of the form $1/\epsilon_i^2$. Figure 3.4 shows the expectation of the relative fitness of the stochastic gradient method in Algorithm 2 after $T = 100$ iterations versus the size of the datasets $n_1 = n_2 = n_3$. Similarly, the slop of the linear lines in the log-log scale in Figure 3.4 is -2 pointing to that $\psi(\bar{\theta}[k]) \propto n_i^{-2}$. This is again a perfect match for our theoretical bound in Theorem 2 (because $n = n_1 + n_2 + n_3 = 3n_i$).

Finally, we consider a few scenarios of collaboration for the data owners. Specifically, we evaluate the performance of the learning algorithm for four distinct scenarios in which the second and the third data owners have: (*i*) small datasets and small privacy budgets (i.e., reluctant to share due to privacy concerns); (*ii*)

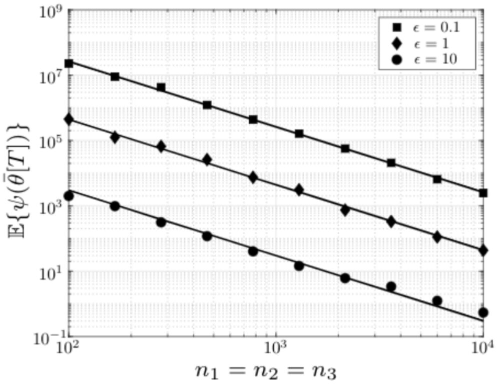


Figure 3.9: Relative fitness of the stochastic gradient method in Algorithm 2 for fraud detection after $T = 100$ iterations versus the size of the datasets. The solid line illustrate the bound in Theorem 2.

small datasets and large privacy budgets (i.e., eager to share); (*iii*) large datasets and small privacy budgets; (*iv*) large datasets and large privacy budgets. For each case, we vary the privacy budget and the size of the dataset of the first data owner. This allows us to investigate the potential benefit to data owners in various scenarios. Figure 3.5 illustrates the expectation of the relative fitness of the stochastic gradient method in Algorithm 2, after $T = 100$ iterations, versus the size of the dataset n_1 and the privacy budget ϵ_1 for four distinct scenarios of collaboration.

The first scenario in Figure 3.5 (the left most plot) shows that there is no point in collaboration with small data owners, even if the size of the dataset of the first data owner is large and it is eager to share its data. We could foresee this from the bound in Theorem 3 without running Algorithm 2. This bound shows that $\psi(\bar{\theta}[k]) \propto 1/(2000 + n_1)\sqrt{200 + 1/\epsilon_1^2}$; hence, no matter how large ϵ_1 gets (even if $\epsilon_1 = \infty$), the error's coefficient remains large due to small privacy budgets of the other two data owners and n_1 must become considerably large to compensate for it. In the second scenario (the second left most plot in Figure 3.5), the effect of ϵ_1 and n_1 are more pronounced. This is because, although the other two data owners are small, they do not hinder the learning process by adding large amounts

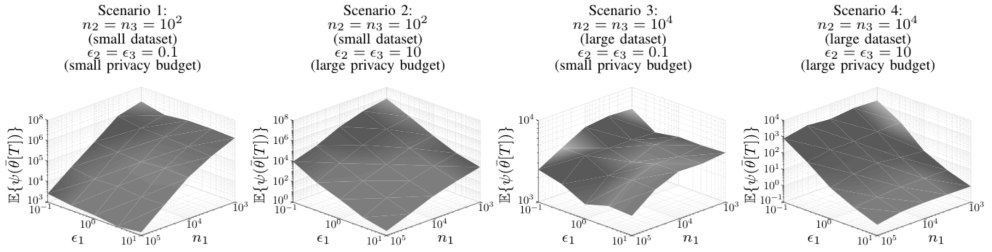


Figure 3.10: Relative fitness of the stochastic gradient method in Algorithm 2 for a trained ML model determining lending interest rates after $T = 100$ iterations versus the size of the dataset and the privacy budget of the first data owner for four distinct scenarios of collaboration.

of privacy-preserving noise because of their conservatively small privacy budgets. The third scenario is similar to the first one, albeit with better relative fitness as conservative data owners are relatively larger. The best scenario for collaboration, unsurprisingly, is the fourth scenario in which phenomenal performances can be achieved even without much consideration towards the size of the first dataset or its privacy budget as the other two datasets are large and eager to collaborate for learning.

3.3.2 Credit Card Fraud Detection

In this subsection, we use a credit card dataset with a L-SVM classifier to further demonstrate the value of the methodology and to validate the theoretical results.

3.3.2.1 Dataset Description

The datasets contains transactions made by European credit card holders in September 2013 available on Kaggle [39]. The inputs are vectors extracted by PCA (to avoid confidentiality issues) as well as the amount of the transaction. The output is a class, determining if the transactions was deemed fraudulent or not. The dataset is highly unbalanced, as the positive class (frauds) account for 0.172% of all transactions.

3.3.2.2 Experiment Setup

The experiments demonstrate the outcome of collaborations among $N = 3$ financial institutes for training a SVM classifier to detect fraudulent activities automatically and rapidly. Each institute has access to a private dataset of n_i historical credit card transactions and their authenticity. The value of ϵ_i for each institute determines eagerness for collaboration. In L-SVM, the model is $\mathfrak{M}(x;\theta) := \theta^\top[x^\top\ 1]^\top$, and $g_1(\theta) := (1/2)\theta^\top\theta$ and $g_2(\mathfrak{M}(x;\theta), y) := \max(0, 1-\mathfrak{M}(x;\theta)y)$.

3.3.2.3 Results

First, we investigate the transient behaviour of the iterates of Algorithm 2. Assume that $n_1 = n_2 = n_3 = 30,000$. Figure 3.7 shows the statistics of the relative fitness of the iterates of Algorithm 2 for training a fraud detection SVM classifier, $\psi(\bar{\theta}[k])$, versus the iteration number k for $T = 100$ for three choices of privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. The boxes, i.e., the vertical lines at each iterations, illustrate the range of 25% to 75% percentiles of relative fitness extracted from one-hundred runs of the algorithm and the black lines show the median relative fitness. As expected from Theorem 3, the performance of the trained SVM classifier gets closer to the SVM classifier trained with no privacy constraints θ^* as the privacy budgets increases.

Now, we can demonstrate the effect of the size of the datasets and the privacy budgets on the performance of the trained SVM classifier at the end of T training iterations. Figure 3.6 shows the expectation of the relative fitness of the stochastic gradient method in Algorithm 2 after $T = 100$ iterations versus the size of the datasets $n_1 = n_2 = n_3$ and the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. Similar to the theoretical results in Theorem 3, the fitness improves by increasing the size of the datasets $n_1 = n_2 = n_3$ and the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. We can also isolate the effects of the size of the datasets and the privacy budgets. Figure 3.8 illustrates the expectation of the relative fitness of the iterates of Algorithm 2 after

$T = 100$ iterations versus the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. As all linear slopes in the log-log scale in Figure 3.8 are -2, the bound in Theorem 2 seems to be a perfect fit. Figure 3.4 shows the expectation of the relative fitness of the iterates of Algorithm 2 after $T = 100$ iterations versus the size of the datasets $n_1 = n_2 = n_3$ revealing the exact behaviour predicted in the bound in Theorem 2.

Finally, we evaluate the performance of the learning algorithm for four distinct scenarios, in which the second and the third data owners have: (*i*) small datasets and small privacy budgets; (*ii*) small datasets and large privacy budgets; (*iii*) large datasets and small privacy budgets; (*iv*) large datasets and large privacy budgets. Figure 3.10 illustrates the expectation of the relative fitness of Algorithm 2 after $T = 100$ iterations versus the size of the dataset n_1 and the privacy budget ϵ_1 for four distinct scenarios of collaboration. The first scenario in Figure 3.10 (the left most plot) illustrates that there is no point in collaboration with small data owners even if the size of the dataset of the first data owner is large and it is eager to share its data. In the second scenario (the second left most plot in Figure 3.10), the effect of ϵ_1 and n_1 are more pronounced because the privacy budgets of the second and the third data owners are large and thus they do not degrade the performance of the learning algorithm by injecting excessive privacy-preserving noise. The third scenario is again similar to the first one, albeit with better results as conservative data owners are relatively larger. The best scenario for collaboration, similar to the loan example, is the fourth scenario in which the training performances with and without privacy constraints are identical, so long as the dataset of the first subsystem is large, or its privacy budget is not too small.

Chapter 4

Game Theory in Privacy preserving Machine Learning

As mentioned in the previous sections, the differential privacy mechanism used is simplified and min-entropy measures the probability of correct guessing for one value at one attribute of datasets. We will extend the model to a more realistic and complicated one. In this section, the problem is moved to solving the utility-leakage tradeoffs in privacy-aware machine learning data analysis. First, the system model of the machine learning is explained. Then, a real problem of data training with two financial databases is introduced. In this thesis, the utility is measured by the loss function in machine learning while the leakage is measured by differential entropy. By combining the utility and leakage quantifications, a compensation policy is proposed to build a utility-leakage game for the machine learning model.

4.1 System model

As shown in Figure 4.1, there are several database owners A, B, C, and D sharing information at aggregator. At each database, a finite set $Ind = 1, 2, ..., u$ of u individuals participate with a finite set $Val = v_1, v_2, ..., v_v$ of v different possible values for the sensitive attribute of each individual (e.g. disease-name in medical database). The absence of an individual from the database can be modelled with a special value in Val. Then, the database could be modelled as a u-tuple $\mathscr{D} = d_1, d_2, ..., d_u$, where $d_i \in Val$ is the value of individual i. $\mathscr{D}$ and $\mathscr{D}'$ are adjacent if

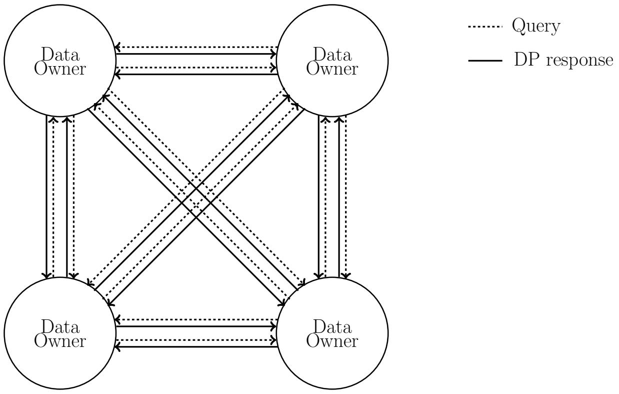


Figure 4.1: System model of multiple users sharing information by answering queries

they differ for the value of exactly one individual.

The process of information sharing between data holders is as following: Dataset $\mathscr{X}$ is input to the channel, it first gives a true answer of $\mathscr{Y}$ to the query f, where $\mathscr{Y} = Range(f)$. Then, this true answer is processed by a randomise channel and output a reported answer $\mathscr{Z}$, following the probability distribution of the channel. Intuitively, the correlation between $\mathscr{X}$ and $\mathscr{Z}$ measures how much information about the complete database the other could learn about by observing the reported answer $\mathscr{Z}$. So this could be regarded as the leakage of the channel. This could be quantified by using the min-entropy concept to calculate the mutual information between $\mathscr{X}$ and $\mathscr{Z}$. The correlation between $\mathscr{Y}$ and $\mathscr{Z}$ measures how much others can learn about the real answer from the reported answer. So this is regarded as the utility of the channel.

4.2 Compensation for Accessing Private Data

We assume that the learning agent is in a position to compensate its neighbours for softening their privacy constraints (i.e., using a larger ϵ_ℓ in ϵ_ℓ-differential privacy). We use the notation $\tau_\ell(\epsilon_\ell, \epsilon_{-\ell})$ to denote the compensation value of the learning agent to data owner $\ell \in \mathcal{N}$, where $\epsilon_{-\ell} := (\epsilon_j)_{j \in \mathcal{N} \setminus \{\ell\}}$.

Each data owner $\ell \in \mathcal{N}$ is strategic, self-interested, and wants minimize to its overall cost $V_\ell(\epsilon_\ell, \epsilon_{-\ell}) := wf \cdot U_\ell(\epsilon_\ell) - L_\ell(\epsilon_\ell, \epsilon_{-\ell})$, where wf is weighting factor to normalise units and utility and leakage value's magnitude. U_ℓ here is defined by the predicted relative loss function in Theorem 2 in Chapter 3.2, while L_ℓ is by using information theoretical method in Chapter 2.4.1.1.

This setup results in a game-theoretic problem. For any compensation policy $(\tau_\ell)_{\ell \in \mathcal{N}}$, a *privacy-compensation game* is defined by the tuple $(\mathcal{N}, (\mathbb{R}_{\geq 0})_{\ell \in \mathcal{N}}, (V_\ell)_{\ell \in \mathcal{N}})$ encoding the set of players[1] $\mathcal{N}$ each with action space $\mathbb{R}_{\geq 0}$ (positive real numbers from which ϵ_ℓ can be selected), and cost functions $V_\ell(\epsilon_\ell, \epsilon_{-\ell})$. Naturally, we are interested in studying the behaviour of data owners at the equilibrium, a set of behaviours from which no data owner is inclined to deviate unilaterally.

Definition 2 (Equilibrium). *$(\epsilon^*_\ell)_{\ell \in \mathcal{N}}$ constitutes an equilibrium of the privacy-compensation game with compensation policy $(\tau_\ell)_{\ell \in \mathcal{N}}$ if $\epsilon^*_\ell \in \arg\min_{\epsilon_\ell \geq 0} V_\ell(\epsilon_\ell, \epsilon^*_{-\ell})$ for all $\ell \in \mathcal{N}$. Let $\Upsilon((\tau_\ell)_{\ell \in \mathcal{N}})$ denote the set of all the equilibria with the payment policy $(\tau_\ell)_{\ell \in \mathcal{N}}$.*

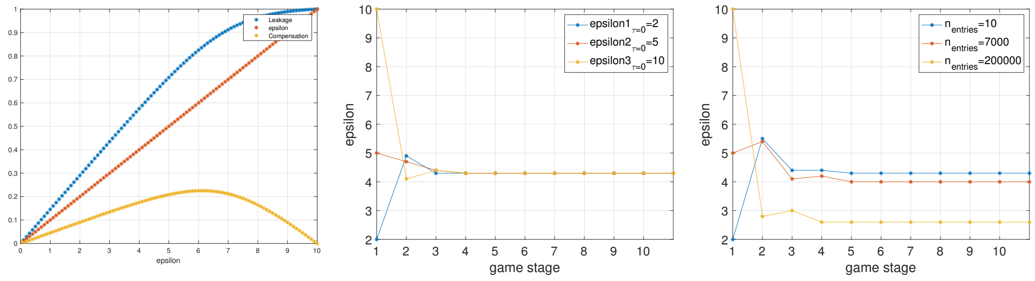
In general, the existence of a (pure strategy Nash) equilibrium for a privacy-compensation game as in Definition 2 can only be guaranteed if the cost functions $(V_\ell)_{\ell \in \mathcal{N}}$ are continuous and quasi-convex (in the decision variables of the corresponding data owners) [6, 12]. However, for a set of specific compensation policies, the existence of an equilibrium requires fewer conditions.

Proposition 1. *For any compensation policy $(\tau_\ell)_{\ell \in \mathcal{N}}$ such that $\tau_\ell(\epsilon_\ell, \epsilon_{-\ell}) = \tau_\ell(\epsilon_\ell, \epsilon'_{-\ell})$, $\forall \epsilon_{-\ell}, \epsilon'_{-\ell}$ (i.e., τ_ℓ is only a function of ϵ_ℓ) an equilibrium exists if the cost functions are continuous.*

Proof. See Appendix A.4. □

Although the existence of an equilibrium is easier to guarantee in the case of Proposition 1, proving uniqueness is still non-trivial. Proving the uniqueness of

[1] players (a common term within the game-theory literature) denote agents or data owners in this thesis



(a) (b) (c)

Figure 4.2: Equilibrium analysis and game convergence in multiple players game with Leakage Compensation tradeoff function (a)Payoff function plot with unique Equilibrium point; (b)Privacy budget Convergence of multiple users with different initial values; (c)Privacy budget Convergence of multiple users with different initial values and different data size.

the equilibrium generally requires the cost functions to be continuous and strictly convex [47]. For the case discussed in Proposition 1, any equilibrium is also a dominant strategy, i.e., each player implements its optimal action irrespective of the actions of the others. This makes the equilibrium robust to collusion.

Noting that, in a liberal society, the data owners are free to not cooperate if there is no hope for receiving compensation in return for their efforts, there exist compensation policies for which all the data owners cease to communicate and the privacy-aware ML algorithms may not be implemented. To avoid such trivial compensation policies, we define the set of individually rational policies.

Definition 3 (Individually Rational)**.** *The compensation policy $(L_\ell)_{\ell\in\mathcal{N}}$ is individually rational if $L_\ell(\epsilon_\ell,\epsilon_{-\ell}) \geq U_\ell(\epsilon_\ell)$, for all $\ell\in\mathcal{N}$ and $(\epsilon_\ell)_{\ell\in\mathcal{N}} \in \Upsilon((L_\ell)_{\ell\in\mathcal{N}})$.*

In the next two sections, discrete queries examples of security game with compensation function are implemented in semi-hostile models of multiple data owners.

4.3 Leakage-Compensation Game

For a database $\mathscr{X}$ with u individuals and the $v \in Val$ possible values for one sensitive attribute. The leakage between $\mathscr{X}$ and $\mathscr{Z}$ by using min-entropy quantification has a tight upper bounds as $I_\infty(\mathscr{X},\mathscr{Z}) \leq ulog2\frac{v\cdot e^\epsilon}{v-1+e^\epsilon}$ derived in 2.4.1.1.

We consider the worst case which is the exact upper bound of the mutual information as the leakage and calculate the increase trends by doing differential equation $\frac{dL}{d\epsilon} = \frac{u}{In2}\cdot\frac{v-1}{(v-1+e^\epsilon)}$ it can be found that $\frac{dL}{d\epsilon} \geq 0$, when $\epsilon > 0$. From the above equation, the leakage increases with the increase of ϵ. Thus, the more accuracy of the reported answer, the more leakage the system would be.

The payoff function is that, the more the leakage the database gives, the more payoff it will gets. The cost is intuitively the privacy budget it offers. The weighting factor wf is carefully selected to rescale the payoff and cost into a range $[0,1]$, then the privacy budget from other database drives itself to contribute more accuracy. The compensation function τ_ℓ is as:

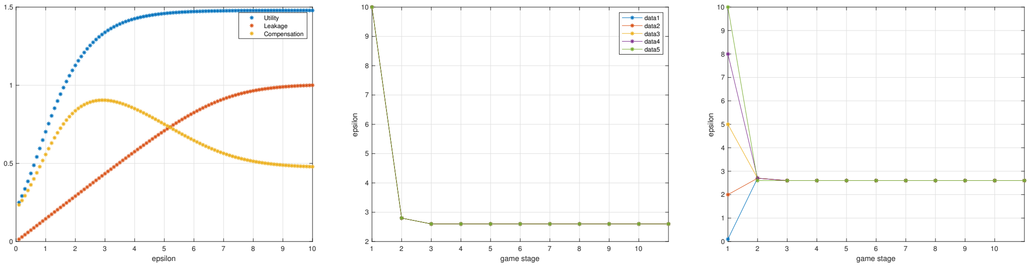
$$\tau_\ell = wf(\epsilon_\ell,\epsilon_{-\ell})\cdot u\cdot\log 2\frac{v\cdot e^\epsilon}{v-1+e^\epsilon} - e^\epsilon \tag{4.1}$$

where wf is weighting factor, $wf = \frac{\sum_{-\ell}\epsilon_\ell n^c_\ell}{\sum\epsilon_\ell n^c_\ell}\cdot max(ulog2\frac{v\cdot e^\epsilon}{v-1+e^\epsilon})/max(e^\epsilon)$. The compensation function (4.1) is a concave continuous function, which implies the Equilibrium in Definition 2 is unique.

4.4 Utility-Compensation Game

In a randomisation mechanism, the real answer $y \in \mathscr{Y}$ is mapped into a reported answer $z \in \mathscr{Z}$ according to the given probability distribution $p_{\mathscr{Z}|\mathscr{Y}}$. Sometimes, the user doesn't take z as the guess for the real answer. Bayesian post-processing is used to maximise the probability of a right guess. Thus, for a reported answer z, a remapping function $\rho(z) : \mathscr{Z} \rightarrow \mathscr{Y}$ gives a guess of $y' \in \mathscr{Y}$. For each pair (y, y') there is an associated value regarded as gain function $g(y, y')$ represents the utility.

The distance d between two elements $y, y' \in \mathscr{Y}$ has a maximum distance of n. The upper bound in such conditions is as $U(\mathscr{Y},\mathscr{Z}) \leq \frac{(e^\epsilon)^n(1-e^\epsilon)}{(e^\epsilon)^n(1-e^\epsilon)+c(1-(e^\epsilon)^n)}$. The



(a) (b) (c)

Figure 4.3: Equilibrium analysis and game convergence in multiple players game with Utility-Leakage tradeoff function (a)Utility function plot with unique Equilibrium point; (b)Privacy budget Convergence of multiple users; (c)Privacy budget Convergence of multiple users with different initial values.

bound provided by the above theorem is strict in the sense that for every ϵ and $\mathscr{Y}$ there exist an adjacency relation $\sim$ for which we can construct a randomization mechanism H that provides ϵ-differential privacy and whose utility achieves the bound of Theorem 3. This randomization mechanism is therefore optimal, in the sense that it provides the maximum possible utility for the given ϵ. Intuitively, the condition on $\sim$ is that $|Borderd(y)|$must be exactly c or 0 for every $d > 0$. Intuitively, by picking the worst case with largest utility value for the upper bound, the payoff function τ could be as following:

$$\tau_\ell = wf(\epsilon_\ell, \epsilon_{-\ell}) \cdot \frac{(e^\epsilon)^n(1-e^\epsilon)}{(e^\epsilon)^n(1-e^\epsilon)+c(1-(e^\epsilon)^n)} - e^\epsilon, \tag{4.2}$$

where wf is weight factor $wf = \frac{\sum_{-\ell}\epsilon_{-\ell}n_{-\ell}^c}{\sum \epsilon_\ell n_\ell^c} \cdot \frac{max(\frac{(e^\epsilon)^n(1-e^\epsilon)}{(e^\epsilon)^n(1-e^\epsilon)+c(1-(e^\epsilon)^n)})}{max(e^\epsilon)}$.

The simulation results are shown in Figure 4.2 and Figure 4.3. The Equilibrium point as defined in Definition 2 is unique. With different initial value of strategy ϵ, players with the same size of dataset converge at the end of the game. If the size for datasets are different, the convergence value for each player is accordingly differs to balance the payoff value at each other's best response, which guarantee in the case of Proposition 1.

Chapter 5

Conclusions and Future Research

We considered privacy-aware optimization-based ML on distributed private datasets. We assumed that the data owners provide DP responses to gradient queries. The theoretical analysis of the proposed DP gradient descent algorithms provided a way for predicting the quality of ML models based on the privacy budgets and the size of the datasets. We proved that the difference between the training model with and without considering privacy constrains of the data owners is bounded by $(\sum_{\ell\in\mathcal{N}} n_\ell)^{-2}\sum_{\ell\in\mathcal{N}}\epsilon_\ell^{-2}$ in our proposed algorithms under smoothness and strong-convexity assumptions for the fitness cost. The empirical results with real-world financial datasets split between multiple institutes/banks while using regression and support vector machine models demonstrated that the relative fitness in fact follows ϵ_i^{-2} and n_i^{-2} for the proposed algorithm. This shows the tightness of the upper bounds on the difference between the trained ML models with and without privacy constraints from the theoretical analysis, which can be utilized for quantification of the privacy-utility trade-off in privacy-preserving ML. These results can be used or extended in multiple directions for future research:

- We can extend the framework to multiple learners aiming to train separate privacy-aware ML models with similar structures based on their own datasets and DP responses from other learners and private data owners. This is closer in nature to the distributed or federated ML framework over an arbitrary connected communication network. Note that, in this thesis, the communication structure among the learner and the data owners is over a star graph with the learner at the center.

- The results of this thesis can be used to understand the behaviour of data owners and learners in a data market for ML training. The utility-privacy trade-off in this thesis, in terms of the quality of the trained ML models, can be used in conjunction with the cost of sharing private data of costumers with the learner (in terms of loss of reputation, legal costs, implementation of privacy-preserving mechanisms, and communication infrastructure) to setup a game-theoretic framework for modeling interactions across a data market. The learner can compensate the data owners for access to their private data, by essentially paying them for choosing larger privacy budgets. After negotiations between the data owners and the learners for setting the privacy budgets, the algorithm of this thesis can be used to the n train ML models, while knowing in advance the expected quality of the trained model.

- Synchronous updates of the algorithm is indeed a bottleneck of the proposed algorithm. Future work can focus on extending the results of this paper to asynchronous gradient updates where, at each iteration, only a subset of the data owners update the ML model. To be able to ensure the convergence of the asynchronous algorithm, we need to ensure that all the data owners update the model as frequently as required.

- Another direction for future research is to extend the framework of this paper to adversarial learning scenarios that can admit more general adversaries (than the case of curious-but-honest adversaries in this paper).

APPENDIX

A.1 Proof of Theorem 1

First, note that $\|\overline{\mathfrak{Q}}_\ell(\mathscr{D}_\ell; k) - \overline{\mathfrak{Q}}_\ell(\mathscr{D}'_\ell; k))\|_1 = (1/n_\ell)\| \sum_{\{x,y\}\in\mathscr{D}_\ell} \xi_{\bar{g}_2^{x,y}}(\theta[k]) - \sum_{\{x,y\}\in\mathscr{D}'_\ell} \xi_{\bar{g}_2^{x,y}}(\theta[k])\|_1 = (1/n_\ell)\|\xi_{\bar{g}_2^{x,y}}(\theta[k])|_{\{x,y\}\in\mathscr{D}_\ell\subseteq\mathscr{D}'_\ell} - \xi_{\bar{g}_2^{x,y}}(\theta[k])|_{\{x,y\}\in\mathscr{D}'_\ell\subseteq\mathscr{D}_\ell}\|_1$. This implies that $\|\overline{\mathfrak{Q}}_\ell(\mathscr{D}_\ell; k) - \overline{\mathfrak{Q}}_\ell(\mathscr{D}'_\ell; k))\|_1 \leq (2/n_\ell)\max_{\{x,y\}\in\mathscr{D}'_\ell\subseteq\mathscr{D}_\ell\cup\mathscr{D}_\ell\subseteq\mathscr{D}'_\ell} \|\xi_{\bar{g}_2^{x,y}}(\theta[k])\|_1 \leq 2\Xi/n_\ell$. The rest follows from $p((\overline{\mathfrak{Q}}_\ell(\mathscr{D}_\ell; k))_{k=1}^T)/p((\overline{\mathfrak{Q}_\ell}(\mathscr{D}'_\ell; k))_{k=1}^T) = \prod_{k=1}^T \exp(\|\overline{\mathfrak{Q}}_\ell(\mathscr{D}'_\ell; k)\|_1/b - \|\overline{\mathfrak{Q}}_\ell(\mathscr{D}_\ell; k)\|_1/b) \leq \prod_{k=1}^T \exp(2\Xi/bn_\ell) = \exp(2\Xi T/bn_\ell)$, where, by some abuse of notation, $p(\cdot)$ denotes the probability density of the variable in its argument.

A.2 Proof of Theorem 2

First, note that

$$
\begin{aligned}
\mathbb{E}\{\|w[k]\|_2^2\} =& \mathbb{E}\left\{\left\|\left(\frac{1}{\sum_{\ell\in\mathcal{N}} n_j}\right)\sum_{j\in\mathcal{N}} n_\ell w_\ell[k]\right\|_2^2\right\} \\
=& \left(\frac{1}{\sum_{\ell\in\mathcal{N}} n_j}\right)^2 \sum_{\ell\in\mathcal{N}} n_\ell^2 \mathbb{E}\{\|w_\ell[k]\|_2^2\} \\
=& \left(\frac{1}{\sum_{\ell\in\mathcal{N}} n_j}\right)^2 \sum_{\ell\in\mathcal{N}} \frac{8\Xi^2 T^2}{\epsilon_\ell^2} \\
=& \frac{8\Xi^2 T^2}{n^2} \sum_{\ell\in\mathcal{N}} \frac{1}{\epsilon_\ell^2}.
\end{aligned}
$$

Because ∇f is λ-Lipschitz, $f(z_1) \leq f(z_2) + \nabla f(z_2)^\top (z_1 - z_2) + 0.5\lambda \|z_2 - z_1\|_2^2$ for all z_1, z_2 [44] and therefore

$$\begin{aligned}
\mathbb{E}\{f(\theta[k+1])\} \leq & \mathbb{E}\{f(\theta[k])\} \\
& + \mathbb{E}\{\nabla f(\theta[k])^\top (\theta[k+1] - \theta[k])\} \\
& + \frac{\lambda}{2} \mathbb{E}\{\|\theta[k+1] - \theta[k]\|_2^2\} \\
\leq & \mathbb{E}\{f(\theta[k])\} \\
& + \rho_k \left(\frac{\lambda \rho_k}{2} - 1 \right) \mathbb{E}\{\|\nabla f(\theta[k])\|_2^2\} \\
& + \rho_k^2 \frac{8\Xi^2 T^2}{n^2} \sum_{\ell \in \mathcal{N}} \frac{1}{\epsilon_\ell^2}.
\end{aligned}$$

For all $\rho_k \leq 1/\lambda$, we have

$$\begin{aligned}
\mathbb{E}\{f(\theta[k+1])\} \leq & \mathbb{E}\{f(\theta[k])\} - \frac{\rho_k}{2} \mathbb{E}\{\|\nabla f(\theta[k])\|_2^2\} \\
& + \rho_k^2 \frac{8\Xi^2 T^2}{n^2} \sum_{\ell \in \mathcal{N}} \frac{1}{\epsilon_\ell^2}.
\end{aligned} \tag{A.1}$$

For $\varepsilon > 0$, we may define

$$k_0 := \inf_k \left\{ k \,\middle|\, \mathbb{E}\{\|\nabla f(\theta[k])\|_2^2\} \leq \frac{16\Xi^2 T^2 \rho_k}{n^2} \sum_{\ell \in \mathcal{N}} \frac{1}{\epsilon_\ell^2} + \varepsilon \right\}.$$

If T is large enough, we can easily show that there exists $k_0 < \infty$. This can be proved by contrapositive. Assume that this not the case. Therefore,

$$\begin{aligned}
\lim_{k \to 0} \mathbb{E}\{f(\theta[k])\} = & \mathbb{E}\{f(\theta[1])\} \\
& + \sum_{t=2}^{k} (\mathbb{E}\{f(\theta[t])\} - \mathbb{E}\{f(\theta[t-1])\}) \\
\leq & \mathbb{E}\{f(\theta[1])\} - \sum_{t=2}^{k} \varepsilon \rho_k \\
= & -\infty.
\end{aligned}$$

This is however not possible. Since f is L-strongly convex, Polyak-Lojasiewicz inequality [44] implies that

$$\begin{aligned}\mathbb{E}\{f(\theta[k_0])\} - f(\theta^*) \leq & \frac{1}{2L}\mathbb{E}\{\|\nabla f(\theta[k])\|_2^2\} \\ \leq & \frac{8\Xi^2 T^2 \rho_k}{Ln^2} \sum_{\ell\in\mathcal{N}} \frac{1}{\epsilon_\ell^2} + \frac{\varepsilon}{2L}.\end{aligned}$$

Now, because $k_0 \leq T$, we get

$$\begin{aligned}\min_{1\leq k\leq T} \mathbb{E}\{f(\theta[k])\} - f(\theta^*) \leq & \mathbb{E}\{f(\theta[k_0])\} - f(\theta^*) \\ \leq & \frac{8\Xi^2 T^2 \rho_k}{Ln^2} \sum_{\ell\in\mathcal{N}} \frac{1}{\epsilon_\ell^2} + \frac{\varepsilon}{2L}.\end{aligned}$$

Again, because f is L-strongly convex, we can see that

$$\begin{aligned}f(\theta^*) \leq & f(t\theta + (1-t)\theta^*) \\ & \leq tf(\theta) + (1-t)f(\theta^*) - \frac{L}{2}t(t-1)\left\|\theta - \theta^*\right\|_2^2,\end{aligned}$$

for all $t \in (0,1)$. Setting $t = 1/2$ results in

$$\left\|\theta - \theta^*\right\|_2^2 \leq 4(f(\theta) - f(\theta^*))/L. \tag{A.2}$$

Hence,

$$\begin{aligned}\min_{1\leq k\leq T} \|\theta[k] - \theta^*\|_2^2 \leq & \frac{4}{L}\left(\min_{1\leq k\leq T} \mathbb{E}\{f(\theta[k])\} - f(\theta^*)\right) \\ \leq & \frac{32\Xi^2 T^2 \rho_k}{L^2 n^2} \sum_{\ell\in\mathcal{N}} \frac{1}{\epsilon_\ell^2} + \frac{\varepsilon}{8L^2}.\end{aligned}$$

This concludes the proof.

A.3 Proof of Theorem 3

Under all these assumptions, the inequality in (3.9) follows from the result of [50] using the optimal selection of c in [24]. The only difference with the proofs in [50] is to appreciate that

$$\zeta_k - \zeta_{k-1} \leq \frac{2}{\sqrt{T}T(T+1)},$$

where

$$\zeta_k := \frac{1/\sqrt{T}+1}{1/\sqrt{T}+k} \prod_{m=k+1}^{T} \frac{m-1}{1/\sqrt{T}+m}.$$

The inequality follows from that

$$\begin{aligned}
\zeta_k - \zeta_{k-1} &= \frac{(1/\sqrt{T})(1/\sqrt{T}+1)}{(k-1+1/\sqrt{T})(k+1/\sqrt{T})} \\
&\quad \times \prod_{m=k+1}^{T} \frac{m-1}{1/\sqrt{T}+m} \\
&= \frac{(1/\sqrt{T})(1/\sqrt{T}+1)}{(k-1+1/\sqrt{T})(k+1/\sqrt{T})} \\
&\quad \times \frac{\prod_{m=k+1}^{T}(m-1)}{\prod_{m=k+1}^{T}(1/\sqrt{T}+m)} \\
&= (1/\sqrt{T})(1/\sqrt{T}+1) \frac{\prod_{m=k}^{T-1} m}{\prod_{m=k-1}^{T}(1/\sqrt{T}+m)} \\
&= (1/\sqrt{T})(1/\sqrt{T}+1) \frac{\prod_{m=k}^{T-1} m}{\prod_{m=k}^{T+1}(1/\sqrt{T}+m-1)} \\
&= \frac{(1/\sqrt{T})(1/\sqrt{T}+1)}{T(T+1)} \prod_{m=k}^{T+1} \frac{m}{(1/\sqrt{T}+m-1)} \\
&\leq \frac{2}{\sqrt{T}} \frac{1}{T(T+1)}.
\end{aligned}$$

If f is L-strongly convex, the proof of the inequality in (3.10) follows from (A.2).

A.4 Proof of Proposition 1

The proof is a direct consequence of the extreme value theorem [38, p. 30] and the fact that each player's cost function becomes only the function of its own decision variable.

Bibliography

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[2] Dakshi Agrawal and Charu C Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 247–255. ACM, 2001.

[3] Mário S Alvim, Miguel E Andrés, Konstantinos Chatzikokolakis, Pierpaolo Degano, and Catuscia Palamidessi. Differential privacy: on the trade-off between utility and information leakage. In *International Workshop on Formal Aspects in Security and Trust*, pages 39–54. Springer, 2011.

[4] Mário S Alvim, Miguel E Andrés, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. On the relation between differential privacy and quantitative information flow. In *International Colloquium on Automata, Languages, and Programming*, pages 60–76. Springer, 2011.

[5] Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shiho Moriai, et al. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345, 2018.

[6] Kenneth J. Arrow and Gerard Debreu. Existence of an equilibrium for a competitive economy. *Econometrica*, 22(3):265–290, 1954.

[7] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014.

[8] Colin J Bennett and Charles D Raab. Revisiting the governance of privacy: Contemporary policy instruments in global perspective. *Regulation & Governance*, 2018.

[9] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191. ACM, 2017.

[10] Lawrence Carin, George Cybenko, and Jeff Hughes. Cybersecurity strategies: The queries methodology. *Computer*, 41(8), 2008.

[11] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In *Advances in Neural Information Processing Systems*, pages 289–296, 2009.

[12] Gerard Debreu. A social equilibrium existence theorem. *Proceedings of the National Academy of Sciences*, 38(10):886–893, 1952.

[13] Wenliang Du, Yunghsiang S Han, and Shigang Chen. Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *Proceedings of the 2004 SIAM international conference on data mining*, pages 222–233. SIAM, 2004.

[14] Cynthia Dwork. A firm foundation for private data analysis. *Communications of the ACM*, 54(1):86–95, 2011.

[15] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[16] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[17] Quan Geng and Pramod Viswanath. Optimal noise adding mechanisms for approximate differential privacy. *IEEE Trans. Information Theory*, 62(2):952–969, 2016.

[18] Robert Gibbons. *Game theory for applied economists*. Princeton University Press, 1992.

[19] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning*, pages 201–210, 2016.

[20] Thore Graepel, Kristin Lauter, and Michael Naehrig. ML confidential: Machine learning on encrypted data. In *International Conference on Information Security and Cryptology*, pages 1–21. Springer, 2012.

[21] MT Hale and M Egersted. Differentially private cloud-based multi-agent optimization with constraints. In *Proceedings of the American Control Conference*, pages 1235–1240, 2015.

[22] Samuel N Hamilton, Wendy L Miller, Allen Ott, and O Sami Saydjari. Challenges in applying game theory to the domain of information warfare. In *Information Survivability Workshop (ISW)*. Citeseer, 2002.

[23] Samuel N Hamilton, Wendy L Miller, Allen Ott, and O Sami Saydjari. The role of game theory in information warfare. In *4th Information survivability workshop (ISW-2001/2002)*, 2002.

[24] Shuo Han, Ufuk Topcu, and George J Pappas. Differentially private distributed constrained optimization. *IEEE Transactions on Automatic Control*, 62(1):50–64, 2017.

[25] Zhu Han, Dusit Niyato, Walid Saad, Tamer Başar, and Are Hjørungnes. *Game theory in wireless and communication networks: theory, models, and applications*. Cambridge University Press, 2012.

[26] Justin Hsu, Marco Gaboardi, Andreas Haeberlen, Sanjeev Khanna, Arjun Narayan, Benjamin C Pierce, and Aaron Roth. Differential privacy: An economic method for choosing epsilon. In *Computer Security Foundations Symposium (CSF), 2014 IEEE 27th*, pages 398–410. IEEE, 2014.

[27] Zhenqi Huang, Sayan Mitra, and Nitin Vaidya. Differentially private distributed optimization. In *Proceedings of the 2015 International Conference on Distributed Computing and Networking*, page 4, 2015.

[28] Zonghao Huang, Rui Hu, Yanmin Gong, and Eric Chan-Tin. DP-ADMM: ADMM-based distributed learning with differential privacy. *Preprint:* `arXiv preprint arXiv:1808.10101`, 2018.

[29] Tyler Hunt, Congzheng Song, Reza Shokri, Vitaly Shmatikov, and Emmett Witchel. Chiron: Privacy-preserving machine learning as a service. *arXiv preprint arXiv:1803.05961*, 2018.

[30] Geetha Jagannathan and Rebecca N Wright. Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In *Proceedings of the*

eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pages 593–599. ACM, 2005.

[31] Kousha Kalantari, Lalitha Sankar, and Anand D Sarwate. Robust privacy-utility tradeoffs under differential privacy and hamming distortion. *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*, 13(11), 2018.

[32] Wendy Kan. Lending club loan data: Analyze lending club's issued loans. `https://www.kaggle.com/wendykan/lending-club-loan-data`, Date Accessed: 17 Oct 2018.

[33] Jaewoo Lee and Chris Clifton. How much is enough? choosing ε for differential privacy. In *International Conference on Information Security*, pages 325–340. Springer, 2011.

[34] Ping Li, Jin Li, Zhengan Huang, Tong Li, Chong-Zhi Gao, Siu-Ming Yiu, and Kai Chen. Multi-key privacy-preserving deep learning in cloud computing. *Future Generation Computer Systems*, 74:76–85, 2017.

[35] Xiannuan Liang and Yang Xiao. Game theory for network security. *IEEE Communications Surveys & Tutorials*, 15(1):472–486, 2013.

[36] Yehuda Lindell and Benny Pinkas. Privacy preserving data mining. In Mihir Bellare, editor, *Advances in Cryptology — CRYPTO 2000*, pages 36–54, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[37] Yu Liu, Cristina Comaniciu, and Hong Man. A bayesian game approach for intrusion detection in wireless ad hoc networks. In *Proceeding from the 2006 workshop on Game theory for communications and networks*, page 4. ACM, 2006.

[38] T. W. Ma. *Classical Analysis on Normed Spaces*. World Scientific, Singapore, 1995.

[39] Machine Learning Group–ULB. Credit card fraud detection: Anonymized credit card transactions labeled as fraudulent or genuine. https://www.kaggle.com/mlg-ulb/creditcardfraud/home, Date Accessed: 27 Nov 2018.

[40] Mohammad Hossein Manshaei, Quanyan Zhu, Tansu Alpcan, Tamer Bacşar, and Jean-Pierre Hubaux. Game theory meets network security and privacy. *ACM Computing Surveys (CSUR)*, 45(3):25, 2013.

[41] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

[42] José Moura and David Hutchison. Game theory for multi-access edge computing: Survey, use cases, and future trends. *IEEE Communications Surveys & Tutorials*, 2018.

[43] John Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

[44] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Springer US, 2013.

[45] Erfan Nozari, Pavankumar Tallapragada, and Jorge Cortés. Differentially private distributed convex optimization via functional perturbation. *IEEE Transactions on Control of Network Systems*, 5(1):395–408, 2018.

[46] Maxim Raya, Reza Shokri, and Jean-Pierre Hubaux. On the tradeoff between trust and privacy in wireless ad hoc networks. In *Proceedings of the third ACM conference on Wireless network security*, pages 75–80. ACM, 2010.

[47] J Ben Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica*, 33(3):520–534, 1965.

[48] Sankardas Roy, Charles Ellis, Sajjan Shiva, Dipankar Dasgupta, Vivek Shandilya, and Qishi Wu. A survey of game theory as applied to network security. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1–10. IEEE, 2010.

[49] Anand D Sarwate and Kamalika Chaudhuri. Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data. *IEEE signal processing magazine*, 30(5):86–94, 2013.

[50] Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International Conference on Machine Learning*, pages 71–79, 2013.

[51] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321. ACM, 2015.

[52] Naum Zuselevich Shor. *Minimization methods for non-differentiable functions*, volume 3 of *Springer Series in Computational Mathematics*. Springer, Berlin, Heidelberg, 2012.

[53] Jaideep Vaidya and Chris Clifton. Privacy preserving association rule mining in vertically partitioned data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 639–644. ACM, 2002.

[54] Jaideep Vaidya, Murat Kantarcıoğlu, and Chris Clifton. Privacy-preserving naive bayes classification. *The VLDB Journal*, 17(4):879–898, 2008.

[55] Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional mechanism: regression analysis under differential privacy. *Proceedings of the VLDB Endowment*, 5(11):1364–1375, 2012.

[56] Tao Zhang and Quanyan Zhu. Dynamic differential privacy for ADMM-based distributed classification learning. *IEEE Transactions on Information Forensics and Security*, 12(1):172–187, 2017.

[57] Tianwei Zhang, Zecheng He, and Ruby B. Lee. Privacy-preserving machine learning through data obfuscation. *arXiv preprint arXiv:1807.01860*, 2018.

[58] Zuhe Zhang, Benjamin I P Rubinstein, and Christos Dimitrakakis. On the differential privacy of bayesian inference. In *AAAI Conference on Artificial Intelligence*, pages 2365–2371, 2016.