Optimized Data Sharing with Differential Privacy: A Game-theoretic Approach

Nan Wu, [*,‡], David Smith, [*,§], and Mohamed Ali Kaafar, [*,‡]
[‡]Macquarie University, [*]CSIRO's Data61, [§]Australian National University
nan.wu5@hdr.mq.edu.au, david.smith@data61.csiro.au, dali.kaafar@mq.edu.au

Abstract

We study and optimize the differentially private learning outcomes from data shared amongst multiple separate data-owners according to a classical privacy versus accuracy trade-off using a game-theoretic approach. A dynamic non-cooperative game, with imperfect information, provides this optimal tradeoff, with differentially private models that learn from the data. In this model, we make an optimal choice of privacy budget parameter from the Laplace mechanism, according to pure differential privacy. The data analysis model uses differentially private gradient queries, as privacy aware supervised-machine learning. Then, we use non-cooperative game theory to analyse and optimize the utility-leakage trade-off to minimize learning loss achieving a unique Nash equilibrium (mutual best response). We quantify the quality of the trained model with a novel method to capture the trade-off between privacy and utility (accuracy). Our novel method uses fixed-point theory in gradient descent learning to predict the contraction mapping of the outcomes. We validate the collaborative learning method applied with our non-cooperative game over a partitioned real financial dataset, demonstrating benefits of sharing data for all data-owners, with significant benefits in social welfare from applying our game.

Introduction

For the sake of enhancing efficiency and capacity of the internet of things (IoT), edge computing allows data to be transferred and processed at the edge of the network such as at a cloud aggregator or end devices (Dwork and Pappas 2017; Shi et al. 2016). In such networks, data analysis methods using machine learning (ML) can unlock valuable insights for improving revenue or quality-of-service from, potentially proprietary, private datasets (Hunt et al. 2018; Graepel, Lauter, and Naehrig 2012). The shared information from the data owners in a particular IoT network can then contribute to training ML models.

Due to the nature of learning, having large high-quality datasets improves the quality of trained ML models in terms of the accuracy of predictions on potentially untested data (Dwork, Roth et al. 2014). The subsequent improvements in quality motivate multiple data owners to share and merge their datasets in order to create larger training datasets for federated training (Li et al. 2017; Konečnỳ et al. 2016). For instance, financial institutions 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, this shared information between data owners will inevitably have sensitive data which the owners wish to protect. As such data owners are independent of each other, they will be concerned about their own data safety in a collaborative learning settings.

We consider multiple learners aim to train separate privacy-aware ML models with similar structures based on their own datasets and differentially private (DP) responses from other learners and private data owners as shown in Figure 1. Each data owner trains a separate ML model and sends the differentially private response to other participants for collaborative learning. This is similar to distributed ML on arbitrary connected graphs. This way, we can extend the results to more general communication structures with the learner not necessarily at the center. Note that the latter configuration where the communication structure among the learner and the data owners is set up as a star graph with the learner at the center has been considered in prior research, e.g. (Wu et al. 2020; Farokhi et al. 2020).

In this paper we first use Banach fixed point theory to get a more accurate prediction of the learning loss (Jung 2017). The next challenge is to significantly improve the utility-privacy trade-off in terms of the quality of the trained ML models; so our results of learning outcome prediction can be used in conjunction with the cost of sharing private data of consumers with the learner (in terms of loss of reputation, legal costs, implementation of privacy-preserving mechanisms, and communication infrastructure). To address this, we establish a game-theoretic framework for modelling 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(i.e.,more relaxed privacy). After negotiations between the data owners and the learners for setting privacy budgets, the ML models can be trained and tradeoff between learning loss and privacy level.

Related Work. Optimization of the trade-off between privacy and utility has been discussed well in the literature (Kalantari, Sankar, and Sarwate 2018; Brenner and Nissim 2010; Ghosh, Roughgarden, and Sundararajan 2012; Gupte and Sundararajan 2010). In a previous paper, the problem

of optimizing utility for differential privacy using linear programming has been solved (Bordenabe, Chatzikokolakis, and Palamidessi 2014). In another work (Xu et al. 2015) , the trade-off between data utility and privacy preservation is discussed with respect to how game theory can be used to complete this trade-off. In (Xu et al. 2015) a sequential game model is constructed between data user and data collector followed by backward induction reaching a subgame perfect Nash Equilibrium. In another work, (Xu et al. 2016), the authors focus on the idea of exchanging private information for money or other incentives provided to the data owner by the data collector. And then they discuss how to use game theory to obtain an agreement between the parties involved in this trade.

Distributed/Collaborative Privacy-Preserving Machine Learning (ML) has been investigated in (Shokri and Shmatikov 2015; Huang et al. 2018; Zhang, He, and Lee 2018; Zhang and Zhu 2017; Wu et al. 2020). Stochastic gradient descent is utilized in distributed ML models with additive Gaussian/Laplace noise to ensure differential privacy. By appropriately selecting step size in the stochastic gradient descent, the quality of the trained ML model based on the privacy budget can be forecast according to (Wu et al. 2020). In a work by (Jung 2017), authors study the basic gradient descent iterations in ML models from the contraction view of a specific operator with a differentiable objective function. They show how gradient descent can be accelerated in ML models, preserving fixed-points with faster convergence, by the contraction mapping theorem.

Contributions

In this paper, we evaluate the collaborative learning model with DP from a fixed-point view of linear regression contraction problem. This way, we make a precise prediction of the learning parameter in ML model. We distributedly optimize the trade-off of utility and privacy in collaborative learning using a non-cooperative game, with imperfect information, between multiple data owners. More specifically, this paper makes the following contributions:

- We build a non-cooperative game model for these learners to optimally trade-off accuracy and privacy, according to privacy budget and gain with minimised learning loss.
- We demonstrate a unique Nash equilibrium for this game, providing a mutual best response in terms of the differentially private shared data and its learning loss. Moreover, this unique Nash equilibrium is demonstrated to be maintained with imperfect information with data owners simultaneously sharing, and learning from, each others' data.
- We use a Banach fixed-point of view on gradients responses to modify the learning algorithm and to evaluate learning iteration speed.
- Our numerical tests built on real financial datasets, where each learner is training for an expectation of annual loan rate with a list of users' data including sensitive information, demonstrate the significant benefits of learning using our game with differentially-private collaborative machine learning.

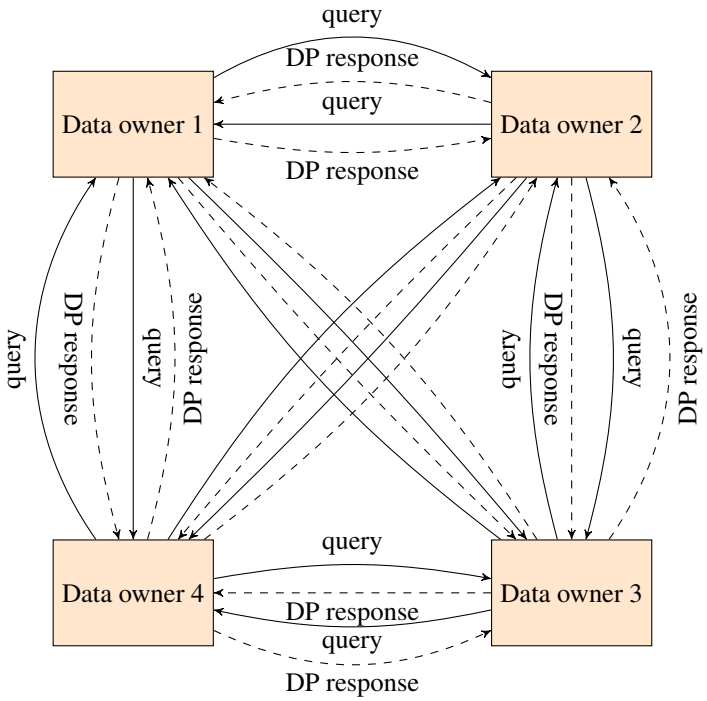

Figure 1: The communication structure between the distributed data owners (learners) for submitting queries and providing differentially-private (DP) responses to each other. The dashed lines are DP responses and solid lines are queries.

System Model

We assume that there are several data owners training their own machine learning parameters with both their own dataset and other private datasets. They send queries to each other and respond with a randomised answer as in Figure 1.

Thus, the system setup incorporates multiple data owners sharing data by sending a query and answering with good, but not precise accuracy. For brevity, a data owner that trains their model according to others responses is denoted as an analyst, whilst when a data owner replies to a query, by adding noise to the query, we denote that owner as an agent. Importantly, a data owner trains its own ML model simultaneously in our federated learning model.

In this work, we assume that an agent i perturb query's exact answer by adding Laplace noise with variance σ_i^2: $z_i = y_i + n_i$, where n_i is a zero-mean random variable with respect to Differential Privacy. The agent sends to the analyst both the perturbed query answer and the the variance that provides the accuracy.

We extend the existing framework to multiple learners aiming to train their own ML models according to each others' separate privacy aware sub-gradient responses.

Definitions

A group of $N \in \mathbb{N}$ private agents or data owners $\mathcal{N} := \{1, \ldots, N\}$ are connected to each other and train their own ML model over an undirected communication graph as in Figure 1.

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.

In linear regression for training home loan datasets, each data owner i computes its private datasets $\mathbb{X}_i$, and use a randomised privacy algorithm to perturb the answer then replies to the query sender. Let $(D_i, i \in \mathcal{N})$ denote the aggregate of all the training datasets. 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 gender, age, salary, and employment status). Categorical attributes, such as gender, can always be translated into numerical ones according to a rule as inputs, and historically approved interest rates per annum by the bank are outputs.

The participants in a collaborative machine learning model are interested in finding the relationship between explanatory variables and real valued outcomes by using a ML model $\mathfrak{M} : \mathbb{R}^{\|\mathbb{X}\|} \to \mathbb{R}^{\|\mathbb{Y}\|}$, where $\mathbb{X}$ are inputs and $\mathbb{Y}$ are outputs come from the available training datasets $\mathcal{D}_i, \forall i \in \mathcal{N}$.

The training model is to solve the optimization problem of minimising the learning loss in

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

where $g_1(\theta)$ is a regularizing term, $n := \sum_{\ell \in \mathcal{N}} n_\ell$, and $\Theta := \{\theta \in \mathbb{R}^{p_\theta} \mid \|\theta\|_\infty \leq \theta_{\max}\}$. $g_2(\mathfrak{M}(x;\theta), y)$ is the loss function which measures the distance between the ML outcome $\mathfrak{M}(x;\theta)$ and the real output y. In a linear regression training model, this problem can become to optimize with $g_1(\theta) = 0$ and $g_2(\mathfrak{M}(x;\theta), y) = \|\mathfrak{M}(x;\theta) - y\|_2^2$ which is a Mean Squared Error(MSE) function in equation (1).

Also, in a collaborative training model with a similar structure to (Wu et al. 2020), the data owners not only train with their local dataset but also with other learning agents' datasets. So the data owners send queries to each other. $\mathcal{Q}$ denotes the output space of the query. In this paper, the query used is the sub-gradient of the loss function $g_2(\cdot)$.

Because we use a MSE function for g_2 loss function, it is a convex function of θ. In the absence of privacy concerns, every data owner i trains its ML model with both the dataset from itself and the original private dataset from neighbouring data owners with unfettered access. The ML model for data owner i can be trained by following the projected sub-gradient descent iteration :

$$\theta_i[k+1] = \Pi_{\Theta_i}[\theta_i[k] - \rho_k \xi_{g_2}^{x,y}(\theta_i[k])], \tag{2}$$

where $\rho_k > 0$ is the step-size at iteration k, $\xi_{g_2}^{x,y}(\theta_i[k]) = \partial_\theta g_2(\mathfrak{M}(x;\theta), y)$ is a sub-gradient of the loss function $g_2(\cdot)$ with respect to the variable θ at $\theta[k]$ (Shor 2012), and $\Pi_\Theta[\cdot]$ denotes a projection operator into the set Θ defined as $\Pi_\Theta[a] := \arg\min_{b \in \Theta} \|a - b\|_2$ to prevent overfitting. For continuously differentiable functions, the gradient is the only sub-gradient.

The update law for learning iterations can be rewritten as

$$\begin{aligned} \theta[k+1] &= \Pi_\Theta \left[\theta[k] - \frac{\rho_k}{n} \sum_{\ell \in \mathcal{N}_j} \sum_{\{x,y\} \in \mathcal{D}_\ell} \xi_{g_2^{x,y}}(\theta[k]) \right], \\ &= \Pi_\Theta \left[\theta[k] - \frac{\rho_k}{n} \sum_{\ell \in \mathcal{N}_j \setminus \{j\}} n_\ell \mathfrak{Q}_\ell(\mathcal{D}_\ell; k) \right], \end{aligned} \tag{3}$$

where $\xi_{g_2^{x,y}}$ is a sub-gradient of $g_2^{x,y}$, and $\mathfrak{Q}_\ell(\mathcal{D}_\ell; k)$ is a query that can be submitted by the learning agent to data owner $\ell \in \mathcal{N}$ in order to provide the aggregate sub-gradient:

$$\mathfrak{Q}_\ell(\mathcal{D}_\ell; k) = \frac{1}{n_\ell} \sum_{\{x,y\} \in \mathcal{D}_\ell} \xi_{g_2^{x,y}}(\theta[k]). \tag{4}$$

Responding to the query $\mathfrak{Q}_\ell(\mathcal{D}_\ell; k)$ clearly intrudes on the privacy of the individuals in dataset $\mathcal{D}_\ell$. Therefore, data owner ℓ only responds in a differentially-private manner by reporting the noisy aggregate:

$$\overline{\mathfrak{Q}}_\ell(\mathcal{D}_\ell; k) = \mathfrak{Q}_\ell(\mathcal{D}_\ell; k) + w_\ell[k], \tag{5}$$

where $w_\ell[k]$ is an additive noise to establish differential privacy with privacy budget ϵ_ℓ over the total iterations number T.

The response of data owner is ϵ-differentially private over the learning iterations T.

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

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

where $\mathcal{Y}$ is any Borel-measurable subset of $\mathcal{Q}^T$ (Wu et al. 2020).

Banach fixed point contraction

Now we are going to interpret gradient methods as fixed-point iterations to analyze convergence properties and contraction rates.

In this work, we assume collaborative learning with multiple data owners who are training their own machine learning models respectively. If there is no concern for privacy, the trained dataset is the same for each data owner and is the aggregate of all data $(D_j, j \in \mathcal{N})$. However in our case we assume that data owners do not wish to share all dataset details, rather each learner i trains their ML model with their own dataset D_i and the sub-gradient response from other data owners. When there is no differentially-private noise added to the query responses, we use $D_{all} = (D_j, j \in \mathcal{N})$ to denote the aggregate of the training datasets in the collaborative ML model.

The object of the interest is to minimize the loss function $g_2(\cdot)$ in equation (1). In linear regression, we wish to predict the output Y by a linear combination of the features:

$$Y' = \theta X'.$$

The loss function is to minimize the following Mean Square Error $f(\theta)$:

$$f(\theta) = \|\theta X' - Y'\|_2^2$$

Our linear regression model in equation (3) can be turned into be the following iteration problem:

$$\theta_i[k+1] = \theta_i[k] - \frac{\rho_k}{n_\ell}\Bigg[\frac{2}{n_i}(\theta_i[k]X_i' - Y_i')X_i + \sum_{j\in\mathcal{N}\setminus\{i\}}(\frac{2}{n_j}(\theta_j[k]X_j' - Y_j')X_j + w(j;k;\epsilon_j))\Bigg], \tag{6}$$

Then we define this iteration problem for linear regression model as a contraction problem from a fixed point of view on gradient methods.

Definition 2. *The operation of the fixed point iterations problem is* $\mathcal{T}^\rho : \mathbb{R}^n : \mathbb{R} : \theta \to \mathcal{T}(\theta)$

$$\mathcal{T}(\theta_i[k+1]) = \theta_i[k] - \frac{\rho_k}{n_\ell}\left[\frac{2}{n_i}(\theta_i[k]X_i' - Y_i')X_i + Q_i[k]\right] \tag{7}$$

where $Q_i[k]$ *is the received query responses from all other data owners at iteration* k*:*

$$Q_i[k] = \sum_{j\in\mathcal{N}\setminus\{i\}}(\frac{2}{n_j}(\theta_j[k]X_j' - Y_j')X_j + w(j;k;\epsilon_j)) \tag{8}$$

For brevity, we extract the first part in equation (8) together with the sub-gradient of data owner i to be the sub-gradient of the aggregation of all datasets $(D_i, i \in \mathcal{N})$. Then $D_\ell = (D_i, i \in \mathcal{N})$, $X_\ell = (X_i, i \in \mathcal{N})$,$Y_\ell = (Y_i, i \in \mathcal{N})$.

Lemma 1. *We have* $\nabla f(\theta) = 0$ *if and only if the vector* $\theta \in \mathbb{R}^n$ *is a fixed point of the operator* $\mathcal{T}^\rho$*. Thus,* $\nabla f(\theta) = 0$ *if and only if* $\mathcal{T}^\rho(\theta) = \theta$*,* $\theta_\ell = (Y_\ell' - \frac{n_\ell - 1}{2}\mathbb{E}(\epsilon_\ell)X_\ell^{-1})X_\ell'^{-1}$*. See proof in Appendix.*

$$\frac{\partial\mathcal{T}(\theta)}{\partial\theta} = \mathbb{I}\otimes\mathbb{I} - \frac{2\rho_k}{n_\ell^2}\cdot(X_\ell^\top\cdot X_\ell)^\top\otimes\mathbb{I} \tag{9}$$

By properly select the value for step size ρ_k, $\frac{\partial\mathcal{T}(\theta)}{\partial\theta} = 0$ can be reached.

A straightforward approach to finding the fixed-point of an operator $\mathcal{T}^\rho$ is by the fixed-point iteration

$$\theta^{(k+1)} = \mathcal{T}^\rho\theta^{(k)}.$$

However, because all data owners are training their own learning model and update their learning parameter θ_j after each iteration, Q_i changes in each iteration. Also noise power $w(j;k;\epsilon_j)$ from each data owner j is changing in each DP query response.

Then, we look into the contraction rate for this fixed-point iteration problem.

Lemma 2. *Assume that for some* $q \in [0,1)$*, we have*

$$\|\mathcal{T}^\rho\mathbf{a} - \mathcal{T}^\rho\mathbf{b}\| \le q\|\mathbf{a} - \mathbf{b}\|, \tag{10}$$

for any $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$*. Then, the operator* $\mathcal{T}^\rho$ *has a unique fixed point* θ_0 *and the iterates* $\theta^{(k)}$ *satisfy*

$$\|\theta^{(k)} - \theta_0\| \le q^k\|\theta^{(0)} - \theta_0\|. \tag{11}$$

The contraction rate q is as following:

$$q \ge \|\mathbb{I} - \frac{\rho n_i}{n_\ell}\nabla^2 f(\theta)\| \tag{12}$$

See proof in Appendix.

Following equation (11) and (12), we can find the smallest q and then get a minimum value for total ML iteration time T.

The learning loss parameter can be predicted by using Lemma 2, with a knowledge of previous iteration $k-1$ contracting parameter $\mathcal{T}^\rho(\theta_{k-1})$ and θ_{k-1}. We thus note, that then, when the game is played between data owners, this implies imperfect information because each owner is updating there chosen epsilon according to the last known parameter $k-1$ as opposed to the kth parameter.:

$$\begin{aligned}\mathcal{T}^\rho(\theta_k) &= \mathcal{T}^\rho(\theta_{k-1}) + (\theta_k - \theta_{k-1})(\mathbb{I} - \rho\nabla^2 f(\theta) \\ &\quad - \frac{2\rho}{n_\ell}\sum_{j\in\mathbb{N}\setminus\{i\}} n_j\mathbb{E}(w(j;k;\epsilon_j))) \\ &= \mathcal{T}^\rho(\theta_{k-1}) + (\theta_k - \theta_{k-1})(\mathbb{I} - \frac{2\rho}{n_\ell}X_\ell'X_\ell \\ &\quad - \frac{2\rho}{n_\ell}\sum_{j\in\mathcal{N}\setminus\{i\}} n_j\mathbb{E}(w(j;k;\epsilon_j)))\end{aligned} \tag{13}$$

Following equation (13), the learning parameter θ at next iteration $k+1$ can be predicted in iteration k, and thus the loss function. This is related to the total number of data size n_ℓ and the privacy budget value ϵ_j from all other data owners $j \in \mathcal{N}\setminus\{i\}$.

Theorem 1. *The learning parameter* θ_{k+1} *for next iteration can be predicted by current and previous learning parameter* θ_{k-1} *and* θ_k*:*

$$\begin{aligned}\theta_{k+1} &= \theta_k + (\theta_k - \theta_{k-1})(\mathbb{I} - \frac{2\rho}{n_\ell}X_\ell'X_\ell \\ &\quad - \frac{2\rho}{n_\ell}\sum_{j\in\mathcal{N}\setminus\{i\}} n_j\mathbb{E}(w(j;k;\epsilon_j)))\end{aligned} \tag{14}$$

Corollary 1. *When k is large enough and the ML models for all players have reach a dynamic fixed point, the learning parameter* θ_k *can be predicted as:*

$$\theta_{k+1} = \theta_k - (\theta_k - \theta_{k-1})\frac{2\rho}{n_\ell}\sum_{j\in\mathcal{N}\setminus\{i\}} n_j\mathbb{E}(w(j;k;\epsilon_j)) \tag{15}$$

Algorithm 1 Non-cooperative game implementation

Require: T
Ensure: $(\theta[k])_{k=1}^{T}$
1: Initialize $\theta[1]$
2: **for** $k = 1, \ldots, T-1$ **do**
3: Learner submits query $\mathfrak{Q}_\ell(\mathcal{D}_\ell; k)$ to data owners in $\mathcal{N}$
4: Data owners return DP responses $\overline{\mathfrak{Q}}_\ell(\mathcal{D}_\ell; k)$
5: Learner follows the update rule

$$\theta[k+1] = \theta[k] - \frac{\rho}{T^2 k}\left(\xi_{g_1}(\theta[k]) + \sum_{\ell \in \mathcal{N}} \frac{n_\ell}{n} \overline{\mathfrak{Q}}_\ell(\mathcal{D}_\ell; k)\right),$$

6: **for** $i = 1, \ldots, n_{players}$ **do**
7: Learner i compute $\epsilon_{new,i}$ by

$$\epsilon_{new,i} = \arg\min_{\epsilon} J_i(\epsilon_1, \epsilon_2, ...\epsilon_{n_{players}})$$

8: **end for**
9: **end for**

Game Model

We adopt a dynamic noncooperative repeated game $G = [\mathcal{N}, \epsilon_{\mathbf{i}}, J_{i,\tau}(\cdot)]$ at each time stage τ of gameplay, where $n = 1, ..., N$ are the player/data-owners in $\mathcal{N}$; $\boldsymbol{\epsilon_i} = [0.1, 10]$ is the (pure) strategy set for the ith data owner; and $J_{i,\tau}(\cdot)$ is the cost function for data owner i . The space of action profiles for the N players in each stage is $\boldsymbol{\epsilon} = \epsilon_1 \times \epsilon_2 \times, \ldots \epsilon_N$.

The game G is finitely repeated $T < \infty$ times, with imperfect information. With respect to imperfect information, all data owners send data to each other, and decisions are made concurrently by data owner i, without the knowledge of others' $-\mathbf{i}$ decisions; this process is then repeated T times.

A data owner may be reluctant to give a less noisy response to the analyst. However, this will in return incurs a cost for its own training model. We model these considerations into cost functions. The higher the variance of the perturbation noise is, the lower the cost for privacy violation is. On the other hand, high noise variance reduce the accuracy of the learning model and hence incurs a higher training loss.

Each data owner $i \in \mathcal{N}$ chooses an action $\epsilon_i \in [0.1, 10]$ to minimize cost

$$J_i(\epsilon_i, \epsilon_{-i}) = c_i(\epsilon_i) + f(\epsilon_i) \tag{16}$$

Assumption 1. *For all participants i, cost function $c_i(\epsilon_i)$ is assumed to be non-negative continuous non decreasing function regarding to privacy budget ϵ_i.*

The first component $c_i : \mathbb{R} \to \mathbb{R}$ in the cost function is referred as disclosure cost , which is thus non-negative continuous decreasing (Zhan et al. 2020; Taghizadeh, Kebriaei, and Niyato 2020). We use $c_i = a * \epsilon_i^b$, where a and $b \in [2, 3, 4]$ are scale constant value constant selected with respect to the term $f(\epsilon_i)$. It measures how much it costs to perturb the response. A lower noise variance, will then increase the cost. The second component is the loss function in the learning model. It gives the distance between the prediction and the real value in dataset. The more accurate learning prediction is, the lower the cost.

The second component f at time slot k represents the Banach contraction parameter at k, hence dynamic, and is non-negative continuous and increasing given as follows:

$$\begin{aligned} f(\epsilon_i) =^{(13)}\ & \mathcal{T}^\rho(\theta_k) \\ =\ & \mathcal{T}^\rho(\theta_{k-1}) + (\theta_k - \theta_{k-1})(\mathbb{I} - \frac{2\rho}{n_\ell} X_\ell' X_\ell \\ & - \frac{2\rho}{n_\ell} \sum_{j \in \mathcal{N} \setminus \{i\}} n_j w(j, k, \epsilon_j)) \end{aligned} \tag{17}$$

As stochastic gradient descent is utilized, we use the averaging magnitude of the noise power for brevity.

$$\overline{w}(i, k, \epsilon_i) = \frac{m_{i,k}}{n^2} \sum_{i \in \mathcal{N}} \frac{1}{\epsilon_i^2}. \tag{18}$$

where $m_{i,k}$ is a parameter chosen according to the step size in each iteration in learning.

That is increasing the precision (lower noise variance) leads to a higher disclosure cost. In contrast, increasing the accuracy helps training learning model and decrease the prediction cost.

Unique Nash Equilibrium

A Nash Equilibrium is a strategy profile ε^* satisfying

$$\epsilon_i^* \in \arg\min J_i(\epsilon_i, \epsilon_{-i}) \tag{19}$$

for all $i \in \mathcal{N}$.

We first observe that ϵ_i is a nonempty, convex, compact subspace of a Euclidean space $\mathbb{R}^N$. At each iteration, each has a strategy space that is continuous and defined by a minimum, a maximum, and all ϵ in between.

Proposition 1. *ϵ^* is a unique Nash equilibrium with cost functions $J_i(\epsilon_i, \epsilon_{-i})$ that are continuous and strictly convex. See proof in Appendix.*

Experimental Validation

In this section, numerical tests are in two scenarios.

The first scenario is collaborative learning with privacy concerns. In this case, there are multiple data owners training with both their own datasets and the Sub-gradient response which contains useful information in ML training model from each other. Differentially private noise is added to the query response to ensure privacy.

The second scenario is collaborative learning with privacy concerns with game theory adjusting the magnitude of differentially private noise. In this case, for each data owner i, the privacy budget ϵ_i is updated in each game stage so as to balance the utility and privacy leakage regarding to the learning loss and privacy budget respectively.

The Home Loans dataset and the interest rates prediction Application We use a lending dataset with a linear

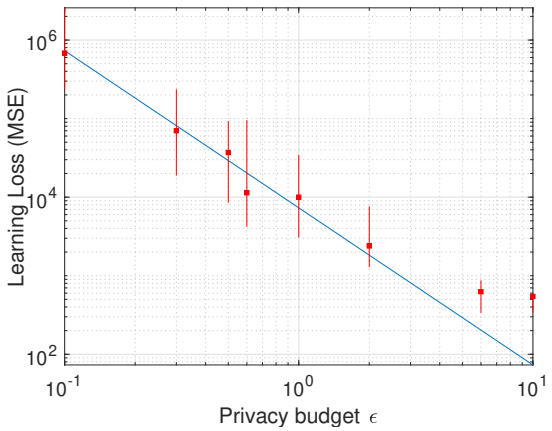

Figure 2: Statistic of Learning loss (MSE) versus privacy budget ϵ. The solid blue line is the prediction of the learning loss value by fixed point contraction method

Number of players	3 players	4 players	5 players
SC(with game)	1.3230	1.2060	0.5640
RMSE(with game)	$2.62e^2$	$1.70e^2$	83.6
Average ϵ	0.6433	0.5250	0.3160
SC(without game)	0.03	0.04	0.05
RMSE(without game)	$1.60e^3$	$8.26e^2$	$4.87e^2$
ϵ for all players	0.1	0.1	0.1
SC(without game)	2.3234	2.7233	3.5824
RMSE(without game)	$1.85e^2$	$1.00e^2$	55.4
ϵ for all players	1	1	1
SC(without game)	3.3446	4.8803	4.9494
RMSE(without game)	23.33	18.42	24.02
ϵ for all players	10	10	10

Table 1: Social Cost (SC) and Learning Loss(RMSE) for different players with and without game

regression model to demonstrate the value of the methodology and to validate the theoretical results. The dataset contains information regarding nearly 890,000 loans made on a peer-to-peer lending platform, called the Lending Club. 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 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 split this dataset into several non-overlapping datasets for different data owners and test datasets.

Then, we compared the learning loss between different data owners in two different scenarios:

1. Collaborative ML training with privacy concerns,
2. Collaborative learning with privacy concerns with game theory adjusting the magnitude of DP noise.

We use five data owners with 1000, 2783, 7743, 21545, 59949 entities for each of them. This is to make sure there

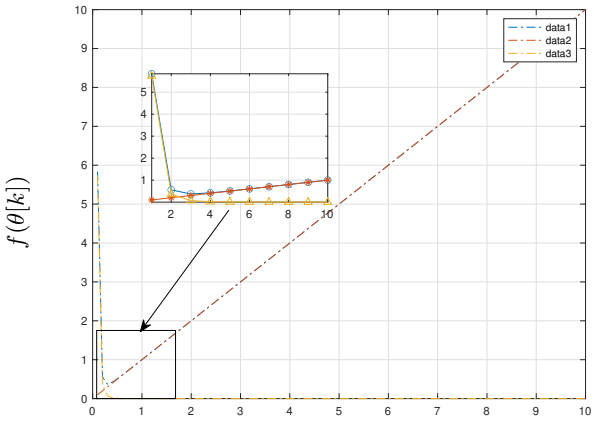

Figure 3: Utility and cost of one data owner as a function of the perturbation ϵ.

are relative large and small datasets in our data market. The learning loss for collaborative learning with no privacy is set as the baseline, where DP noise magnitude is set to be zero.

First, we demonstrate the performance of the learning parameter prediction, which is shown as learning loss in Mean Square Error (MSE), versus privacy budget value in Figure 2. The result is stochastic because the data owners provide differentially-private responses to the gradient queries. The prediction of the learning loss value is fitted to the real learning outcome. Then, we test the outcome of the game reaches equilibrium. In Figure 3, data1, data2 and data3 are plots for payoff function, cost, and utility respectively. There exists one minimum value for the payoff function at the end of the game. After the game for multiple data owners with collaborative learning is performance as desired, we invest how game helps in choosing the best response of privacy budget value ϵ for all players as in Table 1.

As shown in Figure 4, the average learning loss with $\epsilon = 10$ for all data owners is the lowest. With the increase of the DP noise magnitude, the average learning loss increases non-linearly. The Equilibrium result between the learning loss and privacy budget ϵ is shown in red plots in this figure. The average learning loss is close to the case when privacy budget $\epsilon = 1$. There is a significant improvement in privacy, which is about 47.5%.

Social Welfare

Then we evaluate the social welfare for all players at the end of the game. We pick three cases with 3, 4, and 5 players for validation. We define the social cost per player :

$$SC = \frac{1}{\mathbb{N}} \sum_{i \in \mathbb{N}} J_i(\epsilon_{i,k=T}).$$

As shown in Figure 5, the social cost for collaborative learning with game is lower in the case when $\epsilon = 1$ with any number of players. The outcome of the game shows that the

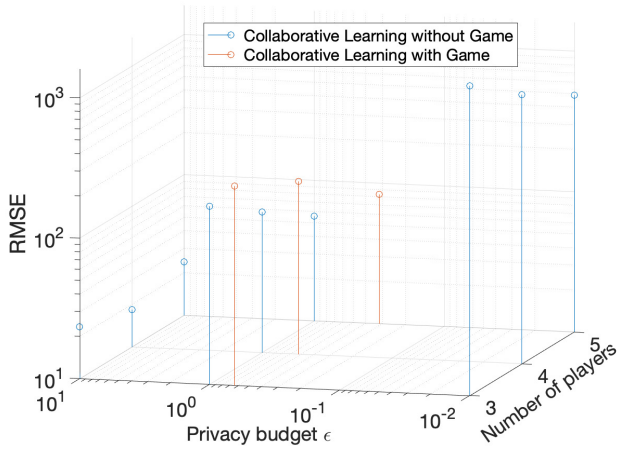

Figure 4: Learning loss (RMSE) versus privacy budget ϵ for 3, 4, and 5 data owners in collaborative learning with and without game

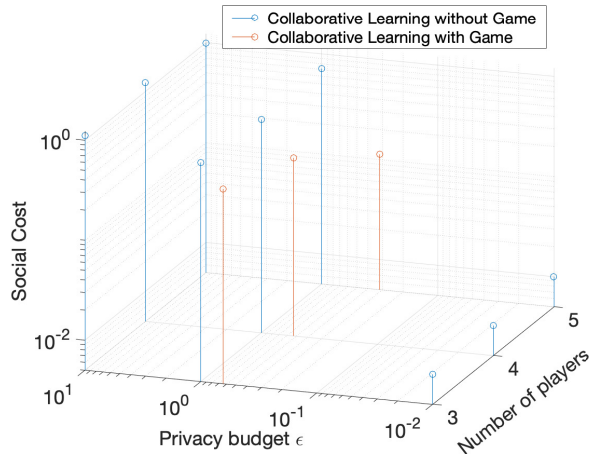

Figure 5: Social cost (SC) versus privacy budget ϵ for 3, 4, and 5 data owners in collaborative learning with and without game

learning loss and privacy budget not only reaches a balance at the equilibrium but also minimises the social cost per player. The social cost without game is higher than with game for all cases other than for $\epsilon = 0.1$. This is due to the DP noise at $\epsilon = 0.1$ is way too large and the learning loss value for all players are equivalent high. Thus, social welfare is preserved as desired.

Conclusion

In this paper, we interpreted gradient methods as fixed-point iterations, and used the concept of Banach contraction to make a prediction of training loss among a privacy concerned collaborative machine learning model. Then, we constructed a game based on linear regression. Our analysis of the Nash equilibrium assumed complete information, that is that agents know the costs and features of other agents. Our model also assumed that the private budget chosen by each player is known by each other player. We established the existence of a unique Nash Equilibrium for the game, which also gave good social welfare across data owners.

References

Bordenabe, N. E.; Chatzikokolakis, K.; and Palamidessi, C. 2014. Optimal geo-indistinguishable mechanisms for location privacy. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, 251–262.

Brenner, H.; and Nissim, K. 2010. Impossibility of differentially private universally optimal mechanisms. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, 71–80. IEEE.

Dwork, C.; and Pappas, G. J. 2017. Privacy in information-rich intelligent infrastructure. *arXiv preprint arXiv:1706.01985* .

Dwork, C.; Roth, A.; et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9(3–4): 211–407.

Farokhi, F.; Wu, N.; Smith, D.; and Kaafar, M. A. 2020. The Cost of Privacy in Asynchronous Differentially-Private Machine Learning. *arXiv preprint arXiv:2003.08500* .

Ghosh, A.; Roughgarden, T.; and Sundararajan, M. 2012. Universally utility-maximizing privacy mechanisms. *SIAM Journal on Computing* 41(6): 1673–1693.

Graepel, T.; Lauter, K.; and Naehrig, M. 2012. ML confidential: Machine learning on encrypted data. In *International Conference on Information Security and Cryptology*, 1–21. Springer.

Gupte, M.; and Sundararajan, M. 2010. Universally optimal privacy mechanisms for minimax agents. In *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 135–146.

Huang, Z.; Hu, R.; Gong, Y.; and Chan-Tin, E. 2018. DP-ADMM: ADMM-based Distributed Learning with Differential Privacy. *Preprint:* `arXiv preprint arXiv:1808.10101` .

Hunt, T.; Song, C.; Shokri, R.; Shmatikov, V.; and Witchel, E. 2018. Chiron: Privacy-preserving machine learning as a service. *arXiv preprint arXiv:1803.05961* .

Jung, A. 2017. A fixed-point of view on gradient methods for big data. *Frontiers in Applied Mathematics and Statistics* 3: 18.

Kalantari, K.; Sankar, L.; and Sarwate, A. D. 2018. Robust Privacy-Utility Tradeoffs under Differential Privacy and Hamming Distortion. *IEEE Transactions on Information Forensics and Security* 13(11).

Konečnỳ, J.; McMahan, H. B.; Yu, F. X.; Richtárik, P.; Suresh, A. T.; and Bacon, D. 2016. Federated learning: Strategies

for improving communication efficiency. *arXiv preprint arXiv:1610.05492* .

Li, P.; Li, J.; Huang, Z.; Li, T.; Gao, C.-Z.; Yiu, S.-M.; and Chen, K. 2017. Multi-key privacy-preserving deep learning in cloud computing. *Future Generation Computer Systems* 74: 76–85.

Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; and Xu, L. 2016. Edge computing: Vision and challenges. *IEEE Internet of Things Journal* 3(5): 637–646.

Shokri, R.; and Shmatikov, V. 2015. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 1310–1321. ACM.

Shor, N. Z. 2012. *Minimization methods for non-differentiable functions*, volume 3 of *Springer Series in Computational Mathematics*. Berlin, Heidelberg: Springer.

Taghizadeh, A.; Kebriaei, H.; and Niyato, D. 2020. Mean Field Game for Equilibrium Analysis of Mining Computational Power in Blockchains. *IEEE Internet of Things Journal* .

Wu, N.; Farokhi, F.; Smith, D.; and Kaafar, M. 2020. The Value of Collaboration in Convex Machine Learning with Differential Privacy. In *2020 IEEE Symposium on Security and Privacy (SP)*, 485–498. Los Alamitos, CA, USA: IEEE Computer Society. ISSN 2375-1207. doi:10.1109/SP40000.2020.00025. URL https://doi.ieeecomputersociety.org/10.1109/SP40000.2020.00025.

Xu, L.; Jiang, C.; Chen, Y.; Wang, J.; and Ren, Y. 2016. A framework for categorizing and applying privacy-preservation techniques in big data mining. *Computer* 49(2): 54–62.

Xu, L.; Jiang, C.; Wang, J.; Ren, Y.; Yuan, J.; and Guizani, M. 2015. Game theoretic data privacy preservation: Equilibrium and pricing. In *2015 IEEE International Conference on Communications (ICC)*, 7071–7076.

Zhan, Y.; Li, P.; Qu, Z.; Zeng, D.; and Guo, S. 2020. A learning-based incentive mechanism for federated learning. *IEEE Internet of Things Journal* .

Zhang, T.; He, Z.; and Lee, R. B. 2018. Privacy-preserving machine learning through data obfuscation. *arXiv preprint arXiv:1807.01860* .

Zhang, T.; and Zhu, Q. 2017. Dynamic differential privacy for ADMM-based distributed classification learning. *IEEE Transactions on Information Forensics and Security* 12(1): 172–187.

Proof of Lemma 1

We have $\nabla f(\theta) = 0$ if and only if the vector $\theta \in \mathbb{R}^n$ is a fixed point of the operator $\mathcal{T}^\rho$. Thus, $\nabla f(\theta) = 0$ if and only if $\mathcal{T}^\rho(\theta) = \theta$, $\theta_i = (Y_i' - \frac{1}{2}Q_i X_i^{-1})X_i'^{-1}$.

Proof. Let θ be a fixed point of $\mathcal{T}^\rho$, i.e.,

$$\mathcal{T}^\rho(\theta) = \theta. \tag{20}$$

Then,

$$\begin{aligned}
&\mathcal{T}^\rho(\theta) - \theta = 0, \\
&\frac{\rho}{n_\ell}\left[2(\theta_i X_i' - Y_i')X_i + Q_i\right] = 0, \\
&(Y_i' - \theta_i X_i')X_i = \frac{1}{2}Q_i, \\
&Y_i' - \theta_i X_i' = \frac{1}{2}Q_i X_i^{-1}, \\
&\theta_i X_i' = Y_i' - \frac{1}{2}Q_i X_i^{-1}, \\
&\theta_i = (Y_i' - \frac{1}{2}Q_i X_i^{-1})X_i'^{-1}.
\end{aligned} \tag{21}$$

□

As Q_i is different in each iteration k, such fixed point is then dynamic regarding to the received responses from all other data owners. In stead, the expectation of the fixed point is calculated with regard to the aggregation of all datasets $D_\ell = (D_i, i \in \mathcal{N})$, $X_\ell = (X_i, i \in \mathcal{N})$,$Y_\ell = (Y_i, i \in \mathcal{N})$, and a expectation value of DP noise $\mathbb{E}(\epsilon_\ell)$ from each data owner.

Proof.

$$\begin{aligned}
&\mathcal{T}^\rho(\theta) - \theta = 0, \\
&\frac{\rho}{n_\ell}\left[2(\theta X_\ell' - Y_\ell')X_i + (n_\ell - 1)\mathbb{E}(\epsilon_\ell)\right] = 0, \\
&(Y_\ell' - \theta X_\ell')X_i = \frac{n_\ell - 1}{2}\mathbb{E}(\epsilon_\ell), \\
&Y_\ell' - \theta X_\ell' = \frac{n_\ell - 1}{2}\mathbb{E}(\epsilon_\ell)X_\ell^{-1}, \\
&\theta X_\ell' = Y_\ell' - \frac{n_\ell - 1}{2}\mathbb{E}(\epsilon_\ell)X_\ell^{-1}, \\
&\theta = (Y_\ell' - \frac{n_\ell - 1}{2}\mathbb{E}(\epsilon_\ell)X_\ell^{-1})X_\ell'^{-1}.
\end{aligned} \tag{22}$$

□

Proof of Lemma 2

Assume that for some $q \in [0, 1)$, we have

$$\|\mathcal{T}^\rho \mathbf{a} - \mathcal{T}^\rho \mathbf{b}\| \leq q\|\mathbf{a} - \mathbf{b}\|, \tag{23}$$

for any $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$. Then, the operator $\mathcal{T}^\rho$ has a unique fixed point θ_0 and the iterates $\theta^{(k)}$ satisfy

$$\|\theta^{(k)} - \theta_0\| \leq q^k\|\theta^{(0)} - \theta_0\|. \tag{24}$$

Proof. Conversely, we assume there are two different fixed points, such that

$$\mathbf{a} = \mathcal{T}^\rho(\mathbf{a}), \mathbf{b} = \mathcal{T}^\rho(\mathbf{b}).$$

$$\|\mathcal{T}^\rho(\mathbf{a}) - \mathcal{T}^\rho(\mathbf{b})\| = \|\mathbf{a} - \mathbf{b}\|$$

Then, by (23),

$$\|\mathbf{a} - \mathbf{b}\| \leq q\|\mathbf{a} - \mathbf{b}\|$$

So, $q \geq 1$ or $\|\mathbf{a}-\mathbf{b}\| = 0$.

However, since $q \in [0,1)$, we must have $\mathbf{a} = \mathbf{b}$. Thus, there is only one fixed point exists.

□

$$\begin{aligned}\mathcal{T}^{\rho}(\mathbf{a}) - \mathcal{T}^{\rho}(\mathbf{b}) &= (\mathbf{a}-\mathbf{b}) - \frac{\rho}{n_\ell}\Big[(n_i\nabla\overline{f(\mathbf{a})} + Q_a) \\ &\quad - (n_i\nabla\overline{f(\mathbf{b})} + Q_b)\Big] \\ &= (\mathbf{a}-\mathbf{b}) - \frac{\rho n_i}{n_\ell}(\nabla\overline{f(\mathbf{a})} - \nabla\overline{f(\mathbf{b})}) \\ &\quad - \frac{\rho}{n_\ell}(Q_a - Q_b)\end{aligned} \tag{25}$$

Because $\nabla f(.)$ is a continuous and differentiable function. So there exist a point $\mathbf{m} = c\mathbf{a} + (1-c)\mathbf{b}, c \in [0,1]$ such that

$$\begin{aligned}\mathcal{T}^{\rho}(\mathbf{a}) - \mathcal{T}^{\rho}(\mathbf{b}) &= (\mathbf{a}-\mathbf{b}) - \frac{\rho n_i}{n_\ell}[(\mathbf{a}-\mathbf{b})\nabla^2 f(\mathbf{m}) \\ &\quad + 2\mathbb{E}(\epsilon_\ell)] - \frac{\rho}{n_\ell}(Q_a - Q_b) \\ &= (\mathbf{a}-\mathbf{b})(I - \frac{\rho n_i}{n_\ell}\nabla^2 f(\mathbf{m})) - 2\frac{\rho n_i}{n_\ell}\mathbb{E}(\epsilon_\ell) \\ &\quad - \frac{\rho}{n_\ell}(Q_a - Q_b)\end{aligned} \tag{26}$$

Assume $Q_a = Q_b$, so

$$\begin{aligned}\mathcal{T}^{\rho}(\mathbf{a}) - \mathcal{T}^{\rho}(\mathbf{b}) &= (\mathbf{a}-\mathbf{b})(I - \frac{\rho n_i}{n_\ell}\nabla^2 f(\mathbf{m})) \\ &\quad - 2\frac{\rho n_i}{n_\ell}\mathbb{E}(\epsilon_\ell)\end{aligned} \tag{27}$$

$$\begin{aligned}\|(\mathbf{a}-\mathbf{b})(I - \frac{\rho n_i}{n_\ell}\nabla^2 f(\mathbf{m}))\| &\leq q\|\mathbf{a}-\mathbf{b}\| \\ q &\geq \|\mathbb{I} - \frac{\rho n_i}{n_\ell}\nabla^2 f(\mathbf{m})\|\end{aligned} \tag{28}$$

Proof.

$$\begin{aligned}\mathbb{E}(\epsilon_\ell) =& \mathbb{E}\left\{\left\|\left(\frac{1}{\sum_{\ell\in\mathcal{N}} n_\ell}\right)\sum_{\ell\in\mathcal{N}} n_\ell w_\ell[k]\right\|_2^2\right\} \\ =& \left(\frac{1}{\sum_{\ell\in\mathcal{N}} n_\ell}\right)^2 \sum_{\ell\in\mathcal{N}} n_\ell^2 \mathbb{E}\{\|w_\ell[k]\|_2^2\} \\ =& \left(\frac{1}{\sum_{\ell\in\mathcal{N}} n_\ell}\right)^2 \sum_{\ell\in\mathcal{N}} \frac{8\Xi^2 T^2}{\epsilon_\ell^2} \\ =& \frac{8\Xi^2 T^2}{n^2}\sum_{\ell\in\mathcal{N}}\frac{1}{\epsilon_\ell^2}.\end{aligned} \tag{29}$$

□

Proof of Proposition 1

Proof.

$$\nabla J(\epsilon) = \nabla(c) + \nabla f(\epsilon) \tag{30}$$

$$\begin{aligned}\nabla J(\epsilon = 0) &\propto \frac{\partial \mathbb{E}(\epsilon_\ell)}{\partial \epsilon} \\ &= -\frac{m}{n^2}\sum_{i\in\mathcal{N}}\frac{2}{\epsilon_i^3}.\end{aligned} \tag{31}$$

$$\begin{aligned}\nabla^2 J(\epsilon = 0) &\propto \frac{\partial^2 \mathbb{E}(\epsilon_\ell)}{\partial \epsilon^2} \\ &= \frac{m}{n^2}\sum_{i\in\mathcal{N}}\frac{6}{\epsilon_i^4}.\end{aligned} \tag{32}$$

$\nabla J(\epsilon = 0) < 0$ and $\nabla^2 J(\epsilon = 0) > 0$

So this cost function $\nabla J(\epsilon)$ is strictly convex with respect to action ϵ. This concludes the proof. □