Asynchronous Interactive Distributed Private Multitask Learning Framework with Trustworthy Data Aggregator

Liyang Xie Manni Liu

Abstract

*In recent years, multi-task learning (MTL) has proved to be a powerful learning framework that promotes performance in supervised learning problems by transferring knowledge among multiple tasks. Distributed MTL is one prevalent setting where the data is separated across different locations. The issue of privacy arises when distributed MTL is applied on the geographically separated data. These datasets contains personal information such as financial and medical records, which is very sensitive and should be away from exposure. Considering the proliferation of various private data, privacy-preserving distributed MTL frameworks are in great need. On the other hand, differential privacy (DP) is one of the most important privacy concepts that are suitable for this genre of frameworks. In this report, we put up with a novel idea: **A**synchronous **I**nteractive **D**istributed **P**rivate **M**ultitask **L**earning **F**ramework with Trustworthy Data Aggregator (AIDPMTLF) to address the privacy issue under distributed MTL setting. The proposed framework adds carefully designed perturbation on the data aggregator. We also provide theoretical guarantees of the proposed framework and extensive empirical results to illustrate our idea.*

1. Introduction

MTL has become a provalent tool in dealing with the problem of learning from the distributed data in the epoch of big data. In spite of its proliferation, challenges also arise. One indispensable challenge is about how the privacy of sensitive data is protected in distributed MTL framework. For example, medical centers in different countries may make a joint effort to conduct medical research, while the data may not be distributed because of its sensitive nature.

On sight, it seems that erasing personal information such as names can protect individual privacy. However, as the development of machine learning algorithms, it is still possible to extract patterns from remaining information and obtain personal information. Back to 2007, the famous media service provider Netflix published an anonymous dataset called Netflix Prize dataset, and encouraged researchers to design a better recommendation system. The dataset contains 10 million movie rankings from 500,000 customers. However, even customers' personal details like usernames and locations are removed and replaced by random numbers, some information is still deanonymized by comparing rankings and timestamps with public resources from the Internet Movie Database (IMDb) [**?**]. Other works also demonstrate it. [**?**] and [**?**] can extract hidden information from an adversary. Even genetic datasets can leak personal information[**?**, **?**].

As early as 1977, [**?**] has defined a concept for the ideal database: nothing about an individual that can not be learned without the database should be learned from the database. Although the idea is too demanding and is later demonstrated theoretically that this desideratum can not be reached[**?**], effort can still be done to reduce the risk of leaking personal information stored in the dataset. In a word, differential privacy aims at maximizing the accuracy of queries to a database while minimizing the possibility of identifying a single record.

In this paper, we proposed a new method to address the issue arising from Smith *et al.* [**?**]. In [**?**], the authors provided the answer to the question: "how much interaction is necessary to optimize convex functions in the local DP model?". Although this non-interactive setting achieves good performance, the testing result under the scenario of distributed MTL with trustworthy aggregator is far from satisfying. This is due to the fact that multitask learning aims at improving each user's performance with the help from all the others, which requires a lot of information exchange. Also in real world cases, heterogeneity of local data and tasks make it very hard to finish training in a few rounds. In addition, a method with few interactive or non-interactive typically requires a large amount of local datasets. This is hard to achieve due to the scarcity of the data under distributed MTL senses.

In this paper, we will show that, with a trustworthy data aggregator, asynchronous interaction performs better than the proposed one in [?] . In addition, [?] mentioned that it is difficult to implement an interactive framework for private data learning because of two reasons: (1) long network latency; (2) the server has to be online for asynchronous updates. The first issue can be addressed with the asynchronous update, whose effect will be further reduced with weighted update mechanism we proposed. The second issue can be addressed by data backup, which is easy to achieve because the complexity of the algorithm in aggregator is moderate.

In summary, this paper makes the following contributions:

- We present the first distributed private learning system with fast light-weight interactions under the condition of a trustworthy central data processor.
- We carefully design the noise perturbation algorithm that is added on the aggregator and prove that the proposed algorithm guarantees local differential privacy (LDF) [?]–one of the most important variants of DP. We check the correctness of our method using state-of-the-art tool [?].
- We reduce the accumulation effect of privacy leakage during iterative updates. We also explore the effect of different distributions of the delay and use weights to eliminate the effect of the delay as well as exploring other delay-reduce methods.
- We demonstrate our method with sufficient empirical evidences, which contain multiple real world settings such as task heterogeneity and data heterogeneity,

The paper is arranged as follows: we first introduce the related work in section 2 with respect to differential privacy and distributed rrivate learning. Next in section 3, we give a detailed description on our model and method. Experiments results are provided in section 4, as well as analysis. We also discuss our technical Weaknesses and potential future work in section 5. We conclude our work in section 6 and provide details on project group members in section 7.

2. Related Work

Differential privacy. Enormous algorithms have been proposed for privacy-preserving data mining[?, ?, ?, ?]. But composition attacks and auxiliary information become big problem for these algorithms. On the other hand, differential privacy acts as an resistance against composition attacks and auxiliary information.

When differential privacy is first put forward by C.Dwork[?], a sensitivity method is also introduced. If we denote the objective function as J and the true query result coming from algorithm $\mathcal{A}$ is represented by $\mathcal{A}(D) = argmin\ J$, the output query result is $\mathcal{A}(D) + b$ where b is an random noise with density $\frac{1}{\alpha}e^{-\beta\|b\|}$. β is a function of ϵ and the L_2-sensitivity of $\mathcal{A}(\cdot)$. The sensitivity method is a typical output perturbation method.

As for the objective perturbation methods, the objective function is turned to be $J(f, D) + \frac{1}{n}b^T f$ where f is the predictor and n is the number of training data points. If the objective function is strongly convex with some constraints on the loss function, objective perturbation proves theoretically better than output perturbation. More details can be found in [?].

Distributed Private Learning. There are also important studies on distributed differentially private Learning. Xie *et al.* [?] for the first time, provide a privacy-preserving distributed MTL framework combined with distributed asynchronous MTL framework. The proposed method successfully address the issue of time delay caused by synchronized optimization algorithms. It is different from our settings because it assumes that the central server is untrustworthy and adding noise from local task side may significantly reduce the overall performance. Xie *et al.* [?] proposed a ensemble learning method for merging binary classifiers (or regressors) trained on local data. This method, though can be applied in our paper's setting with "public-private" case, does not help much due to data heterogeneity. In [?] the authors proved that the method [?] has near-optimal performance under certian conditions. Han *et al.* [?] present a distributed optimization algorithm (with constrained domain) that preserves differential privacy. This method may not achieve good performance due to its exponential mechanism. Rajkumar and Agarwal [?] describe a new differentially private algorithm for the multiparty setting that uses a stochastic gradient descent based procedure to directly optimize the overall multiparty objective. This paper does not address the issue of accumulation effect of privacy leakage. Hamm *et al.* [?] proposed a method of building a global differentially private classifier from locally classifiers from multiple local users without access to their private data. Similar as [?], it does not help due to data heterogeneity. Pathak *et al.* [?] proposed a privacy-preserving framework for composing a differentially private aggregate classifier using local trained classifiers by separate mutually untrusting parties. This method requires other encryption method, which may lead to more cost.

3. Methodology

In this study we aim to provide solution to distributed MTL problems with asynchronous interaction between local learning models and a trustworthy central server. In the following sections we first describe the regularized MTL and analyze its distributed version, which is the foundation of our framework. Next we introduce the concept of DP and describe its importance in machine learning. Then we show that DP is an indispensable part of the proposed framework. With necessary precondition and assumptions, we provide detailed description of our framework.

3.1. Regularized MTL and Its Distributed Version

The relatedness among learning tasks is the foundation of MTL. In our settings we assume that there are totally T tasks. Let d be data dimension and n_t be the number of data points in task t, task t contains a dataset $\mathcal{D}_t = \{X_t, \mathbf{y}_t\}$, where $X_t \in \mathbb{R}^{n_t \times d}$ is the data matrix with feature dimensionality d, $\mathbf{y}_t \in \mathbb{R}^{n_t}$ is the corresponding label vector. For each local task, a model $f(\mathbf{x};\mathbf{w}) : \mathbb{R}^d \to \mathbb{R}$ is learned. To predicts y, we use learned $\mathbf{w}$ and feature vector $\mathbf{x}$ in testing set. Note that we use linear model in this paper and hence $f(\mathbf{x};\mathbf{w}) = \mathbf{x}^T\mathbf{w}$. Let $\ell_{t,i}(\mathbf{w}_t) = \ell(f(\mathbf{x}_{t,i};\mathbf{w}_t), y_{t,i})$ be the loss for the task t's ith sample with loss function ℓ. Let $W = [\mathbf{w}_1, \ldots, \mathbf{w}_T] \in \mathbb{R}^{d\times T}$ be the model matrix whose ith column is the task model $\mathbf{w}_t$. Regularized MTL solves the following problem:

$$\min_W \left\{ \sum_{t=1}^{T} \left(\frac{1}{n_t} \sum_{i=1}^{n_t} \ell_{t,i}(\mathbf{w}_t) \right) + \lambda r(W) \right\} \tag{1}$$

Here $r(\mathbf{W})$ serves as the regularization to induce task relatedness according to different relatedness assumptions [**?**, **?**]. λ is the parameter that determines the strength of knowledge transfer. This is centralized version of MTL.

An alternative representation of 1 is the following:

$$\min_{P,Q} \left\{ \sum_{t=1}^{T} \left(\frac{1}{n_t} \sum_{i=1}^{n_t} \ell_{t,i}(\mathbf{p}_t + \mathbf{q}_t) \right) + \lambda r(P) + \tau g(Q) \right\} \tag{2}$$

where $\mathbf{w}_t = \mathbf{p}_t + \mathbf{q}_t$ and $W = P + Q$. $r(P)$ performs the knowledge transfer and $g(Q)$ regulates the model complexity. The motivation of this representation is that in MTL setting, the knowledge of learned parameter $\mathbf{w}_t$ contains two components: a shared component $\mathbf{p}_t$ that comes from tasks other than t and a task specific component $\mathbf{q}_t$ that contains the local knowledge. This approach provides the flexibility to trade off between the shared one and the task specific one during training.

Our goal now is to make 2 into a distributed version such that it can be solved with distributed learning techniques. Following is the proximal operator that can help transforming 2 into a distributed fashion:

$$\text{prox}_r^{\mu}(X) = \text{argmin}_W \left\{ \tfrac{1}{2}\|W - X\|_F^2 + \mu r(W) \right\}, \tag{3}$$

where μ is a coefficient obtained by the step size and the regularization parameters, and $r(W)$ is required to be a proper and lower semi-continuous function.

With definition 3, representation 2 can be expressed as the following distributed version:

$$Q^+ = Q^- - \alpha \nabla_Q f(Z^-) \tag{4}$$

$$P^+ = \text{prox}_r^{\alpha\lambda}(P^- - \alpha \nabla_P f(Z^-)) \tag{5}$$

where $f(Z)$ is the loss function which has the following expression:

$$\begin{aligned} f(Z) &= f(P, Q) \\ &= \sum_{t=1}^{T} \left(\frac{1}{n_t} \sum_{i=1}^{n_t} \ell_{t,i}(\mathbf{p}_t + \mathbf{q}_t) \right) + \tau g(Q) \end{aligned}$$

where $Z = \begin{bmatrix} P \\ Q \end{bmatrix} \in \mathbb{R}^{2d\times T}$ is the parameter vector.

The above steps are indeed easy to be distributed: Q can be decoupled and distributed for each task for fixed p_t. The tth local task receives the current shared component $\mathbf{p}_t^-$ from the central server, computes the gradient $\nabla_{\mathbf{p}_t} f(Z^-)$ (using task data $\mathcal{D}_t$), sends it back to the server, and finally locally updates $\mathbf{q}_t^+$ using its data. We note that the gradient $\nabla_{\mathbf{p}_t} f(Z^-)$ can be computed locally because of the following:

$$\begin{aligned} \nabla_{\mathbf{p}_t} f(Z) &= \nabla_{\mathbf{p}_t} f(\mathbf{p}_t, \mathbf{q}_t) \\ &= \nabla_{\mathbf{p}_t} \left\{ \frac{1}{n_t} \sum_{i=1}^{n_t} \ell_{t,i}(\mathbf{p}_t + \mathbf{q}_t) + \tau g_i(\mathbf{q}_t) \right\} \end{aligned}$$

where the computation only depends on the task data $\mathcal{D}_t$. After all the gradients $[\nabla_{\mathbf{p}_1} f(Z^-), \ldots, \nabla_{\mathbf{p}_T} f(Z^-)]$ are received, the server immediately performs proximal computation as in Equation (5), and then sends the columns to the corresponding task nodes.

3.2. The Necessity of Privacy and Differential Privacy

Data such as financial records and patients' MRI images contain sensitive information, which raises privacy issues in machine learning. Therefore it is necessary to take the issue of data privacy into consideration. The *fundamental assumptions* in our framework are: (1) data aggregator is trustworthy, (2) communication channels are trustworthy, (3) local tasks are not trustworthy, which is very different from the settings in Xie *et al.* [**?**]. Our setting widely appears in real world scenario and hence solving it will make great contribution. In our framework, the gradient $\nabla_{\mathbf{p}_t} f(\mathbf{p}_t^-, \mathbf{q}_t^+)$ sent to the data aggregator contains private information from local task. However there is not need to add noise before sending out the gradient due to the trustworthy aggregator. After the knowledge transfer in aggregator, the gradient send back to certain local task contains sensitive information from all the other local task, which requires privacy protection.

In this paper, we aim to protect the *differential privacy* of each data point in each local task (formally defined in Definition 1). Differential privacy [**?**] provides a quantifiable level of privacy with respect to the individual data points. First we provide a mathematical definition of ϵ-differential privacy which is very essential to differential privacy:

Theorem 1 ***(ϵ-differential privacy[?])*** *Let ϵ be a positive real number and $\mathcal{A}$ be an randomized algorithm that takes a dataset D as input. Let im$\mathcal{A}(D)$ denote the output of feeding D to $\mathcal{A}$. The algorithm $\mathcal{A}$ is* ***ϵ-differentially private*** *if for any two datasets D_1 and D_2 that differ on a single element (i.e., the data of one person), and all subsets S of im$\mathcal{A}$,*

$$Pr[\mathcal{A}(D_1) \in S] \leq \epsilon \times Pr[\mathcal{A}(D_2) \in S]]$$

where the probability is taken over the randomness used by the algorithm.

ϵ-differential privacy is a privacy measurement which is famous for its robustness to know attacks and has been widely applied in subsequent research. In an intuitive way, an algorithm is regarded as satisfying ϵ-differential privacy if modifying the data of one person doesn't effect the output distribution a lot. Since most differential privacy algorithms are designed on the basis of adding noises to the original dataset, the degree of how much privacy is violated can be quantified with this theoretical definition, and thus we can measure the effectiveness of an algorithm.

3.3. Differentially Private Distributed MTL

As we mentioned in section 3.2, privacy issue in the non DP framework requires us to propose a privacy-preserving framework such that it can not only transfer accurate information among learning task to help improve performance but also provide privacy guarantee. Our idea is to add noise on the matrix (P^+ in 5) such that the returned gradients are perturbated. The perturbation method is under developed. We summarize our idea in Figure 1.

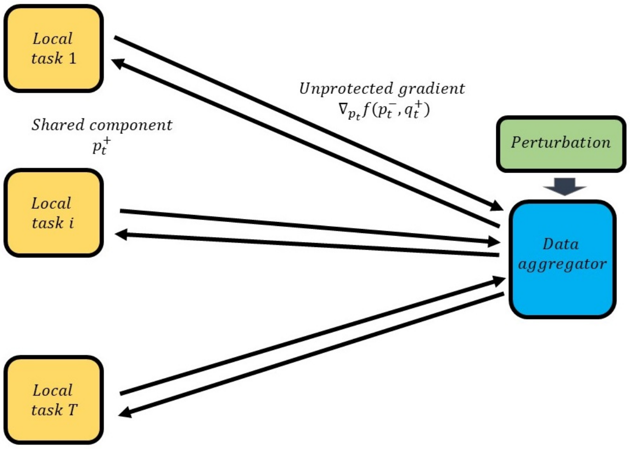

Figure 1. Proposed framework.

4. Experiments and Discussions

We first present the convergence of proposed method and baseline under non-private and private cases, under synthetic data on the centrallized MTL framework.

We then present the convergence of proposed method and baseline under non-private and private cases, under synthetic and real data.

Privacy and delay

5. Future work

AIDPMTLF can learn big data well while protecting the private information as shown in section 4. However, the protection of privacy never stops as attack models arises and develops everday. For the future work, we still consider differential privacy as perspective for multi-task learning problems. Further contribution can be made from: 1) specfic model design for specific datasets. Because each dataset faces different attacks. For example, Netflix Prize dataset is threatened as the existence of public resources from the Internet Movie Database. Financial and medical records all face different information leakage ways. **AIDPMTLF** disolve the privacy issue in a general case. More specific modication can be made on the basis of it when facing different multi-task learning problems. 2) Noises can be analyzed and used to improve the model. Feedback mechanism may produce promotion. 3) Different network structures may incur unexpected results. We use basic neural networks in this paper. Creative network design is promising for amazing achievement for big data analysis.

6. Conclusion

In this paper, we highlight the contribution of multi-task learning on current research of big data analysis. But we also analyze the limitations of it on private datasets. Privacy has been an enduring topic in computer science and appeals a lot of concentration. We introduce differential privacy and emphasize its good performance on privacy-preserving. By combing multi-task learning and differential privacy, we come up with a model called Asynchronous Interactive Distributed Private Multitask Learning Framework with Trustworthy Data Aggregator (AIDPMTLF). Experiments in section 4 reveals its good performance on big data while protecting the data privacy. We further analyze its weakness and discuss the future work.

7. Appendix

Liyang Xie: Methodology part, experiment part, paper revision.

Manni Liu: Background Survey, presentation part, paper revision.