Aggregating Private Sparse Learning Models Using Multi-Party Computation

Lu Tian, Bargav Jayaraman, Quanquan Gu, and David Evans
University of Virginia
https://oblivc.org/ppml

Abstract

We consider the problem of privately learning a sparse model across multiple sensitive datasets, and propose learning individual models locally and privately aggregating them using secure multi-party computation. In this paper, we report some preliminary experiments on distributed sparse linear discriminant analysis, showing both the feasibility and effectiveness of our approach on experiments using heart disease data collected across four hospitals.

1 Introduction

Many applications would benefit from being able to learn models across sensitive datasets owned by different organizations. For example, multi-party data sets such as medical and financial records are increasingly being digitized, stored, and managed by independent hospitals and companies. Previous work in cryptography [11, 24] and privacy-preserving machine learning [28, 4, 5, 12, 13, 16] has developed several different notions of data privacy (most notably differential privacy [7, 6]). Most previous studies [28, 4, 5, 12, 13, 29], however, focus on privacy protection for single-party data sets, where a single operator is entrusted with full access to all of the data and the only concern is limiting how much information is leaked about individual records in the released model or query responses.

The goal of our work is to develop a distributed privacy-preserving machine learning method where multiple parties holding sensitive data can collaboratively learn a model across all of their data sets while minimizing data exposure and information leakage. We are particularly interested in sparse learning in the high dimensional regime, because distributed sparse learning is challenging and under-developed, and has many important potential applications where privacy is critical. The main challenge to developing effective distributed spare learning methods is to develop debiasing methods that allow an accurate joint model to be produced by aggregating the local models.

In our distributed private learning setting, each party produces a local model privately and the local models are aggregated using secure multi-party computation (MPC) to produce the global model without revealing any of the private models. MPC enables two or more parties to jointly perform a computation on their encrypted data and obtain the result, without revealing any information about private inputs or intermediate results.

An instantiation of our method is illustrated in Figure 1. Each individual data owner (P_i in the figure) independently produces a model using its own data set. We assume a threat model where there are two parties (S_1 and S_2) that are trusted not to collude with each other, but otherwise untrusted. The local models are secret-shared between S_1 and S_2. Then, S_1 and S_2 will execute a secure multi-party computation protocol that first obliviously combines the secret shares to obtain the individual models and then aggregates them to produce the global model. This entire process is done while the data is encrypted, so no information about the individual models is leaked to the parties executing MPC.

At the end of the process, the aggregated model could be published (assuming it is determined that the global model does not leak sensitive information about the individual inputs). Alternately, the model could be kept encrypted, and used as part of a MPC query engine that would provide results

Workshop on Private Multi-Party Machine Learning (NIPS 2016), Barcelona, Spain.

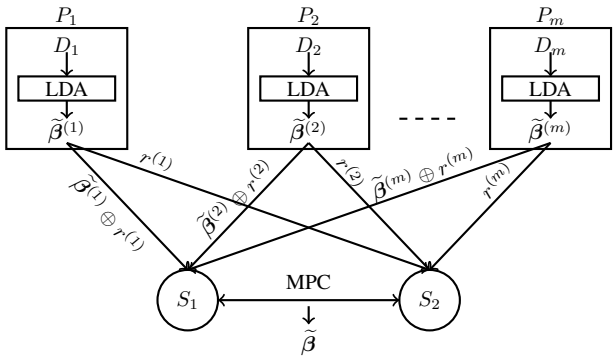

Figure 1: Secure Model Aggregation using Multi-Party Computation

using the model. A third option would be to add noise to the model (for example, to ensure a required level of differential privacy) within the MPC before releasing it. For any of these options, there are substantial advantages to performing the aggregation as a secure computation.

Related Work. Several prior works have proposed using MPC to enable privacy-preserving machine learning by combining the distributed data in encrypted form and performing the full learning process as a secure computation [19, 32, 34]. This approach provides strong privacy and can produce an identical model to the one that would be produced by combining the data sets insecurely, but is very expensive for large datasets and infeasible for complex learning algorithms.

Pathak et al. [23] proposed aggregating the locally-trained models using MPC and then revealing the aggregated model after adding statistical noise. Shokri and Shmatikov [27] proposed iteratively updating a global model by revealing differentially-private local parameter updates. These approaches are more efficient as the local models are computed in ad-hoc and asynchronous way. However, their learned model is less accurate due to the noise that must be added before the models are aggregated. For classification problems, Hamm et al. [9] proposed training a global differentially private classifier from local classifiers with the help of auxiliary unlabeled data. All the above methods are limited to the classical regime, and it is unclear how to extend them to the high dimensional setting.

Compared to existing distributed machine learning [18, 30] and privacy-preserving machine learning [23, 27] methods, we keep both local data and models private and only reveal the global model. In addition, our approach does not need to add noise to the local models since those models are never revealed. In cases where the aggregate model can be released or is used as an encrypted model, this produces a more accurate model than could be achieved if noise is added to the local models before they are aggregated. In cases where it is still necessary to add noise to the aggregated model before it is released, the amount of noise needed is less than what would be needed to protect each local model independently, so the resulting model should be substantially more accurate.

Contributions. The main contribution of this paper is introducing a model for privacy-preserving sparse distributed machine learning (Section 2) in which local models are produced using debiasing methods to enable an accurate aggregate model, and secure multi-party computation is used to perform the aggregation privately. Section 3 describes our implementation for privacy-preserving distributed sparse LDA, and Section 4 reports on results from some preliminary experiments.

2 Privacy-Preserving Distributed Machine Learning

The original goal of distributed machine learning was to scale machine learning by distributing data and computation over multiple computers (but assuming all data is owned by the same organization). A commonly used approach in distributed machine learning is averaging: [22, 40, 39, 1]: each "worker" machine generates a local estimator and sends it to the "master" machine, which averages all the local estimators to form an aggregated estimator. This approach can work for low-dimensional models, but does not produce accurate results for high-dimensional models.

In the high dimensional regime, where the number of features is larger than the number of observations, distributed learning has been done by making structural assumptions on the model parameters, and pursuing regularized estimation methods [31, 26, 3, 21, 8]. For example, Lee et al. [18] and Battey et

al. [2] both proposed distributed Lasso regression [31] methods, which exploit the debiased estimators proposed in [15, 33].

Sparse Linear Discriminant Analysis (sparse LDA) is a widely used classification technique that extends LDA to work on high-dimensional data sets [26, 3, 21, 8], where the number of variables used for classification is much smaller than the training sample size. In recent work [30], we proposed a distributed classification method based on sparse LDA [3], and this paper focuses on adapting that method to work in a setting where the individual data sets and local models must be kept private.

We consider the following high dimensional classification problem: Let $\mathbf{X}$ and $\mathbf{Y}$ be two d-dimensional random vectors following normal distributions, $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}^*)$ and $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}^*)$, respectively. For a new observation $\boldsymbol{Z}$ drawn with equal prior probability from the two normal distributions, the Fisher's linear discriminant rule takes the form

$$\psi(\boldsymbol{Z}) = \mathbb{1}((\boldsymbol{Z} - \boldsymbol{\mu})^\top \boldsymbol{\Theta}^* \boldsymbol{\mu}_d > 0), \tag{1}$$

where $\mathbb{1}(\cdot)$ is the indicator function, $\boldsymbol{\Theta}^* = \boldsymbol{\Sigma}^{*-1}$, also known as the precision matrix, and $\boldsymbol{\mu} = (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)/2$, $\boldsymbol{\mu}_d = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$. From (1), note that the classification rule only depends on the product $\boldsymbol{\Theta}^* \boldsymbol{\mu}_d$, rather than $\boldsymbol{\Theta}^*$ and $\boldsymbol{\mu}_d$. In the high-dimension regime, the estimator $\widehat{\boldsymbol{\Sigma}}$ is often singular and not reliable. To overcome this problem, it is natural to add some structural assumptions on the parameters. For example, Cai and Liu [3] made the assumption that the product $\boldsymbol{\Theta}^* \boldsymbol{\mu}_d$, denoted by $\boldsymbol{\beta}^*$, is sparse, and proposed a direct way to estimate $\boldsymbol{\beta}^*$ as follows

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_1 \quad \text{subject to} \quad \|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta} - \widehat{\boldsymbol{\mu}}_d\|_\infty \leq \lambda, \tag{2}$$

where λ is a tuning parameter. (2) can be solved by linear programming. Given $\widehat{\boldsymbol{\beta}}$, we can classify a new observation $\boldsymbol{Z}$ using the learned discriminant rule $\widehat{\psi}(\boldsymbol{Z}) = \mathbb{1}((\boldsymbol{Z} - \widehat{\boldsymbol{\mu}})^\top \widehat{\boldsymbol{\beta}} > 0)$.

When data are distributed over m owners, each owner will produce a local estimator $\widehat{\boldsymbol{\beta}}^{(l)}$ using Equation (2) based on local data, where l is the index of the owner. Due to the ℓ_1-norm penalty in Equation (2), the resulting estimators $\widehat{\boldsymbol{\beta}}^{(l)}$'s are biased. Since averaging only reduces variances, not the bias, if we directly average all $\widehat{\boldsymbol{\beta}}^{(l)}$'s, the performance of averaged estimator is no better than the local estimator due to the aggregation of bias [25]. To address this problem, we proposed a debiased estimator for LDA [30], which takes the form,

$$\widetilde{\boldsymbol{\beta}}^{(l)} = \widehat{\boldsymbol{\beta}}^{(l)} - \widehat{\boldsymbol{\Theta}}^{(l)\top}(\widehat{\boldsymbol{\Sigma}}^{(l)}\widehat{\boldsymbol{\beta}}^{(l)} - \widehat{\boldsymbol{\mu}}_d^{(l)}),$$

where $\widehat{\boldsymbol{\Theta}}^{(l)}$, $\widehat{\boldsymbol{\Sigma}}^{(l)}$ and $\widehat{\boldsymbol{\mu}}_d^{(l)}$ are estimations of $\boldsymbol{\Theta}^*$, $\boldsymbol{\Sigma}^*$ and $\boldsymbol{\mu}_d$ based on the data owned by the l-th party. In Tian and Gu [30], we prove that, under certain conditions, if we average the debiased estimators obtained by different parties and sparsify the averaged estimator, the final sparse estimator will attain the same statistical rate as the centralized estimator.

Revealing the $\widetilde{\boldsymbol{\beta}}^{(l)}$'s to support traditional distributed learning aggregation would leak substantial information about the local data. Instead, we use secure multi-party computation to aggregate the debiased local estimators while keeping them private.

3 Implementation

Our implementation is built using Obliv-C [37], which provides a high-level language for implementing secure multi-party protocols. For the protocol, we use Obliv-C's implementation of Yao's Garbled Circuit (GC) protocol [35, 36, 20], which incorporates recent improvements in garbled circuit execution [10, 17, 24, 38, 11].

Our implementation is built using a generic protocol that can compute any function securely. Although there could be more efficient custom protocols for particular computations used for these experiments to aggregate LDA models, we prefer to use a generic protocol because it has an established security proof and supporting tools, and because we intend to incorporate additional mechanisms (such as adding privacy-preserving noise within the MPC to the final model) in future work.

To facilitate efficient MPC in our approach, each party P_l first scales its local model parameter $\widetilde{\boldsymbol{\beta}}^{(l)}$ from $\mathbb{R}^d$ to $\mathbb{Z}^d$ by multiplying with a scale factor (10^8) and truncating the remaining fractional part. P_l then generates a d-dimensional random vector $r^{(l)} \in \mathbb{Z}^d$ and creates two shares of $\widetilde{\boldsymbol{\beta}}^{(l)}$ as

Dataset	m	Misclassification Rate				Number of gates for MPC
		Centralized LDA	Naive Averaged	Distributed LDA	Our Approach	
Synthetic	20	0.168 ± 0.002	0.240 ± 0.003	0.182 ± 0.003	0.182 ± 0.003	1,056,800
Synthetic	40	0.167 ± 0.001	0.239 ± 0.002	0.180 ± 0.002	0.180 ± 0.002	1,295,600
Synthetic	60	0.166 ± 0.001	0.240 ± 0.002	0.179 ± 0.002	0.179 ± 0.002	1,559,800
Synthetic	80	0.166 ± 0.001	0.240 ± 0.002	0.179 ± 0.001	0.179 ± 0.001	1,786,400
Synthetic	100	0.165 ± 0.001	0.240 ± 0.002	0.179 ± 0.001	0.179 ± 0.001	2,062,600
Real	4	0.208 ± 0.012	0.329 ± 0.035	0.220 ± 0.017	0.220 ± 0.017	94,200

Table 1: Experimental Results (20 repetitions on Synthetic data and 10 repetitions on Real data)

$\widetilde{\boldsymbol{\beta}}^{(l)} \oplus r^{(l)}$ and $r^{(l)}$ (refer to Figure 1). One share is sent to S_1 and other share is sent to S_2. In this way, S_1 and S_2 obtain the shares of all m parties. Next, S_1 and S_2 execute the MPC protocol to compute the aggregate model of the m parameters.

After the model aggregation, we obtain an averaged estimator $\widetilde{\boldsymbol{\beta}} = (\sum_{l=1}^{m} \widetilde{\boldsymbol{\beta}}^{(l)})/m$, which is not sparse. In order to get a sparse estimator, we truncate the entries of $\widetilde{\boldsymbol{\beta}}$ with small absolute values to zero [30]. The resulting sparse $\widetilde{\boldsymbol{\beta}}$ is broadcasted to all the m parties and they can then locally rescale $\widetilde{\boldsymbol{\beta}}$ from $\mathbb{Z}^d$ back to $\mathbb{R}^d$ by dividing the same scale factor used before. The scaling and rescaling of model parameters may affect the accuracy, and hence it is discussed in the next section.

4 Experiments

We conducted experiments using our method to perform privacy-preserving sparse distributed learning on both synthetic and real data sets.

Synthetic Data Experiments. In the synthetic data experiments, we generate data setting the covariance matrix $\boldsymbol{\Sigma}^*$ as a 200×200 matrix where $\Sigma^*_{ij} = 0.8^{|i-j|}$. We set $\boldsymbol{\mu}_1 = \mathbf{0}$ and $\boldsymbol{\mu}_2 = (1, 1, \ldots, 1, 0, 0, \ldots, 0)^\top$ where the first 10 entries are ones. This means $\boldsymbol{\beta}^*$ is a sparse vector with 11 nonzero entries. We set the sample size in every machine fixed as 200, where half of the data are drawn from $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}^*)$ and the rest are from $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}^*)$. We vary the number of data owners from 20 to 100 and report the average misclassification rate.

Heart Disease Data. To simulate a realistic use of our approach, we performed experiments using a dataset of 920 heart disease patients collected across four hospitals [14]. There are 13 attributes associated with each patient, including age, gender, and laboratory test results. We extend the categorical attributes into dummy variables in preprocessing. Every patient is labeled whether he or she is diagnosed with heart disease. In our experiment, every hospital is treated as an independent data owner (even though in this case the full data set was combined). We randomly choose half of the dataset as the training set and regard the remaining half as the test set. Then an LDA model is trained based on the training set to predict the label of each patient in the test set. We evaluate the misclassification rate of the learned model on the test dataset.

Observations. Table 1 summarizes the results. We observed no loss in accuracy due to scaling and rescaling process. The debiasing step greatly decreases the misclassification rate compared with the naïvely averaged estimator. Also the proposed approach achieves comparable misclassification rate with the centralized LDA estimator. The time to perform the MPC aggregation is under two seconds (our framework executes well over 4M gates/second), and scales linearly with the number of local models to aggregate, so these results would hold for larger datasets.

5 Conclusions

Our proposed framework employs debiased estimators to enable private sparse distributed learning, and uses secure multi-party computation to aggregate the models to provide privacy. We empirically verified that the approach is promising for sparse LDA; our approach can also be applied to high dimensional regression [31, 18, 2]. Several steps remain before these methods can be widely adopted including rigorously analyzing the accuracy loss for different scenarios, and understanding how much information is leaked by the aggregated model and how to incorporate noise to eliminate this leakage.

References

[1] Maria-Florina Balcan, Avrim Blum, Shai Fine, and Yishay Mansour. Distributed Learning, Communication Complexity and Privacy. *arXiv preprint arXiv:1204.3514*, 2012.

[2] Heather Battey, Jianqing Fan, Han Liu, Junwei Lu, and Ziwei Zhu. Distributed Estimation and Inference with Statistical Guarantees. *arXiv preprint arXiv:1509.05457*, 2015.

[3] Tony Cai and Weidong Liu. A Direct Estimation Approach to Sparse Linear Discriminant Analysis. *Journal of the American Statistical Association*, 106(496), 2011.

[4] Kamalika Chaudhuri and Claire Monteleoni. Privacy-Preserving Logistic Regression. In *Advances in Neural Information Processing Systems*, pages 289–296, 2009.

[5] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially Private Empirical Risk Minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

[6] Cynthia Dwork. Differential Privacy: A Survey of Results. In *International Conference on Theory and Applications of Models of Computation*, 2008.

[7] Cynthia Dwork and Kobbi Nissim. Privacy-Preserving Datamining on Vertically Partitioned Databases. In *Annual International Cryptology Conference*, pages 528–544. Springer, 2004.

[8] Jianqing Fan, Yang Feng, and Xin Tong. A Road to Classification in High Dimensional Space: the Regularized Optimal Affine Discriminant. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(4):745–771, 2012.

[9] Jihun Hamm, Paul Cao, and Mikhail Belkin. Learning Privately from Multiparty Data. *arXiv preprint arXiv:1602.03552*, 2016.

[10] Yan Huang, David Evans, Jonathan Katz, and Lior Malka. Faster Secure Two-Party Computation Using Garbled Circuits. In *USENIX Security Symposium*, 2011.

[11] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending Oblivious Transfers Efficiently. In *Annual International Cryptology Conference*, 2003.

[12] Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially Private Online Learning. *arXiv preprint arXiv:1109.0105*, 2011.

[13] Prateek Jain and Abhradeep Thakurta. Differentially Private Learning with Kernels. *International Conference on Machine Learning*, 28:118–126, 2013.

[14] Andras Janosi, William Steinbrunn, Matthias Pfisterer, and Robert Detrano. Heart Disease Data Set. UCI Machine Learning Repository (`https://archive.ics.uci.edu/ml/datasets/Heart+Disease`), 1988.

[15] Adel Javanmard and Andrea Montanari. Confidence Intervals and Hypothesis Testing for High-Dimensional Regression. *The Journal of Machine Learning Research*, 15(1):2869–2909, 2014.

[16] Daniel Kifer, Adam Smith, and Abhradeep Thakurta. Private Convex Empirical Risk Minimization and High-Dimensional Regression. *Journal of Machine Learning Research*, 1:41, 2012.

[17] Vladimir Kolesnikov and Thomas Schneider. Improved Garbled Circuit: Free XOR Gates and Applications. In *International Colloquium on Automata, Languages, and Programming*, 2008.

[18] Jason D Lee, Yuekai Sun, Qiang Liu, and Jonathan E Taylor. Communication-Efficient Sparse Regression: a One-Shot Approach. *arXiv preprint arXiv:1503.04337*, 2015.

[19] Yehuda Lindell and Benny Pinkas. Privacy Preserving Data Mining. In *Advances in Cryptology—-CRYPTO*, pages 36–54. Springer, 2000.

[20] Yehuda Lindell and Benny Pinkas. Secure Multiparty Computation for Privacy-Preserving Data Mining. *Journal of Privacy and Confidentiality*, 1(1):5, 2009.

[21] Qing Mai, Hui Zou, and Ming Yuan. A Direct Approach to Sparse Discriminant Analysis in Ultra-High Dimensions. *Biometrika*, page asr066, 2012.

[22] Ryan Mcdonald, Mehryar Mohri, Nathan Silberman, Dan Walker, and Gideon S Mann. Efficient Large-Scale Distributed Training of Conditional Maximum Entropy Models. In *NIPS*, 2009.

[23] Manas Pathak, Shantanu Rane, and Bhiksha Raj. Multiparty Differential Privacy via Aggregation of Locally Trained Classifiers. In *NIPS*, 2010.

[24] Benny Pinkas, Thomas Schneider, Nigel P Smart, and Stephen C Williams. Secure Two-Party Computation Is Practical. In *International Conference on the Theory and Application of Cryptology and Information Security*, 2009.

[25] Jonathan Rosenblatt and Boaz Nadler. On the Optimality of Averaging in Distributed Statistical Learning. *arXiv preprint arXiv:1407.2724*, 2014.

[26] Jun Shao, Yazhen Wang, Xinwei Deng, Sijian Wang, et al. Sparse Linear Discriminant Analysis by Thresholding for High Dimensional Data. *The Annals of Statistics*, 39(2):1241–1265, 2011.

[27] Reza Shokri and Vitaly Shmatikov. Privacy-Preserving Deep Learning. In *ACM Conference on Computer and Communications Security*, 2015.

[28] Adam Smith and Abhradeep Thakurta. Differentially Private Feature Selection via Stability Arguments, and the Robustness of the Lasso. In *Proceedings of Conference on Learning Theory*, 2013.

[29] Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Nearly Optimal Private Lasso. In *NIPS*, 2015.

[30] Lu Tian and Quanquan Gu. Communication-efficient distributed sparse linear discriminant analysis. *arXiv preprint arXiv:1610.04798*, 2016.

[31] Robert Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[32] Jaideep Vaidya, Murat Kantarcıoğlu, and Chris Clifton. Privacy-Preserving Naive Bayes Classification. *The VLDB Journal*, 17(4), 2008.

[33] Sara Van de Geer, Peter Bühlmann, Ya'acov Ritov, Ruben Dezeure, et al. On Asymptotically Optimal Confidence Regions and Tests for High-Dimensional Models. *The Annals of Statistics*, 42(3):1166–1202, 2014.

[34] Zhiqiang Yang, Sheng Zhong, and Rebecca N Wright. Privacy-Preserving Classification of Customer Data without Loss of Accuracy. In *SIAM International Conference on Data Mining*, 2005.

[35] Andrew C Yao. Protocols for Secure Computations. In *Symposium on Foundations of Computer Science*, 1982.

[36] Andrew C Yao. How to Generate and Exchange Secrets. In *Symposium on Foundations of Computer Science*, 1986.

[37] Samee Zahur and David Evans. Obliv-C: A Language for Extensible Data-Oblivious Computation. Cryptology ePrint Archive, Report 2015/1153, 2015. `http://eprint.iacr.org/2015/1153`.

[38] Samee Zahur, Mike Rosulek, and David Evans. Two Halves Make a Whole: Reducing Data Transfer in Garbled Circuits using Half Gates. In *EuroCRYPT*, 2015.

[39] Yuchen Zhang, John C Duchi, and Martin J Wainwright. Divide and Conquer Kernel Ridge Regression: A Distributed Algorithm with Minimax Optimal Rates. *arXiv preprint arXiv:1305.5029*, 2013.

[40] Yuchen Zhang, Martin J Wainwright, and John C Duchi. Communication-Efficient Algorithms for Statistical Optimization. In *NIPS*, 2012.