DATA-WEIGHTED ENSEMBLE LEARNING FOR PRIVACY-PRESERVING DISTRIBUTED LEARNING

Liyang Xie[⋆] *Sergey Plis*[†] *Anand D. Sarwate*[⋆]

[⋆] Rutgers University, Piscataway, NJ USA
[†] The Mind Research Network, Albuquerque NM USA

ABSTRACT

In collaborative medical research settings, a moderate number of groups (sites) may wish to merge local analyses of private subject data. Differential privacy offers one way to guarantee privacy for these local analyses. We describe a novel ensemble learning method that we call the "feature method" for aggregating binary classifiers or regressors trained on local data. Our method leverages a public data set available at the aggregator to optimize a linear combination of local predictors. We provide some analysis of the method and show how it is effective when the local sites are required to learn classifiers that are differentially private. We prove that this method has near-optimal performance when local data sets are large enough under certain requirements on the parameters. Experimentally, we give a comparison of the feature method and the standard approach of averaging the local classifiers.

Index Terms— distributed learning, classification, differential privacy, empirical risk minimization, convex optimization

1. INTRODUCTION

This work is motivated by learning problems where private or sensitive data is distributed across different sites. The sites wish to collaborate to collectively learn something from their data – leveraging a larger sample size – but must respect the privacy constraints governing their data. Such setting arises in healthcare and medical research systems: with the help of machine learning tools, these systems can be made more efficient and accurate. Likewise, in a research setting, researchers would like to take advantage of multiple data sets to enable better predictions. The extensive existing work on distributed learning either ignores privacy constraints [1–8] or assumes an asymptotic scenario where each site has a large number of samples [9–11]. This is rarely true in medical research settings, so we must be careful about how we aggregate the information from local sites. Our method is a form of *ensemble learning* [12–14] in which the goal is to construct a new classifier by training several classifiers (an ensemble) and combining them. Methods such as bagging [15], boosting [16], and many others have been proposed – our method is a form of ensemble weighting [14] based on treating classifiers learned from local data as new features.

In this paper we examine in more detail an aggregation mechanism for aggregating differentially private classifiers studied in Sarwate et al. [17]. In our approach, each local site trains its own linear classifier and transmits it to an aggregation site. A standard approach for aggregation with a large amount of data is to average the resulting weight vectors from the classifiers [4, 5, 8]. However, if the aggregation site has its own data, it can use this data to weigh the classifiers from the local sites, thereby achieving an improvement in accuracy [17]. The aggregation site does this by treating the local classifiers as features, projecting its data onto these features, and training a lower-dimensional classifier in this new feature space. As we show, this is equivalent to taking a weighted average of the local classifiers – we call this the *feature method*.

We also study the scenario in which the local classifiers are trained using a differentially private training algorithm [18]. Differential privacy provides a way to quantify the privacy risk incurred by running an algorithm on sensitive data [19]: it measures the risk of an individual data point being identified from the output of the algorithm. Differential privacy basically guarantees that an analyst observing the output does not learn too much about any individual's membership in the database. Algorithms that guarantee differential privacy are randomized by introducing noise or randomness to the computation – this noise masks the contribution of individual data points [20, 21].

Experimental validations show that feature method has a significantly better performance than the traditional average method. More importantly, this method has higher empirical stability when the local classifiers are trained using differential privacy.

2. PROBLEM MODEL AND DEFINITIONS

We consider a distributed system where there are $N \geq 2$ *local sites* with data sets $D_i = \{(x_{i,j}, y_{i,j}) : j = 1, 2, \ldots, m_i\}$

The work of the authors was supported by the NSF under award CCF-1453432 and by the NIH under award 1R01DA040487-01A1.

consisting of m_i pairs of feature vectors $x_{i,j} \in \mathbb{R}^d$ and binary labels $y_{i,j} \in \{-1, +1\}$. We focus on problem of *linear classification*: the goal is to find a vector $f \in \mathbb{R}^d$ such that $\mathrm{sgn}(f^\top x)$ is a good predictor of the label y. If we model the data as being drawn independently and identically distributed from a fixed but unknown distribution $\mathcal{P}(x, y)$, a good training algorithm for finding such an f is the regularized empirical risk minimization (ERM), which outputs the minimizer of the following objective function.

$$J(f, D_i) = \frac{1}{m_i} \sum_{j=1}^{m_i} \ell(f^\top x_j, y_j) + \Lambda R_i(f), \qquad (1)$$

where $\ell(\cdot, \cdot)$ is a loss function (e.g. hinge loss for support vector machines) and $R_i(f)$ is a regularization term. In section 4 we will focus on the case where ℓ is the hinge loss and logistic loss. We also set $R_i(f) = \|f\|_2^2$.

We are also interested in the case where each site trains a local classifier under *differential privacy* [19], a privacy framework that has received significant research attention in recent years (see a recent monograph [20] and survey [21]). Differentially private algorithms are randomized in order to prevent someone observing the output from identifying individual data points from the training set D. A classification algorithm Alg guarantees ϵ differential privacy if for all D and D' differing in a single point:

$$\mathbb{P}\left(\mathsf{Alg}(D) \in \mathcal{F}\right) \leq e^{\epsilon} \mathbb{P}\left(\mathsf{Alg}(D') \in \mathcal{F}\right) \qquad (2)$$

for any measurable set $\mathcal{F}$. We consider local classification rules trained using the objective perturbation method [18], which is a differentially private version of ERM.

In the algorithms we consider, each local site i uses either the non-private ERM in (1) or its differentially private version [18] to train a local classifier f_i based on its data set D_i. It then transmits these classifiers to an *aggregation site*. In our model we further assume the aggregation site has an ancillary data set D_0 of m_0 pairs $(x_{0,j}, y_j)$. Our goal is to take advantage of this extra data to design a better algorithm for aggregating $\{f_i\}$. For simplicity, we assume that D_0 is a public data set so that algorithms on that data set need not guarantee differential privacy.

3. AGGREGATION METHODS

We are interested in the problem of ensemble learning, or classifier aggregation. In applications such as neuroimaging for mental health, each local site's data set D_i often has few data points due to the expense of measuring and the rareness of the condition. A classifier f_i trained on local data using ERM (private or non-private) may have high classification error, and the variance across sites may be large. A large-scale machine learning approach [1,4,5,8] is to simply average the N classifiers trained at the local sites:

$$\bar{f} = \frac{1}{N} \sum_{i=1}^{N} f_i. \qquad (3)$$

We call this the *average method*. This method does reduce the variance across the $\{f_i\}$ but may still lead to poor performance when the local classifiers are themselves poor. Boosting [16] may yield better performance but is more effective when the number of sites N is large, which is not the case in our motivating application. In addition, neither of these methods takes advantage of the data D_0 available at the aggregation site.

Algorithm 1 Feature method for classifier aggregation

1: **Inputs:** Data sets $\{D_i\}$, local training algorithms Alg_i, $i = 1, \cdots, N$, aggregation set D_0.
2: **for** i=1,2,..., N **do**
3: Compute $f_i = \mathsf{Alg}_i(D_i)$ and send to aggregation site.
4: **end for**
5: Form matrix M_f whose ith row is classifier $f_i^\top$.
6: Form transformed data set:

$$\tilde{D}_0 = \{(M_f x_{0,j}, y_{0,j}) : j = 1, 2, \ldots, m_0\}.$$

7: Train $\omega_{\mathsf{Feat}} = \mathrm{argmin}_\omega \, J(\omega, \tilde{D}_0)$.
8: **Output:** Classifier $f_{\mathsf{Feat}} = M_f^\top \omega_{\mathsf{Feat}}$.

The *feature method* in Algorithm 1 does take advantage of the local data. It takes each local classifier f_i and computes a new feature $f_i^\top x_{0,j}$ and then trains classifier based only on the locally learned (one-dimensional) features from the local sites. Although this method throws away a lot of information, an empirical study [17] showed that it was effective at merging classifiers from structural MRIs used to classify schizophrenia patient from healthy controls. From the algorithm, we can see that the aggregation site is computing a *weighted linear combination* of local classifiers with the weights in the coefficient vector ω_{Feat}. The feature method lets the data determine the importance of each local classifier rather than the equal weights used in the average method.

As we can see, both the $\bar{f}$ and f_{Feat} lie in the subspace $\mathrm{span}\{f_i\}$ spanned by local classifiers. We define the optimum linear combination as:

$$f^*_{\mathrm{span}} = \operatorname*{argmin}_{f \in \mathrm{span}\{f_i\}} L(f), \qquad (4)$$

where $L(f) = \mathbb{E}_{(x,y)\sim\mathcal{P}}[\ell(f^T x, y)]$ is the expected loss under a classifier with inputs (x, y) drawn according to $\mathcal{P}(x, y)$. We define the corresponding coefficient vector ω^* by $f^*_{\mathrm{span}} = M_f^\top \omega^*$. We would like to characterize how close f_{Feat} is to f^*_{span}. The global minimizer $f^* = \mathrm{argmin}_f \, L(f)$ does not necessarily belong to $\mathrm{span}\{f_i\}$.

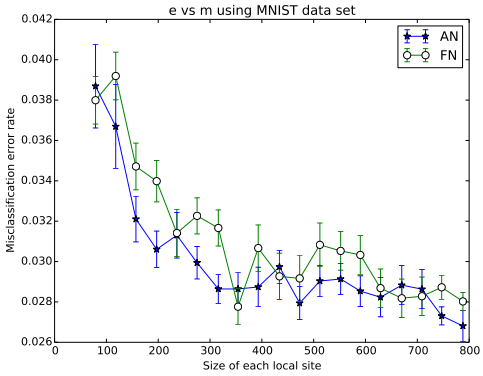


Fig. 1. e vs. m, non-private for MNIST

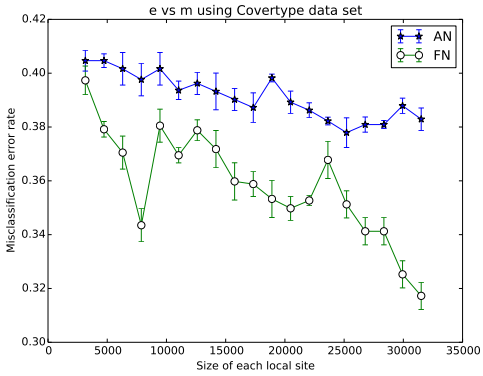


Fig. 2. e vs. m, non-private for Covertype

We make following assumptions: (1) the loss $\ell(\cdot,\cdot)$ is convex with respect to (w.r.t) the first parameter, (2) the regularizers $R_i(\cdot)$ is λ-strongly convex and μ-smooth, (3) the expected gradient of the loss is bounded as $\mathbb{E}\left[\|\nabla l(\cdot,\cdot)\|^2\right] \leq C$, and (4) the regularization function at the aggregation site is bounded at the optimum as $\mathbb{E}\left[\|\nabla R_0(\omega^*)\|^2\right] \leq C'$.

We are interested in the impact of requiring the local classifiers to be learned under differential privacy [18]. We compare two cases: a "baseline" case with all non-private data, and a "public-private" case with private data at the sources.

Theorem 1. (Feature method in two cases) *We have the following upper bound on the expectation (over the data distribution) of the error between f_{Feat} and f^*_{span} for local classifiers learned without differential privacy:*

$$\mathbb{E}\left[\|f_{\mathsf{Feat}} - f^*\|^2\right] = \mathcal{O}\left(\frac{N}{mm_0}\right) + \mathcal{O}\left(\frac{N}{m}\right) + \mathcal{O}\left(\frac{N}{m_0}\right) + C_N. \quad (5)$$

For local classifiers learned under ϵ-differential privacy, in expectation over the data and the privacy mechanism,

$$\mathbb{E}\left[\|f_{\mathsf{Feat}} - f^*\|^2\right] = \mathcal{O}\left(\frac{N}{m^2\epsilon^2}\right) + \mathcal{O}\left(\frac{N}{m_0m^2\epsilon^2}\right) + \mathcal{O}\left(\frac{N}{mm_0}\right) + \mathcal{O}\left(\frac{N}{m}\right) + \mathcal{O}\left(\frac{N}{m_0}\right) + 2C_N. \quad (6)$$

where C_N is a constant that depends on the number of local sites N.

The Theorem shows the effect of enforcing ϵ-differential privacy on local sites in terms of the excess error. Because N is finite in our model, there is an unavoidable gap C_N that depends on N but not m, m_0 and ϵ; the dependence on those parameters is captured in the order-terms. We can generalize these bounds when each site has its own local ϵ value.

4. EXPERIMENTS

We performed extensive experiments to see how our method performed against average-at-the-end methods for learning under differential privacy. We primarily focus on the "Public-Private" setting in which the aggregator's data is public but the local data sites are private. We performed experiments on MNIST [22] and Covertype [23] data sets.

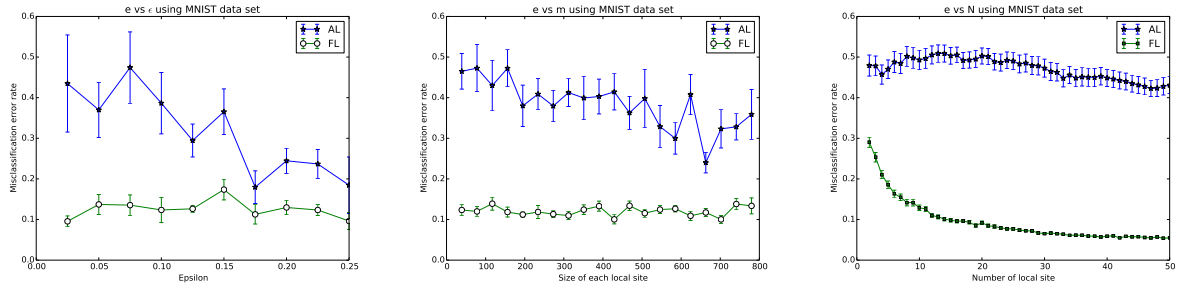
We use misclassification error rate to measure performance in our experiments. For any linear classifier f and a data point (x_j, y_j), we define misclassification error e_j as:

$$\mathrm{e}_j = \begin{cases} 1 & \text{if} \quad y_j f^T x_j < 0 \\ 0 & \text{if} \quad y_j f^T x_j \geq 0. \end{cases}$$

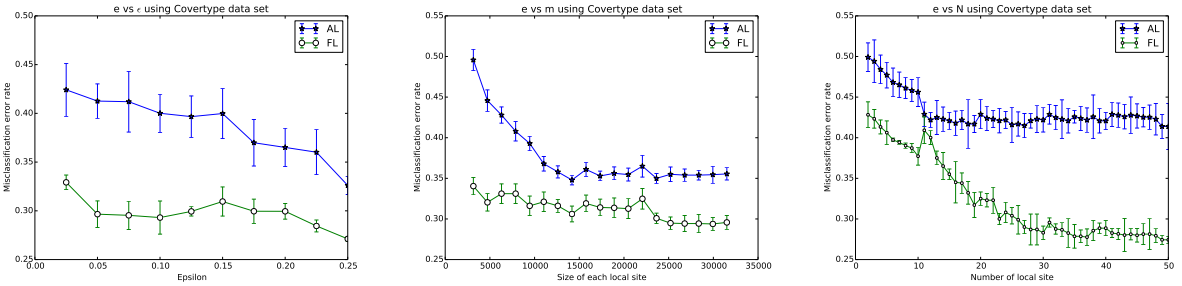
The error rate is the average of e_j over the test set.

From now on, we denote by $\mathrm{e}(\epsilon, N, m, m_0, h, \Lambda)$ the error rate and associated parameters. We use 'FL', 'AL' as error rates of the feature method and the average method with privacy only at the local sites in legend, respectively. With same meaning, 'AN', 'FN' are the corresponding error rates for non-private case. In our experiments we use objective perturbation [18] with Huber loss [24] at all local sites, where h is the Huber constant. We use regularized ERM with logistic loss $l(z) = \log(1 + e^{-z})$ at the aggregation site. We use digits 3 and 7 in MNIST data set with 8,678 points in training set (all local sites + aggregation site) and 3,718 in testing set. We use cover types 1 and 2 in Covertype data set with 346,598 points in the training set and 148,543 in the testing set. We reduce the dimension of data in MNIST data set from 784 to 50 using PCA. Each experiment was repeated 10 times for fixed parameters and plots are shown with error bars.

As a baseline, we evaluated the performances of the feature method and average method using non private classification at the local sites. The results are shown in Figure 1 and 2 . While the feature method outperforms the average method on Covertype, the algorithms are nearly equivalent on MNIST. The story becomes significantly more interesting when we insist on privacy at the local sites.



(a) e vs. ϵ from 0.025 to 0.25 (b) e vs. m from 39 to 789 (c) e vs. N from 2 to 50



(d) e vs. ϵ from 0.025 to 0.25 (e) e vs. m from 3151 to 31509 (f) e vs. N from 2 to 50

Fig. 3. Performance of algorithms for MNIST (top row, $\Lambda = 10^{-2}$) and Covertype (bottom row, $\Lambda = 10^{-7}$) for $N = 10$, $h = 0.5$. Error e versus $\epsilon \in [0.025, 0.25]$ (3a,3d) , m (3b from 39 to 789,3e from 3151 to 31509), and N (3c,3f)

.

Our goal of first experiment is to test the performance of the two methods w.r.t m. We compare the two methods by plotting curves of error rate in Figure 1 and 2. Introducing noise for differential privacy can significantly increase the error of the overall classifier, as suggested by our theoretical result. Here, however, the feature method is significantly better as m increases. Figures 3a and 3d show how performance is affected by the privacy parameter ϵ. From these two figures we can see that the feature method performs much better than average method and less affected by the privacy-preserving noise. Since the average method does not use auxiliary information in the data set, it has significantly poorer performance. Under the "Public-Private" condition, for both methods, increasing ϵ makes the final performance better and the feature method works better than the average method.

Thirdly, we want to know how m affects the performance of the average method and the feature method. We compare two methods in Figure 3b and 3e. We have a similar stability for feature method shown as that in the last experiment. Unsurprisingly, increasing m improves both methods, but the feature method has more consistent performance.

Finally, we compare the feature method with the average method in Figure 3c and 3f as a function of N. Our analysis does not indicate how performance should scale with N. For sufficiently large $N \geq d$ the span of the local classifiers will in general be $\mathbb{R}^d$ so the optimal linear combination would be the distribution-optimal classifier. However, for increasing $N < d$ under a fixed amount of data, the local classifiers will become worse (m is lower) and the feature method appears to be a significantly better way of combining the local classifiers than the average method. We may conclude that under proper choice of N, increasing N makes the final performance better and the feature method works better than the average method.

5. CONCLUSION AND FUTURE WORK

In this paper, we have analyzed a novel aggregation method to improve the performance of distributed classification system. The key idea is to take the advantages of public-accessible data site and use it to weight the local classifiers. We tested three important parameters in the system and the differentially private version of this method shows significantly better performance and stability property compared with averaging the local classifiers. In particular, when the local classifiers are learned in an ϵ-differentially private way, the average method has significantly worse variance since it cannot take advantage of the local data. The major difference between our method and other ensemble learning approaches [14] is that we have an auxiliary public data set. However, it would be interesting to see how the feature method can be improved using ensemble learning techniques. In particular, for larger N with a fixed total data set, boosting may yield significant empirical improvements.

6. REFERENCES

[1] Ohad Shamir, Nathan Srebro, and Tong Zhang, "Communication efficient distributed optimization using an approximate newton-type method," in *31st International Conference on Machine Learning (ICML 2014)*, 2014.

[2] Yuchen Zhang and Lin Xiao, "Communication-efficient distributed optimization of self-concordant empirical loss," *arXiv preprint arXiv:1501.00263*, 2015.

[3] Yuchen Zhang, Martin J Wainwright, and John C Duchi, "Communication-efficient algorithms for statistical optimization," in *Advances in Neural Information Processing Systems*, 2012, pp. 1502–1510.

[4] Ryan Mcdonald, Mehryar Mohri, Nathan Silberman, Dan Walker, and Gideon S Mann, "Efficient large-scale distributed training of conditional maximum entropy models," in *Advances in Neural Information Processing Systems*, 2009, pp. 1231–1239.

[5] Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J Smola, "Parallelized stochastic gradient descent," in *Advances in Neural Information Processing Systems*, 2010, pp. 2595–2603.

[6] 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.

[7] John C Duchi, Michael I Jordan, Martin J Wainwright, and Yuchen Zhang, "Information-theoretic lower bounds for distributed statistical estimation with communication constraints," *arXiv preprint arXiv:1405.0782*, 2014.

[8] Yuchen Zhang, John Duchi, Michael I Jordan, and Martin J Wainwright, "Information-theoretic lower bounds for distributed statistical estimation with communication constraints," in *Advances in Neural Information Processing Systems*, 2013, pp. 2328–2336.

[9] Zhenqi Huang, Sayan Mitra, and Nitin Vaidya, "Differentially private distributed optimization," *arXiv preprint arXiv:1401.2596*, 2014.

[10] Zhanglong Ji, Xiaoqian Jiang, Shuang Wang, Li Xiong, and Lucila Ohno-Machado, "Differentially private distributed logistic regression using private and public data," *BMC medical genomics*, vol. 7, no. Suppl 1, pp. S14, 2014.

[11] Shuo Han, Ufuk Topcu, and George J Pappas, "Differentially private distributed constrained optimization," *arXiv preprint arXiv:1411.4105*, 2014.

[12] Thomas G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems: First International Workshop, MCS 2000*, Lecture Notes in Computer Science. Springer, 2000.

[13] João Mendes-Moreira, Carlos Soares, Alípio Mário Jorge, and Jorge Freire De Sousa, "Ensemble approaches for regression: A survey," *ACM Computing Surveys*, vol. 45, no. 1, pp. Article 10, November 2012.

[14] Lior Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1, pp. 1–39, February 2010.

[15] Leo Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, August 1996.

[16] Robert E. Schapire and Yoav Freund, *Boosting: Foundations and Algorithms*, MIT Press, Cambridge, MA, USA, 2012.

[17] Anand D Sarwate, Sergey M Plis, Jessica A Turner, Mohammad R Arbabshirani, and Vince D Calhoun, "Sharing privacy-sensitive access to neuroimaging and genetics data: a review and preliminary validation," *Frontiers in neuroinformatics*, vol. 8, 2014.

[18] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate, "Differentially private empirical risk minimization," *The Journal of Machine Learning Research*, vol. 12, pp. 1069–1109, 2011.

[19] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, pp. 265–284. Springer, 2006.

[20] Cynthia Dwork and Aaron Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2013.

[21] Anand D. Sarwate and Kamalika Chaudhuri, "Signal processing and machine learning with differential privacy: theory, algorithms, and challenges," *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 86–94, September 2013.

[22] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[23] M. Lichman, "UCI machine learning repository," 2013.

[24] Olivier Chapelle, "Training a support vector machine in the primal," *Neural Computation*, vol. 19, no. 5, pp. 1155–1178, 2007.