Stochastic gradient descent with differentially private updates

Shuang Song
Dept. of Computer Science and Engineering
University of California, San Diego
La Jolla, CA USA
shs037@eng.ucsd.edu

Kamalika Chaudhuri
Dept. of Computer Science and Engineering
University of California, San Diego
La Jolla, CA USA
kamalika@eng.ucsd.edu

Anand D. Sarwate
Toyota Technological Institute
Chicago, IL USA
asarwate@ttic.edu

***Abstract*—Differential privacy is a recent framework for computation on sensitive data, which has shown considerable promise in the regime of large datasets. Stochastic gradient methods are a popular approach for learning in the data-rich regime because they are computationally tractable and scalable. In this paper, we derive differentially private versions of stochastic gradient descent, and test them empirically. Our results show that standard SGD experiences high variability due to differential privacy, but a moderate increase in the batch size can improve performance significantly.**

I. Introduction

As data becomes easier to acquire and aggregate in digitized formats, designing efficient algorithms that can operate on this data has emerged as a central challenge for signal processing, machine learning, and related fields. Often this data may be *private* or sensitive, such as medical or financial records. Differential privacy [1] is rapidly becoming a popular framework for designing algorithms that can guarantee a quantifiable level of privacy. An α-differentially private algorithm guarantees that the log-likelihood ratio of the outputs of the algorithm under two databases differing in a single individual's data is smaller than α. For small α an adversary's inferences about an individual will be similar regardless of whether that individual participates in the dataset or not.

Guaranteeing differential privacy involves *approximating* some desired algorithm or computation. The approximate nature impacts the performance, or *utility* of the resulting procedure. For example, in parameter estimation, guaranteeing privacy may increase the mean-squared error of the estimator. One way to interpret this effect is that the sample size required for a target utility level increases with the privacy constraint.

In the data-rich setting, at first blush it appears that learning algorithms can enjoy both low privacy risk and high utility. However, optimization methods for large data sets must also be scalable. Stochastic gradient descent (SGD) algorithms have received significant attention recently because they are simple and satisfy the same asymptotic guarantees as more computationally intensive learning methods [2], [3]. However, because these guarantees are asymptotic, to obtain reasonable performance on finite data sets practitioners must take care in setting parameters such as the learning rate (step size) for the updates. To alleviate some of this sensitivity and improve the performance of SGD in the finite sample setting, several works [4]–[6] have suggested grouping updates into "mini-batches." This can improve the robustness of the updating at a moderate expense in terms of computation, but also introduces the batch size as a free parameter.

In this paper we derive differentially private versions of single-point SGD and mini-batch SGD and evaluate them on real and synthetic data sets. These algorithms work for gradient descent for general convex objectives – we illustrate the approach using logistic regression for classification. We demonstrate that differentially private single-point SGD has high variance, but a moderate increase in the batch size can improve the performance significantly. For low-dimensional problems, the private algorithm's performance is close to non-private SGD. However, we show that there is a limit to how much the batch size can help, and that the performance is dependent on the learning rate.

II. Preliminaries

While the methods we propose work for general optimization methods, we will describe the problem in terms of a classification problem. There, the data are n labelled examples $(x_1, y_1), \ldots, (x_n, y_n)$, where $x_i \in \mathbb{R}^d$, and $y_i \in \{-1, 1\}$. We assume that for all i, the norm $\|x_i\| \leq 1$. In linear classification, our goal is to find a hyperplane through the origin that largely separates the examples labeled 1 from those labeled -1. The most popular method of training such a linear classifier based on labelled data is by solving a regularized convex optimization problem:

$$w^* = \operatorname*{\mathbf{argmin}}_{w \in \mathbb{R}^d} \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \ell(w, x_i, y_i) \qquad (1)$$

Here w is the normal vector to the hyperplane separator, and ℓ is a convex loss function. Popular choices for ℓ in the machine learning literature are the logistic loss $\ell(w, x, y) = \log(1 + e^{-yw^\top x})$, which leads to Logistic Regression, and the hinge loss $\ell(w, x, y) = \max(0, 1 - yw^\top x)$, which leads to Support Vector Machines (SVMs).

SGD is an iterative algorithm for solving the regularized convex optimization problem in 1. SGD begins with an initial point w_0, and at step t, updates the iterate as:

$$w_{t+1} = w_t - \eta_t(\lambda w_t + \nabla \ell(w_t, x_t, y_t)) \qquad (2)$$

Here η_t is a learning rate, and the (sub)gradient $\nabla\ell(w_t, x_t, y_t)$ is computed based on a single example (x_t, y_t).

In SGD with mini-batch updates, instead of a single example, the update at each step t is based on a small subset B_t of examples of size b. Specifically,

$$w_{t+1} = w_t - \eta_t \left(\lambda w_t + \frac{1}{b} \sum_{(x_i,y_i)\in B_t} \nabla\ell(w_t, x_i, y_i) \right) \quad (3)$$

Both of these methods are approximations of a full gradient update – if the point(s) at each time t are sampled uniformly from $\{1, 2, \ldots, n\}$ then the expected gradient step at each iteration is equal to a gradient step on the full objective function in (1). More generally, we can consider general empirical risk minimization with convex loss functions. We study the L_2-regularized objective because strong convexity allows favorable theoretical guarantees.

Our algorithms guarantee *differential privacy*, a crypographically-motivated notion of privacy due to [1]. Differential privacy has gained significant attention over the past few years in the computer science community, and has spawned a growing literature. The privacy parameter $\alpha > 0$ quantifies privacy risk; lower α means higher privacy.

Definition 1: A (randomized) algorithm $\mathcal{A}$ whose outputs lie in a domain $\mathcal{S}$ is said to be α-differentially private if for all $S \subseteq \mathcal{S}$, for all datasets D and D' that differ in the value of a single individual, it is the case that: $\Pr(\mathcal{A}(D) \in S) \leq e^\alpha \Pr(\mathcal{A}(D') \in S)$

Dwork and Smith reviewed much of the early theoretical work on differential privacy [7]. Sarwate and Chaudhuri provide a tutorial from a signal processing perspective [8]. Signal processing methods for differential privacy have recently been investigated by Le Ny and Pappas [9], [10]. The work most connected to the current paper are those on differentially private classification [11], [12]. Duchi et al. proposed an SGD method for *local privacy* [13]. Stochastic gradient methods are on example of online learning methods. Another approach to differentially private online learning was proposed by Jain et al. [14]; however, their algorithm is more computationally intensive than ours. The PINQ [15] package uses a noisy sum operation to compute full noisy gradient steps for logistic regression [16]. There the goal was exchanging iterations for accuracy. The noisy perceptron method [17] also uses iterative noisy updates to learn a classifier. We focus here on the effect of step size and batch size for SGD methods, so we do not compare the performance for these specific classification methods.

III. SGD with Differential Privacy

A differentially-private version of the SGD update can be written as:

$$w_{t+1} = w_t - \eta_t \left(\lambda w_t + \nabla\ell(w_t, x_t, y_t) + Z_t \right), \quad (4)$$

where each Z_t is a random noise vector in $\mathbb{R}^d$ drawn independently from the density:

$$\rho(z) \propto e^{-(\alpha/2)\|z\|} \quad (5)$$

A differentially-private version of the mini-batch update using a batch B_t of examples of size b can be written as:

$$w_{t+1} = w_t - \eta_t \left(\lambda w_t + \frac{1}{b} \sum_{(x_i,y_i)\in B_t} \nabla\ell(w_t, x_i, y_i) + \frac{1}{b} Z_t \right). \quad (6)$$

where Z_t is again drawn from the density in (5).

Theorem 1 shows that provided the initialization point w_0 is determined independent of the sensitive data, the batches B_t are disjoint, and the $\|\nabla\ell(w, x, y)\|$ is bounded for all w, these updates are α-differentially private.

Theorem 1 (Privacy of SGD and Mini-Batch Updates): Suppose we run SGD with mini-batch updates in (6) for T batches $B_1, \ldots, B_T$. If the initialization point w_0 is chosen independent of the sensitive data, the batches B_t are disjoint, and if $\|\nabla\ell(w, x, y)\| \leq 1$ for all w, and all (x_i, y_i), then SGD with mini-batch updates is α-differentially private.

Due to space limitations, we provide a sketch of the proof here and defer the details for the full version of this work. The key idea of the proof is the observation that provided the conditions of the theorem hold, the global sensitivity of each update is $\frac{2\eta_t}{b}$. The proof now follows by combining this observation with results of Dwork et al. [1], and using the fact that the privacy guarantee does not degrade across batches as the samples used in the batches are disjoint.

Because we add noise at each iteration, the SGD procedure guarantees differential privacy in a "local" sense – each individual i may choose an α_i and this method can guarantee differential privacy at different levels α_i for different individuals by adjusting the distribution of Z_t. A slightly different notion of local privacy was also studied by Duchi et al. [13] in the statistical setting: there the algorithm can sample individuals from a distribution with unknown parameter and the goal is to estimate the parameter. At each time their algorithm can samples a new individual and receives a noisy subgradient estimate. They use mirror descent to guarantee privacy under a variant of differential privacy based on a mutual information criterion.

IV. Experiments

A. Datasets

We consider three classification tasks – one on a synthetic dataset and two on real data. Our synthetic dataset consists of $n = 10,000$ samples drawn uniformly from a 5-dimensional sphere, and is linearly separable with margin 0.001. For our first classification task on real data, we use the `KDDCup99` dataset [18], an intrusion detection dataset on network connections. We address the normal vs. malicious classification task, and use a subsample of size $50,000$. For our second task on real data, we address the "1 vs. all" classification task on the MNIST dataset [19], which consists of $60,000$ training examples and $10,000$ test examples of images of handwritten digits 0 to 9. In both cases, we preprocess the data by normalizing each feature, projecting each row to the unit ball, and then reducing the data dimension by random

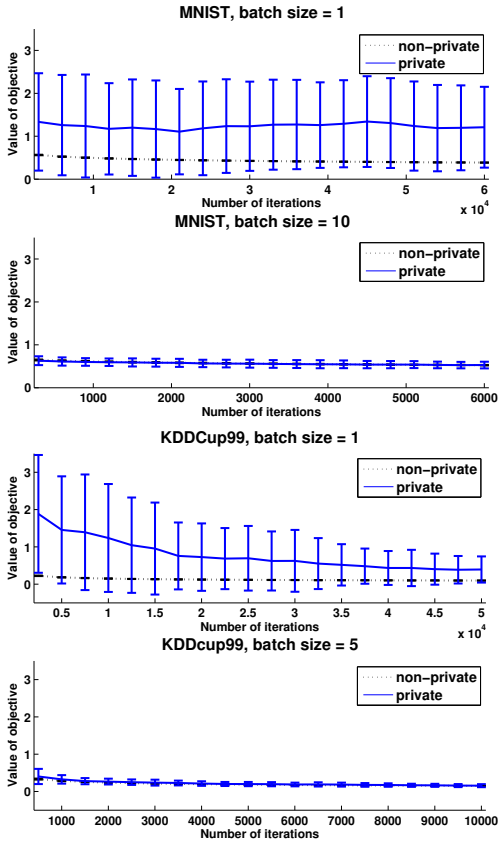

Fig. 1: Objective function value vs. number of iterations for private and non-private algorithms on the MNIST and KDDCup99 data sets. Horizontal axis is scaled so that the number of samples is the same.

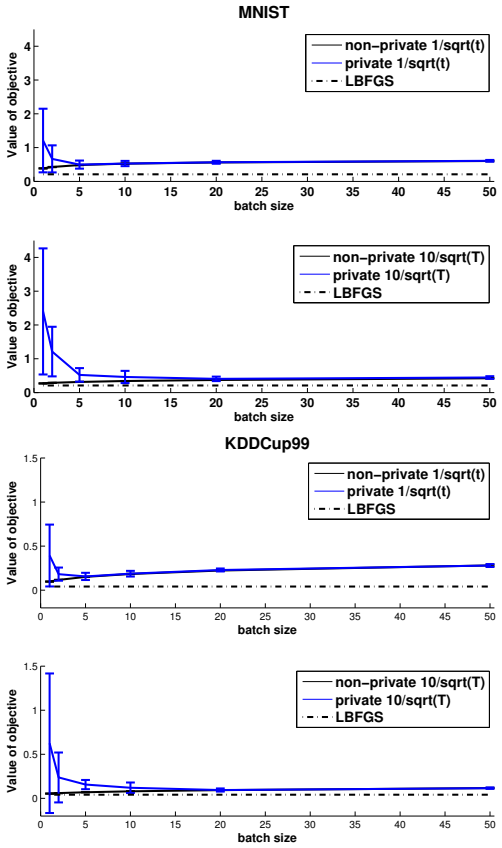

Fig. 2: Objective function value vs. batch size b for private and non-private algorithms on MNIST and KDDCup99.

projections, which preserve differential privacy. We use a reduced dimension of $d = 9$ for KDDCup, and $d = 15$ for MNIST.

B. Procedure

We use SGD to train a logistic regression model. For each update, we use the mini-batch update from (6) for batch sizes $b \in \{1, 2, 5, 10, 20, 50\}$, with regularization parameter $\lambda = 0.0001$ and $\alpha = 1$. In each case, we make a *single pass* over the entire training data. To maintain numerical stability, after each update, we project the iterate w_t onto a ball of radius $1/\lambda$. For each experiment we investigated a few different schemes for setting the learning rates. We averaged the objective function values obtained over 20 random permutations of the training data as well as fresh random samples of the noise Z_t for the private algorithm. The error bars are at a single standard deviation. Since we are interested in the optimization performance, we plot the objective function value – in future work we will also investigate the classification accuracy.

C. Mini-batching reduces variance

The first question we ask is how SGD would fare with batch size 1, since this is the case which has been most studied in the literature. The top half of Figure 1 shows the objective value for the MNIST data set versus the number of samples in the algorithm for learning rate $\eta_t = 1/\sqrt{t}$. For batch size $b = 1$, differentially private SGD is far from the non-private objective and furthermore has high variance. That is, the noise added in each iteration prevents the algorithm from converging. However, a modest batch size $b = 10$, as shown in the lower half of the figure, reduces the variance of differentially private SGD to the point of matching non-private SGD, even for a moderate number of data points.

The other plots in Figure 1 show that this behavior also holds for the KDDCup99 data set. Although the variance of the differentially private algorithm decreases slowly, choosing $b = 5$ makes the mini-batch SGD performance nearly identical to that of the non-private mini-batch SGD. What these two experiments indicate is that in terms of objective value, guaranteeing differential privacy can come for "free" using SGD with moderate batch size. We emphasize here that all of these examples are low-dimensional problems, and the privacy parameter $\alpha = 1$. It is well-known that differentially private learning algorithms often have a sample complexity that scales linearly with the data dimension d and inversely with the privacy risk α. Thus a moderate reduction in α or increase in d may require more data. It would be interesting to see if increasing the batch size can still make private SGD

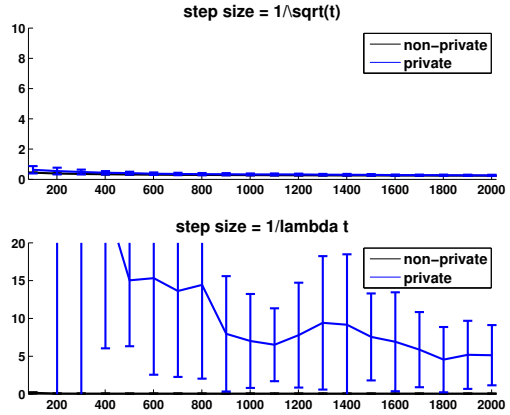

Fig. 3: Objective function value vs. number of data points for private and non-private algorithms on synthetic data for batch size $b = 5$.

match non-private SGD in these settings.

D. Choosing appropriate parameters

Our next experiment is to find the impact of batch size on the performance of these algorithms. Figure 2 shows the objective value as a function of batch size for private SGD, non-private SGD, and a centralized learning procedure which solves the optimization using all of the data points. In all cases, increasing the batch size improved the performance of private SGD, but there is a limit – for step size $1/\sqrt{t}$, much larger batch sizes actually degrade performance. Because we choose to make a single pass over the data to limit the noise per iteration, increasing the batch size decreases the number of iterations, and therefore there is an optimal choice of b for each problem. With a larger learning rate $10/\sqrt{t}$, the performance for larger batches does not degrade as much, and the end value of the objective is closer to that of the centralized learning algorithm.

Many analyses of SGD in the strongly convex case suggest that a learning rate $\eta_t = 1/\lambda t$ guarantees fast convergence rates [2]. In our case λ is quite small, meaning the objective is not very strongly convex. To see the impact of the noise added for differential privacy, we simulated two learning rates, $1/\sqrt{t}$ and $1/\lambda t$, on the synthetic data with $b = 5$. The results in Figure 3 show that choosing a rapidly decreasing step size dramatically increases the variance of private SGD. In practice, choosing the step size in stochastic approximation schemes is often a matter of art, and differentially private noise complicates this choice.

V. Conclusions

In this paper we investigated how differential privacy affects mini-batched stochastic gradient descent (SGD). When data is plentiful, privacy is "affordable," and SGD strategies are more computationally efficient. We showed that in many cases the performance of differentially private SGD was close to that of non-private SGD, especially with larger batch sizes. In stochastic optimization, both the variability of the algorithm and the impact of privacy-preserving noise can be ameliorated by processing groups of points together. Our experiments show that privacy affects both the optimal batch size b and learning rate η_t. Some interesting future directions suggested by our work include: quantifying the impact of the dimension d and privacy parameter α, allowing different α_i's for each point, and using multiple passes through the data to trade off iterations, total privacy loss (via composition results for differential privacy), and error. These modifications could make differentially private learning more effective in practical settings.

Acknowledgement

KC and SS would like to thank NIH U54-HL108460, the Hellman Foundation, and NSF IIS 1253942 for support.

References

[1] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, vol. 3876, March 2006, pp. 265–284.

[2] A. Rakhlin, O. Shamir, and K. Sridharan, "Making gradient descent optimal for strongly convex stochastic optimization," ArXiV, Tech. Rep. arXiv:1109.5647 [cs.LG], 2012.

[3] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: Primal Estimated sub-GrAdient SOlver for SVM," *Mathematical Programming, Series B*, vol. 127, no. 1, pp. 3–30, 2011.

[4] A. Cotter, O. Shamir, N. Srebro, and K. Sridharan, "Better mini-batch algorithms via accelerated gradient methods," in *Adv. NIPS 24*, 2011, pp. 1647–1655.

[5] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, "Optimal distributed online prediction using mini-batches," *J. Mach. Learn. Res.*, vol. 13, pp. 165–202, 2012.

[6] M. Takáč, A. Bijral, P. Richtárik, and N. Srebro, "Mini-batch primal and dual methods for SVMs," in *Proc. ICML*, 2013.

[7] C. Dwork and A. Smith, "Differential privacy for statistics: What we know and what we want to learn," *Journal of Privacy and Confidentiality*, vol. 1, no. 2, pp. 135–154, 2009.

[8] A. D. Sarwate and K. Chaudhuri, "Signal processing and machine learning with differential privacy : Algorithms and challenges for continuous data," *IEEE Sig. Proc. Mag.*, vol. 30, no. 5, pp. 86–94, September 2013.

[9] J. Le Ny and G. J. Pappas, "Differentially private filtering," in *Proc. CDC*, Maui, HI, USA, December 2012, pp. 3398–3403.

[10] ——, "Differentially private Kalman filtering," in *Proc. Allerton*, Monticello, IL, USA, October 2012, pp. 1618–1625.

[11] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *J. Mach. Learn. Res.*, vol. 12, pp. 1069–1109, March 2011.

[12] B. I. P. Rubinstein, P. L. Bartlett, L. Huang, and N. Taft, "Learning in a large function space: Privacy-preserving mechanisms for SVM learning," *Journal of Privacy and Confidentiality*, vol. 4, no. 1, pp. 65–100, 2012.

[13] J. Duchi, M. Jordan, and M. Wainwright, "Privacy aware learning," in *Adv. NIPS 25*, 2012, pp. 1439–1447.

[14] P. Jain, P. Kothari, and A. Thakurta, "Differentially private online learning," in *Proc. COLT*, June 2012.

[15] F. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," *Communications of the ACM*, vol. 53, no. 9, pp. 89–97, September 2010.

[16] O. Williams and F. McSherry, "Probabilistic inference and differential privacy," in *Adv. NIPS 23*, J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds., 2010, pp. 2451–245.

[17] A. Blum, C. Dwork, F. McSherry, and K. Nissim, "Practical privacy: the SuLQ framework," in *Proc. PODS*. New York, NY, USA: ACM, 2005, pp. 128–138.

[18] S. Rosset and A. Inger, "KDD-cup 99: knowledge discovery in a charitable organization's donor database," *SIGKDD Explor. Newsl.*, vol. 1, no. 2, pp. 85–90, Jan. 2000.

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proc. IEEE*, 1998, pp. 2278–2324.