# Bridging the Gap between SG-MCMC and Differential Privacy

Bai Li, Changyou Chen, Hao Liu

Duke University

October 6, 2017

# Overview

1. Differential Privacy
   - Motivation
   - Definition

2. Differentially Private SG-MCMC
   - Previous Work
   - Differentially Private SGLD
   - Empirical Results

# Motivation

The training data could be recovered by only manipulating the model output, thus is not 'private'.

Figure: An image recovered using an inversion attack (left) and a training set image of the victim (right) from Matt Fredrikson et al.(2015)

# Differential Privacy (Dwork [2008])

## Differential Privacy

For two data sets $D$ and $D'$ that only differ by one record, a randomized algorithm $\mathcal{M}$ mapping from data space to range($\mathcal{M}$) satisfies ($\epsilon, \delta$)-Differential privacy if for all measurable $\mathcal{S} \subset$ range($\mathcal{M}$)

$$Pr(\mathcal{M}(\mathcal{D}) \in \mathcal{S}) \leq e^{\epsilon} Pr(\mathcal{M}(\mathcal{D}') \in \mathcal{S}) + \delta.$$

where $\epsilon$ and $\delta$ are two positive parameters which indicate the privacy loss.

Good properties:

- quantitatively evaluate the privacy loss
- protect privacy from all kinds of attacks, thus acknowledged as "the strongest privacy guarantee"

Problem: the utility of its output is not guaranteed
Goal: Keep a good balance between the privacy and the utility

# Previous Work

The idea to privately release stochastic gradient has been well-studied. Song et al. [2013], Bassily et al. [2014] and Abadi et al. [2016] all proposed differentially private stochastic gradient descent (SGD) algorithms:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\eta_t}{\tau} \left( \sum_{i \in J} \nabla \log \ell(\boldsymbol{\theta}_t | x_i) + N(0, \sigma^2 I) \right)$$

it satisfies $(\epsilon, \delta)$-DP if $\sigma$ is above **a certain threshold**. Here $\boldsymbol{\theta}$ are the parameters, $x_i$ are the data, $\eta_t$ is the stepsize, $\tau$ is the batch size.
It essentially adds a normal noise after computing a new gradient.

# Differentially Private SG-MCMC

However, there is no theoretical guarantee showing optimization methods with noise will work on non-convex problems. On the other hand, in Bayesian inference, posterior sampling naturally introduce randomness, and further satisfies DP "for free" (Wang et al. [2015]).
For example, stochastic gradient Langevin dynamics (SGLD):

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \left( \frac{\nabla r(\theta_t)}{N} + \frac{1}{\tau} \left( \sum_{i \in J} \nabla \log \ell(\boldsymbol{\theta}_t | x_i) \right) \right) + N(0, \frac{\eta_t}{N} I)$$

where $r$ is the prior distribution, $T$ is the number of iterations, $\eta_t$ is the stepsize, $N$ is the size of the training set, and $\tau$ is the batch size. It's guaranteed to converge to the true posterior distribution as $t \to \infty$ in theory.

# Differentially Private SG-MCMC

Wang et al. [2015] proved the SGLD algorithm satisfies $(\epsilon, \delta)$ Differential Privacy if:

$$\eta_t < \frac{\epsilon^2 N}{128\, T L^2 \log\left(\frac{2.5T}{\delta}\right) \log(2/\delta)} \tag{1}$$

where $T$ is the number of iterations, $L$ is the gradient norm bound, $\eta_t$ is the stepsize, $N$ is the size of the training set.
However, a stepsize that satisfies the above condition is often too small to be practically useful:

1. With a small stepsize, the Markov chain will mix slowly.
2. The bound is a function of the number of iterations $T$, which means we are limited to run certain number of iterations due to the privacy constraints.

# Our results

We improve the bound from Wang et al. [2015]:

$$\underbrace{\eta_t < \frac{\epsilon^2 N}{128 TL^2 \log\left(\frac{2.5T}{\delta}\right)\log(2/\delta)}}_{\text{Wang et al.}} \rightarrow \underbrace{\eta_t < \frac{\epsilon^2 N t^{-1/3}}{c^2 T^{2/3} L^2 \log(1/\delta)}}_{ours}$$

where $c$ is a real number that depends on $N, L, T, \epsilon, \delta$. Note we let the stepsize decrease in $o(t^{-1/3})$.

This result is surprisingly good as our bound allows us to choose a stepsize that is practically useful and even optimal.

## How Good Is It?

For MNIST data set, we have $N = 50k$, and if we let $\epsilon = 0.1$, $\delta = 10^{-5}$, $T = 10000$, and $L = 4$, the upper bound is $\eta_t < 0.106$. Therefore, the standard SGLD with $\eta_t = 0.1$ satisfies $(\epsilon, \delta)$-DP already. Note $\eta_t = 0.1$ is often the optimal stepsize for many problems.
As a comparison, we would get $\eta_t < 1.54 \times 10^{-6}$ using the bound in Wang et al. [2015].

Note the optimal stepsize usually takes value in $(10^{-4}, 10^{-1})$. We argue that for most problems, even when the privacy constraints are strong, this range falls below our upper bound.

# Upper Bounds

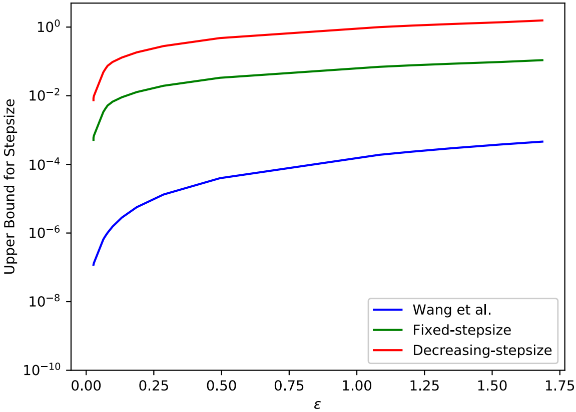


Figure: Upper bounds for fixed-stepsize and decreasing-stepsize with different privacy loss $\epsilon$, as well as the upper bound from Wang et al. [2015].

# Bayesian Logistic Regression on the UCI Adult Dataset

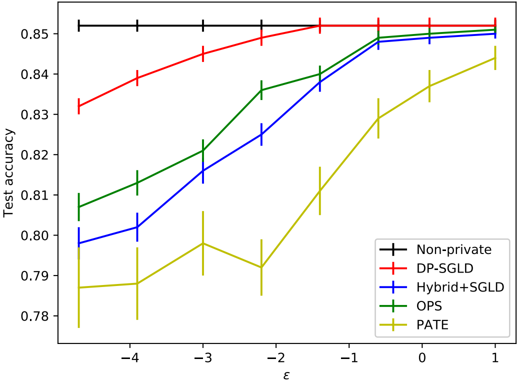


Figure: Test accuracies on a classification task based on Bayesian logistic regression for One-Posterior sample (OPS), Hybrid Posterior sampling based on SGLD, and our proposed DP-SGLD with different choice of privacy loss $\epsilon$. The non-private baseline is obtained by standard SGLD.

# Deep Neural Networks

Table: Test accuracies on MNIST and and SVHN for different methods.

| Dataset | Methods | $\epsilon$ | $\delta$ | Accuracy |
|---|---|---|---|---|
| MNIST | Non-Private | | | 99.23% |
| | PATE(100) | 2.04 | $10^{-5}$ | 98.00% |
| | PATE(1000) | 8.03 | $10^{-5}$ | 98.10% |
| | **DP-SGLD** | 0.10 | $10^{-5}$ | 99.12% |
| | **DP-SGHMC** | 0.24 | $10^{-5}$ | **99.28%** |
| SVHN | Non-Private | | | 92.80% |
| | PATE(100) | 5.04 | $10^{-6}$ | 82.76% |
| | PATE(1000) | 8.19 | $10^{-6}$ | 90.66% |
| | **DP-SGLD** | 0.12 | $10^{-6}$ | 92.14% |
| | **DP-SGHMC** | 0.43 | $10^{-6}$ | **92.84%** |

# Summary

Previous works have to modify existing algorithms or build complicated frameworks and sacrifice a certain amount of performance to achieve $(\epsilon, \delta)$-DP, even when $\epsilon, \delta$ are relatively large.

Our results essentially show the standard SG-MCMC methods with an optimal stepsize guarantees strong (state-of-the-art) DP.

# References

Martín Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Differentially private empirical risk minimization: Efficient algorithms and tight error bounds. *arXiv preprint arXiv:1405.7085*, 2014.

Cynthia Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer, 2008.

Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, pages 245–248. IEEE, 2013.

Yu-Xiang Wang, Stephen Fienberg, and Alex Smola. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*,

# Appendix: Private Aggregation of Teacher Ensembles (PATE)

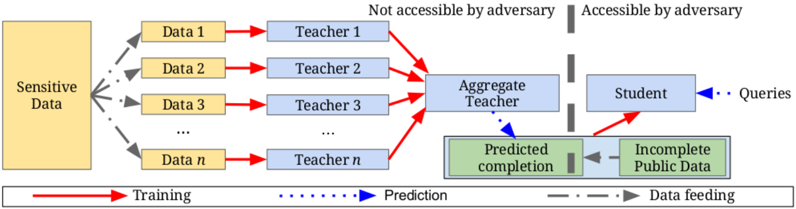


Figure: Overview of this approach: (1) an ensemble of teachers is trained on disjoint subsets of the sensitive data, (2) a student model is trained on public data labeled using the ensemble plus public unlabeled data with semi-supervised learning.

This approach requires extra public unlabeled data.

# Appendix: DP-SGD

## DP-SGD (Abadi et al. [2016])

During SGD updates, use the following

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\eta_t}{\tau}\left(\sum_{i \in J} \nabla \log \ell(\boldsymbol{\theta}_t | x_i) + N(0, \sigma^2 I)\right)$$

then for $T$ iterations, it satisfies $(\epsilon, \delta)$-DP if

$$\sigma \geq c \frac{qL\sqrt{T \log(\frac{1}{\delta})}}{\epsilon}$$

where $c$ is a constant, $q = \frac{\text{batch size}}{\text{data size}}$.