Bolt-on Differential Privacy for Scalable Stochastic Gradient Descent-based Analytics

Xi Wu
wuxi@google.com

Joint work with Fengan Li, Arun Kumar,
Kamalika Chaudhuri, Somesh Jha and Jeffrey Naughton

May 15, 2017

Theme of the Talk

- Better differentially private Stochastic Gradient Descent (SGD).
  - SGD is a popular optimization algorithm for machine learning.
  - Differential privacy is the de facto standard in formalizing privacy.

- Improve private SGD on the following aspects simultaneously:
  - Easier to implement: “Bolt on” with an existing implementation.
  - Run faster,
  - Better convergence/accuracy and
  - Support a stronger privacy model.

- Essence behind the “all-win” improvements: A novel analysis of the L_2-sensitivity of SGD.

Background: Differential Privacy

- [Dwork, McSherry, Nissim and Smith, TCC 2006]
  - A formal notion on how to anonymize participation.
  - Gödel Prize 2017.

- Intuition for differential privacy:
  - Participation is anonymized if it causes little change to the output.
- Has become the de-facto standard of protecting data privacy.
  - Differential privacy will be in your pocket (iOS 10)!
  - Google's RAPPOR.

Background: More Differential Privacy (1/2)

- ε-differentially privacy
  - A stability property of a randomized algorithm $\mathcal{M}$.
  - For any neighboring $S \sim S'$, and any event E,

$$
\begin{aligned}
S' &= \{z_1, \ldots, z_{i-1}, z'_i, z_{i+1}, \ldots, z_m\} \\
S &= \{z_1, \ldots, z_{i-1}, z_i, z_{i+1}, \ldots, z_m\} \\
\Pr[\mathcal{M}(S) \in E] &\le e^{\varepsilon} \cdot \Pr[\mathcal{M}(S') \in E]
\end{aligned}
$$

- (ε, δ)-differential privacy: A relaxation.
  - $\Pr[\mathcal{M}(S) \in E] \le e^{\varepsilon} \Pr[\mathcal{M}(S') \in E] + \delta$
  - Qualitatively weaker privacy model.

Background: More Differential Privacy (2/2)

- ε is a ratio bound that measures the strength of privacy.
  - Smaller ε, stronger privacy.
- We inject random noise to ensure privacy.
  - Typically: Smaller $\varepsilon \leftrightarrow$ More noise $\leftrightarrow$ Less accurate statistics.
- The "game" of finding better differentially private algorithms:
  - For the same ε we want less noise and better accuracy.
  - The key challenge: How to inject noise?

Background: Optimization and Machine Learning

- Setup:
  - $Z = X \times Y$: a sample space.
  - Let $S = \{(x_i, y_i) \ : \ i \in [m]\}$, a training set.
  - $\mathcal{W} \subseteq \mathbb{R}^d$, a hypothesis space.
  - $\ell : \mathcal{W} \times Z \mapsto \mathbb{R}$, a loss function.

- Empirical Risk Minimization (ERM): Find $w \in \mathcal{W}$ that minimizes:

$$\frac{1}{m} \sum_{i=1}^{m} \ell(w, (x_i, y_i))$$

m: training set size.

Stochastic Gradient Descent

- A fundamental algorithm for ERM,
- An iterative procedure: At iteration t, sample $i_t \sim [m]$, and

$$w_{t+1} = w_t - \eta_t \nabla \ell_{i_t}(w_t).$$

- Problem Statement: How to inject noise for SGD to get both *private* and *accurate models*?
  - Focus on convex optimization (ℓ_i is convex).
  - Some remarks on non-convex optimization in the backup slides.

A Remark: Why Differentially Private SGD?

- SGD is fundamental for training machine learning models.
    - In particular on large scale datasets.
    - Private SGD implies automatic privacy for all these models.

- More robust privacy guarantees
    - Many previous work on private ERM requires assumptions in finding the exact minimizer, which is too idealistic.
    - Making SGD private avoids any such assumption.

Previous Private SGD

A common paradigm: Inject noise at each iteration.

- Each step locally private, global privacy follows from composition.

Previous Private SGD

A common paradigm: Inject noise at each iteration.

- Each step locally private, global privacy follows from composition.

[+]: Pros, [-]: Cons.

- [Song, Chaudhuri and Sarwate (GlobalSIP 2013)]
  - [-] A lot of noise for each iteration, very "inaccurate" model.
- [Bassily, Smith and Thakurta (STOC 2014)]
  - [+] Reduces noise for each iteration, and improves composition.
  - [-] The composition only works for (ε, δ)-differential privacy.
  - [-] (Their proof) needs $\Theta(m^2)$ iterations to converge.
- Both approaches
  - [-] Relatively hard to implement.
  - [-] Large runtime overhead.

Our Proposal

- Use the classic "output perturbation" method.
  - Inject noise only at the end to the result of non-private SGD.

- Analyze "global stability" of SGD:

$$L_2\text{-sensitivity} : \Delta_2 = \max_{S,S',r,r'} \left\| SGD(r, S) - SGD(r', S') \right\|_2$$

[Challenge] *Upper bound Δ_2 by a small quantity.*

Our Proposal

- Use the classic "output perturbation" method.
  - Inject noise only at the end to the result of non-private SGD.

- Analyze "global stability" of SGD:

$$L_2\text{-sensitivity} : \Delta_2 = \max_{S,S',r,r'} \left\| SGD(r,S) - SGD(r',S') \right\|_2$$

[Challenge] *Upper bound Δ_2 by a small quantity.*

- [Our Contribution] *Address the challenge by a novel analysis of Δ_2.*

Our Proposal

- Use the classic "output perturbation" method.
  - Inject noise only at the end to the result of non-private SGD.

- Analyze "global stability" of SGD:

$$L_2\text{-sensitivity} : \Delta_2 = \max_{S,S',r,r'} \left\| SGD(r,S) - SGD(r',S') \right\|_2$$

[Challenge] *Upper bound Δ_2 by a small quantity.*

- [Our Contribution] *Address the challenge by a novel analysis of Δ_2.*

- Automatic benefits
  - [+] Easier to implement: "Bolt on" with an existing implementation.
  - [+] Low runtime overhead.

Our Algorithms: The New Part is How to Set Δ_2

Algorithm 1 Private Convex Permutation-based SGD

Require: $\ell(\cdot, z)$ is convex for every z, $\eta \leq 2/\beta$.
Input: Data S, parameters k, η, ε

```
function PrivateConvexPSGD(S, k, ε, η)
    w ← PSGD(S) with k passes and η_t = η
    Δ_2 ← 2kLη
    Sample noise vector κ according to (3).
    return w + κ
```

Our Algorithms: The New Part is How to Set Δ_2

Algorithm 1 Private Convex Permutation-based SGD

Require: $\ell(\cdot, z)$ is convex for every z, $\eta \leq 2/\beta$.
Input: Data S, parameters k, η, ε

```
function PrivateConvexPSGD(S, k, ε, η)
    w ← PSGD(S) with k passes and η_t = η
    Δ_2 ← 2kLη
    Sample noise vector κ according to (3).
    return w + κ
```

Algorithm 2 Private Strongly Convex Permutation-based SGD

Require: $\ell(\cdot, z)$ is γ-strongly convex for every z
Input: Data S, parameters k, ε

```
function PrivateStronglyConvexPSGD(S, k, ε)
    w ← PSGD(S) with k passes and η_t = min(1/β, 1/(γt))
    Δ_2 ← 2L/(γm)
    Sample noise vector κ according to (3).
    return w + κ
```

Theoretical Guarantees of Our Algorithms

With output perturbation...

Theorem (Informal)

There is a private SGD algorithm based on output perturbation that gives both ε-differential privacy and convergence, even for 1 epoch over the data.

Intuition: Convergence with stronger privacy model (ε-DP).

Theoretical Guarantees of Our Algorithms

With output perturbation...

Theorem (Informal)

There is a private SGD algorithm based on output perturbation that gives both ε-differential privacy and convergence, even for 1 epoch over the data.

Intuition: Convergence with stronger privacy model (ε-DP).

Theorem (Informal)

For (ε, δ)-differential privacy and constant epochs, there is a private SGD algorithm based on output perturbation that gives $(\log m)^{O(1)}$-factor improvement in excess empirical risk over BST14.

Intuition: Better convergence for $O(1)$ passes and (ε, δ)-DP.

Empirical Study

- Datasets: MNIST (for this talk).
  - Recognize digits in images.
  - More datasets in the paper: KDDCup-2004 Protein, Forest Covertype.
- Model: Build logistic regression models (using SGD).
- Key Experimental Results:
  - Much faster running time.
  - Substantially better model accuracy.

Implementation

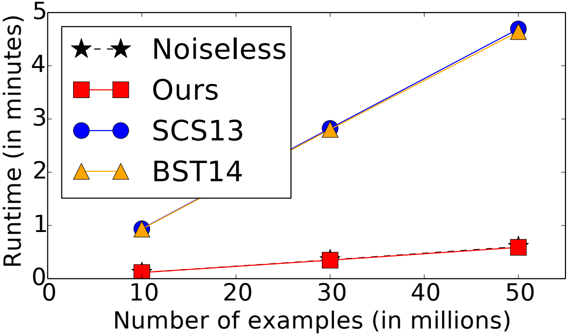
- Implemented using Bismarck
  - An in-RDBMS analytics system.
  - [Feng, Kumar, Recht and Re (SIGMOD 2012)].
  - Using Permutation-based SGD to unify in-RDBMS analytics.

- Integration effort.
  - Our algorithms: Trivial to integrate.
  - SCS13, BST14: Needs to re-implement sampling functions inside Bismarck core.

Experimental Results: Running Time

Much faster when CPU cost dominates the runtime:

- Negligible overhead compared to the noiseless version.



Experimental Results: ε-Differential Privacy

More accurate for the same privacy guarantee (ε):

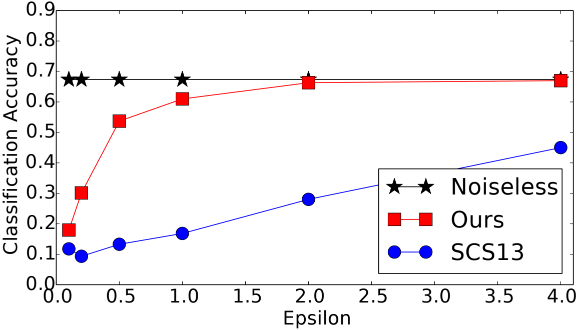


Figure : Convex case. Mini-batch size is 50, 10 epochs

Experimental Results: (ε, δ)-Differential Privacy

Up to 4X better test accuracy:

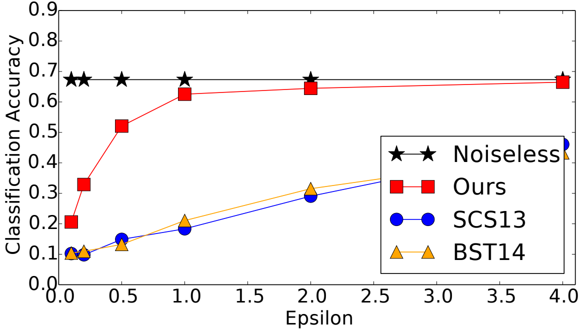


Figure : Convex case. $\delta = 1/m^2$. Mini-batch size is 50, 10 epochs

Very Roughly: How the Theory Works

- Sharpen and combine two recent theory advancements:
  - Stability of SGD in expectation: [Hardt, Recht and Singer, ICML 2016].
  - Convergence of Permutation-based SGD (PSGD): [Shamir, NIPS 2016].

Very Roughly: How the Theory Works

- Sharpen and combine two recent theory advancements:
  - Stability of SGD in expectation: [Hardt, Recht and Singer, ICML 2016].
  - Convergence of Permutation-based SGD (PSGD): [Shamir, NIPS 2016].

- Part 1: From "stability in expectation" to ε-differential privacy.
  - Have to use PSGD.
  - Key: If the randomness does not depend on S, then it suffices to bound

$$\max_{S,S',r} \left\| SGD(r,S) - SGD(r,S') \right\|.$$

  - Differential privacy is really a notion of worst-case stability.

Very Roughly: How the Theory Works

- Sharpen and combine two recent theory advancements:
  - Stability of SGD in expectation: [Hardt, Recht and Singer, ICML 2016].
  - Convergence of Permutation-based SGD (PSGD): [Shamir, NIPS 2016].
- Part 1: From "stability in expectation" to ε-differential privacy.
  - Have to use PSGD.
  - Key: If the randomness does not depend on S, then it suffices to bound

$$\max_{S,S',r} \left\| SGD(r,S) - SGD(r,S') \right\|.$$

  - Differential privacy is really a notion of worst-case stability.
- Part 2: Convergence of private PSGD.
  - Convergence of PSGD is poorly understood in theory.
  - We mitigate this issue using Shamir's results.

Important Details that We Do Not Cover

- Please refer to the paper for the following important details:
  - Proofs.
  - How batch sizes improve accuracy under the same privacy guarantee.
  - How to set hyperparameters.
  - How to do private parameter tuning.
  - Reduce dimensionality via random projection.
  - More lessons we learned (e.g. Our algorithms are easier to tune).
  - More implementation details (differential privacy can be very subtle).
  - More experimental results.
  - ...

Summary and Future Directions

- Better differentially private stochastic gradient descent
  - Bolt-on implementation, more efficient, produces more accurate models and supports a stronger privacy model.
- Many interesting things to do:
  - Better understanding of convergence of constant-epoch private SGD.
  - Principled ways to set batch size for private SGD?
  - Systematic comparison of different approaches to private ERM.
  - How does our work fit into the larger context of implementing a differential privacy system?
  - ...

Thanks!

?

Backup Slides

Better Analysis of L_2-Sensitivity of SGD

- Denote A the non-private SGD algorithm.
  - $A(r, S)$: r the randomness part, S the input training set.
  - R : random variable where r is sampled from.

- Step 1: Reduce to the "same randomness" case.
  - In general, we need to bound

$$\max_{S,S',r,r'} \|A(r,S) - A(r',S')\|.$$

  - Key: If the random variable R does not depend on S, then we can bound

$$\max_{S,S',r} \|A(r,S) - A(r,S')\|.$$

“Same Randomness” $\Rightarrow$ “Almost Identical Gradient Updates”

- Step 2: Analyze the “same randomness” case:
  - Permutation-based SGD (PSGD): We sample a random permutation r of $[m]$, and cycle through S according to r.

"Same Randomness"
$\Rightarrow$ "Almost Identical Gradient Updates"

- Step 2: Analyze the "same randomness" case:
    - Permutation-based SGD (PSGD): We sample a random permutation r of $[m]$, and cycle through S according to r.
    - We have the following diagram (G_i are functions)

$$
\begin{array}{c}
S : w_0 \xrightarrow{G_1} w_1 \xrightarrow{G_2} \cdots \xrightarrow{G_t} w_t \xrightarrow{G_{t+1}} \cdots \xrightarrow{G_T} w_T \\
\uparrow \\
\delta_t = \|w_t - w'_t\| \\
\downarrow \\
S' : w'_0 \xrightarrow{G'_1} w'_1 \xrightarrow{G'_2} \cdots \xrightarrow{G'_t} w'_t \xrightarrow{G'_{t+1}} \cdots \xrightarrow{G'_T} w'_T
\end{array}
$$

“Same Randomness” $\Rightarrow$ “Almost Identical Gradient Updates”

- Step 2: Analyze the “same randomness” case:
  - Permutation-based SGD (PSGD): We sample a random permutation r of $[m]$, and cycle through S according to r.
  - We have the following diagram (G_i are functions)

$$
\begin{array}{c}
S : w_0 \xrightarrow{G_1} w_1 \xrightarrow{G_2} \cdots \xrightarrow{G_t} w_t \xrightarrow{G_{t+1}} \cdots \xrightarrow{G_T} w_T \\
\uparrow \\
\delta_t = \|w_t - w'_t\| \\
\downarrow \\
S' : w'_0 \xrightarrow{G'_1} w'_1 \xrightarrow{G'_2} \cdots \xrightarrow{G'_t} w'_t \xrightarrow{G'_{t+1}} \cdots \xrightarrow{G'_T} w'_T
\end{array}
$$

  - Key: Due to “same randomness,” in each pass we only encounter once the differing gradient update function $G_{t^*} \neq G'_{t^*}$.

Expansion Properties of Gradient Operators

[Key Quantity] $\delta_t = \|w_t - w_t'\|$

Definition (Expansiveness)

An operator $G : \mathcal{W} \mapsto \mathcal{W}$ is *ρ-expansive* if $\sup_{w,w'} \frac{\|G(w)-G(w')\|}{\|w-w'\|} \leq \rho$.

Intuition: Measure how δ_t gets stretched/contracted.

Expansion Properties of Gradient Operators

[Key Quantity] $\delta_t = \|w_t - w_t'\|$

Definition (Expansiveness)

An operator $G : \mathcal{W} \mapsto \mathcal{W}$ is *ρ-expansive* if $\sup_{w,w'} \frac{\|G(w)-G(w')\|}{\|w-w'\|} \leq \rho$.

Intuition: Measure how δ_t gets stretched/contracted.

Lemma (Nesterov, Polyak)

Assume that ℓ is β-smooth. Then, the following hold.

1. *If ℓ is convex, then for any $\eta \leq 2/\beta$, $G_{\ell,\eta}$ is 1-expansive.*
2. *If ℓ is γ-strongly convex, then for $\eta \leq \frac{2}{\beta+\gamma}$, $G_{\ell,\eta}$ is $(1 - \frac{2\eta\beta\gamma}{\beta+\gamma})$-expansive.*

Intuition: δ_t is either unchanged or is shrinking!

Expansion Properties of Gradient Operators

[Key Quantity] $\delta_t = \|w_t - w'_t\|$

Definition (Expansiveness)

An operator $G : \mathcal{W} \mapsto \mathcal{W}$ is *ρ-expansive* if $\sup_{w,w'} \frac{\|G(w)-G(w')\|}{\|w-w'\|} \leq \rho$.

Intuition: Measure how δ_t gets stretched/contracted.

Lemma (Nesterov, Polyak)

Assume that ℓ is β-smooth. Then, the following hold.

1. *If ℓ is convex, then for any $\eta \leq 2/\beta$, $G_{\ell,\eta}$ is 1-expansive.*
2. *If ℓ is γ-strongly convex, then for $\eta \leq \frac{2}{\beta+\gamma}$, $G_{\ell,\eta}$ is $(1 - \frac{2\eta\beta\gamma}{\beta+\gamma})$-expansive.*

Intuition: δ_t is either unchanged or is shrinking!

Our Results on Bounding δ_T

- Step 3: Using the expansion properties, and that most of the time we are contracting or unchanged (thanks to "same randomness!"),

Our Results on Bounding δ_T

- Step 3: Using the expansion properties, and that most of the time we are contracting or unchanged (thanks to "same randomness!"),

Theorem (Convex)

Consider k-passes PSGD for L-Lipschitz, convex and β-smooth optimization. Let $\eta_1 = \eta_2 = \cdots = \eta_T = \eta \leq \frac{2}{\beta}$. Then $\sup_{S \sim S'} \sup_r \delta_T \leq 2kL\eta$.

Intuition: $\delta_T = O(k\eta)$.

Our Results on Bounding δ_T

- Step 3: Using the expansion properties, and that most of the time we are contracting or unchanged (thanks to "same randomness!"),

Theorem (Convex)

Consider k-passes PSGD for L-Lipschitz, convex and β-smooth optimization. Let $\eta_1 = \eta_2 = \cdots = \eta_T = \eta \le \frac{2}{\beta}$. Then $\sup_{S\sim S'} \sup_r \delta_T \le 2kL\eta$.

Intuition: $\delta_T = O(k\eta)$.

Theorem (Strongly Convex)

Consider k-passes PSGD for L-Lipschitz, γ-strongly convex and β-smooth optimization. Let $\eta_t = \min(\frac{1}{\gamma t}, \frac{1}{\beta})$. Then $\sup_{S\sim S'} \sup_r \delta_T \le \frac{2L}{\gamma m}$.

Intuition: $\delta_T = O(\frac{1}{m})$.

More Remarks on Implications of Our Results

- A recent paper [Zhang, Zheng, Mou and Wang, ArXiv 2017]
- Batch size m can lead to optimal excess empirical risk:
  - Note that this is nothing but Gradient Descent.
  - No need of Shamir's results as no randomness in gradient steps.
- Non-convex Optimization:
  - Basically, by choosing a "random" starting point and then SGD, one can get (ε, δ)-differential privacy with convergence to a stationary point.