Differentially Private Machine Learning

Theory, Algorithms, and Applications

UC San Diego

Kamalika Chaudhuri (UCSD) Anand D. Sarwate (Rutgers)
Logistics and Goals

• Tutorial Time: 2 hr (15 min break after first hour)

• What this tutorial will do:
  • Motivate and define differential privacy
  • Provide overview of common methods and tools to design differentially private ML algorithms

• What this tutorial will not do:
  • Provide detailed results on the state of the art in differentially private versions of specific problems
Learning Outcomes

At the end of the tutorial, you should be able to:

• Explain the definition of differential privacy,

• Design basic differentially private machine learning algorithms using standard tools,

• Try different approaches for introducing differential privacy into optimization methods,

• Understand the basics of privacy risk accounting,

• Understand how these ideas blend together in more complex systems.
Motivating
Differential Privacy
Sensitive Data

Medical Records

Genetic Data

Search Logs
AOL Violates Privacy
A Face Is Exposed for AOL Searcher No. 4417749

By MICHAEL BARBARO and TOM ZELLER Jr.
Published: August 9, 2006

Buried in a list of 20 million Web search queries collected by AOL and recently released on the Internet is user No. 4417749. The number was assigned by the company to protect the searcher’s anonymity, but it was not much of a shield.

No. 4417749 conducted hundreds of searches over a three-month period on topics ranging from “numb fingers” to “60 single men” to “dog that urinates on...
Netflix Violates Privacy [NS08]

2-8 movie-ratings and dates for Alice reveals:
Whether Alice is in the dataset or not
Alice’s other movie ratings
High-dimensional Data is Unique

Example: UCSD Employee Salary Table

<table>
<thead>
<tr>
<th>Position</th>
<th>Gender</th>
<th>Department</th>
<th>Ethnicity</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faculty</td>
<td>Female</td>
<td>CSE</td>
<td>SE Asian</td>
<td>-</td>
</tr>
</tbody>
</table>

One employee (Kamalika) fits description!
Simply anonymizing data is unsafe!
**Disease Association Studies [WLWTZ09]**

Correlation (R² values), Alice’s DNA reveals:

If Alice is in the **Cancer** set or **Healthy** set
Simply anonymizing data is unsafe!
Statistics on small data sets is unsafe!
Privacy Definition
The Setting

Private data set $D$ → Privacy-preserving sanitizer

- non-public
- public

Privacy barrier

- summary statistic
- synthetic dataset
- ML model
Property of Sanitizer

Private data set $D$ → Privacy-preserving sanitizer → non-public summary statistic, synthetic dataset, ML model → public

Aggregate information computable
Individual information protected (robust to side-information)
Differential Privacy

Participation of a person does not change outcome

Since a person has agency, they can decide to participate in a dataset or not
Prior Knowledge:
A’s Genetic profile
A smokes
Adversary

Prior Knowledge:
A’s Genetic profile
A smokes

Case 1: Study

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>2.16</td>
<td>2.51</td>
<td>1.00</td>
</tr>
<tr>
<td>1.05</td>
<td>1.17</td>
<td>0.87</td>
</tr>
<tr>
<td>1.54</td>
<td>1.70</td>
<td>0.93</td>
</tr>
<tr>
<td>1.15</td>
<td>1.40</td>
<td>1.02</td>
</tr>
<tr>
<td>2.70</td>
<td>2.15</td>
<td>2.94</td>
</tr>
<tr>
<td>1.01</td>
<td>0.85</td>
<td>1.70</td>
</tr>
<tr>
<td>2.29</td>
<td>0.71</td>
<td>1.65</td>
</tr>
<tr>
<td>0.47</td>
<td>0.17</td>
<td>2.45</td>
</tr>
<tr>
<td>0.19</td>
<td>0.20</td>
<td>1.32</td>
</tr>
</tbody>
</table>

A has cancer

Cancer

[ Study violates A’s privacy ]
Case 1: Study

| 0.106 | .251 | 1.00 |
| 0.166 | .047 | 1.00 |
| 0.154 | .193 | .033 | 1.00 |
| 0.150 | .056 | .095 | .109 | 1.00 |
| 0.270 | .294 | .246 | .140 | .141 | 1.00 |
| 0.101 | .170 | .056 | .234 | .039 | .175 | 1.00 |
| 0.071 | .155 | .161 | .093 | .199 | .197 | 1.00 |
| 0.171 | .243 | .094 | .144 | .123 | .283 | .216 | 2.74 | 1.00 |
| 0.179 | .202 | .152 | .094 | .087 | .159 | .207 | .108 | .092 | .204 | 1.00 |

A has cancer

Cancer

[ Study violates A’s privacy ]

Case 2: Study

A probably has cancer

Smoking causes cancer

[ Study does not violate privacy ]

Prior Knowledge:
A’s Genetic profile
A smokes
Differential Privacy [DMNS06]

Data + Algorithm → Outcome

Data + Algorithm → Outcome

Participation of a person does not change outcome

Since a person has agency, they can decide to participate in a dataset or not
How to ensure this?

…through randomness

\[ A(\text{Data} + \text{Random variables}) \]

\[ A(\text{Data} + \text{Random variables}) \]

have close distributions
How to ensure this?

Random variables

\[ A\left( \text{Data} + \right) \]

\[ A\left( \text{Data} + \right) \]

have close distributions

Randomness: Added by randomized algorithm A

Closeness: Likelihood ratio at every point bounded
Differential Privacy [DMNS06]

For all $D, D'$ that differ in one person’s value, if $A = \epsilon$-differentially private randomized algorithm, then:

$$\sup_{t} \left| \log \frac{p(A(D) = t)}{p(A(D') = t)} \right| \leq \epsilon$$
For all $D, D'$ that differ in one person's value,

If $A = \epsilon$-differentially private randomized algorithm, then:

\[
\sup_t \left| \log \frac{p(A(D) = t)}{p(A(D') = t)} \right| \leq \epsilon
\]
Approx. Differential Privacy [DKM+06]

For all $D, D'$ that differ in one person's value, if $A = (\epsilon, \delta)$-differentially private randomized algorithm, then:

$$\max_{S, \Pr(A(D) \in S) > \delta} \left[ \log \frac{\Pr(A(D) \in S)}{\Pr(A(D') \in S)} - \delta \right] \leq \epsilon$$
Properties of Differential Privacy
Property 1: Post-processing Invariance

Risk doesn’t increase if you don’t touch the data again
Property 2: Graceful Composition

Total privacy loss is the sum of privacy losses
(Better composition possible — coming up later)
How to achieve Differential Privacy?
Tools for Differentially Private Algorithm Design

- Global Sensitivity Method [DMNS06]
- Exponential Mechanism [MT07]

Many others we will not cover [DL09, NRS07, …]
Global Sensitivity Method [DMNS06]

Problem:
Given function $f$, sensitive dataset $D$
Find a differentially private approximation to $f(D)$

Example: $f(D) = \text{mean of data points in } D$
The Global Sensitivity Method [DMNS06]

Given: A function f, sensitive dataset D
Define: $\text{dist}(D, D') = \#\text{records that } D, D' \text{ differ by}$
The Global Sensitivity Method [DMNS06]

Given: A function $f$, sensitive dataset $D$

Define: $\text{dist}(D, D') = \#\text{records that } D, D' \text{ differ by}$

Global Sensitivity of $f$: 
The Global Sensitivity Method [DMNS06]

Given: A function $f$, sensitive dataset $D$

Define: $\text{dist}(D, D') = \#\text{records that } D, D' \text{ differ by}$

Global Sensitivity of $f$:

$$S(f) = |f(D) - f(D')|$$

Domain($D$)
The Global Sensitivity Method [DMNS06]

Given: A function $f$, sensitive dataset $D$

Define: $\text{dist}(D, D') = \#\text{records that } D, D' \text{ differ by}$

Global Sensitivity of $f$:

$$S(f) = \left| f(D) - f(D') \right|$$

$\text{dist}(D, D') = 1$
The Global Sensitivity Method [DMNS06]

Given: A function \( f \), sensitive dataset \( D \)

Define: \( \text{dist}(D, D') = \# \text{records that } D, D' \text{ differ by} \)

Global Sensitivity of \( f \):

\[
S(f) = \max_{\text{dist}(D, D') = 1} | f(D) - f(D') |
\]
Laplace Mechanism

Global Sensitivity of $f$ is $S(f) = \max_{\text{dist}(D, D') = 1} |f(D) - f(D')|$

Output $f(D) + Z$, where

$Z \sim \frac{S(f)}{\epsilon} \text{Lap}(0, 1)$

$\epsilon$-differentially private

Laplace distribution:

$p(z|\mu, b) = \frac{1}{2b} \exp\left(-\frac{|z - \mu|}{b}\right)$
Global Sensitivity of $f$ is $S(f) = \max_{\text{dist}(D, D') = 1} |f(D) - f(D')|$

Output $f(D) + Z$, where

$$Z \sim \frac{S(f)}{\epsilon} \mathcal{N}(0, 2 \ln(1.25/\delta)) \quad (\epsilon, \delta)$-differentially private
Example 1: Mean

\[ f(D) = \text{Mean}(D), \text{ where each record is a scalar in } [0, 1] \]
Example 1: Mean

\[ f(D) = \text{Mean}(D), \text{ where each record is a scalar in } [0, 1] \]

Global Sensitivity of \( f \) = \( 1/n \)
Example 1: Mean

\[ f(D) = \text{Mean}(D), \text{ where each record is a scalar in } [0, 1] \]

Global Sensitivity of \( f = 1/n \)

**Laplace Mechanism:**

Output \( f(D) + Z, \) where \( Z \sim \frac{1}{n\epsilon} \text{Lap}(0, 1) \)
How to get Differential Privacy?

- Global Sensitivity Method [DMNS06]
- Two variants: Laplace and Gaussian
- Exponential Mechanism [MT07]

Many others we will not cover [DL09, NRS07, …]
Exponential Mechanism [MT07]

Problem:
Given function \( f(w, D) \), Sensitive Data \( D \)
Find differentially private approximation to

\[
    w^* = \arg\max_w f(w, D)
\]

Example: \( f(w, D) = \) accuracy of classifier \( w \) on dataset \( D \)
The Exponential Mechanism [MT07]

Suppose for any $w$,

$$|f(w, D) - f(w, D')| \leq S$$

when $D$ and $D'$ differ in 1 record. Sample $w$ from:

$$p(w) \propto e^{\epsilon f(w, D)/2S}$$

for $\epsilon$-differential privacy.
Example: Parameter Tuning

Given validation data $D$, $k$ classifiers $w_1, \ldots, w_k$ (privately) find the classifier with highest accuracy on $D$

Here, $f(w, D) = \text{classification accuracy of } w \text{ on } D$

For any $w$, any $D$ and $D'$ that differ by one record,

$$|f(w, D) - f(w, D')| \leq \frac{1}{|D|}$$

So, the exponential mechanism outputs $w_i$ with prob:

$$\Pr(w_i) \propto e^{\varepsilon |D| f(w_i, D)/2}$$

[CMS11, CV13]
Summary

— Motivation

— What is differential privacy?

— Basic differential privacy algorithm design tools:
  — The Global Sensitivity Method
    — Laplace Mechanism
    — Gaussian Mechanism
  — Exponential Mechanism
Differential privacy in estimation and prediction
Estimation and prediction problems

Statistical estimation: estimate a parameter or predictor using private data that has good expected performance on future data.

Goal: Good privacy-accuracy-sample size tradeoff
Privacy and accuracy make different assumptions about the data

Privacy – differential privacy makes no assumptions on the data distribution: privacy holds unconditionally.

Accuracy – accuracy measured w.r.t a “true population distribution”: expected excess statistical risk.
Statistical Learning as Risk Minimization

\[ w^* = \arg\min_w \frac{1}{n} \sum_{i=1}^{n} \ell(w, (x_i, y_i)) + \lambda R(w) \]

• *Empirical Risk Minimization* (ERM) is a common paradigm for prediction problems.

• Produces a predictor \( w \) for a label/response \( y \) given a vector of features/covariates \( x \).

• Typically use a convex loss function and regularizer to “prevent overfitting.”
Why is ERM not private?

easy for adversary to tell the difference between $D$ and $D'$

[CMS11, RBHT12]
Kernel learning: even worse

- Kernel-based methods produce a classifier that is a function of the data points.

- Even adversary with black-box access to w could potentially learn those points.

\[
w(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i)
\]
Privacy is compatible with learning

• Good learning algorithms generalize to the population distribution, not individuals.

• *Stable learning algorithms* generalize [BE02].

• Differential privacy can be interpreted as a form of stability that also implies generalization [BNS+15].

• Two parts of the same story: Privacy implies generalization asymptotically. Tradeoffs between privacy-accuracy-sample size for finite $n$. 
Revisiting ERM

\[ w^* = \arg\min_w \frac{1}{n} \sum_{i=1}^{n} \ell(w, (x_i, y_i)) + \lambda R(w) \]

• Learning using (convex) optimization uses three steps:

  1. read in the data
  2. form the objective function
  3. perform the minimization

• We can try to introduce privacy in each step!
Revisiting ERM

\[ w^* = \arg\min_w \frac{1}{n} \sum_{i=1}^{n} \ell(w, (x_i, y_i)) + \lambda R(w) \]

• Learning using (convex) optimization uses three steps:

1. read in the data  \textcolor{red}{\textit{input perturbation}}
2. form the objective function
3. perform the minimization

• We can try to introduce privacy in each step!
Revisiting ERM

\[ w^* = \arg\min_{w} \frac{1}{n} \sum_{i=1}^{n} \ell(w, (x_i, y_i)) + \lambda R(w) \]

- Learning using (convex) optimization uses three steps:
  1. read in the data  \textit{input perturbation}
  2. form the objective function  \textit{objective perturbation}
  3. perform the minimization
- We can try to introduce privacy in each step!
Revisiting ERM

\[ w^* = \arg \min_w \frac{1}{n} \sum_{i=1}^{n} \ell(w, (x_i, y_i)) + \lambda R(w) \]

• Learning using (convex) optimization uses three steps:

1. read in the data \textit{input perturbation}

2. form the objective function \textit{objective perturbation}

3. perform the minimization \textit{output perturbation}

• We can try to introduce privacy in each step!
Privacy in ERM: options
Privacy in ERM: options

- input perturbation
- input perturbation
- input perturbation
- input perturbation
- input perturbation
- input perturbation

sanitized database

non-private algorithm

$(\varepsilon, \delta)$ privacy barrier
Privacy in ERM: options

Privacy barrier

\[ D \rightarrow \text{private database} \rightarrow \text{DP preprocessing} \rightarrow \text{sanitized dataset} \rightarrow \text{non-private learning} \rightarrow \hat{W}_\text{input} \]
Privacy in ERM: options
Privacy in ERM: options

Private database $D$ → non-private preprocessing → non-private optimization → noise addition or randomization → $w^*$ → $\hat{w}_{\text{output}}$

Privacy barrier $(\varepsilon, \delta)$
Local Privacy

- Local privacy: data contributors sanitize data before collection.
- Classical technique: *randomized response* [W65].
- Interactive variant can be minimax optimal [DJW13].
Input Perturbation

- Input perturbation: add noise to the input data.
- Advantages: easy to implement, results in reusable sanitized data set.

\[ \hat{W}_{\text{input}} \]

\[ (\varepsilon, \delta) \]

private database

DP preprocessing

sanitized dataset

non-private learning

[DJW13, FTS17]
Output Perturbation

- Compute the minimizer and add noise.
- Does not require re-engineering baseline algorithms

Noise depends on the sensitivity of the argmin.

[CMS11, RBHT12]
Objective Perturbation

private database $D$

non-private preprocessing

DP optimization $(\varepsilon, \delta)$

$\hat{W}_{\text{objective}}$

privacy barrier
Objective Perturbation

\[ J(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(w, (x_i, y_i)) + \lambda R(w) \]

A. Add a random term to the objective:

\[ \hat{w}_{\text{priv}} = \arg\min_w (J(w) + w^\top b) \]

B. Do a randomized approximation of the objective:

\[ \hat{w}_{\text{priv}} = \arg\min_w \hat{J}(w) \]

Randomness depends on the sensitivity properties of \( J(w) \).

[CMS11, ZZX+12]
Sensitivity of the argmin

\[ w^* = \arg \min_w J(w) \]
Sensitivity of the argmin

\[ w^* = \arg\min_w J(w) \]

• Non-private optimization solves \( \nabla J(w) = 0 \implies w^* \)
Sensitivity of the argmin

\[ \hat{w}_{\text{priv}} = \arg\min_w (J(w) + w^\top b) \]

- Non-private optimization solves \( \nabla J(w) = 0 \implies w^* \)
- Generate vector analogue of Laplace: \( b \sim p(z) \propto e^{-\epsilon/2\|z\|} \)
Sensitivity of the argmin

\[ \hat{w}_{\text{priv}} = \arg\min_w (J(w) + w^\top b) \]

- Non-private optimization solves \( \nabla J(w) = 0 \implies w^* \)
- Generate vector analogue of Laplace: \( b \sim p(z) \propto e^{-\varepsilon/2\|z\|} \)
- Private optimization solves \( \nabla J(w) = -b \implies w_{\text{priv}} \)
Sensitivity of the argmin

\[ \hat{w}_{\text{priv}} = \arg\min_w (J(w) + w^\top b) \]

• Non-private optimization solves \( \nabla J(w) = 0 \implies w^* \)

• Generate vector analogue of Laplace: \( b \sim p(z) \propto e^{-\varepsilon/2\|z\|} \)

• Private optimization solves \( \nabla J(w) = -b \implies w_{\text{priv}} \)

• Have to bound the sensitivity of the gradient.
Theoretical bounds on excess risk

Same important parameters:

• privacy parameters \((\varepsilon, \delta)\)

• data dimension \(d\)

• data bounds \(\|x_i\| \leq B\)

• analytical properties of the loss (Lipschitz, smoothness)

• regularization parameter \(\lambda\)
Theoretical bounds on excess risk

$$\tilde{O} \left( \frac{\sqrt{d \log(1/\delta)}}{n\varepsilon} \right)$$

(input perturbation)

Same important parameters:

- privacy parameters \((\varepsilon, \delta)\)
- data dimension \(d\)
- data bounds \(\|x_i\| \leq B\)
- analytical properties of the loss (Lipschitz, smoothness)
- regularization parameter \(\lambda\)

(quadratic loss)

[FTS17]
Theoretical bounds on excess risk

\[ \tilde{O}\left(\frac{\sqrt{d \log(1/\delta)}}{n\varepsilon}\right) \]

\[ \tilde{O}\left(\frac{d}{n\varepsilon^{3/2}}\right) \]

output perturbation

Same important parameters:

- privacy parameters \((\varepsilon, \delta)\)
- data dimension \(d\)
- data bounds \(\|x_i\| \leq B\)
- analytical properties of the loss (Lipschitz, smoothness)
- regularization parameter \(\lambda\)

[CMS11, BST14]
Theoretical bounds on excess risk

Same important parameters:

- privacy parameters \((\varepsilon, \delta)\)
- data dimension \(d\)
- data bounds \(\|x_i\| \leq B\)
- analytical properties of the loss (Lipschitz, smoothness)
- regularization parameter \(\lambda\)

Objective perturbation

\[
\tilde{O}\left(\frac{\sqrt{d \log(1/\delta)}}{n\varepsilon}\right)
\]

\[
\tilde{O}\left(\frac{d}{n\varepsilon}\right)
\]

[CMS11, BST14]
Typical empirical results

In general:

- **Objective perturbation empirically outperforms output perturbation.**
- **Gaussian mechanism with** $(\varepsilon, \delta)$ **guarantees outperform Laplace-like mechanisms with** $\varepsilon$-guarantees.
- **Loss vs. non-private methods is very dataset-dependent.**
Gaps between theory and practice

- Theoretical analysis is for fixed privacy parameters – how should we choose them in practice?
- Given a data set, can I tell what the privacy-utility-sample-size tradeoff is?
- What about more general optimization problems/algorithms?
- What about scaling (computationally) to large data sets?
Summary

• Training does not on its own guarantee privacy.

• There are many ways to incorporate DP into prediction and learning using ERM with different privacy-accuracy-sample size tradeoffs.

• Good DP algorithms should generalize since they learn about populations, not individuals.

• Theory and experiment show that \((\varepsilon, \delta)\)-DP algorithms have better accuracy than \(\varepsilon\)-DP algorithms at the cost of a weaker privacy guarantee.
Break
Differential privacy and optimization algorithms
Scaling up private optimization
Scaling up private optimization

- Large data sets are challenging for optimization:
  ⇒ batch methods not feasible
Scaling up private optimization

• Large data sets are challenging for optimization:
  ⇒ batch methods not feasible

• Using more data can help our tradeoffs look better:
  ⇒ better privacy and accuracy
Scaling up private optimization

• Large data sets are challenging for optimization:  
  ⇒ batch methods not feasible

• Using more data can help our tradeoffs look better:  
  ⇒ better privacy and accuracy

• Online learning involves multiple releases:  
  ⇒ potential for more privacy loss
Scaling up private optimization

• Large data sets are challenging for optimization:
  $\Rightarrow$ batch methods not feasible

• Using more data can help our tradeoffs look better:
  $\Rightarrow$ better privacy and accuracy

• Online learning involves multiple releases:
  $\Rightarrow$ potential for more privacy loss

Goal: guarantee privacy using the optimization algorithm.
Stochastic Gradient Descent

- Stochastic gradient descent (SGD) is a moderately popular method for optimization

- Stochastic gradients are random
  ⇒ already noisy ⇒ already private?

- Optimization is iterative
  ⇒ intermediate results leak information
Non-private SGD

\[ J(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(w, (x_i, y_i)) + \lambda R(w) \]

\[ w_0 = 0 \]

For \( t = 1, 2, \ldots, T \)

\[ i_t \sim \text{Unif}\{1, 2, \ldots, n\} \]

\[ g_t = \nabla \ell(w_{t-1}, (x_{i_t}, y_{i_t})) + \lambda \nabla R(w_{t-1}) \]

\[ w_t = \Pi_{W}(w_{t-1} - \eta_t g_t) \]

\[ \hat{w} = w_T \]
Private SGD with noise

\[ J(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(w, (x_i, y_i)) + \lambda R(w) \]

\[ w_0 = 0 \]

For \( t = 1, 2, \ldots, T \)

\[ i_t \sim \text{Unif}\{1, 2, \ldots, n\} \]

\[ z_t \sim p(\varepsilon, \delta)(z) \]

\[ \hat{g}_t = z_t + \nabla \ell(w_{t-1}, (x_{i_t}, y_{i_t})) + \lambda \nabla R(w_{t-1}) \]

\[ w_t = \Pi_{\mathcal{W}}(w_{t-1} - \eta_t \hat{g}_t) \]

\[ \hat{w} = w_T \]

- select random data point
- add noise to gradient

[SSC15]
Choosing a noise distribution

“Laplace” mechanism

\[ p(z) \propto e^{(\varepsilon/2)\|z\|} \]

\( \varepsilon \) — DP

Gaussian mechanism

\[ p(z) \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \tilde{O}\left(\frac{\log(1/\delta)}{\varepsilon^2}\right)\right) \]

\((\varepsilon, \delta)\) — DP

- Have to choose noise according to the sensitivity of the gradient:

\[ \max_{D, D'} \max_{w} \| \nabla J(w; D) - \nabla J(w; D') \| \]

- Sensitivity depends on the data distribution, Lipschitz parameter of the loss, etc.
Private SGD with randomized selection

\[
J(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(w, (x_i, y_i)) + \lambda R(w)
\]

\[
\begin{align*}
\mathbf{w}_0 &= \mathbf{0} \\
\text{For } t = 1, 2, \ldots, T \\
&\quad i_t \sim \text{Unif}\{1, 2, \ldots, n\} \\
&\quad g_t = \nabla \ell(w_{t-1}, (x_{i_t}, y_{i_t})) + \lambda \nabla R(w_{t-1}) \\
&\quad \hat{g}_t \sim p(\epsilon, \delta), g(z) \\
&\quad w_t = \Pi_{\mathcal{W}}(w_{t-1} - \eta_t \hat{g}_t) \\
\hat{w} &= w_T
\end{align*}
\]

- select random data point
- randomly select unbiased gradient estimate
Randomized directions

$= \frac{L}{\|g\|} g \text{ w.p. } \frac{1}{2} + \frac{\|g\|}{2L}$

Select hemisphere in direction of gradient or opposite. Pick uniformly from the hemisphere and take a step

- Need to have control of gradient norms: $\|g\| \leq L$
- Keep some probability of going in the wrong direction.
Why does DP-SGD work?

Noisy Gradient

Choose noise distribution using the sensitivity of the gradient.

Random Gradient

Randomly select direction biased towards the true gradient.

Both methods

• Guarantee DP at each iteration.
• Ensure unbiased estimate of $g$ to guarantee convergence.
Making DP-SGD more practical

“SGD is robust to noise”

- True up to a point — for small epsilon (more privacy), the gradients can become too noisy.

- **Solution 1**: more iterations ([BST14]: need $O(n^2)$)

- **Solution 2**: use standard tricks: mini-batching, etc. [SSC13]

- **Solution 3**: use better analysis to show the privacy loss is not so bad [BST14][ACG+16]
Randomly sampling data can amplify privacy

\[ \text{Private data set } D \rightarrow \text{Differentially private algorithm} \]

\[ \epsilon \]

[BBKN14, BST14]
Randomly sampling data can amplify privacy

- Suppose we have an algorithm $A$ which is $\varepsilon$-differentially private for $\varepsilon \leq 1$.

[BBKN14,BST14]
Randomly sampling data can amplify privacy

- Suppose we have an algorithm $A$ which is $\varepsilon$-differentially private for $\varepsilon \leq 1$.

- Sample $\gamma n$ entries of $D$ uniformly at random and run $A$ on those.

[BBKN14,BST14]
Randomly sampling data can amplify privacy

- Suppose we have an algorithm A which is $\varepsilon$-differentially private for $\varepsilon \leq 1$.
- Sample $\gamma n$ entries of $D$ uniformly at random and run A on those.
- Randomized method guarantees $2\gamma \varepsilon$-differential privacy.

[BBKN14,BST14]
Summary

- Stochastic gradient descent can be made differentially private in several ways by randomizing the gradient.

- Keeping gradient estimates unbiased will help ensure convergence.

- Standard approach for variance reduction/stability (such as minibatching) can help with performance.

- Random subsampling of the data can amplify privacy guarantees.
Accounting for total privacy risk
Measuring total privacy loss

Post processing invariance: risk doesn’t increase if you don’t touch the data again

- more complex algorithms have multiple stages
  ⇒ all stages have to guarantee DP

- need a way to do privacy accounting: what is lost over time/multiple queries?
A simple example

Private data set $\mathcal{D}$

Preprocessing $(\varepsilon_1, \delta_1)$

Training $(\varepsilon_2, \delta_2)$

Cross-validation $(\varepsilon_3, \delta_3)$

Testing $(\varepsilon_4, \delta_4)$
Composition property of differential privacy

Basic composition: privacy loss is additive:

- Apply R algorithms with \((\epsilon_i, \delta_i) : i = 1, 2, \ldots, R\)
- Total privacy loss:
  \[
  \left( \sum_{i=1}^{R} \epsilon_i, \sum_{i=1}^{R} \delta_i \right)
  \]
- Worst-case analysis: each result exposes the worst privacy risk.
What composition says about multi-stage methods

Total privacy loss is the sum of the privacy losses…
An open question: privacy allocation across stages

Compositions means we have a *privacy budget*.

How should we allocate privacy risk across different stages of a pipeline?

- Noisy features + accurate training?
- Clean features + sloppy training?

It’s application dependent! Still an open question…
A closer look at $\varepsilon$ and $\delta$

Gaussian noise of a given variance produces a spectrum of $(\varepsilon, \delta)$ guarantees:
A closer look at $\varepsilon$ and $\delta$

Gaussian noise of a given variance produces a spectrum of $(\varepsilon, \delta)$ guarantees:

$$p(\tau_1 | D) \leq \exp(\varepsilon_1) \frac{p(\tau_1 | D')}{p(\tau_1 | D')}$$
A closer look at $\varepsilon$ and $\delta$

Gaussian noise of a given variance produces a spectrum of $(\varepsilon, \delta)$ guarantees:

$$p(\tau_2 | D) \leq \exp(\varepsilon_2)$$

$${p(\tau_2 | D) \over p(\tau_2 | D')} \leq \exp(\varepsilon_2)$$
Privacy loss as a random variable

Actual privacy loss is a random variable that depends on $D$:

$$Z_{D, D'} = \log \frac{p(A(D) = t)}{p(A(D') = t)} \text{ w.p. } p(A(D) = t)$$

Spectrum of $(\varepsilon, \delta)$ guarantees means we can trade off $\varepsilon$ and $\delta$ when analyzing a particular mechanism.
Random privacy loss

\[ Z_{D,D'} = \log \frac{p(A(D) = t)}{p(A(D') = t)} \quad \text{w.p.} \quad p(A(D) = t) \]

- Bounding the max loss over (\(D,D'\)) is still a random variable.

- Sequentially *computing* functions on private data is like sequentially *sampling* independent privacy losses.

- Concentration of measure shows that the loss is much closer to its expectation.
Strong composition bounds

\[\sum_{\ell=0}^{i-1} \binom{k}{\ell} (e^{(k-\ell)\varepsilon} - e^{(k-2i+\ell)\varepsilon}) \left(1 + e^{\varepsilon}\right)^k\]

- Given only the \((\varepsilon, \delta)\) guarantees for \(k\) algorithms operating on the data.

- Composition again gives a family of \((\varepsilon, \delta)\) tradeoffs: can quantify privacy loss by choosing any valid \((\varepsilon, \delta)\) pair.

[DRV10, KOV15]
Moments accountant

**Basic Idea:** Directly calculate parameters \((\varepsilon, \delta)\) from composing a sequence of mechanisms.

More efficient than composition theorems.

[ACG+16]
### How to Compose Directly?

Given datasets $D$ and $D'$ with one different record, mechanism $A$, define privacy loss random variable as:

$$Z_{D,D'} = \log \frac{p(A(D) = t)}{p(A(D') = t)}, \quad \text{w.p. } p(A(D) = t)$$

- **Properties of $Z_{D,D'}$ related to privacy loss of $A$**

$$-$$

- If max absolute value of $Z_{D,D'}$ over all $D, D'$ is $\varepsilon$, then $A$ is $(\varepsilon,0)$-differentially private
How to Compose Directly?

Given datasets $D$ and $D'$ with one different record, mechanism $A$, define privacy loss random variable as:

$$Z_{D,D'} = \log \frac{p(A(D) = t)}{p(A(D') = t)}, \quad \text{w.p.} \ p(A(D) = t)$$

Challenge: To reason about the worst case over all $D, D'$

Key idea in [ACG+16]: Use moment generating functions
Accounting for Moments...

Three Steps:

1. Calculate moment generating functions for $A_1, A_2, ..$
2. Compose
3. Calculate final privacy parameters

$D$ [ACG+16]
I. Stepwise Moments

Define: Stepwise Moment at time $t$ of $A_t$ at any $s$:

$$\alpha_{A_t}(s) = \sup_{D,D'} \log \mathbb{E}[e^{sZ_{D,D'}}]$$

(D and $D'$ differ by one record)

[ACG+16]
2. Compose

Theorem: Suppose $A = (A_1, \ldots, A_T)$. For any $s$:

$$\alpha_A(s) \leq \sum_{t=1}^{T} \alpha_{A_t}(s)$$

[ACG+16]
3. Final Calculation

Theorem: For any \( \varepsilon \), mechanism \( A \) is \((\varepsilon, \delta)\)-DP for

\[
\delta = \min_s \exp(\alpha_A(s) - s\varepsilon)
\]

Use theorem to find best \( \varepsilon \) for a given \( \delta \) from closed form

or by searching over \( s_1, s_2, .., s_k \)

[ACG+16]
Example: composing Gaussian mechanism

Suppose $A_t$ answers a query with global sensitivity 1 by adding $N(0, 1)$ noise
I. Stepwise Moments

Suppose $A_t$ answers a query with global sensitivity 1 by adding $N(0, 1)$ noise.

Simple algebra gives for any $s$: $\alpha_{A_t}(s) = \frac{s(s + 1)}{2}$
2. Compose

Suppose $A_t$ answers a query with global sensitivity 1 by adding $N(0, 1)$ noise

$$\alpha_A(s) \leq \sum_{t=1}^{T} \alpha_{A_t}(s) = \frac{T \cdot s \cdot (s + 1)}{2}$$
3. Final Calculation

Find lowest $\delta$ for a given $\epsilon$ (or vice versa) by solving:

$$\tilde{\delta} = \min_s \exp\left(\frac{T s(s + 1)}{2} - s \epsilon\right)$$

In this case, solution can be found in closed form.
How does it compare?

[DRV10] (better than linear)

Moments
Accountant

[ACG+16]
How does it compare on real data?

EM for MOG with Gaussian Mechanism

\[ \delta = 10^{-4} \]

[1] PFCW17
Summary

• Practical machine learning looks at the data many times.

• Post-processing invariance means we just have to track the cumulative privacy loss.

• Good composition methods use the fact that the actual privacy loss may behave much better than the worst-case bound.

• The Moments Accountant method tracks the actual privacy loss more accurately: better analysis for better privacy guarantees.
Applications to modern machine learning
When is differential privacy practical?

Differential privacy is best suited for understanding population-level statistics and structure:

• Inferences about the population should not depend strongly on individuals.

• Large sample sizes usually mean lower sensitivity and less noise.

To build and analyze systems we have to leverage post-processing invariance and composition properties.
Differential privacy in practice

Google: RAPPOR for tracking statistics in Chrome.

Apple: various iPhone usage statistics.

Census: 2020 US Census will use differential privacy.

mostly focused on count and average statistics
Challenges for machine learning applications

Differentially private ML is complicated because real ML algorithms are complicated:

• Multi-stage pipelines, parameter tuning, etc.

• Need to “play around” with the data before committing to a particular pipeline/algorithm.

• “Modern” ML approaches (= deep learning) have many parameters and less theoretical guidance.
For today, we will describe some recent examples:

1. Differentially private deep learning [ACG+16]
2. Differential privacy and Bayesian inference
Differential privacy and deep learning

Main idea: train a deep network using differentially private SGD and use moments accountant to track privacy loss.

Additional components: gradient clipping, minibatching, data augmentation, etc.

[ACG+16]
Overview of the algorithm

Private data set $D$

select random lot $L_1, L_2, \ldots$

compute gradients $x_{L_1}, x_{L_2}, \ldots$

clip and add noise

new network weights $\theta_T$

update parameters

moments accountant $(\varepsilon, \delta)$

privacy barrier

[ACG+16]
Effectiveness of DP deep learning

Empirical results on MNIST and CIFAR:

• Training and test error come close to baseline non-private deep learning methods.

• To get moderate loss in performance, epsilon and delta are not “negligible”

[ACG+16]
Effectiveness of DP deep learning

Empirical results on MNIST and CIFAR:

- Training and test error come close to baseline non-private deep learning methods.

- To get moderate loss in performance, epsilon and delta are not “negligible”

[ACG+16]
Moving forward in deep learning

This is a good proof of concept for differential privacy for deep neural networks. There are lots of interesting ways to expand this.

• Just used one NN model: what about other architectures? RNNs? GANs?

• Can regularization methods for deep learning (e.g. dropout) help with privacy?

• What are good rules of thumb for lot/batch size, learning rate, # of hidden units, etc?

[ACG+16]
Differentially Private Bayesian Inference

Data $X = \{ x_1, x_2, \ldots \}$
Model Class $\Theta$

$\{\}$ Related through likelihood $p(x|\theta)$

Prior $\pi(\theta)$
Data $X$

$=$

Posterior $p(\theta|X)$

Find differentially private approx to posterior
Differentially Private Bayesian Inference

- General methods for private posterior approximation
- A Special Case: Exponential Families
- Variational Inference
How to make posterior private?

Option 1: Direct posterior sampling [DMNR14]

Not differentially private except under restrictive conditions - likelihood ratios may be unbounded!

[GSC17] Answer changes under a new relaxation Rényi differential privacy [M17]
How to make posterior private?

Option 2: One Posterior Sample (OPS) Method [WFS15]

1. Truncate posterior so that likelihood ratio is bounded in the truncated region.

2. Raise truncated posterior to a higher temperature
How to make posterior private?

Option 2: One Posterior Sample (OPS) Method:

Advantage: General

Pitfalls:
— Intractable - only exact distribution private
— Low statistical efficiency even for large $n$
How to make posterior private?

Option 3: Approximate the OPS distribution via Stochastic Gradient MCMC [WFS15]

**Advantage:** Noise added during stochastic gradient MCMC contributes to privacy

**Disadvantage:** Statistical efficiency lower than exact OPS
Differentially Private Bayesian Inference

- General methods for private posterior approximation
- A Special Case: Exponential Families
- Variational Inference
(Non-private) posterior comes from exp. family:

\[ p(\theta | x) \propto e^{\eta(\theta) ^\top ( \sum_i T(x_i)) - B(\theta)} \]

given data \( x_1, x_2, \ldots \)

Posterior depends on data through sufficient statistic \( T \)
Exponential Family Posteriors

(Non-private) posterior comes from exp. family:

\[ p(\theta|x) \propto e^{\eta(\theta)^\top (\sum_i T(x_i)) - B(\theta)} \]

given data \( x_1, x_2, \ldots \), sufficient statistic \( T \)

Private Sampling:

1. If \( T \) is bounded, add noise to \( \sum_i T(x_i) \) to get private version \( T' \)

2. Sample from the perturbed posterior:

\[ p(\theta|x) \propto e^{\eta(\theta)^\top T' - B(\theta)} \]

[ZRDI6, FGWC16]
How well does it work?

Statistically efficient

Performance worse than non-private, better than OPS

Can do inference in relatively complex systems by building up on this method — eg, time series clustering in HMMs
Differentially Private Bayesian Inference

- General methods for private posterior approximation
- A Special Case: Exponential Families
- Variational Inference
Variational Inference

**Key Idea:** Start with a stochastic variational inference method, and make each step private by adding Laplace noise. Use moments accountant and subsampling to track privacy loss.

[JDH16, PFCW16]
Summary

- Two examples of differentially private complex machine learning algorithms
- Deep learning
- Bayesian inference
Summary, conclusions, and where to look next
Summary

1. Differential privacy: basic definitions and mechanisms

2. Differential privacy and statistical learning: ERM and SGD

3. Composition and tracking privacy loss

4. Applications of differential privacy in ML: deep learning and Bayesian methods
Things we didn’t cover…

• Synthetic data generation.

• Interactive data analysis.

• Statistical/estimation theory and fundamental limits

• Feature learning and dimensionality reduction

• Systems questions for large-scale deployment

• … and many others…
Where to learn more

Several video lectures and other more technical introductions available from the Simons Institute for the Theory of Computing:

https://simons.berkeley.edu/workshops/bigdata2013-4

Monograph by Dwork and Roth:

http://www.nowpublishers.com/article/Details/TCS-042
Final Takeaways

• Differential privacy measures the risk incurred by algorithms operating on private data.

• Commonly-used tools in machine learning can be made differentially private.

• Accounting for total privacy loss can enable more complex private algorithms.

• Still lots of work to be done in both theory and practice.
Thanks!