Safer Data Mining: Algorithmic Techniques in Differential Privacy

Speakers: Moritz Hardt and Sasho Nikolov
IBM Almaden and Rutgers
SDM 2014
Disclaimer

This talk is *not* a survey.
Disclaimer

This talk is *not* a history lesson.

---

Millions of years ago, giant marshmallows ruled the earth.

Having no skeletal system, they left no fossil-record.
Focus

A few widely applicable algorithmic techniques

Algorithms you can run on an actual computer
Outline

Part I: Differential Privacy:
Attacks, basic definitions & algorithms

Part II: Unsupervised Learning
Technique: Singular Value Decomposition

Part III: Supervised Learning
Technique: Convex optimization

Part IV: Streaming
Technique: Streaming data structures
Releasing Data: What could go wrong?

Wealth of examples

One of my favorites: Genome-Wide Association Studies (GWAS)

Trust me: You will like this even if you don’t like biology
GWAS

Typical Setup:

1. NIH takes DNA of 1000 test candidates with common disease
2. NIH releases minor allele frequencies (MAF) of test population at 100,000 positions (SNPs)

**Goal:** Find association between SNPs and disease
Attack on GWAS data [Homer et al.]

Can **infer membership in test group** of an individual with known DNA from published data!

<table>
<thead>
<tr>
<th>SNP</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>...</th>
<th>...</th>
<th>100000</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAF</td>
<td>0.02</td>
<td>0.03</td>
<td>0.05</td>
<td></td>
<td></td>
<td>0.02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SNP</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>...</th>
<th>...</th>
<th>100000</th>
</tr>
</thead>
<tbody>
<tr>
<td>MA</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td></td>
<td></td>
<td>YES</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SNP</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>...</th>
<th>...</th>
<th>100000</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAF</td>
<td>0.01</td>
<td>0.04</td>
<td>0.04</td>
<td></td>
<td></td>
<td>0.01</td>
</tr>
</tbody>
</table>

Test population

Moritz’s DNA

Reference population (HapMap data, public)
Can **infer membership in test group** of an individual with known DNA from published data!

<table>
<thead>
<tr>
<th>SNP</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>...</th>
<th>...</th>
<th>100000</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAF</td>
<td>0.02</td>
<td>0.03</td>
<td>0.05</td>
<td>...</td>
<td>...</td>
<td>0.02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SNP</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>...</th>
<th>...</th>
<th>100000</th>
</tr>
</thead>
<tbody>
<tr>
<td>MA</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>...</td>
<td>...</td>
<td>YES</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SNP</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>...</th>
<th>...</th>
<th>100000</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAF</td>
<td>0.01</td>
<td>0.04</td>
<td>0.04</td>
<td>...</td>
<td>...</td>
<td>0.01</td>
</tr>
</tbody>
</table>
Interesting characteristics

• Only **innocuous looking data** was released
  – Data was **HIPAA compliant**
• Data **curator is trusted** (NIH)
• Attack uses **background knowledge** (HapMap data set) available in public domain
• Attack uses **unanticipated algorithm**
• Curator pulled data sets (now hard to get)
How *not* to solve the problem

- **Ad-hoc anonymization** (e.g., replace name by random string)
  - Gone wrong: Netflix user data, AOL search logs, MA medical records, linkage attacks

- **Allow only innocuous looking statistics** (e.g. low-sensitivity counts)
  - Gone wrong: GWAS

- **Use the wrong crypto tool** (e.g., encrypt hard disk, securely evaluate query)
Differential Privacy
[Dwork-McSherry-Nissim-Smith-06]

Meaningful privacy guarantee
• handles attacks with background information
• powerful composition properties

Intuition: Presence or absence of any individual in the data cannot be inferred from output
Differential Privacy
[Dwork-McSherry-Nissim-Smith-06]

Two data sets \(D, D'\) are called *neighboring* if they differ in at most one data record.

Example: \(D = \{\text{GWAS test population}\}, \ D' = D - \{\text{Moritz's DNA}\}\)

**Informal Definition (Differential Privacy):**
A randomized algorithm \(M(D)\) is differentially private if for all neighboring data sets \(D, D'\) and all events \(S\):

\[
\Pr\{M(D) \in S\} \approx \Pr\{M(D') \in S\}
\]
Differential Privacy

Two data sets $D, D'$ are called *neighboring* if they differ in at most one data record.

Example: $D = \{\text{GWAS test population}\}, \ D' = D - \{\text{Moritz's DNA}\}$

**Definition (Differential Privacy):**
A randomized algorithm $M(D)$ is $\varepsilon$-differentially private if for all neighboring data sets $D, D'$ and all events $S$:

$$\Pr\{M(D) \in S\} \leq e^\varepsilon \Pr\{M(D') \in S\}$$

Think: $\varepsilon = 0.01$ and $e^\varepsilon \approx 1 + \varepsilon$
Differential Privacy

[Dwork-McSherry-Nissim-Smith-06]

Two data sets \(D,D'\) are called *neighboring* if they differ in at most one data record.

Example: \(D = \{\text{GWAS test population}\}, D' = D - \{\text{Moritz's DNA}\}\)

**Definition (Differential Privacy):**

A randomized algorithm \(M(D)\) is \((\epsilon, \delta)\)-differentially private if for all neighboring data sets \(D,D'\) and all events \(S\):

\[
\Pr\{M(D) \in S\} \leq e^\epsilon \Pr\{M(D') \in S\} + \delta
\]

Think: \(\epsilon = 0.01\) and \(e^\epsilon \approx 1 + \epsilon\)

\(\delta \ll 1/|D|\)
Definition (Differential Privacy):
A randomized algorithm \( M(D) \) is \( \epsilon \)-differentially private if for all neighboring data sets \( D, D' \) and all events \( S \):

\[
\Pr\{M(D) \in S\} \leq e^\epsilon \Pr\{M(D') \in S\}
\]
Definition (Differential Privacy):
A randomized algorithm \( M(D) \) is \((\varepsilon, \delta)\)-differentially private if for all neighboring data sets \( D, D' \) and all events \( S \):

\[
\Pr\{M(D) \in S\} \leq e^{\varepsilon} \Pr\{M(D') \in S\} + \delta
\]
Sensitivity

Assume $f$ maps databases to real vectors.

**Definition:**

- **$L_1$-sensitivity** $\Delta_1(f) = \max \| f(D) - f(D') \|_1$
  
  – maximum over all neighboring $D, D'$

- **$L_2$-sensitivity** $\Delta_2(f) = \max \| f(D) - f(D') \|_2$

**Intuition:** Low sensitivity implies “compatible with differential privacy”
Exercise

\[ f \text{ maps } D \text{ to } (q_1(D), q_2(D), \ldots, q_k(D)) \]

where each \( q_i(D) \) is a count:

“How many people satisfy predicate \( P_i \)?”

L1-sensitivity? \( k \)

L2-sensitivity? \( k^{1/2} \)
Laplacian Mechanism [DMNS’06]

Given function $f$:
1. Compute $f(D)$
2. Output $f(D) + \text{Lap}(\Delta_1(f)/\varepsilon)^d$

Fact 1: Satisfies $(\varepsilon,0)$-differential privacy

Fact 2: Expected error $\Delta_1(f)/\varepsilon$ in each coord.
Laplacian Mechanism [DMNS’06]

One-dimensional example

Suppose $D, D'$ neighboring

- $q(D) + \text{Lap}(1/\epsilon)$
- $q(D') + \text{Lap}(1/\epsilon)$

$\text{density } \exp(-\epsilon |x-q(D)|)$
$\text{density } \exp(-\epsilon |x-q(D')|)$
Gaussian Mechanism

Given function $f$:
1. Compute $f(D)$
2. Output $f(D) + N(0, \sigma^2)^d$ with $\sigma = \Delta_2(f) \log(1/\delta) / \varepsilon$

Scale noise to $L_2$-sensitivity

Fact: Satisfies $(\varepsilon, \delta)$-differential privacy

Fact: Expected error $\Delta_2(f) \log(1/\delta) / \varepsilon$
Differential privacy composes:
1. in parallel
2. sequentially
3. adaptively
$T$-fold \textbf{composition of} \(\varepsilon_0\)-differential privacy satisfies:

\[
\varepsilon_0 T\text{-differential privacy}
\]

**Answer 1**

[DMNS’06]:

**Answer 2**

[DRV’10]:

\[(\varepsilon,\delta)\text{-differential privacy}
\]

\[
\varepsilon \leq 4\varepsilon_0 \sqrt{T \log(1/\delta)}
\]

\textit{Note: for small enough }\varepsilon
Part II: Private SVD and Applications
We meet the protagonist

$A$ is a real $n \times d$ matrix

thought of as $n$ data points in $d$ dimensions
...and its singular value decomposition

\[ A = U \Sigma V^T \]

\( U, V \) have orthonormal columns
\( \Sigma \) is diagonal \( r \times r \), \( r = \text{rank}(A) \)
The left factor

Columns $u_1,...,u_r$ are the singular vectors of $A$
$\iff$ eigenvectors of the $n \times n$ Gram matrix $AA^T$

By Courant-Fisher:

$\|u_1^TA\| = \max \left\{ \|x^TA\| : \|x\| = 1 \right\}$

$\|u_2^TA\| = \max \left\{ \|x^TA\| : \|x\| = 1, \langle x, u_1 \rangle = 0 \right\}$

...
Columns $v_1, \ldots, v_r$ of $V$ are the eigenvectors of the $d \times d$ covariance matrix $A^T A$

Diagonal entries $\sigma_1, \ldots, \sigma_r$ square roots of the nonzero eigenvalues of $A^T A$

Notation: $\sigma_i(A) = \sigma_i$

Note: Singular values are unique, singular vectors are not (e.g. identity matrix)
Why compute the SVD?

**Principal Component Analysis**
right singular vectors
= principal components

**Low Rank Approximation**
truncated SVD gives best low rank approximation in Frobenius and spectral norm

**Many more:** Spectral clustering, collaborative filtering, topic modeling
Differential Privacy on Matrices

How should we define “neighboring”? 

- Differ in one row
- Differ in one row by norm at most 1
- Differ in at most one entry by at most 1

Stronger privacy \rightarrow \text{Smaller error}
Differential Privacy on Matrices

How should we define “neighboring”? 

Differ in at most one entry by at most 1

• Focus in this talk, others are reasonable
• Reasonable for sparse data with bounded coeffs
• Many entries? Choose smaller $\varepsilon$.
• Algorithms we’ll see also give other guarantees using different noise scaling.
Objectives

• Approximating the top singular vector
  – Objective:
    \[
    \max_{x: \|x\|=1} \|Ax\| \quad \text{OPT} = \sigma_1(A)
    \]

• Approximating \(k\) singular vectors
  – Objective:
    \[
    \min_{B: \text{rk}(B)=k} \|A - B\|_2 \quad \text{OPT} = \sigma_{k+1}(A)
    \]

Goal: Minimize additive error subject to differential privacy
RANDOMIZED RESPONSE: A SURVEY TECHNIQUE FOR ELIMINATING EVASIVE ANSWER BIAS

STANLEY L. WARNER
Claremont Graduate School

For various reasons individuals in a sample survey may prefer not to confide to the interviewer the correct answers to certain questions. In such cases the individuals may elect not to reply at all or to reply with incorrect answers. The resulting evasive answer bias is ordinarily difficult to assess. In this paper it is argued that such bias is potentially removable through allowing the interviewee to maintain privacy through the device of randomizing his response. A randomized response method for estimating a population proportion is presented as an example. Unbiased maximum likelihood estimates are obtained and their mean square errors are compared with the mean square errors of conventional estimates under various assumptions about the underlying population.

1. INTRODUCTION

For reasons of modesty, fear of being thought bigoted, or merely a reluctance to confide secrets to strangers, many individuals attempt to evade certain questions put to them by interviewers. In survey vernacular, these people become the "non-cooperative" group [5, pp. 235–72], either refusing outright to be surveyed, or consenting to be surveyed but purposely providing wrong answers to the questions. In the one case there is the problem of refusal bias [1, pp. 355–61], [2, pp. 33–6], [5, pp. 261–9]; in the other case there is the
Input Perturbation [Warner65, BlumDworkMcSherryNissim05] aka Randomized Response (RR)

**Randomized Response (Input A):**

1. \( N = \text{Lap}(1/\varepsilon)^{n \times d} \)
2. \( B = A + N \)
3. Output \( \text{SVD}(B) \)

Fact: Satisfies \( \varepsilon \)-differential privacy.

How much error does this introduce?
Some matrix theory

**Theorem** (Weyl): Let $A, N$ be $n \times d$ matrices. Then,

$$|\sigma_i(A + N) - \sigma_i(A)| \leq \|N\|_2$$

**Theorem**: Let $N = \text{Lap}(1/\epsilon)^{n \times d}$. Then,

$$\mathbb{E}\|N\|_2 \leq O\left(\frac{\sqrt{n} + \sqrt{d}}{\epsilon}\right)$$
When does RR Work?

n >> d, d relatively small

Run on d x d covariance matrix

Example:
Differentially private recommender system evaluated on Netflix data set [McSherryMironov09]
Here, n = 480189 and d = 17770
When RR doesn’t work

**Problem 1:**
\[ n \text{ and } d \text{ both large} \]

RR generates huge dense noise matrix

**Problem 2:**
\[ \text{matrix sparse, e.g, } \Delta \text{ ones per row} \]

\[ \sigma_1(A) \leq \sqrt{n\Delta} \]

Can we improve on RR?

No! But we’ll do it anyway.
Why not? Dictator example

\[
\begin{array}{c}
\text{n} \\
\begin{array}{c}
111100011111\ldots111110011111 \\
000000000000\ldots000000000000 \\
000000000000\ldots000000000000 \\
000000000000\ldots000000000000 \\
000000000000\ldots000000000000 \\
\ldots \\
000000000000\ldots000000000000 \\
\end{array}
\end{array}
\]

v_1 = top row (up to scaling)

Fact: Differential privacy requires error $\sqrt{n}$

Error $o(n^{1/2})$ definition of “blatant non-privacy”!

How are we going to get around this?
ZUSAMMENFASSENDE BERICHT

Praktische Verfahren der Gleichungsauflösung.


In einer Vorlesung über „Praktische Analysis“, die der erstgenannte Verfasser im Sommer-Semester 1927 hielt, kamen eine Reihe von Rechnungsverfahren zur Sprache, die teils neu, teils wenig bekannt sind. Es erschien zweckmäßig, einiges davon mit kurzer Begründung und Hinweisen auf die wichtigsten Anwendungen zusammenzustellen. Dieser Arbeit hat sich der zweitgenannte Verfasser – zunächst hinsichtlich der Methoden der Gleichungsauflösung – unterzogen; von ihm rühren auch wesentliche Ergänzungen und nähere Ausführungen sowie die Zahlenbeispiele her.


From here on

Assume $A$ is symmetric and $n \times n$

Wlog, consider:

$$
\begin{bmatrix}
0 & A^T \\
A & 0
\end{bmatrix}
$$
Recap: Power Method

**Input:** \( n \times n \) matrix \( A \), parameter \( T \)

Pick random unit vector \( x_0 \)

For \( t = 1 \) to \( T \):

\[
\begin{align*}
y_t &= Ax_{t-1} \\
x_t &= \frac{y_t}{\|y_t\|_2}
\end{align*}
\]

**Output** \( x_T \)

**Fact:** \( x_T \) converges to top SV

for now all we care about!
Senstivity of Matrix-Vector Product

- Suppose $A, A'$ differ in one entry by 1
- Assume $x$ is a unit vector

**Fact:**

$$|| (A-A')x ||_2 \leq || x ||_\infty$$

where $$|| x ||_\infty = \max_i |x_i|$$
**Noisy Power Method**

**Input:** $n \times n$ matrix $A$, parameters $T, \varepsilon, \delta > 0$

Pick random unit vector $x_0$

For $t = 1$ to $T$:

$$y_t = A_{x_{t-1}} + g_t, \quad g_t = \mathcal{N}(0, T \log(1/\delta)/\varepsilon^2 |x_{t-1}|_\infty^2)$$

$$x_t = \frac{y_t}{|y_t|_2}$$

Output $x_T$

**Fact:** Satisfies $(\varepsilon, \delta)$-differential privacy
Bounding the largest entry of $x_t$

Let $v_1,\ldots,v_n$ be singular vectors of $A$

Put $B = \max_{i=1}^n \|v_i\|_\infty$

**Easy bound:** $\|x_t\|_\infty \leq \sqrt{nB}$

Can we do better?

**Lemma [H-Roth13]:**

$$\Pr \left\{ \|x_t\|_\infty \geq 4B \sqrt{\log n} \right\} \leq 1/n^3$$
A pleasant surprise

\[ \mu(A) = n \cdot \max_{i=1}^{r} \|v_i\|_\infty^2 \]

known as coherence of A and widely studied

Theoretically and empirically often small

polylog(n) in random models [CandesTao09]

much less than n for real world data

Previous lemma: \[ \|x_t\|_\infty \lesssim \sqrt{\frac{\mu(A)}{n}} \]

and hence \[ \|g_t\| \lesssim \sqrt{\mu(A)} \]
Performance of Power Method

**Theorem** [H-Roth’13]:
The Noisy Power Method satisfies \((\varepsilon, \delta)\)-differential privacy and with \(T = O(\log(n))\) steps returns a unit vector \(x\) such that

\[
\|Ax\| \geq \sigma_1(A) - O\left(\varepsilon^{-1} \sqrt{\mu(A)} \log(1/\delta) \log n\right).
\]

provided \(A\) satisfies “singular value separation.”

**Theorem:** Nearly matching lower bound for every setting of \(\mu(A)\).

Contrast with \(O\left(\sqrt{n}\right)\) for Randomized Response even if \(\mu(A) = 1\).
When **utility** and **privacy** benefit from the same principle

- **Privacy-preserving PCA**
  - Improve RR using incoherence
  - [H-Roth12,13]

- **Robust PCA**
  - Cope with corrupted entries
  - [CandesLiMaWright09]

- **Matrix Completion**
  - Recover missing entries
  - [CandesTao09,CandesRecht]

- **Netflix Prize**
  - Partial rating matrix released.
  - **Competition:** Improve recommendation system by 10%

- **Privacy Outcry**
  - Users re-identified
  - [NarayananShamatikov08]

- **Differentially Private Recommender System**
  - Building on Randomized Response
  - [McSherryMironov09]

- **Other approaches**
  - [Chaudhuri-Sarwate-Sinha12, Kapralov-Talwar13]
Generalization: Subspace Iteration

**Input:** \( n \times n \) matrix \( A \) symmetric, target rank \( k \), \( X_0 \) random orthonormal matrix

For \( t = 1 \) to \( T \):
- Pick Gaussian perturbation \( G_t \)
- \( Y_t = AX_{t-1} + G_t \)
- \( X_t = \text{Orthonormalize}(Y_t) \)

**Output** \( X_T \) (approx top \( k \) singular vectors)
Principle Angle Between Subspaces

Let $U, X$ subspaces of dimension $k$

$k=1 \quad \cos \Theta(U,X) = |U^TX|$

In general \( \cos \Theta(U,X) = \sigma_{\text{min}}(U^TX) \)

\( \sin \Theta(U,X) = \sigma_{\text{max}}(V^TX) \)

where $V$ orthog. complement of $U$

\( \tan \Theta(U,X) = \sin \Theta(U,X) / \cos \Theta(U,X) \)
Main Convergence Lemma

Let $U$ be spanned by top $k$ singular vectors of $A$.

$$\tan \theta(U, X_t) \leq \frac{\sigma_{k+1} \sin \theta(U, X_{t-1})}{\sigma_k \cos \theta(U, X_{t-1})}$$

If $G_t = 0$
Application 1: Spectral Clustering

Planted multisection model:
- $c$ clusters, intra-cluster edge probability $p$
- inter-cluster probability $q, \quad q < p$

How to recover a cluster?
Simple approach: Cheeger Cut
1. Compute second eigenvector $v_2$ of the graph $G$
2. Sort coordinates in ascending order
3. Pick vertices corresponding to first $\frac{n}{c}$ coordinates
Application 1: Spectral Clustering

Graph with n= 20,000 vertices, p = 0.2

Open Problem:
Explore more sophisticated spectral clustering techniques.
## Application 2: Matrix approximation

Upper left 40 x 100 corner of a 1000 x 1000 matrix.
Differentially Private Approximation

One step Subspace (Non-)Iteration, noisy projection, rounding

< 1000 entries have changed out of 1,000,000
In this example: Matrix has rank 2 and coherence 2 making nearly exact recovery feasible even under differential privacy.
Enter Graph Cuts

• Given graph \( G=(V,E) \), cut query is a subset \( S \) of \( V \), answer is \( E_G(S,S^c) \)

**Goal:** Come up with weighted graph \( G' \) that satisfies differential privacy and approximates all cuts in \( G \).
Synthetic Data for Cuts: What’s known

Goal: Preserve all cuts of size $s$

Randomized Response

Error $O(sn^{1/2})$

Johnson-Lindenstrauss

[BlockiBlumDattaSheffet12]

Error $O(s^{1.5})$

Better for $s << n$

MW+EM (inefficient) gives error:

$O(\sqrt{ms})$
JL Approximation of the Laplacian

Graph Laplacian $L_G$  
Cut query: $\mathbb{1}_S^T L_G \mathbb{1}_S$

Fact: $L_G = E_G^T E_G$  
$E_G$ is the weighted $\binom{n}{2} \times n$ edge-vertex incidence matrix

Suppose we pick random Gaussian $r \times \binom{n}{2}$ matrix $M$ and put: $\tilde{L}_G = \frac{1}{r} E_G^T M^T M E_G$

By JL Theorem, $r = \alpha^{-2} \log(1/\beta)$ preserves single cut up to factor $(1 \pm \alpha)$ with pr $1-\beta$
Adding in Privacy [BBDM12]

\[ H = \frac{w}{n} K_n + \left(1 - \frac{w}{n}\right) G \]

**Theorem [BBDM]:**
For \( w = O\left(\epsilon^{-1} \sqrt{r \log(1/\delta) \log(r/\delta)}\right) \) the JL approximation of \( L_H \) satisfies \((\epsilon,\delta)\)-differential privacy.

Additive error on one cut of size \( s \): \( O(ws) \)
Geometric Intuition

Sample  \( g \sim N(0, 1)^{(n)} \)

**Fact:**  \( E_{H}^{T}g \) Gaussian variable with covariance matrix  \( L_{H} \)

Sparse cuts \( \Leftrightarrow \) directions of small variance

Mixing in \((w/n)K_{n}\) gives variance \( w^{2} \) in every unit direction (thus hiding single edge change in \( G \))
End of Part II

But wait, there’s more...
Part III: Supervised Learning
(Empirical Risk Minimization)
Supervised Learning: Classification

Fruit Classifier

Apple

Orange

http://capitaldisruptivo.files.wordpress.com/2012/04/apples-and-oranges1.jpg
Supervised Learning: Regression
Generalized Linear Model

- **Data** $X_n = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$
  - $(x_i, y_i)$ sampled IID from distribution $D$
  - *data point* $x_i$ is $d$-dimensional, *label* $y_i$ is real

- **Goal**: predict $y$ for new $x$, $w$

- **Hypothesis class**: $H$

- **Loss function**: $l(<w, x>; y)$
  - loss of hypothesis $w$ on $(x, y)$
  - assumption: $y$ can be predicted based on linear measurement of $x$
  - $l$ will be *convex* in $w$
Risk Minimization

- The Risk Minimization Problem:
  \[ w^* = \arg \min_w E(x, y) \sim_D [ l(<w, x>; y) ] \]
- How to do that based on \( X_n \)?
- Minimize empirical risk:

\[
R_n(w) = \frac{1}{n} \sum_i l(<w, x_i>; y_i)
\]

- *Uniform convergence*: if everything is “nice”, empirical risk minimizer \( w \rightarrow w^* \) as \( n \rightarrow \infty \)
Regularization

• Often there are many solutions to the ERM problem
  – many hypotheses fit the data
• Occam’s Razor: pick the “simplest” hypothesis
• Regularized ERM:
  \[ J_n(w) = R_n(w) + \lambda r(w) \]

• Regularizer \( r(w) \): “complexity” of \( w \)
Strong Convexity

• Need regularizer to be strongly convex:
  – unique optimal $w$
  – **robustness** to data perturbation
  – helps **privacy**: output does not depend on any data point too much

$$r(z) \geq r(w) + \langle \nabla r(w), z - w \rangle + \|z - w\|_2^2$$
Strong Convexity

\[ r(z) \]
\[ r(w) + \langle \nabla r(w), z - w \rangle + \|z - w\|^2 \]
Examples

- **SVM:**
  - $l(<w, x>; y) = \max\{0, 1-y<w, x>\}$;
  - $r(w) = 0.5 |w|^2$

- **Logistic Regression**
  - $l(<w, x>; y) = \log(1 - \exp(-y<w, x>))$;
  - $r(w) = 0.5 |w|^2$

- **Ridge Regression**
  - $l(<w, x>; y) = (y - <w, x>)^2$;
  - $r(w) = 0.5 |w|^2$
Private Algorithm: Output Perturbation

1. Compute minimizer $w$ of $J_n(w)$
2. Sample noise $b \sim N(0, c(\epsilon, \delta)/\lambda^2 n^2)^d$
3. Output $w^\sim = w + b$

• $(\epsilon, \delta)$-DP if:
  – $l$ is 1-Lipschitz:
    \[ |l(z, x; y) - l(w, x; y)| \leq |<z-w, x>| \]
  – $|x| \leq 1$

• [Chaudhuri, Monteleoni, Sarwate ‘11]
Sensitivity Analysis

- **Intuition**: strong convexity -> low sensitivity

  \[ X = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \]

  \[ X' = \{(x_1', y_1'), (x_2', y_2'), \ldots, (x'_n, y'_n)\} \]

  \[ J_n(w): \text{risk for } X; \quad J'_n(w): \text{risk for } X' \]

  \[ w: \text{minimizer of } J_n(w); \quad w': \text{minimizer of } J'_n(w); \]

  **Sensitivity**: \[ |w - w'| \leq 2/(\lambda n) \]

  Privacy follows as usual.
Sensitivity Analysis Sketch

• Optimality of $w, w'$ and strong convexity:

\[
J_n'(w) \geq J_n'(w') + \frac{\lambda}{2} \|w - w'\|^2
\]

\[
J_n(w') \geq J_n(w) + \frac{\lambda}{2} \|w - w'\|^2
\]

• Only one data point changed:

\[
J_n(w) - J_n'(w) = \frac{1}{n} (\ell(\langle w, x \rangle; y) - \ell(\langle w, x' \rangle; y'))
\]

• Combine + Lipschitz-ness:

\[
\lambda \|w - w'\|^2 \leq \frac{1}{n} \langle w - w', x \rangle + \frac{1}{n} \langle w - w', x' \rangle \leq \frac{2}{n} \|w - w'\|
\]
Generalization Error

• Expected risk relates to the optimal risk as:

\[ E[\ell(\langle w^\sim, x \rangle; y) - \ell(\langle w^*, x \rangle; y)] \leq \frac{c'(\varepsilon, \delta)}{\sqrt{n}} \]

when \( \lambda = n^{-1/2} \).

• Proof idea [Jain, Thakurta ‘13]
  – Lipschitzness:
    \[ |\ell(\langle w^\sim, x \rangle; y) - \ell(\langle w, x \rangle; y)| \leq \langle b, x \rangle \]
  – \( \langle b, x \rangle \) is Gaussian with variance \( c(\varepsilon, \delta)/(\lambda^2 n^2) \)
  – bounds \( J_n(w^\sim) - J_n(w) \). Suffices by convergence results.
Variants and Extensions

• **Objective Perturbation** [Chaudhuri, Monteleoni, Sarwate ‘11], [Kifer, Smith, Thakurta ‘12], [Jain, Thakurta ’13]
  – minimize $J_n(w) + <b, w>$ for random Gaussian $b$
  – improved guarantees for “nice” data (adapts to convexity of the instance)

• Analysis can be extended to:
  – ERM with more general loss function
  – Structural constraints (sparse regression)
  – But usually a dependence on $d$ creeps in
Other Approaches

• Exploiting robustness
  – Private algorithms from learning algorithms robust to perturbations of the input
    – [Smith, Thakurta ‘13] Private Lasso with optimal sampling complexity

• Online learning: data arrives online, minimize regret
  – [Jain, Kothari, Thakurta ‘12] Sensitivity analysis
  – [Smith, Thakurta ‘13] *Follow the approximate leader*
Part IV: Streaming Models
The Streaming Model

1, 4, 5, 19, 145, 14, 5, 5, 16, 4
+, -, +, -, +, +, -, +, -, +

• Underlying frequency vector $A = A[1], \ldots, A[n]$
  – start with $A[i] = 0$ for all $i$.

• We observe an online sequence of updates:
  – Increments only (cash register):
    • Update is $i_t \rightarrow A[i_t] := A[i_t] + 1$
  – Fully dynamic (turnstile):
    • Update is $(i_t, \pm 1) \rightarrow A[i_t] := A[i_t] \pm 1$

• Requirements: compute statistics on $A$
  – Online, $O(1)$ passes over the updates
  – Sublinear space, $\text{polylog}(n,m)$
Typical Problems

• Frequency moments: \( F_k = |A[1]|^k + \ldots + |A[n]|^k \)
  – related: \( L_p \) norms
• Distinct elements: \( F_0 = \#\{i: A[i] \neq 0\} \)
• k-Heavy Hitters: output all \( i \) such that \( A[i] \geq F_1/k \)
• Median: smallest \( i \) such that \( A[1] + \ldots + A[i] \geq F_1/2 \)
  – Generalize to Quantiles
• Different models:
  – Graph problems: a stream of edges, increments or dynamic
    • matchings, connectivity, triangle count
  – Geometric problems: a stream of points
    • various clustering problems
When do we need this?

- The universe size \( n \) is huge.
- Fast arriving stream of updates:
  - IP traffic monitoring
  - Web searches, tweets
- Large unstructured data, external storage:
  - multiple passes make sense
- Streaming algorithms can provide a *first rough approximation*
  - decide whether and when to analyze more
  - fine tune a more expensive solution
- Or they can be the *only feasible solution*
A taste: the AMS sketch for $F_2$

[Alon Matias Szegedy 96]

$h: [n] \rightarrow \{\pm 1\}$ is 4-wise independent:

\[ h(i_1) = \pm 1 \quad h(i_2) \quad h(i_3) \quad h(i_4) \]

\[ E[X^2] = F_2 \quad E[X^4]^{1/2} \leq O(F_2) \]
The Median of Averages Trick

Average: reduces variance by $\alpha^2$.

Median: reduces probability of large error to $\delta$. 
Defining Privacy for Streams

• We will use differential privacy.
• The database is represented by a stream
  – online stream of transactions
  – offline large unstructured database
• Need to define neighboring inputs:
  – Entry (event) level privacy: differ in a single update
    
    1, 4, 5, 19, 145, 14, 5, 5, 16, 4
    1, 1, 5, 19, 145, 14, 5, 5, 16, 4
  
  – User level privacy: replace some updates to \(i\) with updates to \(j\)
    
    1, 4, 5, 19, 145, 14, 5, 5, 16, 4
    1, 4, 3, 19, 145, 14, 3, 5, 16, 4
  
  – We also allow the modified updates to be placed somewhere else
Streaming & DP?

- Large unstructured database of transactions
- Estimate how many distinct users initiated transactions?
  - i.e. $F_0$ estimation

- Can we satisfy both the streaming and privacy constraints?
  - $F_0$ has sensitivity 1 (under user privacy)
  - Computing $F_0$ exactly takes $\Omega(n)$ space
  - Classic sketches from streaming may have large sensitivity
Flajolet Martin Sketch for $F_0$

- Store a bit map $B$ of $L = O(\log n)$ bits.
  - One computer word
- Randomly hash update to $L$ bits
- Bitmap: information about least significant 1 in hashed values

$h(i_1) = 01110$  $h(i_2) = 00011$  $h(i_3) = 10011$  $h(i_4) = 01010$

- Estimate: $k = \text{index of lowest 0}; \text{Output } f(S) = 2^k$
  - $k = 3; \text{Output } 8$
Oblivious Sketch

- **Accuracy:** $F_0/2 \leq f(S) \leq 2F_0$ with constant probability
- **Obliviousness:** distribution of $f(S)$ is *entirely* determined by $F_0$
  - similar to functional privacy [Feigenbaum Ishai Malkin Nissim Strauss Wright 01]
- **Why it helps:**
  - Pick noise $\eta$ from discretized $\text{Lap}(1/\epsilon)$
  - Create new stream $S'$ to feed to $f$:
    - If $\eta < 0$, ignore first $\eta$ distinct elements
    - If $\eta > 0$, insert elements $n+1, \ldots, n+\eta$
- Distribution of $f(S')$ is a function of $\max\{F_0 + \eta, 0\}$: $\epsilon$-DP (user)
- **Error:** $F_0/2 - O(1/\epsilon) \leq f(S) \leq 2F_0 + O(1/\epsilon)$
- **Space:** $O(1/\epsilon + \log n)$
  - can make $\log n$ w.h.p. by first inserting $O(1/\epsilon)$ elements
Continual Observation

• In an online stream, often need to track the value of a statistic.
  – number of reported instances of a viral infection
  – sales over time
  – number of likes on Facebook

• Privacy under continual observation [Dwork Naor Pitassi Rothblum 10]:
  – At each time step the algorithm outputs the value of the statistic
  – The entire sequence of outputs is $\varepsilon$-DP (usually event level)

• Results:
  – A single counter (number of 1’s in a bit stream) [DNPR10]
  – Time-decayed counters [Bolot Fawaz Muthukrishnan Nikolov Taft 13]
  – Online learning [DNPR10] [Jain Kothari Thakurta 12] [Smith Thakurka 13]
  – Generic transformation for monotone algorithms [DNPR10]
Binary Tree Technique [DPNR10], [Chan Shi Song 10]

Sensitivity of tree: $\log m$

Add $\text{Lap}(\log m/\epsilon)$ to each node
Binary Tree Technique

Each prefix: sum of $\log m$ nodes

$\rightarrow$ polylog error per query
Continuous Counter

- Achieves polylog($m$) error per time step
- Simple variations:
  - the value of $m$ is unknown
  - other statistics decomposable over time intervals
- Improve error for time-decayed statistics:
  - vary the noise on different levels of the tree
- Applications to online learning
  - continuous counters → track gradient of risk function
Thank you!