Credit: Mary McGlohon

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

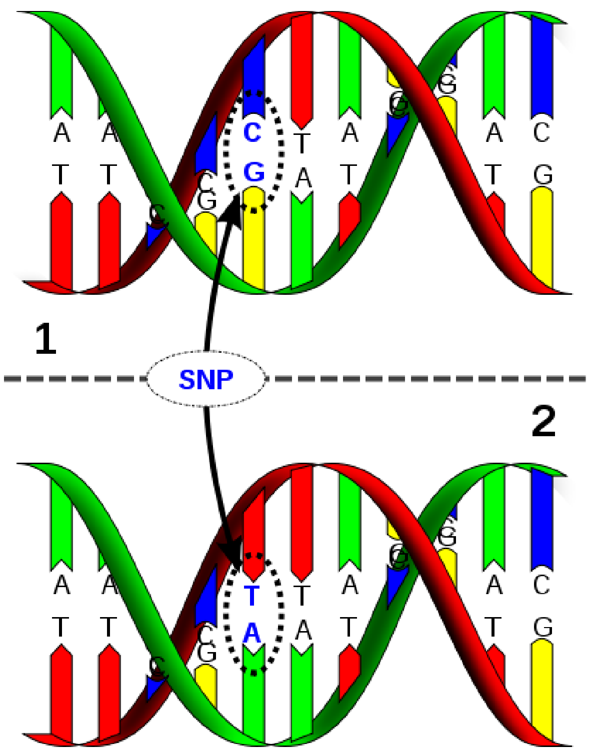
Technique: Streaming data structures

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

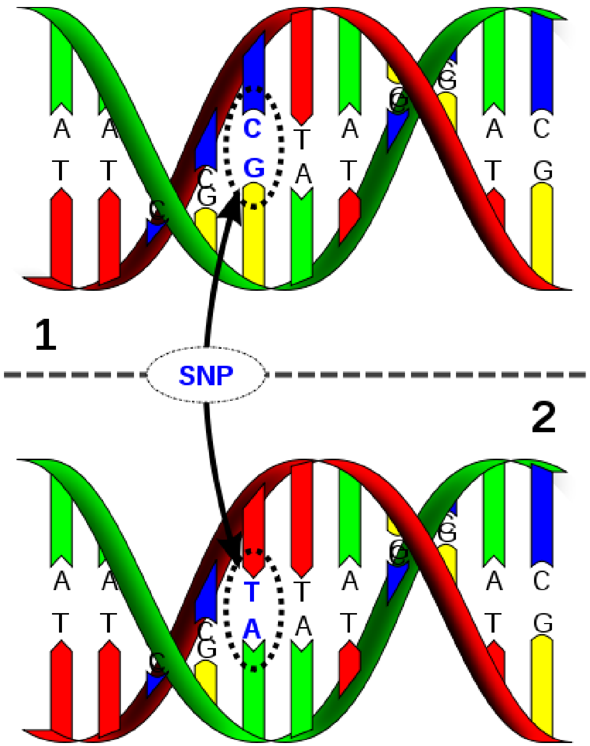


GWAS

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!

SNP	1	2	3	…	…	100000
MAF	0.02	0.03	0.05	…		0.02

Test population

SNP	1	2	3	…	…	100000
MA	NO	NO	YES	…		YES

Moritz's DNA

SNP	1	2	3	…	…	100000
MAF	0.01	0.04	0.04	…		0.01

Reference population (HapMap data, public)

Attack on GWAS data [Homer et al.]

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

Test population

SNP	1	2	3	…	…	100000
MAF	0.02	0.03	0.05	…		0.02

probably

Moritz's DNA

SNP	1	2	3	…	…	100000
MA	NO	NO	YES	…		YES

Reference population (HapMap data, public)

SNP	1	2	3	…	…	100000
MAF	0.01	0.04	0.04	…		0.01

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* = {GWAS test population}, *D'* = *D* – {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

[Dwork-McSherry-Nissim-Smith-06]

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

Example: *D* = {GWAS test population}, *D'* = *D* – {Moritz's DNA}

Definition (Differential Privacy):

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

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

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* = {GWAS test population}, *D'* = *D* – {Moritz's DNA}

Definition (Differential Privacy):

A randomized algorithm *M(D)* is (ε,δ)-differentially private if for all neighboring data sets *D,D'* and all events *S*:

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

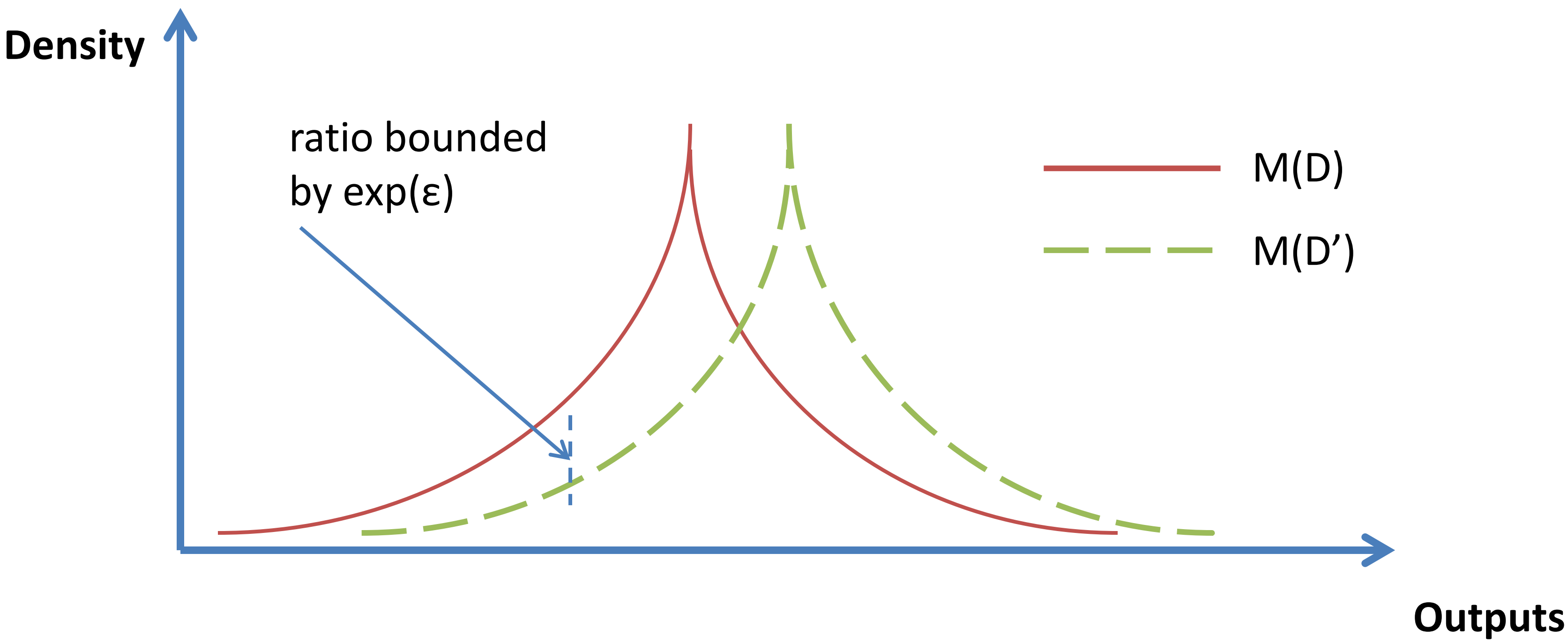
Think: $\varepsilon = 0.01$ and $e^{\varepsilon} \approx 1+\varepsilon$

$\delta << 1/|D|$

Definition (Differential Privacy):

A randomized algorithm *M(D)* is *ε*-differentially private if for all neighboring data sets *D,D’* and all events *S*:

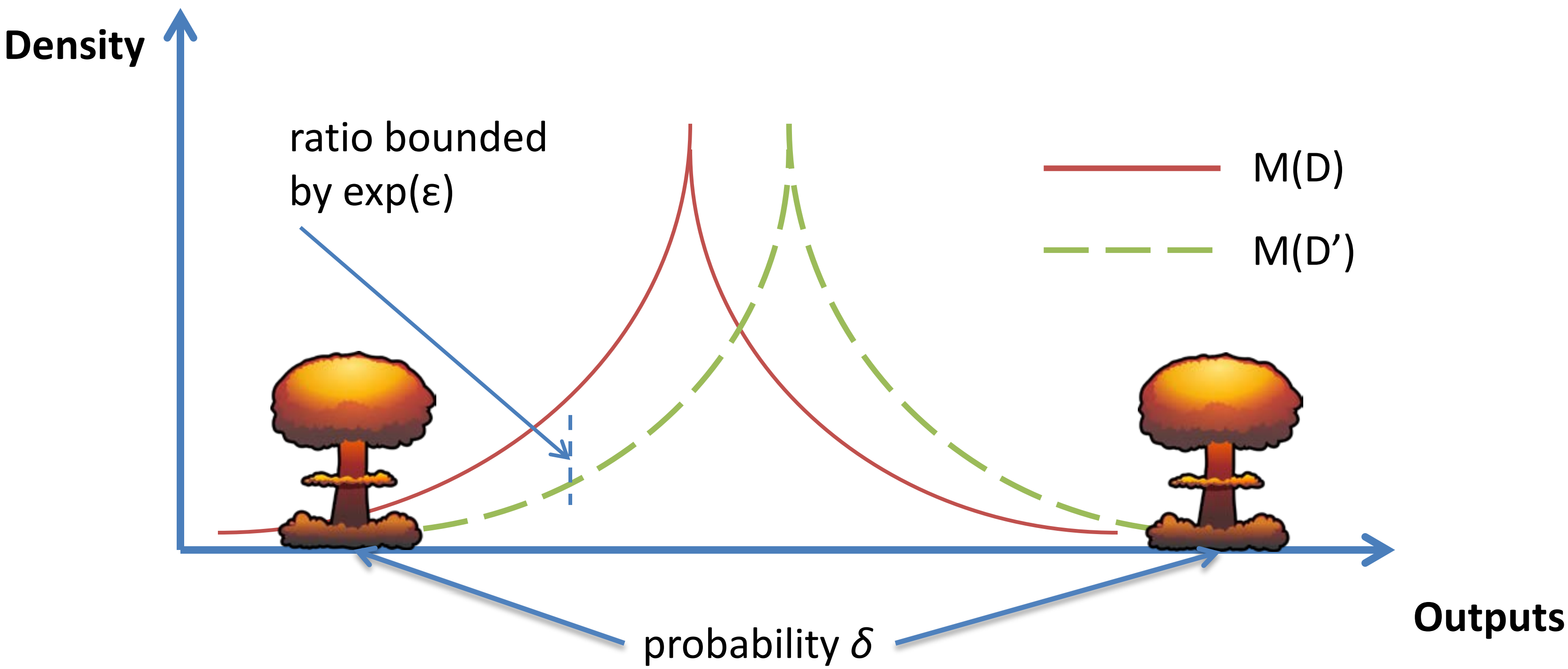
$$\Pr\{M(D) \in S\} \le e^{\epsilon} \Pr\{M(D') \in S\}$$



Definition (Differential Privacy):

A randomized algorithm *M(D)* is (ε,δ)-differentially private if for all neighboring data sets *D,D'* and all events *S*:

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



Sensitivity

Assume f maps databases to real vectors

Definition:

- L1-sensitivity $\Delta_1(f) = \max \| f(D) - f(D') \|_1$
  - maximum over all neighboring D,D'
- L2-sensitivity $\Delta_2(f) = \max \| f(D) - f(D') \|_2$

Intuition: Low sensitivity implies "compatible with differential privacy"

Exercise

f maps D 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?"

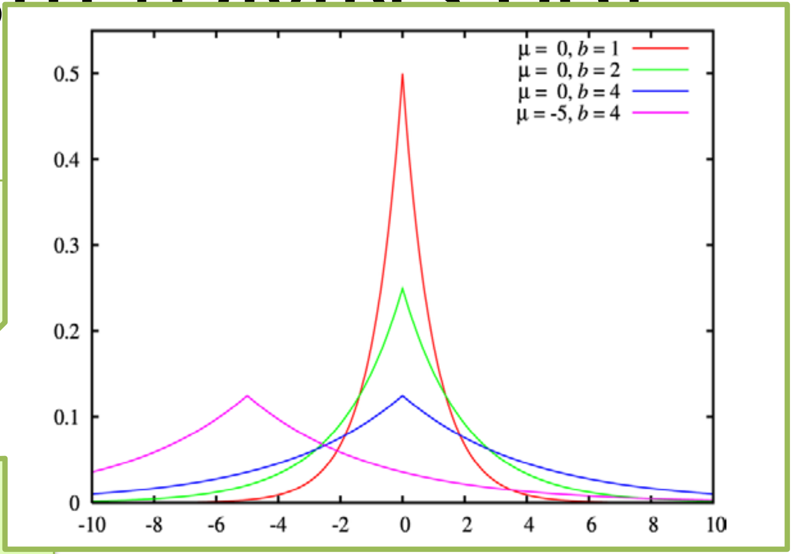
L1-sensitivity? k

L2-sensitivity? $k^{1/2}$

Laplacian Mechanism [DMNS'06]

Given function *f:*

1. Compute *f(D)*
2. *Output f(D) +* $Lap(\Delta_1(f)/\varepsilon)^d$



d-dimensional Laplace distribution

Scale noise to L1-sensitivity

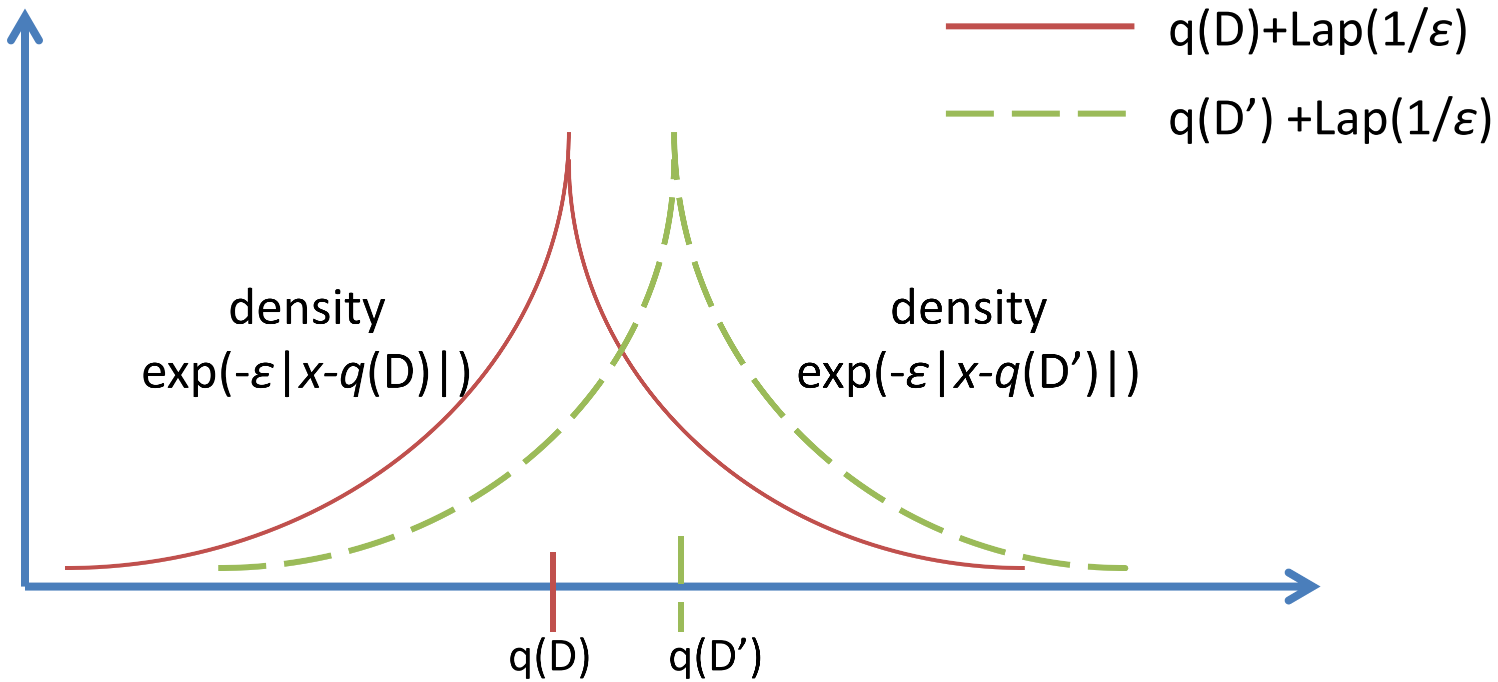
Fact 1: Satisfies $(\varepsilon, 0)$-differential privacy

Fact 2: Expected error $\Delta_1(f)/\varepsilon$ in each coord.

Laplacian Mechanism [DMNS'06]

One-dimensional example

Suppose D,D' neighboring

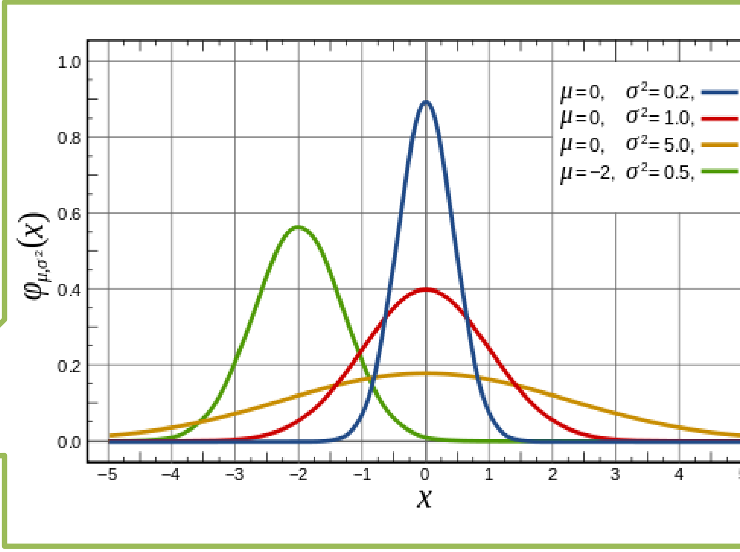


Gaussian Mechanism

Given function *f:*

1. Compute *f(D)*
2. *Output f(D) + N(0,σ²)^d*

   with $\sigma = \Delta_2(f)\log(1/\delta)/\varepsilon$



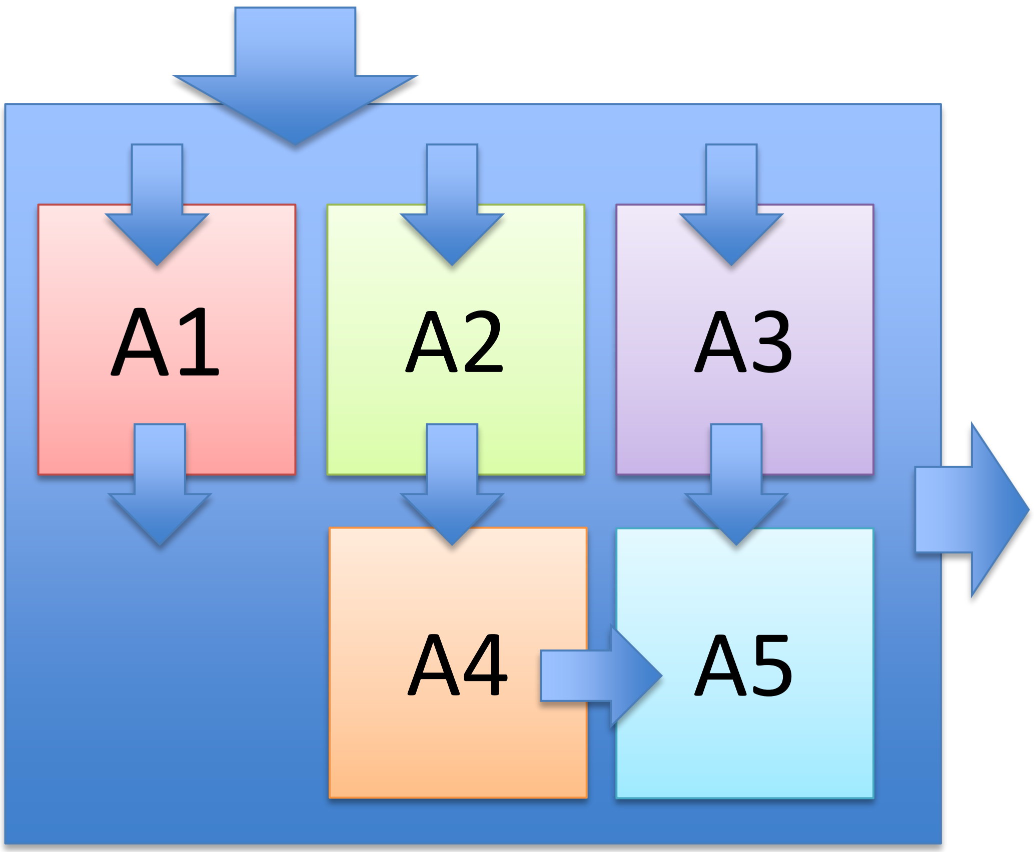
d-dimensional Gaussian distribution

Scale noise to **L2**-sensitivity

Fact: Satisfies (ε, δ)-differential privacy

Fact: Expected error $\Delta_2(f)\log(1/\delta)/\varepsilon$

Composition



Differential privacy composes:

1. in parallel
2. sequentially
3. adaptively

T-fold **composition** of ε_0-differential privacy satisfies:

Answer 1
[DMNS'06]:

$\varepsilon_0 T$-differential privacy

Answer 2
[DRV'10]:

(ε,δ)-differential privacy

$$\epsilon \leq 4\epsilon_0 \sqrt{T \log(1/\delta)}$$

Note: for small enough ε

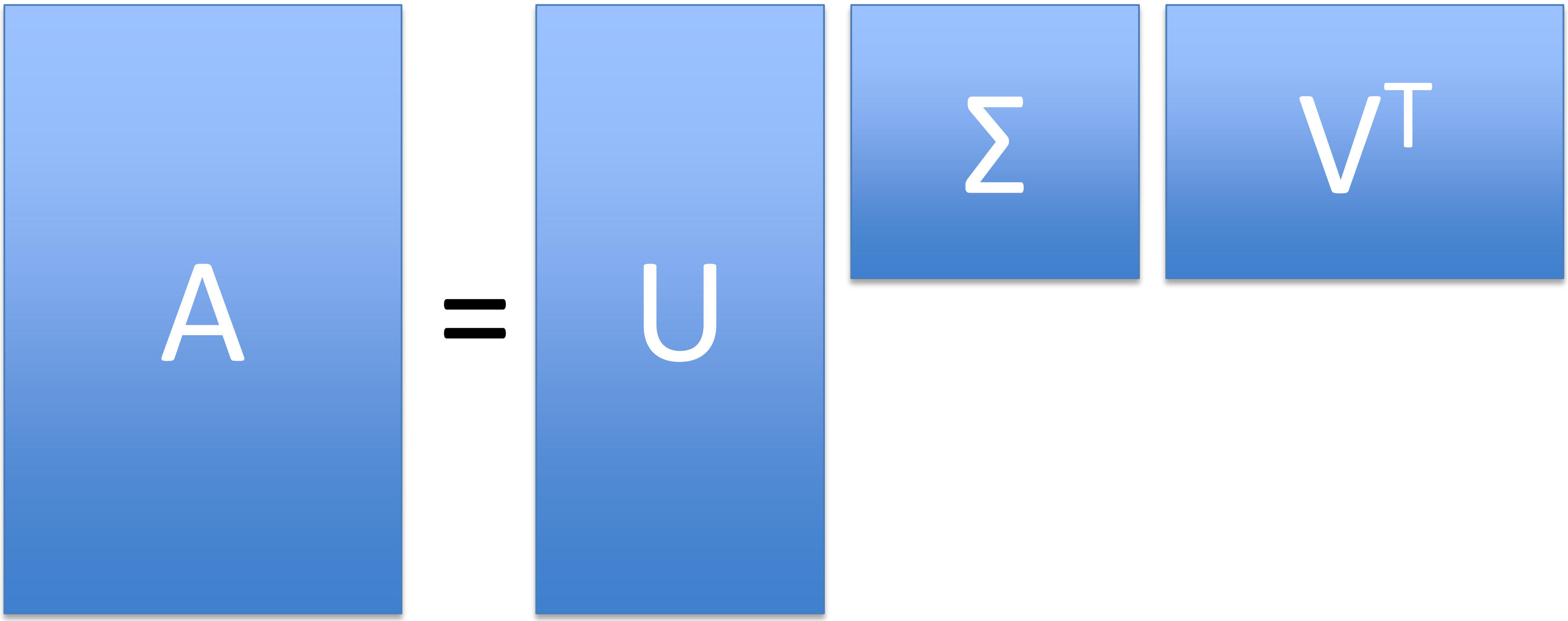
Part II:
Private SVD
and Applications

We meet the protagonist

A is a real n x d matrix

thought of as n data points in d dimensions

...and its singular value decomposition



U,*V* have orthonormal columns

Σ is diagonal *r* x *r*,   *r* = rank(*A*)

The left factor

Columns $u_1,\ldots,u_r$ are the singular vectors of A
$\Leftrightarrow$ eigenvectors of the n x n Gram matrix AA^T

By **Courant-Fisher:**

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

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

…

V^T

Columns $v_1,\ldots,v_r$ of V are the eigenvectors of the d x d covariance matrix A^TA

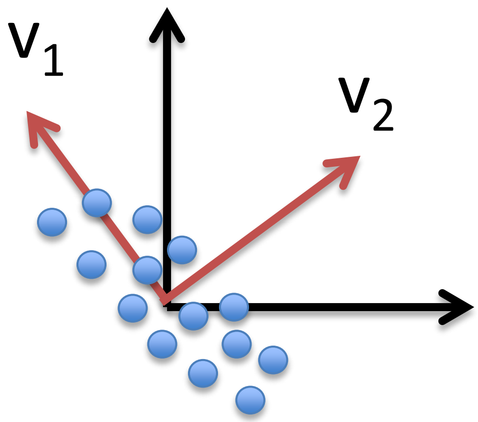
Σ

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

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"?

Differer in one row

Differ in one row by norm at most 1

Differ in at most one entry by at most 1

Stronger privacy

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 ε.
- Algorithms we’ll see also give other guarantees using different noise scaling.

Objectives

- Approximating the top singular vector
  - Objective:

$$\max_{x:\,\|x\|=1} \|Ax\| \qquad \mathrm{OPT} = \sigma_1(A)$$

- Approximating *k* singular vectors
  - Objective:

$$\min_{B:\,\mathrm{rk}(B)=k} \|A - B\|_2 \qquad \mathrm{OPT} = \sigma_{k+1}(A)$$

Goal: Minimize additive error subject to differential privacy

Baseline Algorithm

1965

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. $N = Lap(1/\varepsilon)^{nxd}$
2. $B = A + N$
3. Output $SVD(B)$

Fact: Satisfies ε-differential privacy.

How much error does this introduce?

Some matrix theory

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

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

Theorem: Let *N* = Lap$(1/\varepsilon)^{nxd}$. *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 and d both large

RR generates huge dense noise matrix

Problem 2:
matrix sparse, e.g, Δ 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

n

n

1111000111111...111110011111
0000000000000...000000000000
0000000000000...000000000000
0000000000000...000000000000
0000000000000...000000000000
...
0000000000000...000000000000

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?

Beat **Randomized Response** (1965) with **Power Method** (1929)

ZUSAMMENFASSENDE BERICHTE

Praktische Verfahren der Gleichungsauflösung.

Von **R. v. MISES** und **H. POLLACZEK-GEIRINGER** in Berlin.

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.

Im ersten Abschnitt des vorliegenden Berichtes wird ein Verfahren zur Lösung einer beliebigen Gleichung mit einer Unbekannten entwickelt, das sich von den bisher bekannten dadurch unterscheidet, daß es stets zum Ziele führt (immer konvergiert) und in der Regel weniger Rechenarbeit verlangt. Die Aufsuchung der Wurzeln von Gleichungssystemen wird in den beiden folgenden Abschnitten nur für den Sonderfall der linearen Gleichungen behandelt. Für das »Iterationsverfahren in Gesamtschritten«, welches die unmittelbare Uebertragung des Verfahrens für Eine Unbekannte darstellt, werden in Abschnitt 2 die zum Teil bekannten, meist nicht gehörig beachteten Konvergenzbedingungen zusammengestellt sowie Fehlerabschätzungen gegeben.

Besondere Bedeutung kommt dem in Abschnitt 3 behandelten, als »Iteration in Einzelschritten« bezeichneten, von Ph. Seidel stammenden Ansatz zu. Dieser ist — nach

From here on

Assume A is symmetric and n x n

Wlog, consider:

$$\begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix}$$

Recap: Power Method

Input: n x n matrix A, parameter T

Pick random unit vector x_0

For $t = 1$ ***to*** T:

$y_t = Ax_{t-1}$

$x_t = y_t / |y_t|_2$

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: $||(A\text{-}A')x||_2 \leq ||x||_\infty$
where $||x||_\infty = \max_i |x_i|$

Noisy Power Method

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

Pick random unit vector x_0

For $t = 1$ ***to*** T:

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

$x_t = y_t / |y_t|_2$

Output x_T

largest squared entry of x_{t-1} in [1/n,1]

Fact: *Satisfies* (ε,δ)*-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 : $\|x_t\|_\infty \le \sqrt{n}B$

Can we do better?

Lemma [H-Roth13]:

$$\Pr\left\{\|x_t\|_\infty \ge 4B\sqrt{\log n}\right\} \le 1/n^3$$

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 (ε,δ)-differential privacy and with T = O(log(n)) steps returns a unit vector x such that

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

provided A satisfies "singular value separation."

Contrast with

$$O\left(\sqrt{n}\right)$$

for Randomized Response even if $\mu(A)$=1

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

When **utility** and **privacy** benefit from the same principle

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

Robust PCA
[CandesLiMaWright09]
Cope with corrupted entries

Other approaches
[Chaudhuri-Sarwate-Sinha12, Kapralov-Talwar13]

Low
coherence

Matrix Completion
[CandesTao09,CandesRecht]
Recover missing entries

Differentially Private Recommender System
[McSherryMironov09]
building on
Randomized Response
[BlumDworkMcSherryNissim]

Privacy Outcry
Users re-identified
[NarayananShamatikov08]

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

Generalization: Subspace Iteration

Input: n x n matrix A *symmetric*, target rank k

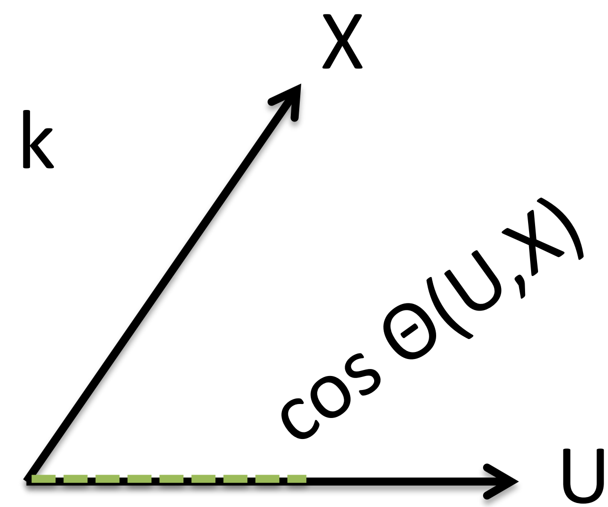
X_0 random orthonormal matrix

For $t = 1$ to T:

- Pick Gaussian perturbation G_t
- $Y_t = AX_{t-1} + G_t$
- X_t = Orthonormalize(Y_t)

Output X_T *(approx top k singular vectors)*

Principle Angle Between Subspaces



Let U, X subspaces of dimension k

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

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

$\sin\Theta(U,X) = \sigma_{max}(V^T X)$
where V orthog. complement of U

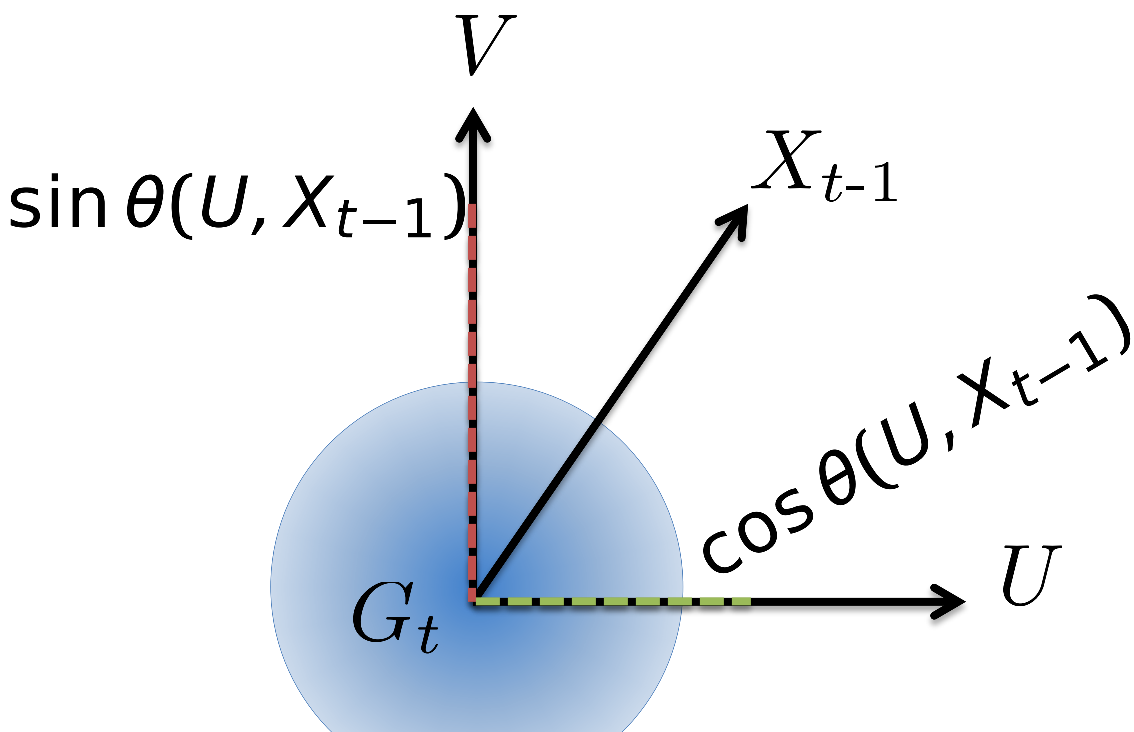
$\tan\Theta(U,X) = \sin\Theta(U,X) / \cos\Theta(U,X)$

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

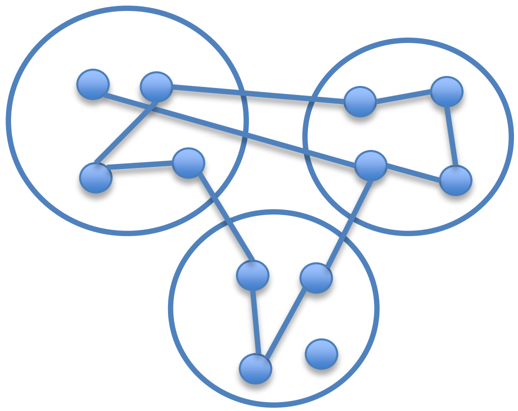


Application 1: Spectral Clustering

Planted multisection model:

c clusters, intra-cluster edge probability p

inter-cluter probability q, $q < p$



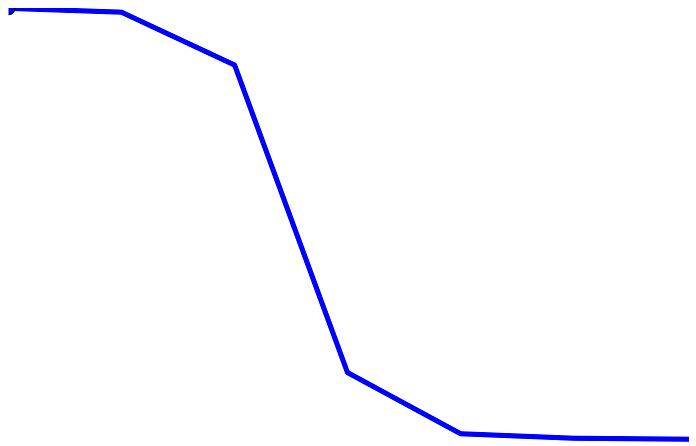
How to recover a cluster?

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 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.

```
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00200102010112012211222101112212111111211101121111011212201011111111122212101211122012011020001112222
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00200102010112012211222101112212111111211101121111011212201011111111122212101211122012011020001112222
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0010000101001100110111100000111110010111010101011000011110001111101001111100111101101101101000111111
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
0010000101001100110111100000111110010111010101011000011110001111101001111100111101101101101000111111
00200102010112012211222101112212111111211101121111011212201011111111122212101211122012011020001112222
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
00200102010112012211222101112212111111211101121111011212201011111111122212101211122012011020001112222
00200102010112012211222101112212111111211101121111011212201011111111122212101211122012011020001112222
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
0010000101001100110111100000111110010111010101011000011110001111101001111100111101101101101000111111
00200102010112012211222101112212111111211101121111011212201011111111122212101211122012011020001112222
00200102010112012211222101112212111111211101121111011212201011111111122212101211122012011020001112222
00200102010112012211222101112212111111211101121111011212201011111111122212101211122012011020001112222
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
00200102010112012211222101112212111111211101121111011212201011111111122212101211122012011020001112222
00200102010112012211222101112212111111211101121111011212201011111111122212101211122012011020001112222
0010000101001100110111100000111110010111010101011000011110001111101001111100111101101101101000111111
00200102010112012211222101112212111111211101121111011212201011111111122212101211122012011020001112222
0010000101001100110111100000111110010111010101011000011110001111101001111100111101101101101000111111
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
0010000101001100110111100000111110010111010101011000011110001111101001111100111101101101101000111111
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
0010000101001100110111100000111110010111010101011000011110001111101001111100111101101101101000111111
00200102010112012211222101112212111111211101121111011212201011111111122212101211122012011020001112222
0010000101001100110111100000111110010111010101011000011110001111101001111100111101101101101000111111
```

Differentially Private Approximation

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

< 1000 entries have changed out of 1,000,000

```
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00200102010112012211222101112212111111211101121111011212201011111111122212101211112201201102000111222
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00200102010112012211222101112212111111211101121111011212201011111111122212101211112201201102000111222
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
001000010100110011011110000011111001011101010101100001111000111110100111110011110110110110100011111
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
001000010100110011011110000011111001011101010101100001111000111110100111110011110110110110100011111
00200102010112012211222101112212111111211101121111011212201011111111122212101211112201201102000111222
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
00200102010112012211222101112212111111211101121111011212201011111111122212101211112201201102000111222
00200102010112012211222101112212111111211101121111011212201011111111122212101211112201201102000111222
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
001000010100110011011110000011111001011101010101100001111000111110100111110011110110110110100011111
00200102010112012211222101112212111111211101121111011212201011111111122212101211112201201102000111222
00200102010112012211222101112212111111211101121111011212201011111111122212101211112201201102000111222
00200102010112012211222101112212111111211101121111011212201011111111122212101211112201201102000111222
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
00200102010112012211222101112212111111211101121111011212201011111111122212101211112201201102000111222
00200102010112012211222101112212111111211101121111011212201011111111122212101211112201201102000111222
001000010100110011011110000011111001011101010101100001111000111110100111110011110110110110100011111
00200102010112012211222101112212111111211101121111011212201011111111122212101211112201201102000111222
001000010100110011011110000011111001011101010101100001111000111110100111110011110110110110100011111
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
001000010100110011011110000011111001011101010101100001111000111110100111110011110110110110100011111
0010010100010101111011110111110101101010100011100101110110100000010111110110010011100100001000000111
001000010100110011011110000011111001011101010101100001111000111110100111110011110110110110100011111
00200102010112012211222101112212111111211101121111011212201011111111122212101211112201201102000111222
001000010100110011011110000011111001011101010101100001111000111110100111110011110110110110100011111
```

In this example: Matrix has

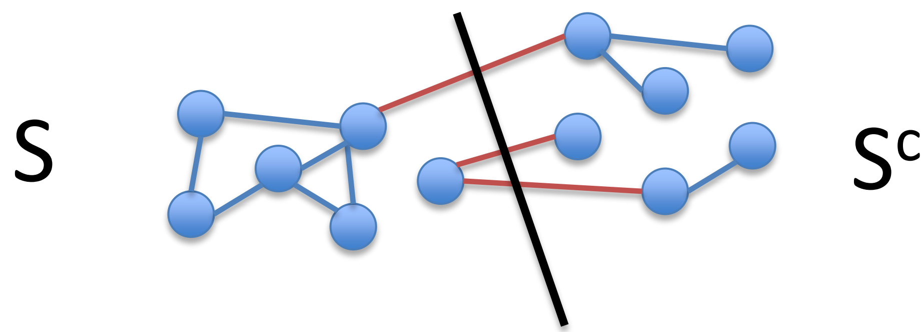
rank *2 and* coherence 2

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: $\widetilde{L}_G = \frac{1}{r} E_G^T M^T M E_G$

By JL Theorem, $r = \alpha^{-2} \log(1/\beta)$ preserves single cut up to factor (1±α) with pr 1-β

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
(ε,δ)-differential privacy.

Additive error on one cut of size s: $O(ws)$

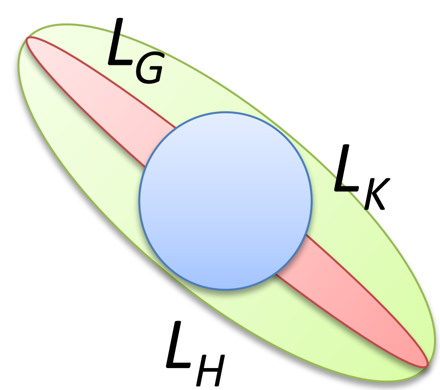
Geometric Intuition

Sample $g \sim N(0,1)^{\binom{n}{2}}$

Fact: $E_H^T g$ Gaussian variable with covariance matrix L_H

Sparse cuts ⇔ 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*)

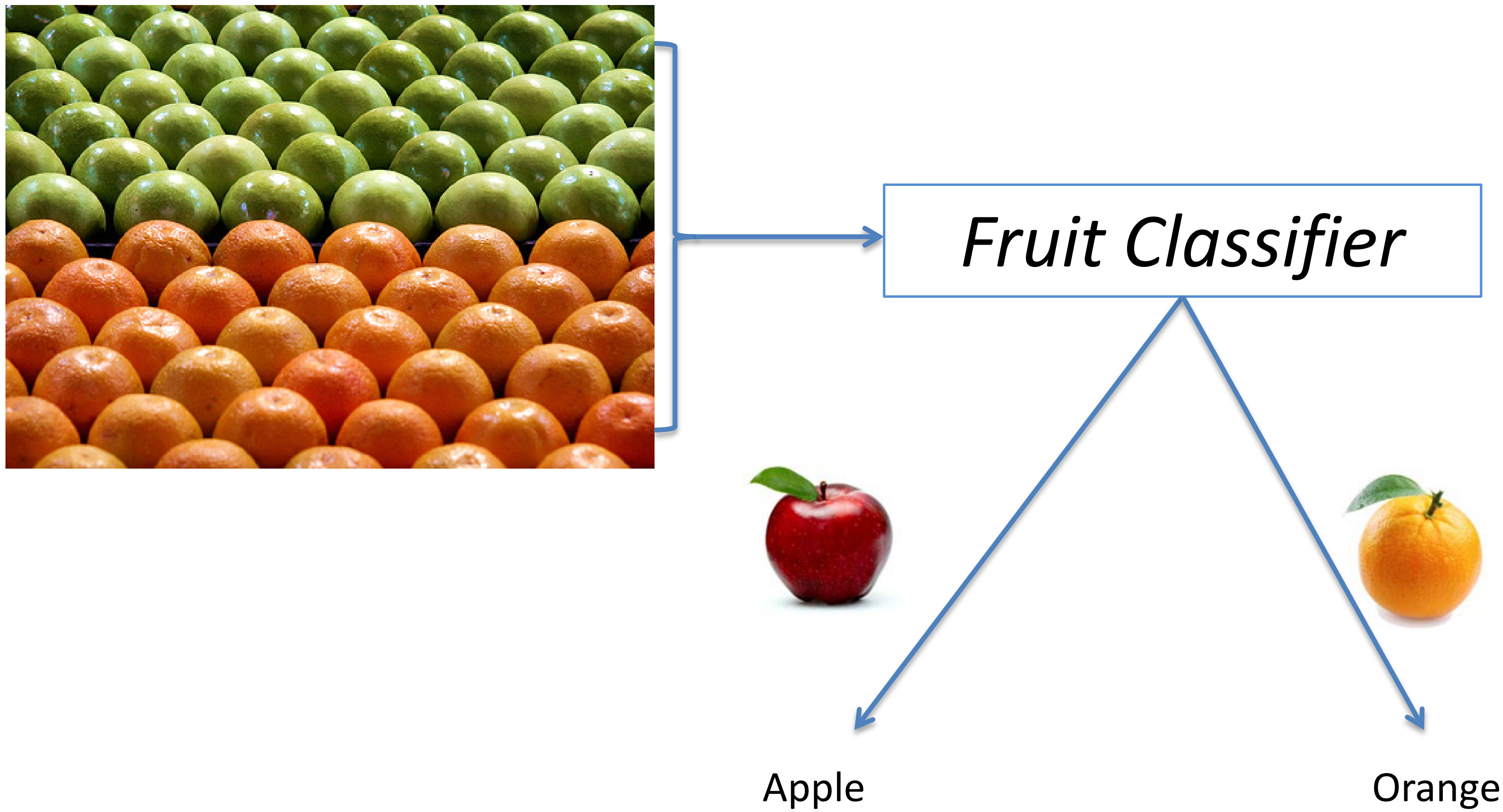


End of Part II

But wait, there’s more…

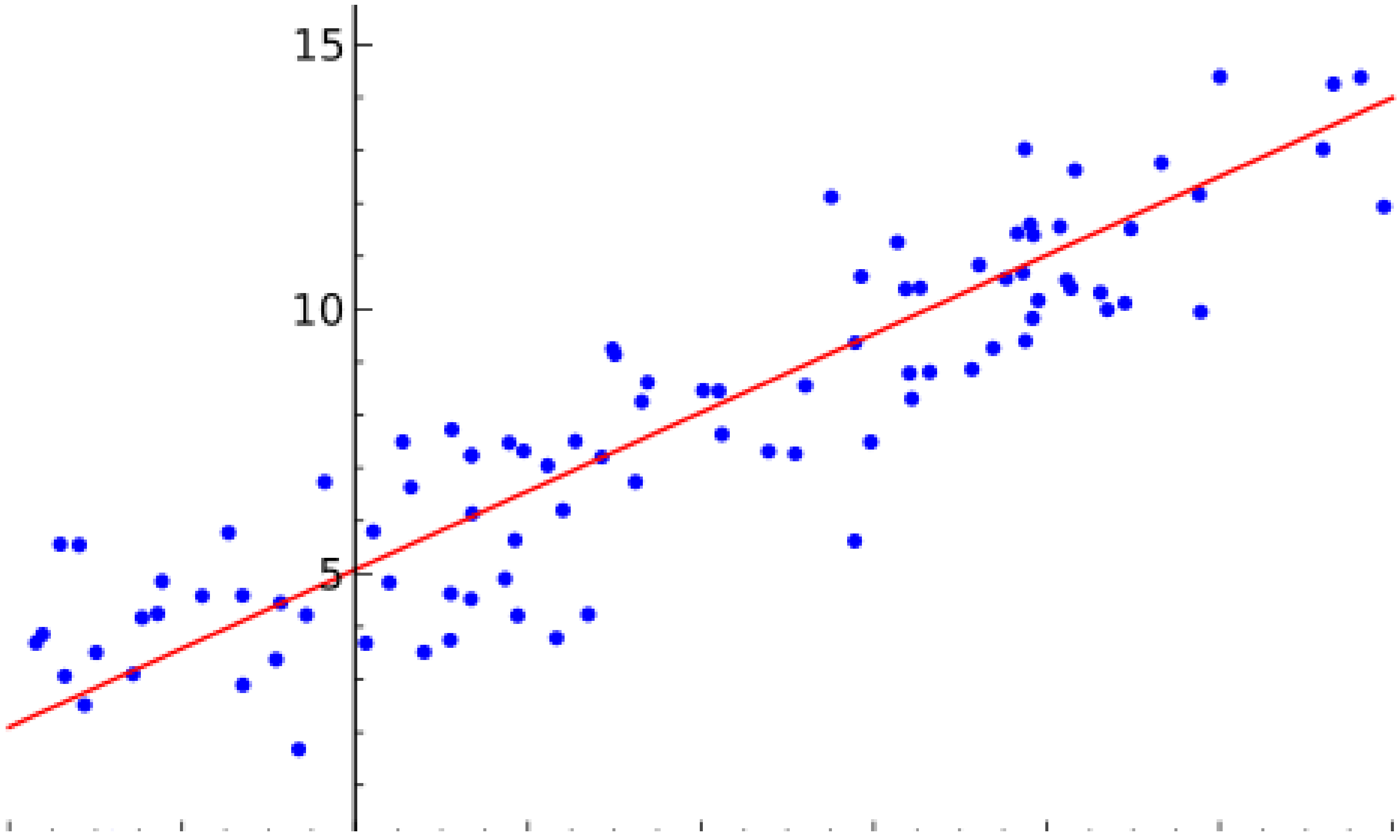
Part III:
Supervised Learning
(Empirical Risk Minimization)

Supervised Learning: Classification



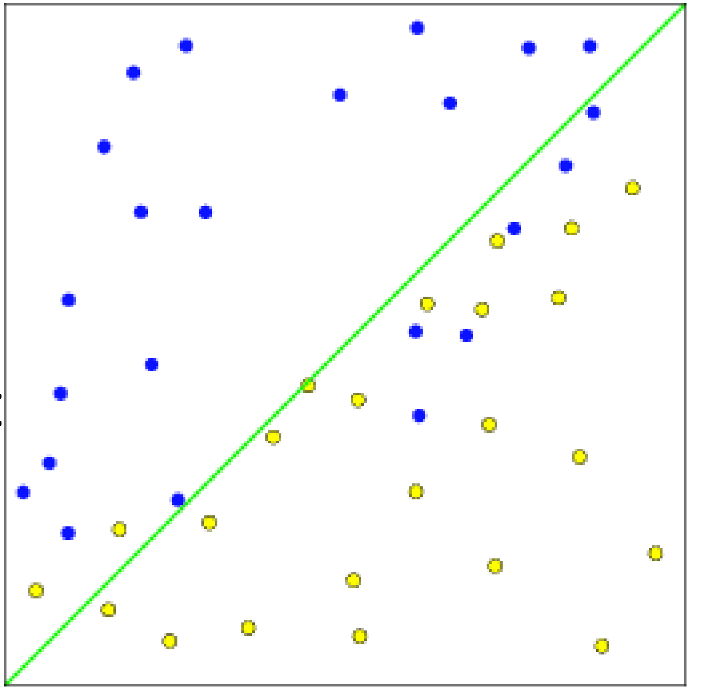
http://www.orangelt.us/info/wp-content/uploads/2010/08/OrangeLT_White_Background.jpg
http://analyticstraining.in/blog/wp-content/uploads/2014/02/red_delicious_apple.jpg
http://capitaldisruptivo.files.wordpress.com/2012/04/apples-and-oranges1.jpg

Supervised Learning: Regression



Generalized Linear Model

- <u>Data</u> $X_n = \{(x_{1,} y_1), (x_{2,} y_2), \ldots, (x_{n,} y_n)\}$
  - $(x_{i,} y_i)$ sampled IID from distribution D
  - *data point* x_i *is* d-dimensional, *label* y_i is real
- <u>Goal</u>: predict y for new x, w
- <u>Hypothesis class:</u> H
- <u>Loss function:</u> l(<w, x>; y)
  - loss of hypothesis w on (x,y)
  - assumption: y can be predict measurement of x
  - l will be *convex* in w



Risk Minimization

- The Risk Minimization Problem:
  - $w^* = \arg\min_w E_{(x,\,y) \sim D} [\, l(\langle w, x \rangle; y) \,]$
- How to do that based on X_n?
- Minimize empirical risk:

$$R_n(w) = \frac{1}{n} \sum_i \ell(\langle w, x_i \rangle; y_i)$$

- *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:

$$J_n(w) = R_n(w) + \lambda r(w)$$

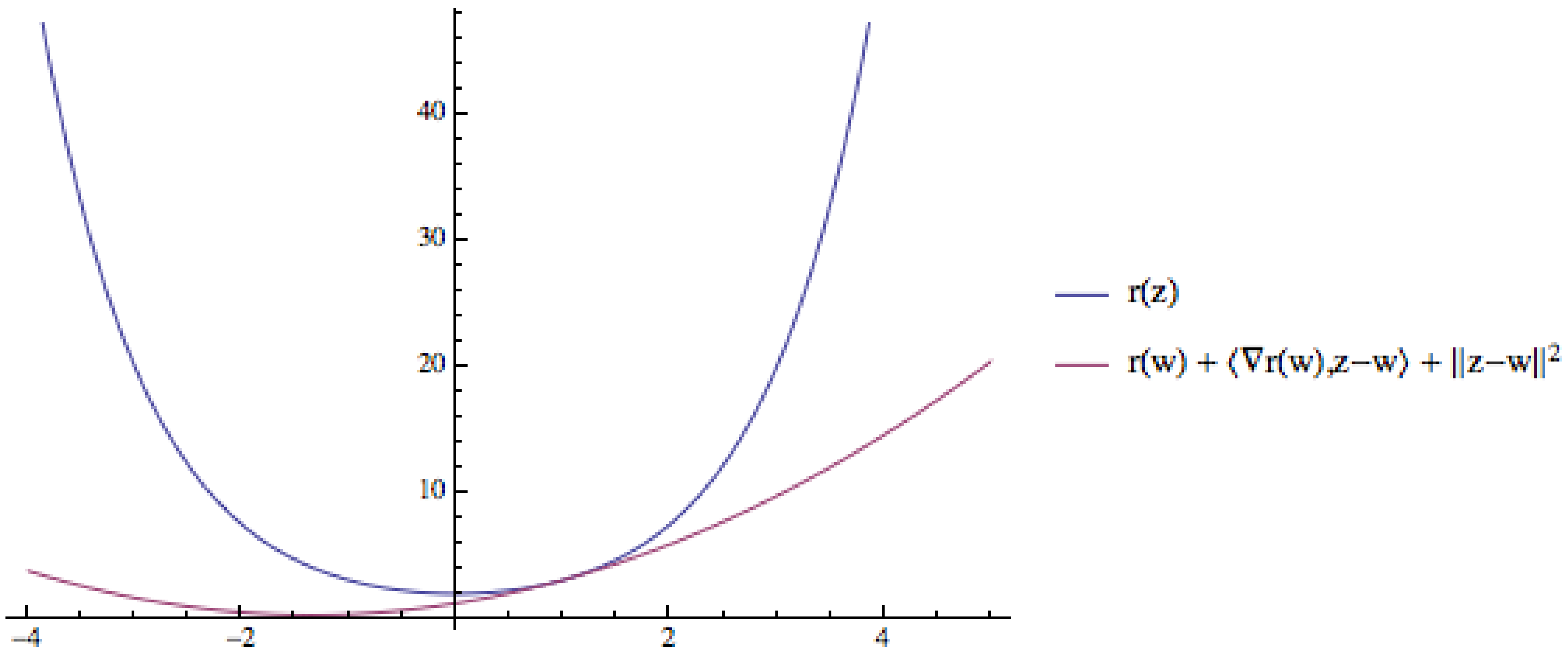
- Regularizer $r(w)$: "complexity" of w

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



Examples

- SVM:
  - $l(\langle w, x\rangle; y) = \max\{0, 1 - y\langle w,x\rangle\}$;
  - $r(w) = 0.5|w|^2$
- Logistic Regression
  - $l(\langle w, x\rangle; y) = \log(1 - \exp(-y\langle w,x\rangle)$;
  - $r(w) = 0.5|w|^2$
- Ridge Regression
  - $l(\langle w, x\rangle; y) = (y - \langle w,x\rangle)^2$;
  - $r(w) = 0.5|w|^2$

Private Algorithm: Output Perturbation

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

- (ε,δ)-DP if:
  - l is 1-Lipschitz:

$$|l(\langle z, x\rangle; y) - l(\langle w, x\rangle; y)| \leq |\langle z-w, x\rangle|$$

  - $|x| \leq 1$
- [Chaudhuri, Monteleoni, Sarwate '11]

Sensitivity Analysis

- *Intuition*: strong convexity -> low sensitivity

$$X = \{(x_{1,} y_1), (x_{2,} y_2), \ldots, (x_{n,} y_n)\}$$

$$X' = \{(x_{1,} y_1), (x_{2,} y_2), \ldots, (x'_{,} y')\}$$

$J_n(w)$: risk for X; $J'_n(w)$: risk for X'

w: minimizer of $J_n(w)$; w': minimizer of $J'_n(w)$;

<u>Sensitivity</u>: $|w - w'| \leq 2/(\lambda n)$

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:

$$\left|\ell(\langle w^{\sim}, x\rangle; y) - \ell(\langle w, x\rangle; y)\right| \leq \left|\langle b, x\rangle\right|$$

  - $\langle b,x\rangle$ 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], …, A[n]
  - start with A[i] = 0 for all i.
- We observe an <u>online sequence of updates:</u>
  - Increments only (cash register):
    - Update is i_t → A[i_t] := A[i_t] + 1
  - Fully dynamic (turnstile):
    - Update is (i_t , ±1) → A[i_t] := A[i_t] ± 1
- <u>Requirements</u>: compute statistics on A
  - Online, O(1) passes over the updates
  - Sublinear space, polylog(n,m)

Typical Problems

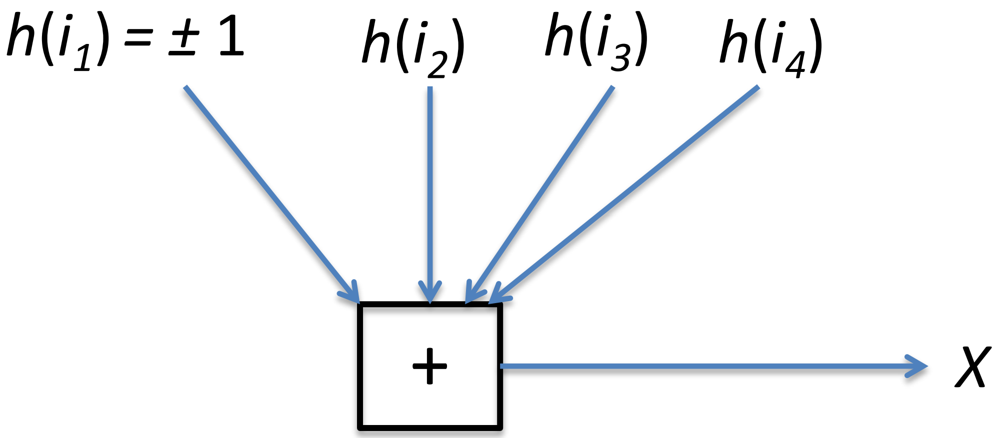
- Frequency moments: $F_k = |A[1]|^k + \ldots + |A[n]|^k$
  - related: L_p norms
- Distinct elements: $F_0 = \#\{i\text{: } A[i] \neq 0\}$
- k-Heavy Hitters: output all i such that $A[i] \geq F_1/k$
- Median: smallest i such that $A[1] + \ldots + A[i] \geq F_1/2$
  - *Generalize to Quantiles*
- Different 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

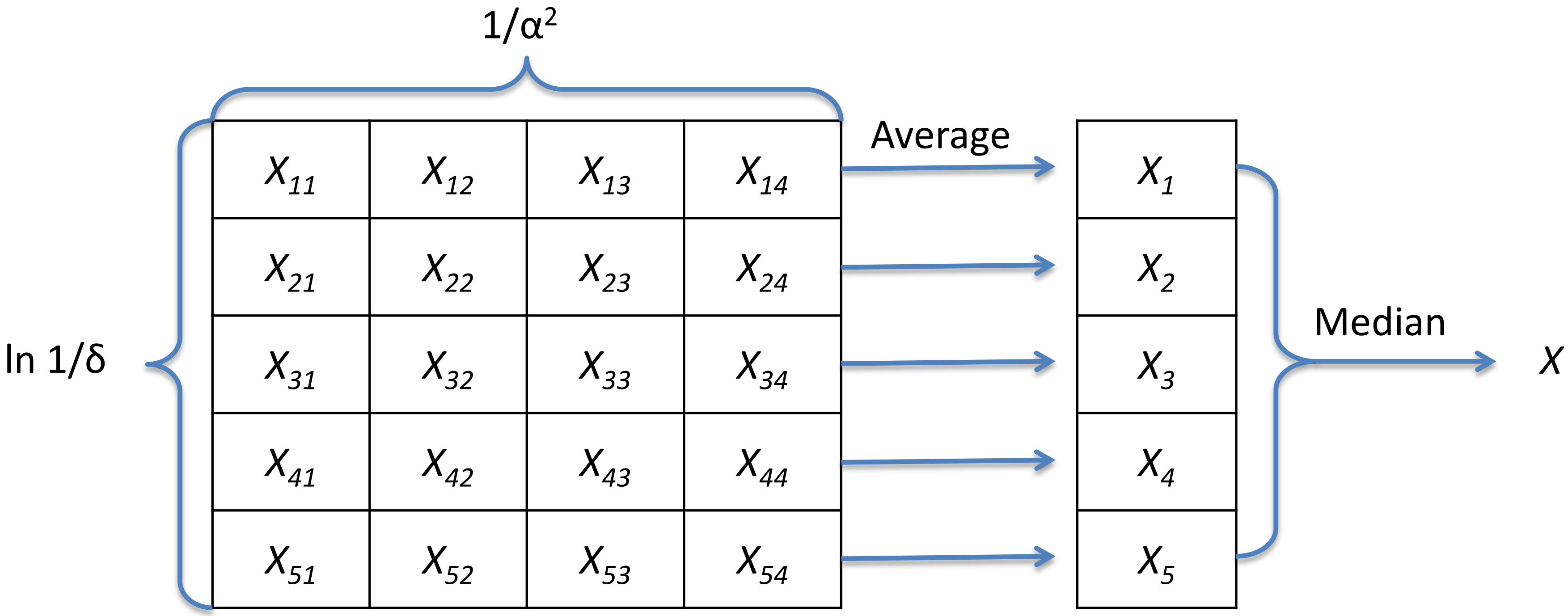
[Alon Matias Szegedy 96]



h:[n] → {± 1} is 4-wise independent:

$E[X^2] = F_2$ $\qquad$ $E[X^4]^{1/2} \leq O(F_2)$

The Median of Averages Trick



Average: reduces variance by α^2.

Median: reduces probability of large error to δ.

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:*
  - <u>Entry (event) level privacy</u>: 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
  - <u>User level privacy</u>: 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 <u>both</u> 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

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

B					
	1	1	0	1	0

- Estimate: k = index of lowest 0; Output $f(S) = 2^k$
  - $k = 3$; Output 8

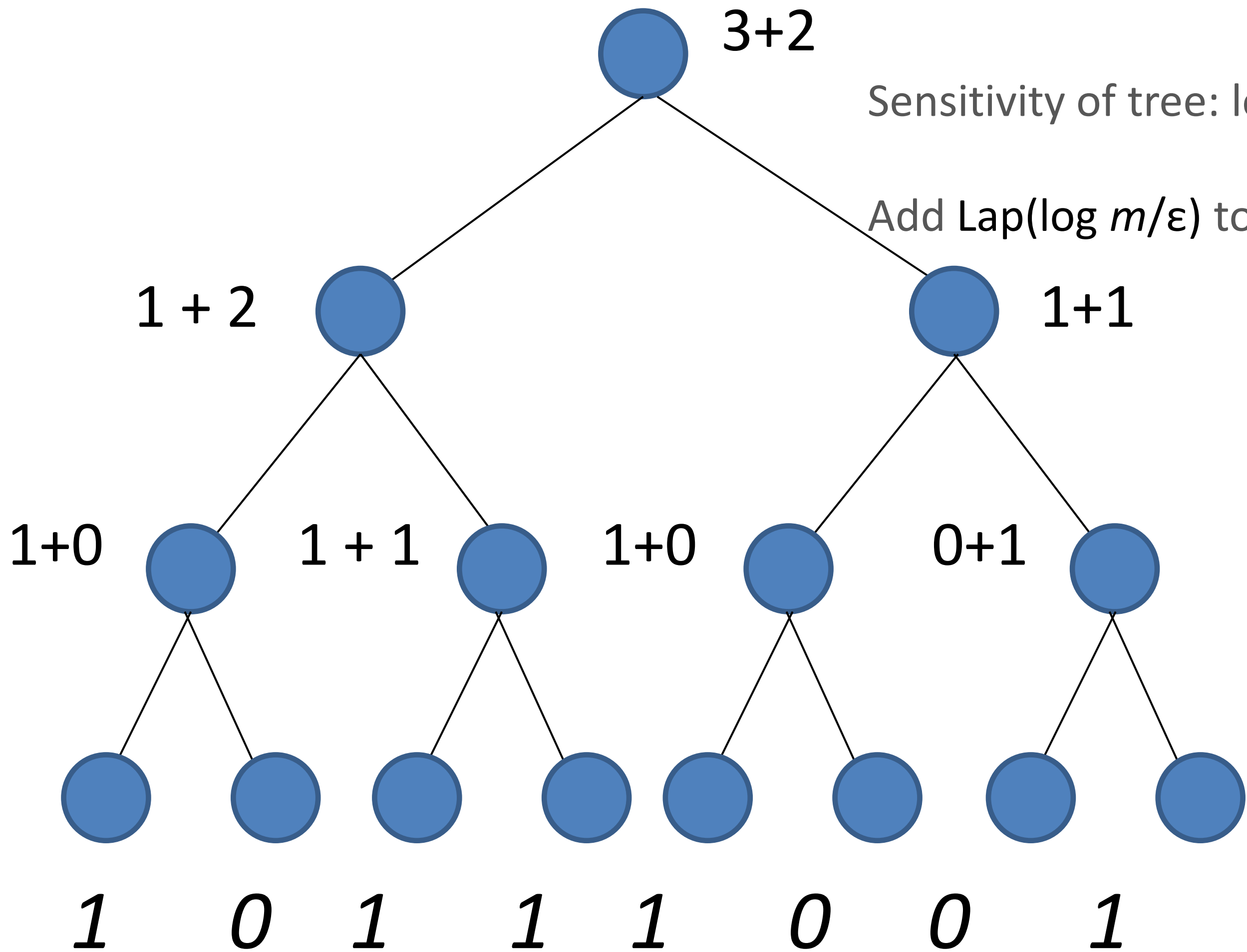
Oblivious Sketch

- 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 η from discretized Lap(1/ε)
  - Create new stream S' to feed to f:
    - If $\eta < 0$, ignore first η distinct elements
    - If $\eta > 0$, insert elements $n+1, \ldots, n+\eta$
- Distribution of $f(S')$ is a function of $\max\{F_0 + \eta, 0\}$: ε-DP (user)
- Error: $F_0/2 - O(1/\varepsilon) \leq f(S) \leq 2F_0 + O(1/\varepsilon)$
- Space: $O(1/\varepsilon + \log n)$
  - can make $\log n$ w.h.p. by first inserting $O(1/\varepsilon)$ 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
- <u>Privacy under continual observation</u> [Dwork Naor Pitassi Rothblum 10]:
  - At each time step the algorithm outputs the value of the statistic
  - The *entire sequence* of outputs is ε-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]

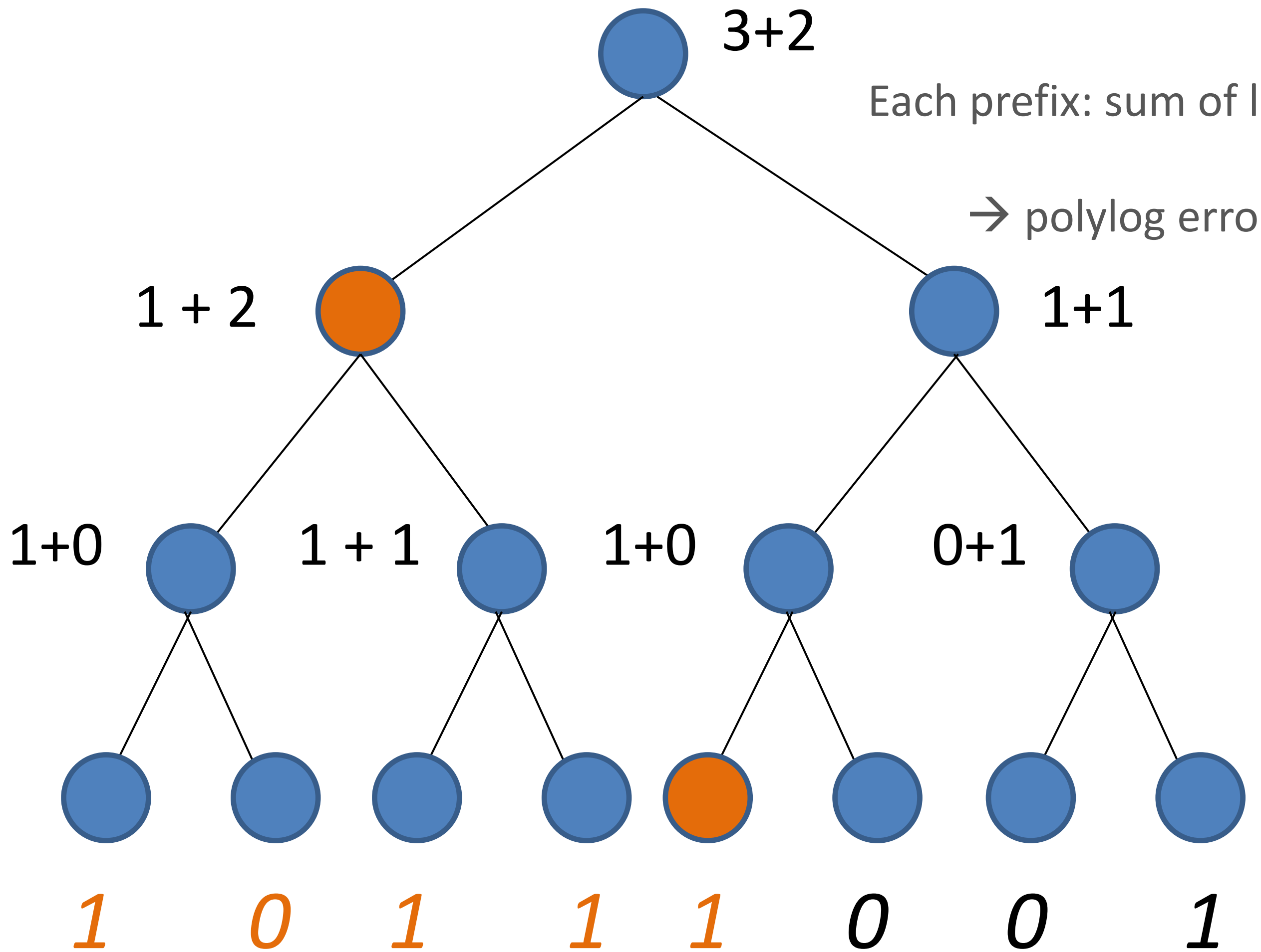
Binary Tree Technique [DPNR10], [Chan Shi Song 10]



Sensitivity of tree: log *m*

Add Lap(log *m*/ε) to each node

Binary Tree Technique



Each prefix: sum of log *m* nodes

→ 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!