DIFFERENTIALLY PRIVATE ALGORITHMS
Some Primitives and Paradigms

Kunal Talwar

Google Brain

DIFFERENTIAL PRIVACY RECAP

-

Databases d and $d' \in D^n$ are neighbors if they differ in one individual's contribution

(ε, δ)-Differential Privacy: For all d, d' neighbors, the distribution of $M(d)$ is (nearly) the same as the distribution of $M(d')$:

$$\forall S, \qquad \Pr[M(d) \in S] \leq \exp(\varepsilon) \cdot \Pr[M(d') \in S] + \delta$$

DIFFERENTIAL PRIVACY RECAP

-

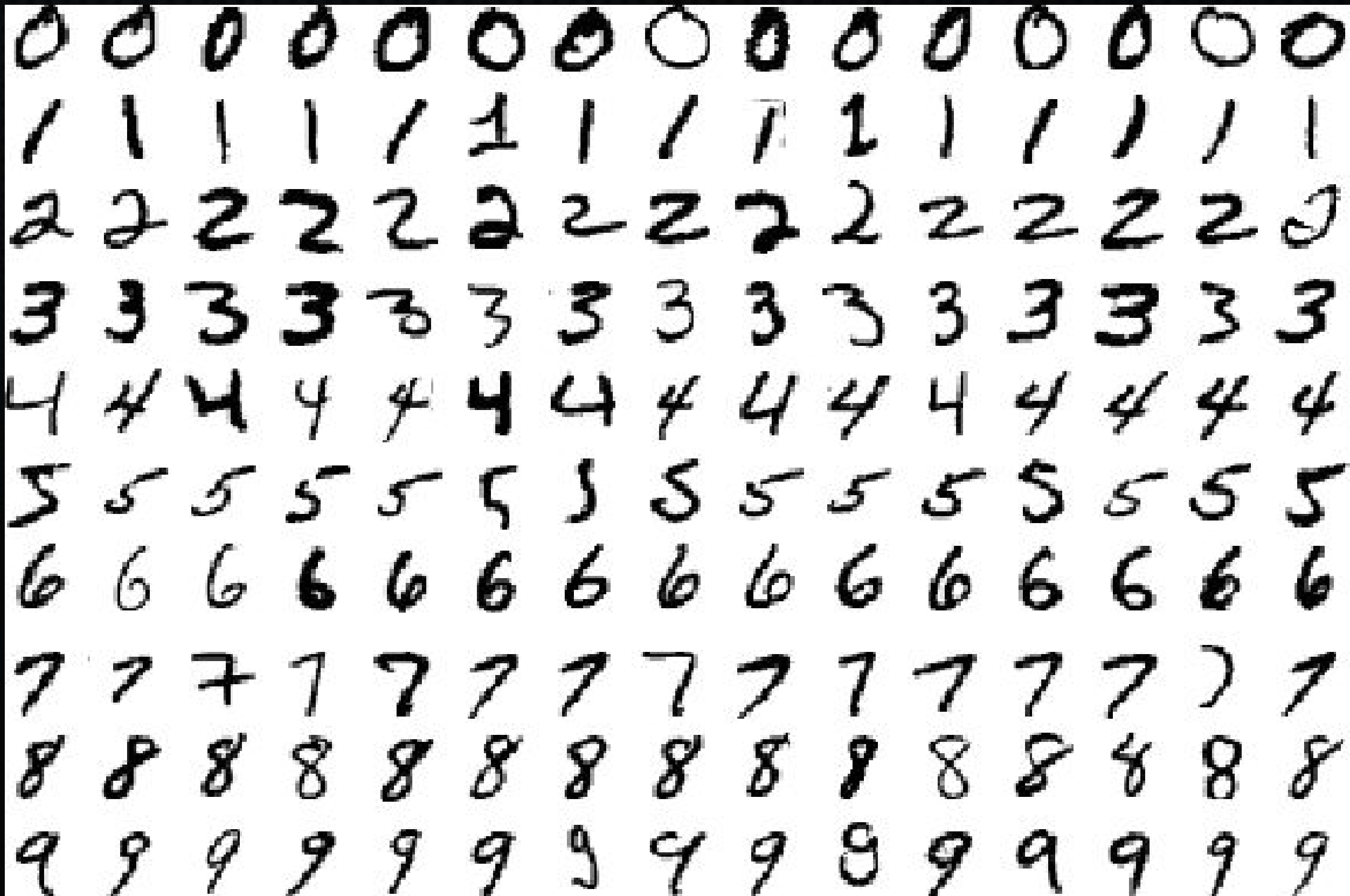
Composition: Allows us to bound privacy cost of sequence of DP algorithms

T runs of (ε, δ)-DP algorithm: $\left(\varepsilon\sqrt{T \ln \frac{1}{\delta}}, \delta(T+1)\right)$-DP

END-TO-END LEARNING A MODEL

Input: MNIST dataset containing handwritten digits, labeled 0-9

Goal: Learn to label fresh digits

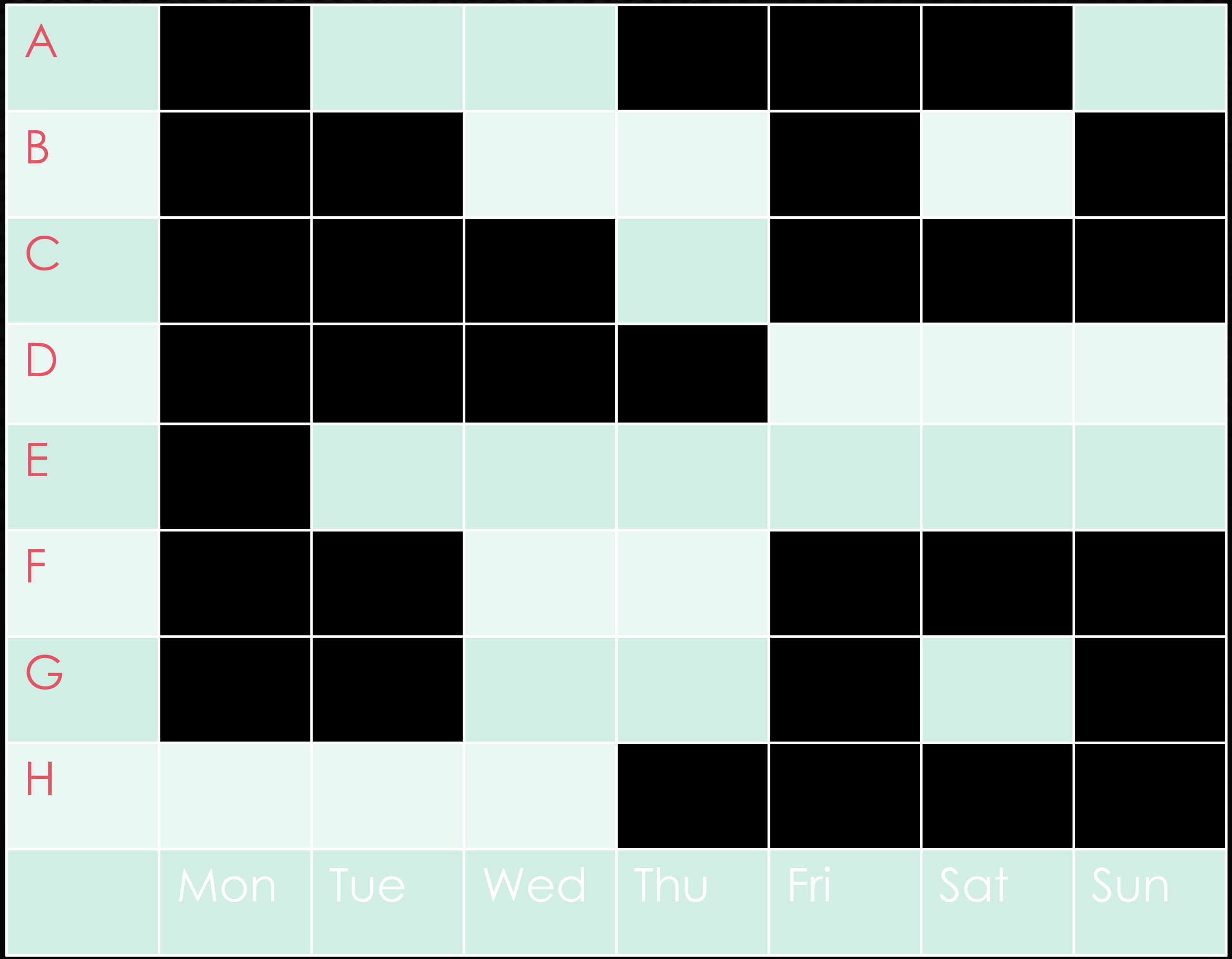


EXAMPLE: PICKING A GOOD DAY

Input: A range of feasible dates for an event.
Input: Conference center availability

Input: For each possible attendee, dates when they are available.

Goal: Find a date when most people are available



ALGORITHMS (with public and private inputs)

Input: A range of feasible dates for an event.
Input: Conference center availability

Input: For each possible attendee, dates when they are available.

Goal: Find a date when most people are available

For each feasible date t:
  If Conference Center available on t :
    Count[t] = number of people available on t
  else:
    Count[t] = 0

Output $argmax_t$ Count[t]

ALGORITHMS (with public and private inputs)

Input: A range of feasible dates for an event.
Input: Conference center availability

Input: For each possible attendee, dates when they are available.

Goal: Find a date when most people are available

```
For each feasible date t:
    If Conference Center available on t :
        Count[t] = number of people available on t
    else:
        Count[t] = 0

Output argmax_t Count[t]
```

ALGORITHMS (with public and private inputs)

Input: A range of feasible dates for an event.
Input: Conference center availability

Input: For each possible attendee, dates when they are available.

Goal: Find a date when most people are available

For each feasible date t:
  If Conference Center available on t :
    Count[t] = SAN(number of people available on t)
  else:
    Count[t] = 0

Output $argmax_t$ Count[t]

ALGORITHMS (with public and private inputs)

Input: A range of feasible dates for an event.
Input: Conference center availability

Input: For each possible attendee, dates when they are available.

Goal: Find a date when most people are available

SAN(
For each feasible date t:
  If Conference Center available on t :
    Count[t] = number of people available on t
  else:
    Count[t] = 0

Output $argmax_t$ Count[t])

PRIMITIVES

Simple San(.) ’s for some simple functions

SANITIZING A COUNT

- $f(d)$ = Number of people in d available on date t

SANITIZING A COUNT

$f(d)$ = Number of people in d available on date t

$$\mathit{Laplace}\left(\frac{1}{\varepsilon}\right)$$

$$\mathit{Density}(y) \propto \exp(-\varepsilon\,|y|)$$

$$\text{San}(\, f(d)\,) = f(d) + \text{Lap}\left(\frac{1}{\varepsilon}\right)$$

[Dwork-McSherry-Nissim-Smith 2006]

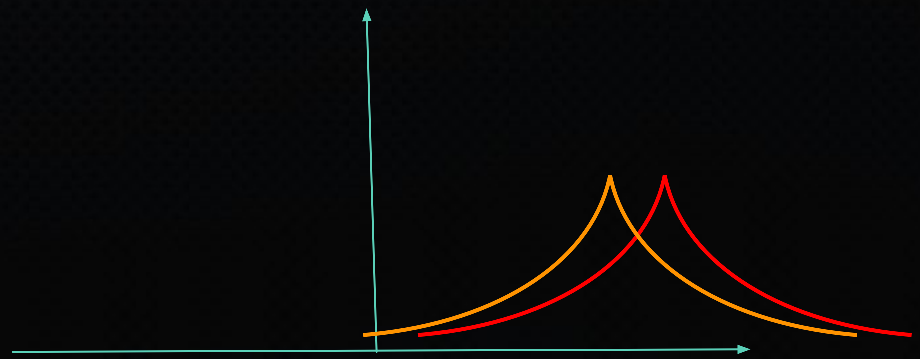
SANITIZING A COUNT

$f(d)$ = Number of people in d available on date t

$$\textit{Laplace}\left(\frac{1}{\varepsilon}\right)$$

$$\textit{Density}(y) \propto \exp(-\varepsilon\,|y|)$$

$\text{San}(\,f(d)\,) = f(d) + \text{Lap}(\frac{1}{\varepsilon})$



$|f(d) - f(d')| \leq 1$

Going from d to d' shifts distribution by 1.

[Dwork-McSherry-Nissim-Smith 2006]

SANITIZING A COUNT

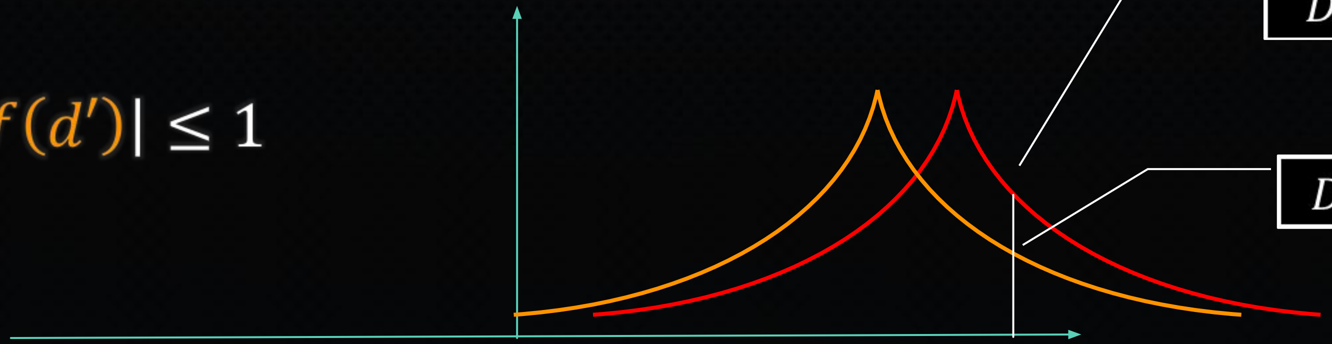
$f(d)$ = Number of people in d available on date t

$$\mathrm{San}(\,f(d)\,) = f(d) + \mathrm{Lap}\left(\frac{1}{\varepsilon}\right)$$

$$\mathit{Laplace}\left(\frac{1}{\varepsilon}\right)$$

$$\mathit{Density}(y) \propto \exp(-\varepsilon\,|y|)$$

$$|f(d) - f(d')| \leq 1$$



$$\mathit{Density}(y) \propto \exp(-\varepsilon\,|y - f(d)|)$$

$$\mathit{Density}(y) \propto \exp(-\varepsilon\,|y - f(d')|)$$

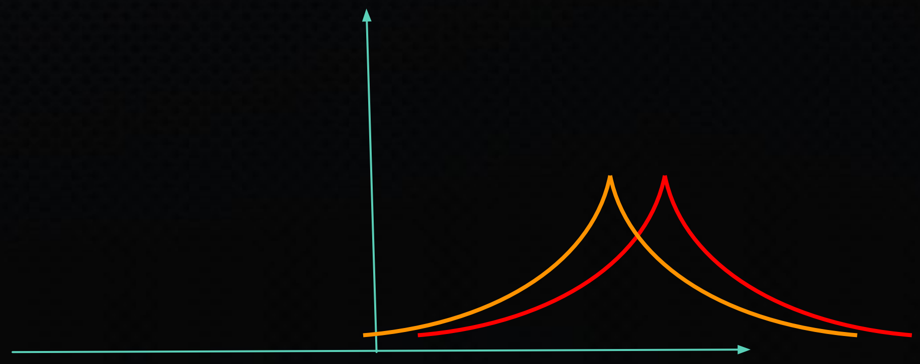
[Dwork-McSherry-Nissim-Smith 2006]

SANITIZING A COUNT

$f(d)$ = Number of people in d available on date t

San($f(d)$) = $f(d)$ + Lap($\frac{1}{\varepsilon}$)

$$|f(d) - f(d')| \leq 1$$



[Dwork-McSherry-Nissim-Smith 2006]

SANITIZING A LOW SENSITIVITY FUNCTION

$f(d)$ = Arbitrary sensitivity-1 function

$$|f(d) - f(d')| \leq 1$$

$$\text{San}(\ f(d)\ ) = f(d) + \text{Lap}(\tfrac{1}{\varepsilon})$$

How large is this noise?

Expected magnitude $\frac{1}{\varepsilon}$

[Dwork-McSherry-Nissim-Smith 2006]

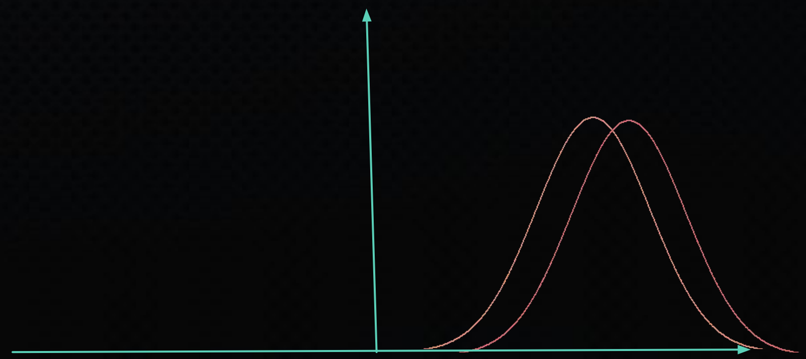
SANITIZING A LOW SENSITIVITY FUNCTION: GAUSSIAN

$f(d)$ = Arbitrary sensitivity-1 function

$$|f(d) - f(d')| \leq 1$$

Gaussian Distribution

San($f(d)$) = $f(d) + N(0, \sigma^2)$



Satisfies (ε, δ)-DP for a suitable σ

[Dwork-McSherry-Nissim-Smith 2006]

SANITIZING A LOW SENSITIVITY FUNCTION: GAUSSIAN

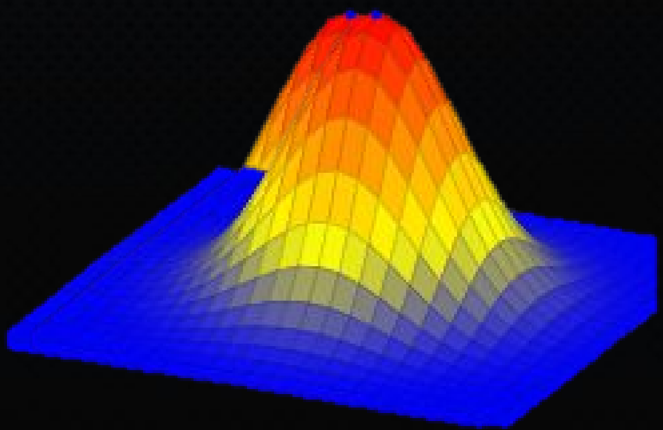
$f(d)$ = Arbitrary sensitivity-1 vector function

$$|f(d) - f(d')|_2 \leq 1$$

Multi-dimensional Gaussian

San($f(d)$) = $f(d) + N(0, \sigma^2 \mathbb{I})$

Satisfies (ε, δ)-DP for a suitable σ



[Dwork-McSherry-Nissim-Smith 2006]

SANITIZING A SELECTION: EXPONENTIAL

- General Output space: a set K of options
- Score function $q : D^n \times K \to \mathbb{R}$
- $|q(d,k) - q(d',k)| \leq 1, \ \forall k \in K, \forall d, d'$ neighboring
- E.g. Select a date to maximize number of attendees

[McSherry-T.- 2007]

SANITIZING A SELECTION: EXPONENTIAL

- General Output space: a set K of options
- Score function $q : D^n \times K \to \mathbb{R}$
- $|q(d,k) - q(d',k)| \leq 1, \ \forall k \in K, \forall d, d'$ neighboring
- E.g. Select a date to maximize number of attendees

San($argmax_k \; q(d,k)$) : Pick $k \in K$ with probability $\propto \exp(\varepsilon \, q(x,k))$

[McSherry-T.- 2007]

SANITIZING A SELECTION: EXPONENTIAL

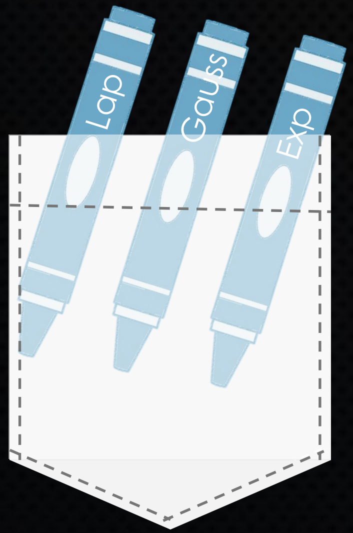
- General Output space: a set K of options
- Score function $q : D^n \times K \to \mathbb{R}$
- $|q(d,k) - q(d',k)| \le 1, \ \forall k \in K, \forall d, d'$ neighboring

Satisfies 2ε-DP

San($argmax_k\ q(d,k)$) : Pick $k \in K$ with probability $\propto \exp(\varepsilon\, q(x,k))$

Utility: $q(x, M(x)) \ge argmax_{k \in K}\ q(x,k) - O(\frac{\log k}{\varepsilon})$

[McSherry-T.- 2007]



PARADIGMS

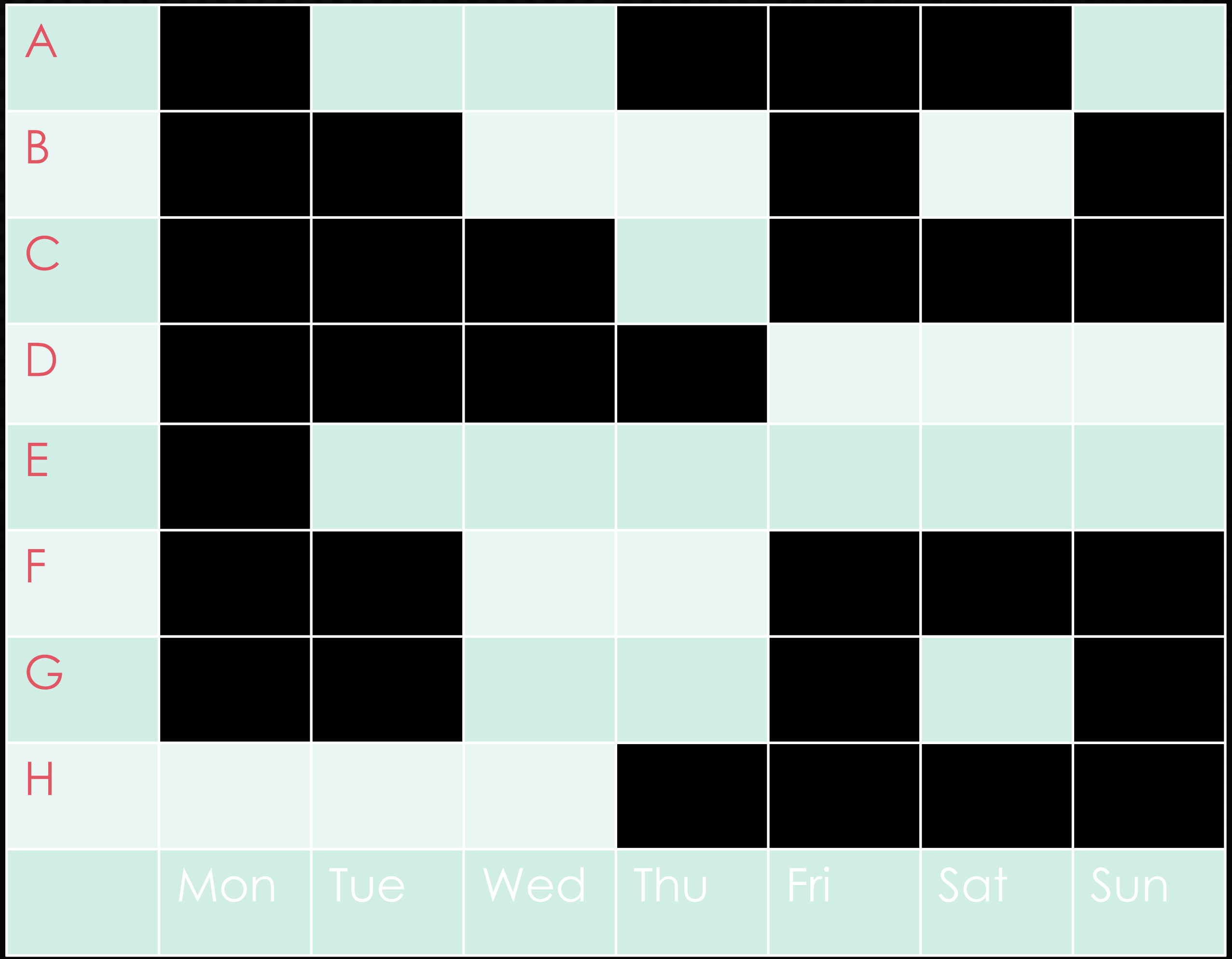
Using San(.) ’s for complex tasks

EXAMPLE: PICKING A GOOD DAY

Input: A range of feasible dates for an event.
Input: Conference center availability

Input: For each possible attendee, dates when they are available.

Goal: Find a date when most people are available



EXAMPLE: PICKING A GOOD DAY

Input: A range of feasible dates for an event.
Input: Conference center availability

Input: For each possible attendee, dates when they are available.

Goal: Find a date when most people are available

SAN(
For each feasible date t:
    If Conference Center available on t :
        Count[t] = number of people available on t
    else:
        Count[t] = 0

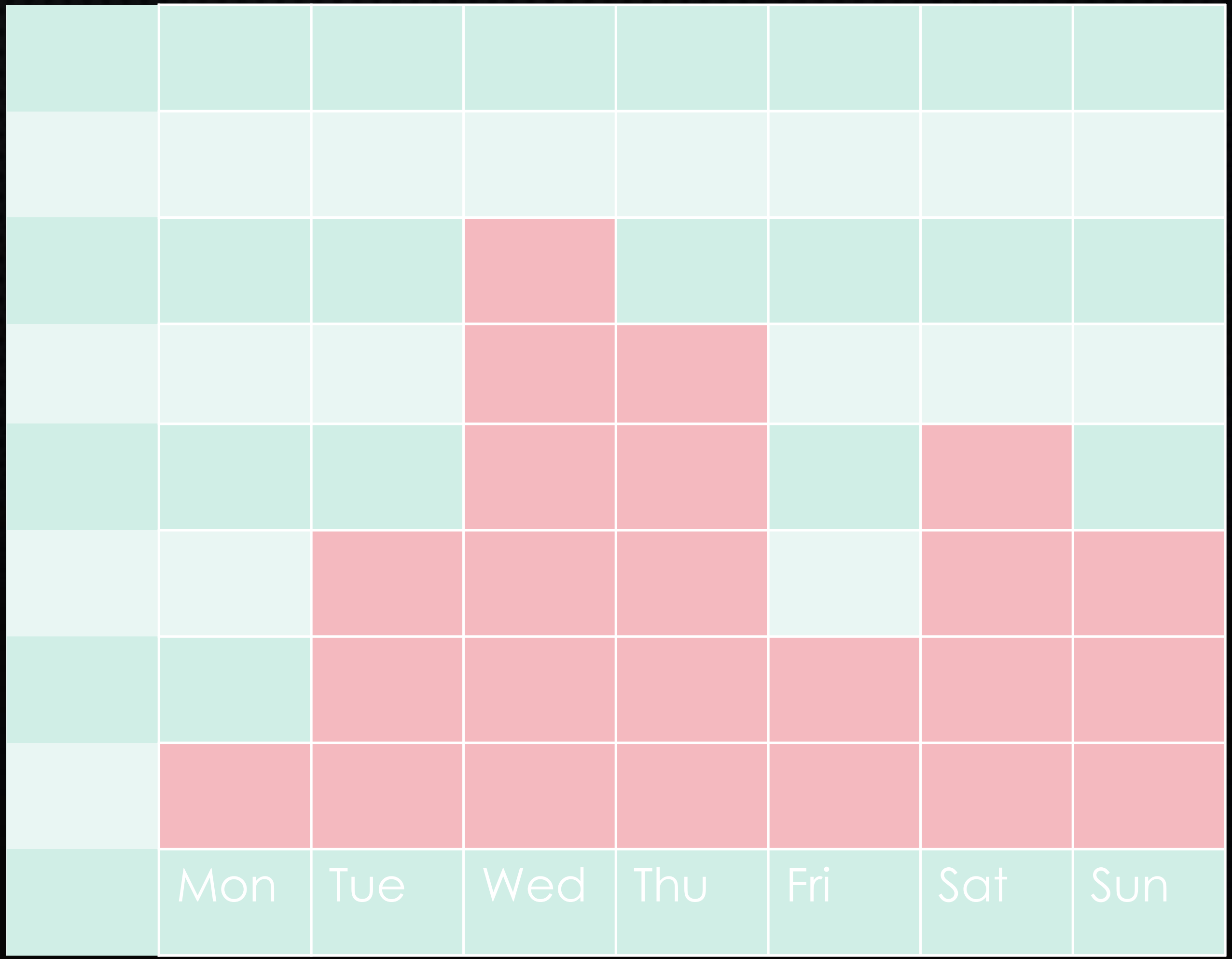
Output $argmax_t$ Count[t])

EXAMPLE: PICKING A GOOD DAY

Input: A range of feasible dates for an event.
Input: Conference center availability

Input: For each possible attendee, dates when they are available.

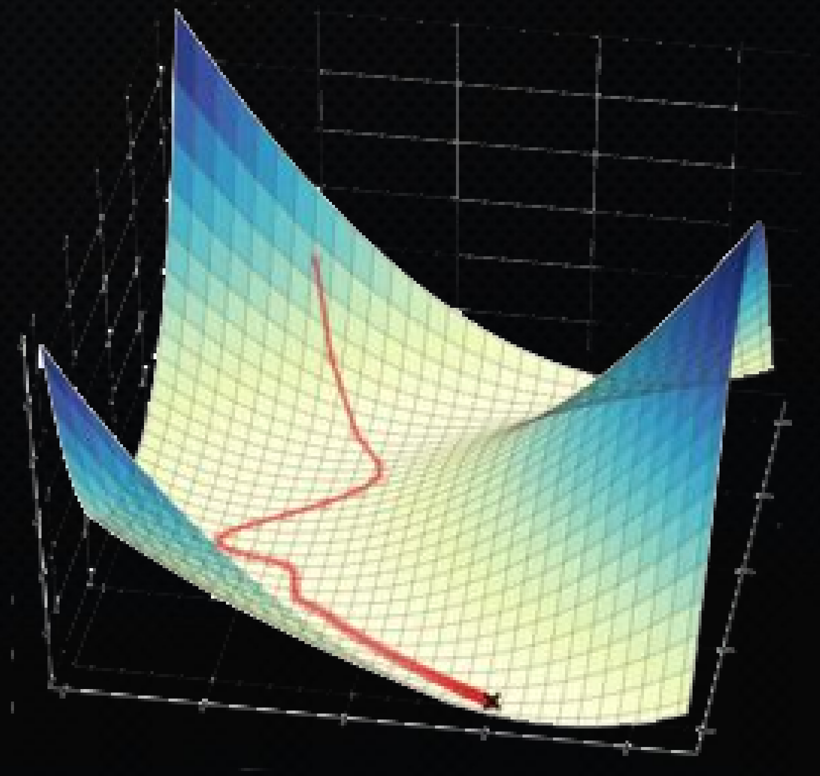
Goal: Find a date when most people are available



THE JUST-ADD-NOISE PARADIGM

Successful in numerous settings

- Releasing simple statistics
- Combinatorial Public Projects, Minimum cuts in graphs [Gupta-Ligett-McSherry-Roth-T.-09]
- Gradient Descent and Stochastic Gradient Descent [Wu-Kumar-Chaudhuri-Jha-Naughton-16]



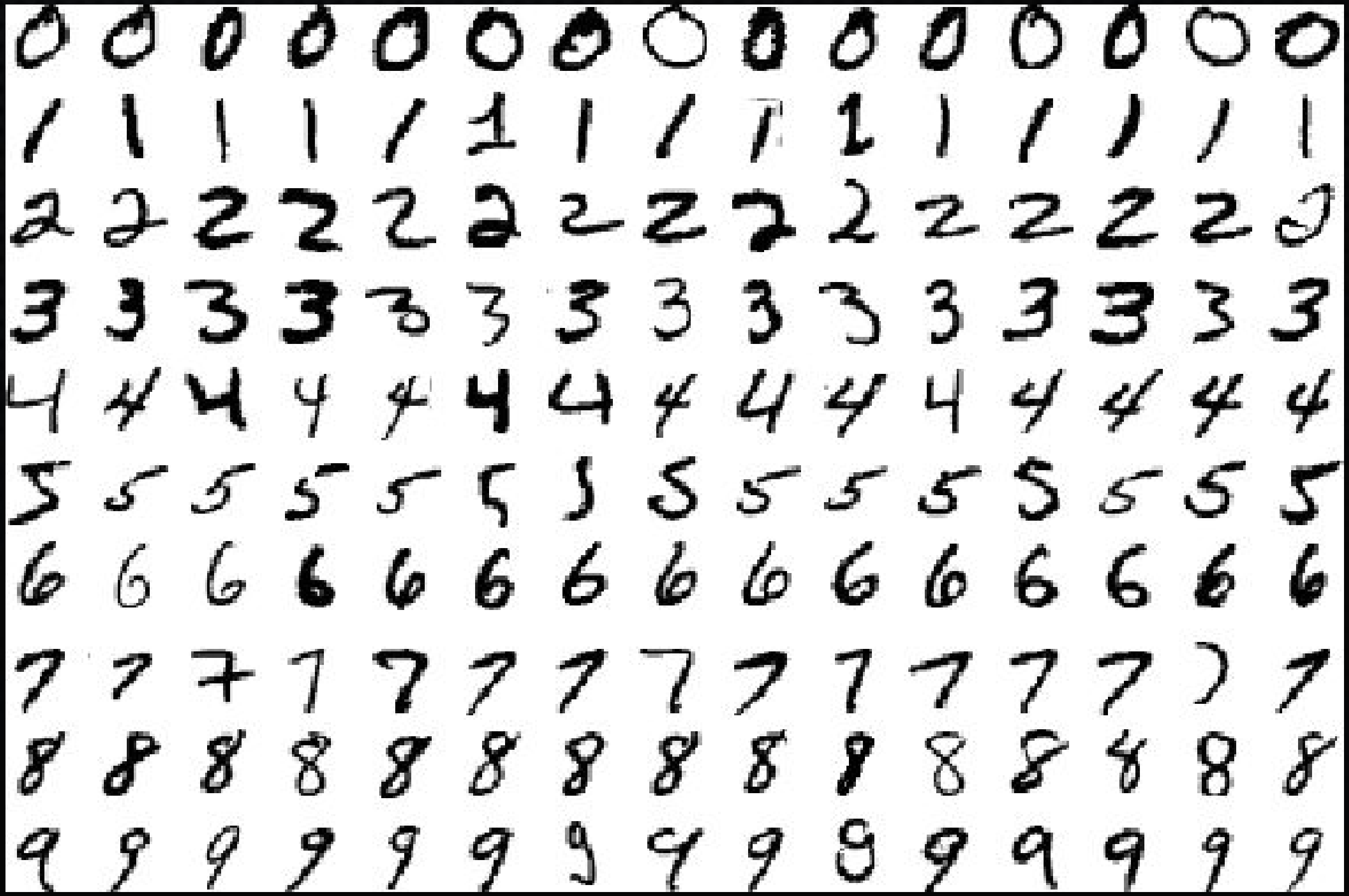
EXAMPLE: PRINCIPAL COMPONENT ANALYSIS

- Input: A number k

Input: For each person i, a vector v_i in $\mathbb{R}^m$, $|v_i|_2 \leq 1$

Goal: Output a rank-k projection Π maximizing the average squared projection length

$$\sum |\Pi v_i|_2^2$$



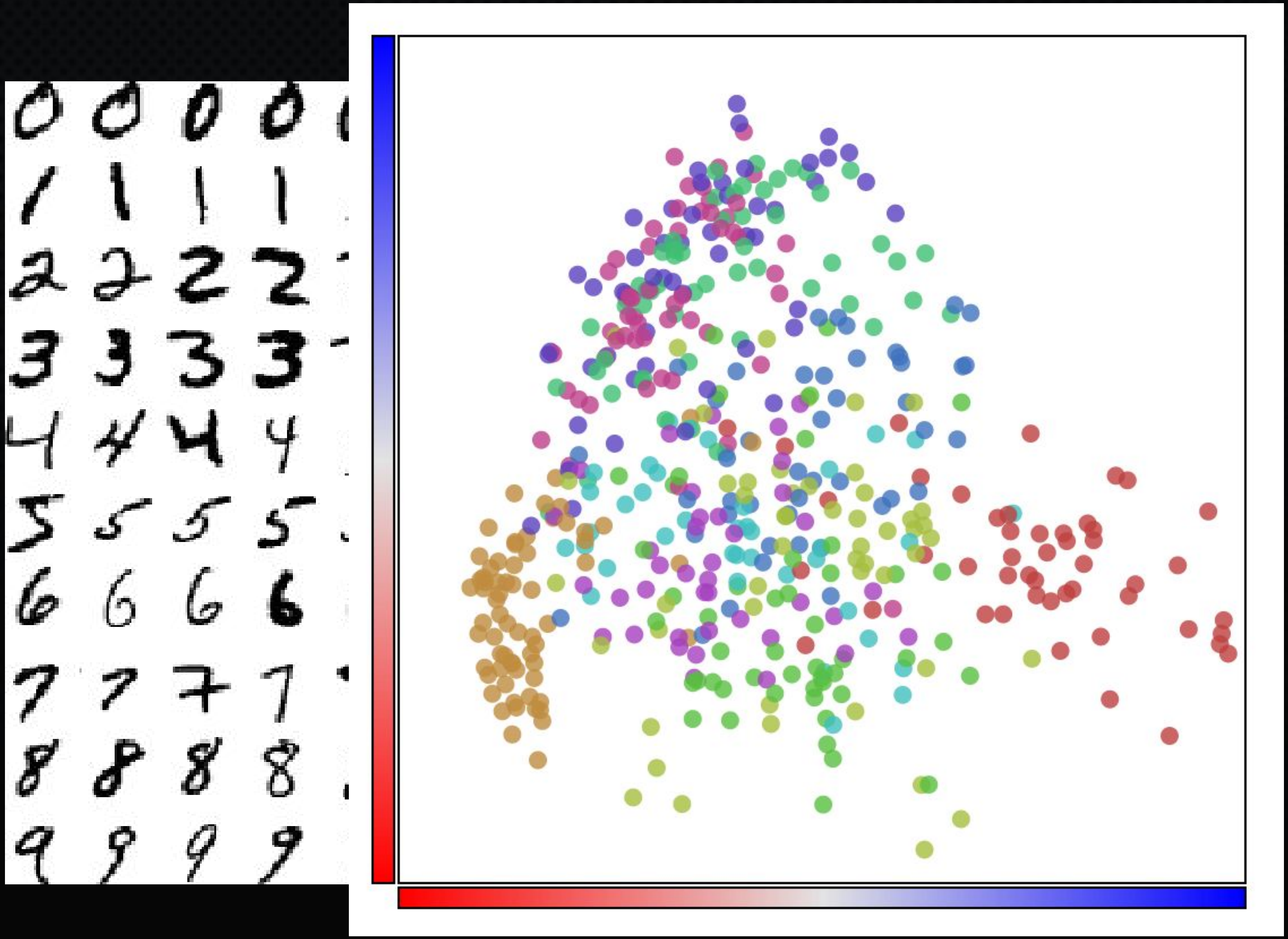
EXAMPLE: PRINCIPAL COMPONENT ANALYSIS

- Input: A number k

Input: For each person i, a vector v_i in $\mathbb{R}^m$, $|v_i|_2 \leq 1$

Goal: Output a rank-k projection Π maximizing the average squared projection length

$$\sum |\Pi v_i|_2^2$$



EXAMPLE: PRINCIPAL COMPONENT ANALYSIS

Input: A number k

Input: For each person i, a vector v_i in $\mathbb{R}^m$, $|v_i|_2 \leq 1$

Goal: Output a rank-k projection Π maximizing the average squared projection length

$$\sum |\Pi v_i|_2^2$$

A lot of work in DP PCA

[Blum-Dwork-McSherry-Nissim-04]
[McSherry-Mironov-09]
[Chaudhuri-Sarwate-Song-12]
[Hardt-Roth-12]
[Kapralov-T.-13]
[Hardt-Roth-13]
[Dwork-Thakurta-T.-Zhang-14]

EXAMPLE: PRINCIPAL COMPONENT ANALYSIS

- Input: A number k

Input: For each person i, a vector v_i in $\mathbb{R}^m$, $|v_i|_2 \leq 1$

Goal: Output a rank-k projection Π maximizing the average squared projection length

$$\sum |\Pi v_i|_2^2$$

$$A = \begin{bmatrix} v_1 \\ v_2 \\ \cdot \\ \cdot \\ \cdot \\ v_n \end{bmatrix}$$

Optimal Π given by top k eigenvectors of

$$C = A^T A = \sum_i v_i v_i^T$$

EXAMPLE: PRINCIPAL COMPONENT ANALYSIS

Input: A number k

Input: For each person i, a vector v_i in $\mathbb{R}^m$, $|v_i|_2 \le 1$

Goal: Output a rank-k projection Π maximizing the average squared projection length

$$\sum |\Pi v_i|_2^2$$

Compute $C = \sum_i v_i v_i^T$
Output the top k eigenvectors of C

EXAMPLE: PRINCIPAL COMPONENT ANALYSIS

- Input: A number k

Input: For each person i, a vector v_i in $\mathbb{R}^m$, $|v_i|_2 \leq 1$

Goal: Output a rank-k projection Π maximizing the average squared projection length

$$\sum |\Pi v_i|_2^2$$

C viewed as an m^2-dim. vector has senstivity 1

Compute $C = \sum_i v_i v_i^T$
Output the top k eigenvectors of C

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \rightarrow \begin{bmatrix} C_{11} \\ C_{12} \\ C_{12} \\ C_{21} \\ C_{22} \\ C_{23} \\ C_{31} \\ C_{32} \\ C_{33} \end{bmatrix}$$

EXAMPLE: PRINCIPAL COMPONENT ANALYSIS

Input: A number k

Input: For each person i, a vector v_i in $\mathbb{R}^m$, $|v_i|_2 \leq 1$

Goal: Output a rank-k projection Π maximizing the average squared projection length

$$\sum |\Pi v_i|_2^2$$

Compute San ($C = \sum_i v_i v_i^T$)
Output the top k eigenvectors of C

[Dwork-Thakurta-T.-Zhang-2014]

EXAMPLE: PRINCIPAL COMPONENT ANALYSIS

- Input: A number k

Input: For each person i, a vector v_i in $\mathbb{R}^m$, $|v_i|_2 \leq 1$

Goal: Output a rank-k projection Π maximizing the average squared projection length

$$\sum |\Pi v_i|_2^2$$

Compute San ($C = \sum_i v_i v_i^T$)
Output the top k eigenvectors of C

Can prove strong error bounds
Optimal under DP constraint

[Dwork-Thakurta-T.-Zhang-2014]

NOISE-UP-THE-RIGHT-OBJECT(S) PARADIGM

Find the ”right” algorithm
Sanitize the appropriate steps

- Recommendation Systems [McSherry-Mironov-09]
- Histogram Release [McSherry-Mironov-T.-10]
- Set Cover [Gupta-Ligett-McSherry-Roth-T.10]
- Gradient Descent [Chaudhuri-Sarwate-Song-12, Bassily-Thakurta-Smith-14]
- LASSO [T.-Thakurta-Zhang-15]

EXAMPLE: COUNT QUERIES

Input: A function $P : D \to [0,1]$

Input: For each person i, a row $d_i \in D$

Goal: Output sum of function evaluations

$$\sum_i P(d_i)$$

San (Compute $\sum_i P(d_i)$)
Output it

EXAMPLE: MULTIPLE COUNT QUERIES

Input: k functions

$$P_1, P_2, \dots, P_k : D \to [0,1]$$

Input: For each person i, a row $d_i \in D$

Goal: Output number of rows that satisfy each predicate

$$\left\{ \sum_i P_j(d_i) \right\}_{j \in [k]}$$

EXAMPLE: MULTIPLE COUNT QUERIES

Input: A vector function

$$P : D \rightarrow [0,1]^k$$

Input: For each person i, a row $d_i \in D$

Goal: Output sums of vector function evaluations

$$\sum_i P(d_i)$$

k-dimensional vector query

Noise = l_2 Sensitivity = $\sqrt{k}$

San (Compute $\sum_i P(d_i)$)

Output it

EXAMPLE: MULTIPLE COUNT QUERIES

Input: A vector function

$$P : D \rightarrow [0,1]^k$$

Input: For each person i, a row $d_i \in D$

Goal: Output sums of vector function evaluations

$$\sum_i P(d_i)$$

Noise = l_2 Sensitivity = $\sqrt{k}$

Can we do better?
In general: NO. Lower bounds via discrepancy.

[Dinur-Nissim-2003, Muthukrishnan-Nikolov-2012]

EXAMPLE: MULTIPLE COUNT QUERIES

Input: A vector function

$$P : D \to [0,1]^k$$

Input: For each person i, a row $d_i \in D$

Goal: Output sums of vector function evaluations

$$\sum_i P(d_i)$$

Noise = l_2 Sensitivity = $\sqrt{k}$

Can we do better?
For Specific Queries?

EXAMPLE: MULTIPLE COUNT QUERIES

Input: A vector function

$$P : D \to [0,1]^k$$

Input: For each person i, a row $d_i \in D$

Goal: Output sums of vector function evaluations

$$\sum_i P(d_i)$$

Noise = l_2 Sensitivity = $\sqrt{k}$

Can we do better?

$$P_1 = P_2 = \ldots = P_k$$

EXAMPLE: MULTIPLE COUNT QUERIES

Input: A vector function

$$P : D \to [0,1]^k$$

Input: For each person i, a row $d_i \in D$

Goal: Output sums of vector function evaluations

$$\sum_i P(d_i)$$

Noise = l_2 Sensitivity = $\sqrt{k}$

Can we do better?

$$P_1 = P_2 = \ldots = P_k$$

Can we exploit dependencies?

EXAMPLE: MULTIPLE COUNT QUERIES

Input: A vector function

$$P : D \to [0,1]^k$$

Input: For each person i, a row $d_i \in D$

Goal: Output sums of vector function evaluations

$$\sum_i P(d_i)$$

Noise = l_2 Sensitivity = $\sqrt{k}$

Can we exploit dependencies?

YES! Depends on the geometry of the vectors $\{P(d)\}_{d \in D}$

[Hardt-T.-10, Bhaskara-Dadush-Krishnaswamy-T.-12, Nikolov-T.-Zhang-12]

EXAMPLE: MULTIPLE COUNT QUERIES

Input: A vector function

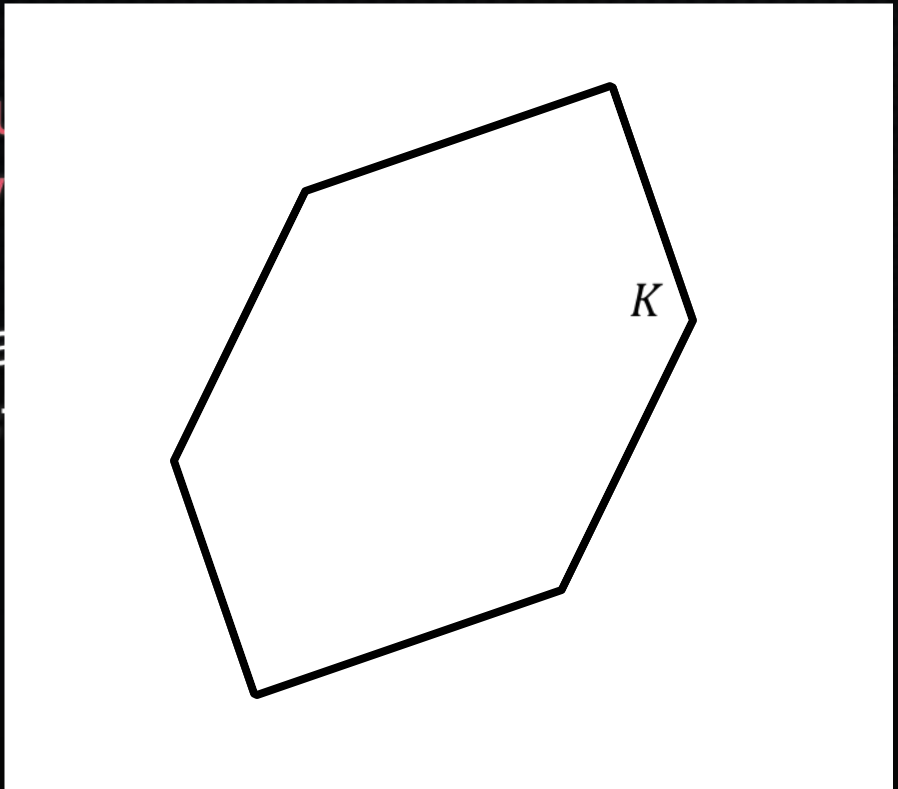
$$P : D \to [0,1]^k$$

Inpu

row

Goa

vec



Noise = l_2 Sensitivity = $\sqrt{k}$

Can we exploit dependencies?

YES! Depends on the geometry of the vectors $\{P(d)\}_{d \in D}$

Convex body

$$K = conv\{\, P(d) : d \in D \,\}$$

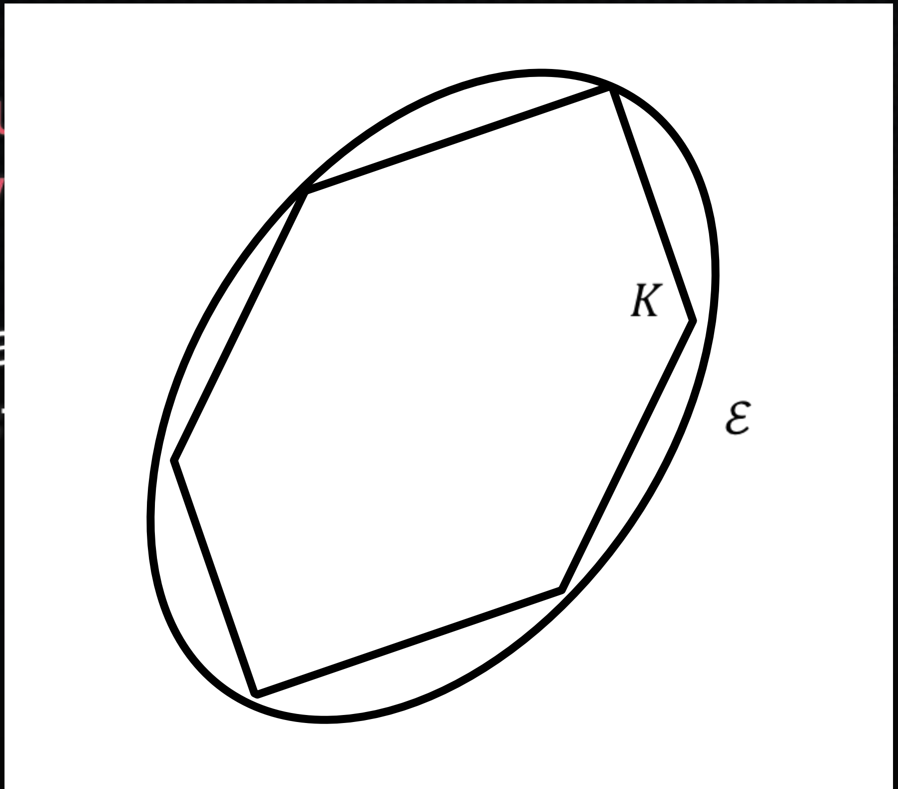
[Hardt-T.-10, Bhaskara-Dadush-Krishnaswamy-T.-12, Nikolov-T.-Zhang-12]

EXAMPLE: MULTIPLE COUNT QUERIES

Input: A vector function
$$P : D \to [0,1]^k$$

Inpu
row

Goa
vec



Noise = l_2 Sensitivity = $\sqrt{k}$

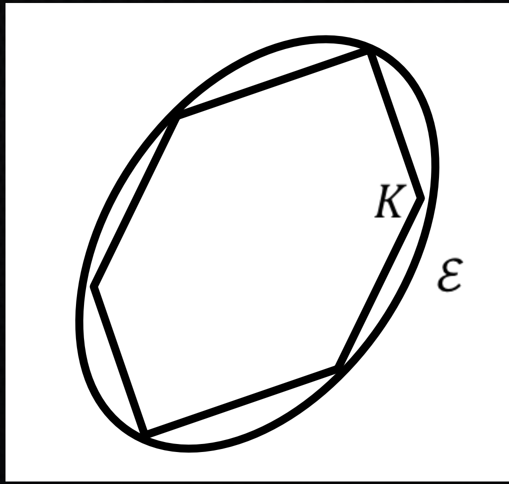
Can we exploit dependencies?

YES! Depends on the geometry of the vectors $\{P(d)\}_{d \in D}$

Tailor the noise to the geometry of the query set

[Hardt-T.-10, Bhaskara-Dadush-Krishnaswamy-T.-12, Nikolov-T.-Zhang-12]

INTERLUDE: GEOMETRY & DISCREPANCY



- Lower bounds use Discrepancy
  - Hereditary Discrepancy
- Upper bounds use Geometry
  - Bourgain-Tzafriri Restricted Invertibility
- Upper and lower bounds match up to logarithmic factors

Leads to new connection between Discrepancy Theory and Geometry

$$HerDisc(A) \approx \gamma_2(A)$$

Polylogarithmic approximation to Hereditary Discrepancy
Progress on the Tusnady problem

[Nikolov-T.-Zhang-12, Matousek-Nikolov-T.-15]

EXAMPLE: MANY MULTIPLE COUNT QUERIES

Input: A vector function

$$P : D \rightarrow [0,1]^k$$

Input: For each person i, a row $d_i \in D$

Goal: Output sums of vector function evaluations

$$\sum P(d_i)$$

Noise = l_2 Sensitivity = $\sqrt{k}$

What if $k \gg n$?

Different kind of dependencies

[Blum-Ligett-Roth-08, Hardt-Rothblum-10]

EXAMPLE: MANY MULTIPLE COUNT QUERIES

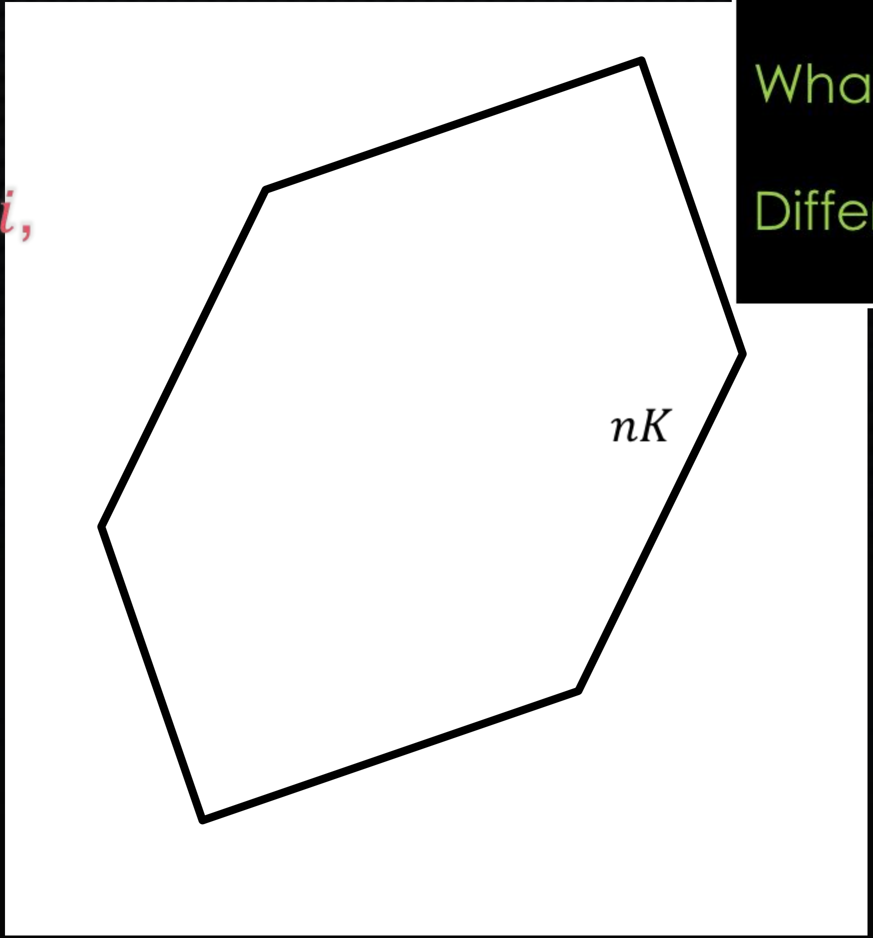
Input: A vector function

$$P : D \to [0,1]^k$$

Input: For each person i, a row $d_i \in D$

Goal: Output sums of vector function evaluations

$$\sum P(d_i)$$



Noise = l_2 Sensitivity = $\sqrt{k}$

What if $k \gg n$?

Different kind of dependencies

EXAMPLE: MANY MULTIPLE COUNT QUERIES

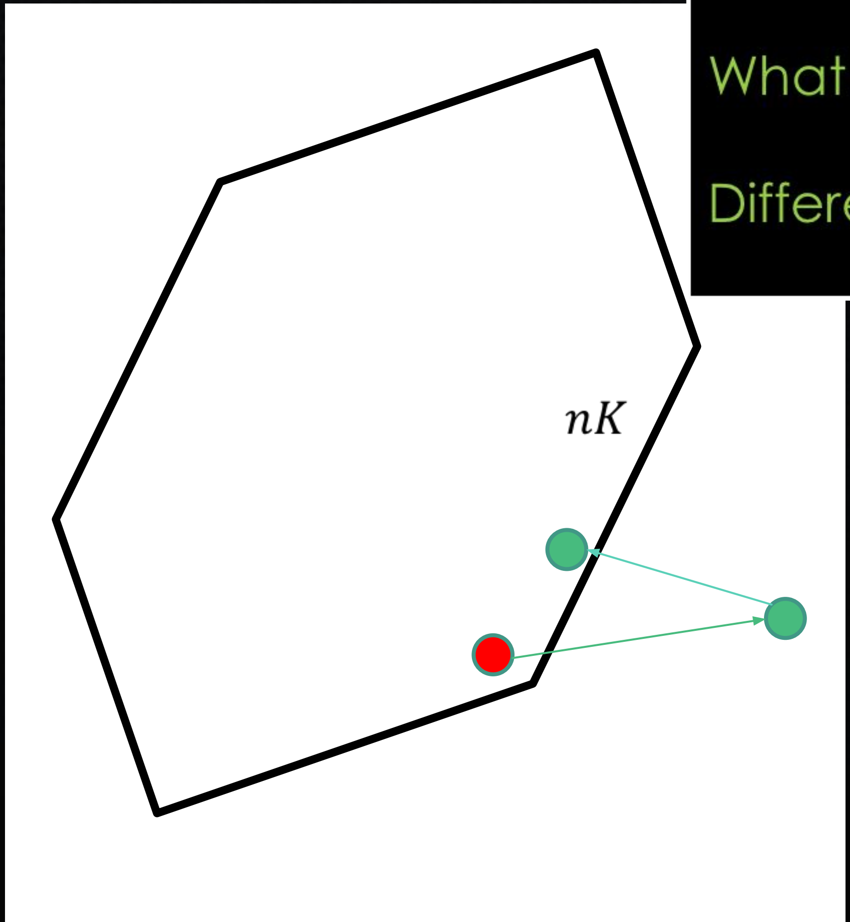
Input: A vector function

$$P : D \to [0,1]^k$$

Input: For each person i, a row $d_i \in D$

Goal: Output sums of vector function evaluations

$$\sum P(d_i)$$



Noise = l_2 Sensitivity = $\sqrt{k}$

What if $k \gg n$?

Different kind of dependencies

Gives nearly optimal error:

$$\sim \sqrt{n} \log k$$

[Nikolov-T.-Zhang-14]

USE-AVAILABLE-INFORMATION-TO-POSTPROCESS

Often answers must satisfy some constraints
Noised-up answers may violate them. Project to enforce constraint.

- Contingency Table Release [Barak-Chaudhuri-Dwork-Kale-McSherry-T.-2007]
- Unattributed Histograms [Hay-Rastogi-Miklau-Suciu-09]
- Degree Distribution [Hay-Li-Miklau-Jensen-09]
- Bayesian Inference [McSherry-Williams-10]
- Covariance matrix release [Sheffet-16]

USE-AVAILABLE-INFORMATION-TO-PREPROCESS

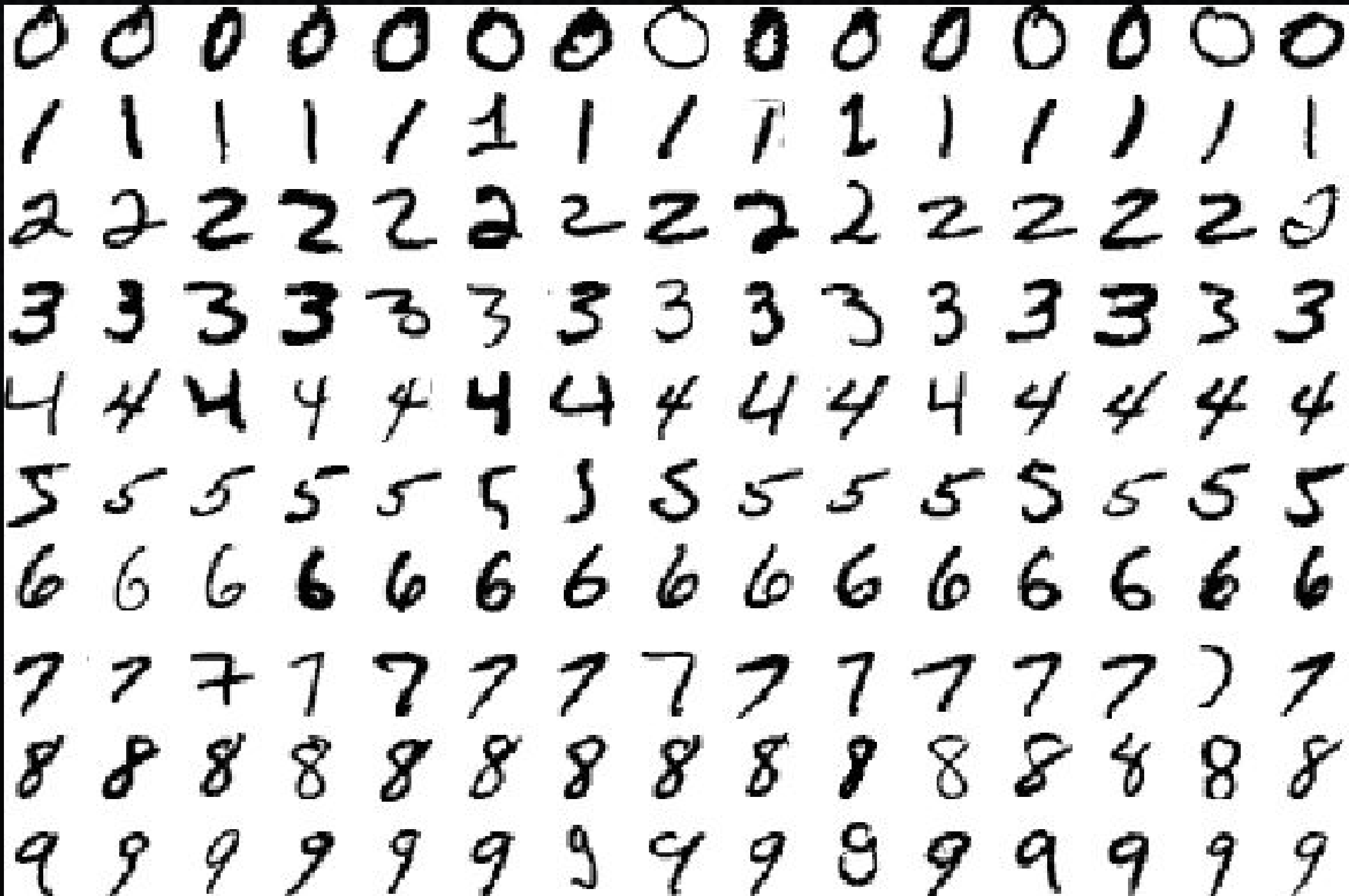
Often we know some property of database that makes problem easier
Use information to transform query to an easier one

- Releasing graph statistics, Graph synthesis [Proserpio-Goldberg-McSherry-13]
- Graph properties under node privacy [Blocki-Blum-Datta-Sheffet-13, Kasiviswanathan-Nissim-Raskhodnikova-Smith-13, Chen-Zhou-13, Raskhodnikova-Smith-15]
- Propose-Test-Release framework [Dwork-Lei-09]

END-TO-END LEARNING A MODEL

Input: MNIST dataset containing handwritten digits, labeled 0-9

Goal: Learn to label fresh digits

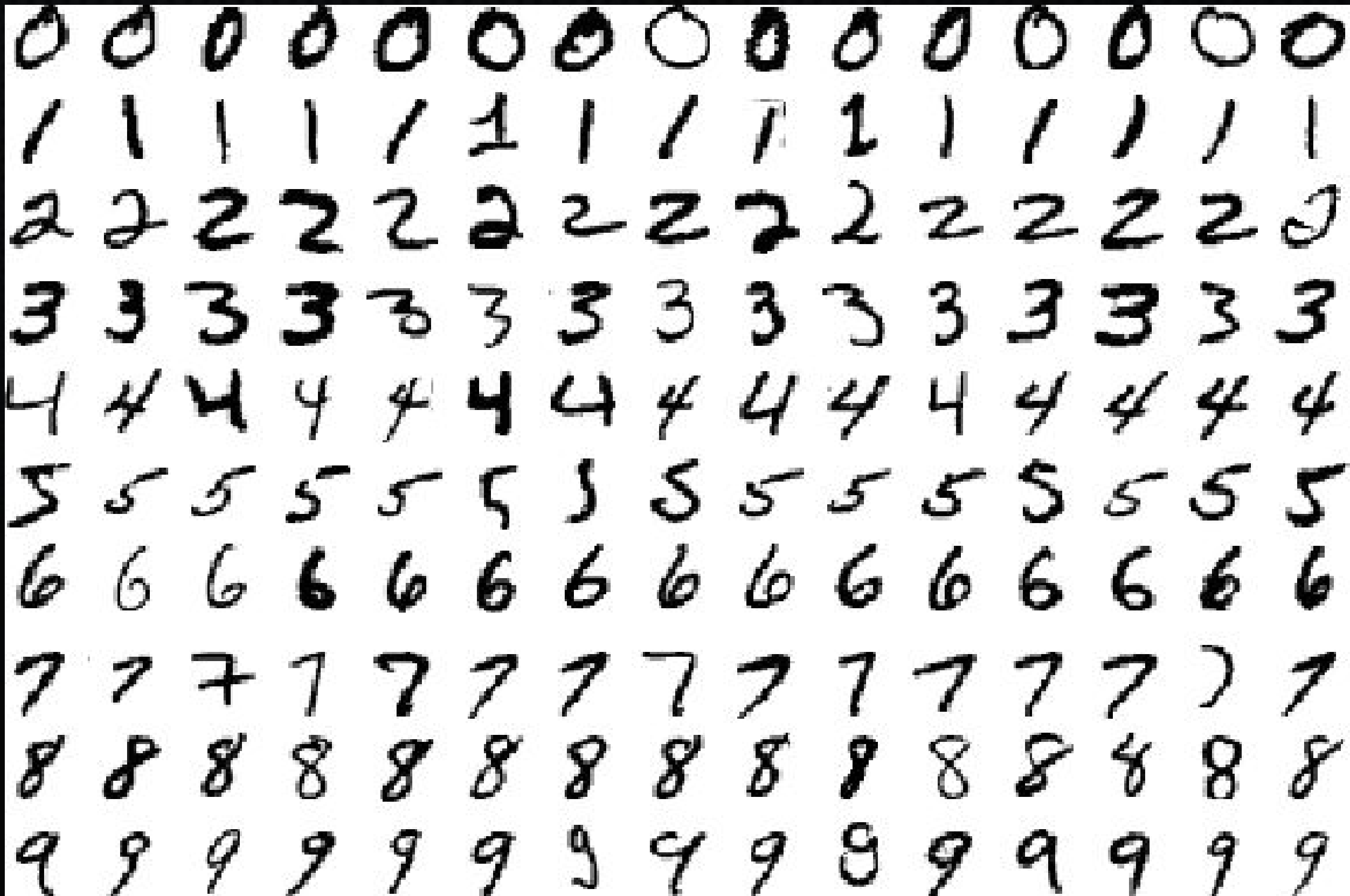


END-TO-END LEARNING A MODEL

Input: Neural Network architecture.

Input: MNIST dataset containing handwritten digits, labeled 0-9

Goal: Learn parameters of a model to label fresh digits



END-TO-END LEARNING A MODEL

Input: Neural Network architecture.

Input: MNIST dataset containing handwritten digits, labeled 0-9

Goal: Learn parameters θ of a model to minimize loss

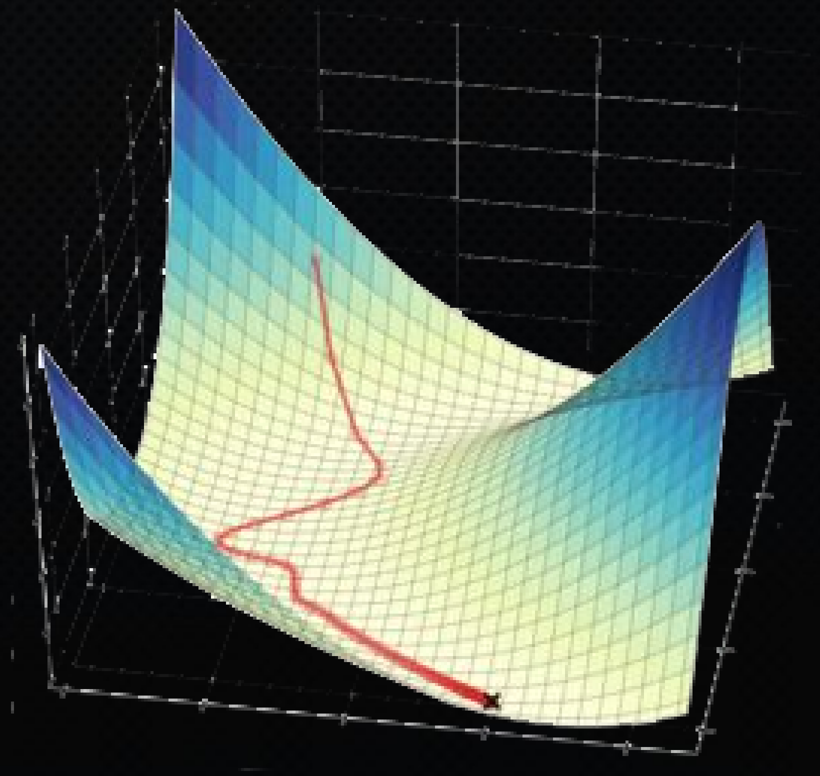
MAKING DEEP NETWORKS PRIVATE

- Just-add-noise paradigm fails
  - Sensitivity is large

- Noise-up-the-right-objects paradigm
  - Can take standard non-private algorithm and add appropriate noise
  - Naïve analysis results in large privacy cost

STOCHASTIC GRADIENT DESCENT

Start at a random point θ
For $t = 1 \ldots T$:
    Pick a small *batch* of examples
    Compute average Gradient g of Loss for examples in batch
    Move in that direction: $\theta \leftarrow \theta - \eta g$



STOCHASTIC GRADIENT DESCENT

Start at a random point θ
For $t = 1 \ldots T$:
    Pick a small *batch* of examples
    San (Compute average Gradient g of Loss for examples in batch)
    Move in that direction: $\theta \leftarrow \theta - \eta g$

Naïve privacy analysis:

Bound privacy cost of each step
Use Strong composition

T usually huge. Get bad bounds

PRIVACY AMPLIFICATION BY SAMPLING

- Take your favorite (ε, δ) –DP San
- Run it on a random q fraction of the data

- This new San is $(2q\varepsilon, q\delta)$-DP

[Kasiviswanathan-Lee-Nissim-Raskhodnikova-Smith-08]

STOCHASTIC GRADIENT DESCENT

Start at a random point θ
For $t = 1 \ldots T$:
    Pick a small *batch* of examples
    San (Compute average Gradient g of Loss for examples in batch)
    Move in that direction: $\theta \leftarrow \theta - \eta g$

Sampling by amplification helps.

We prove a stronger composition theorem for such mechanisms

[Abadi-Chu-Goodfellow-McMahan-Mironov-[illegible]ang-16]

STOCHASTIC GRADIENT DESCENT

San(Compute PCA of data)
Start at a random point θ
For $t = 1 \ldots T$:
    Pick a small *batch* of examples
    San (Compute av. Gradient g of Loss for PCA-projected examples in batch)
    Move in that direction: $\theta \leftarrow \theta - \eta g$

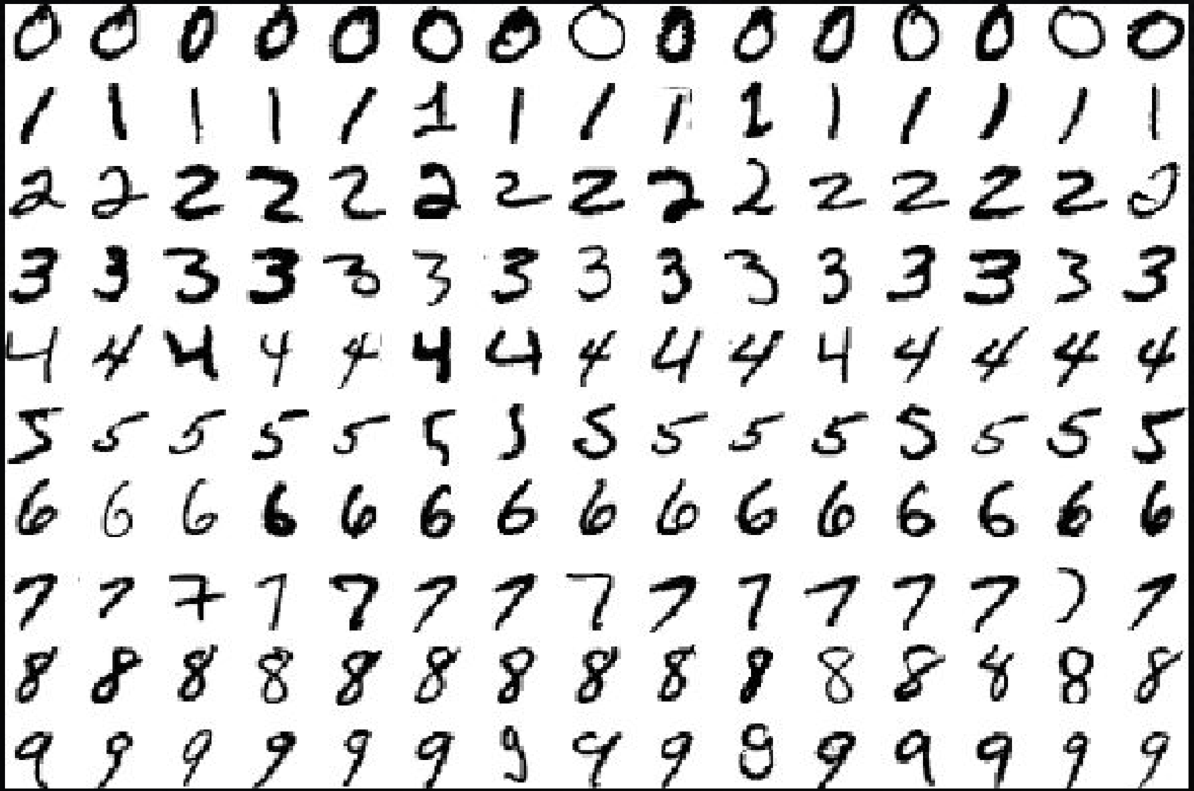
Sampling by amplification helps.

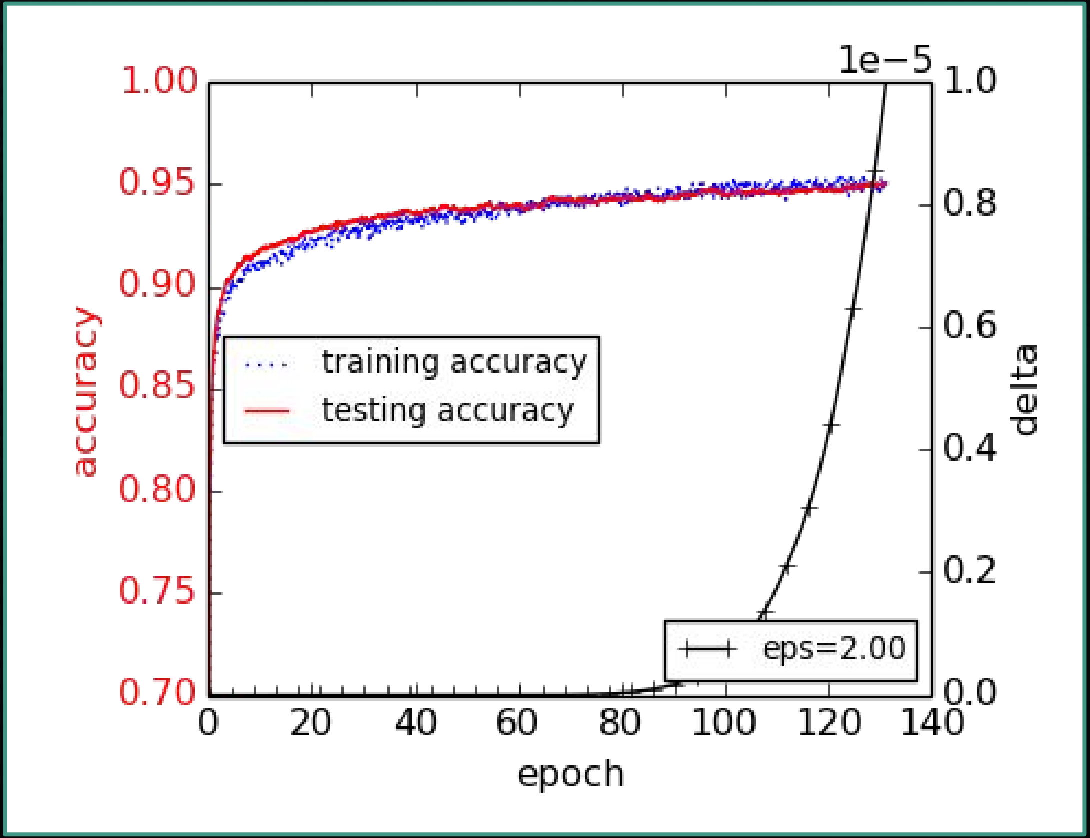
We prove a stronger composition theorem for such mechanisms

[Abadi-Chu-Goodfellow-McMahan-Mironov-Talwar-Zhang-16]

END-TO-END MODEL TRAINING

95 % accuracy with $(2, 10^{-5})$-DP





[Abadi-Chu-Goodfellow-McMahan-Mironov-T.-Zhang-16]

NOISE-ON-SAMPLE PARADIGM

SGD is the most common learning method for a large class of problems
Other algorithms such as EM can also be made private using this approach

- Convex loss Empirical Risk Minimization [Bassily-Thakurta-Smith-14]
- Topic Modeling [Park-Foulds-Chaudhuri-Welling-16]
- Expectation Maximization [Park-Foulds-Chaudhuri-Welling-16]
- Variational Inference [Jalko-Dikmen-Honkela-16, Park-Foulds-Chaudhuri-Welling-16]

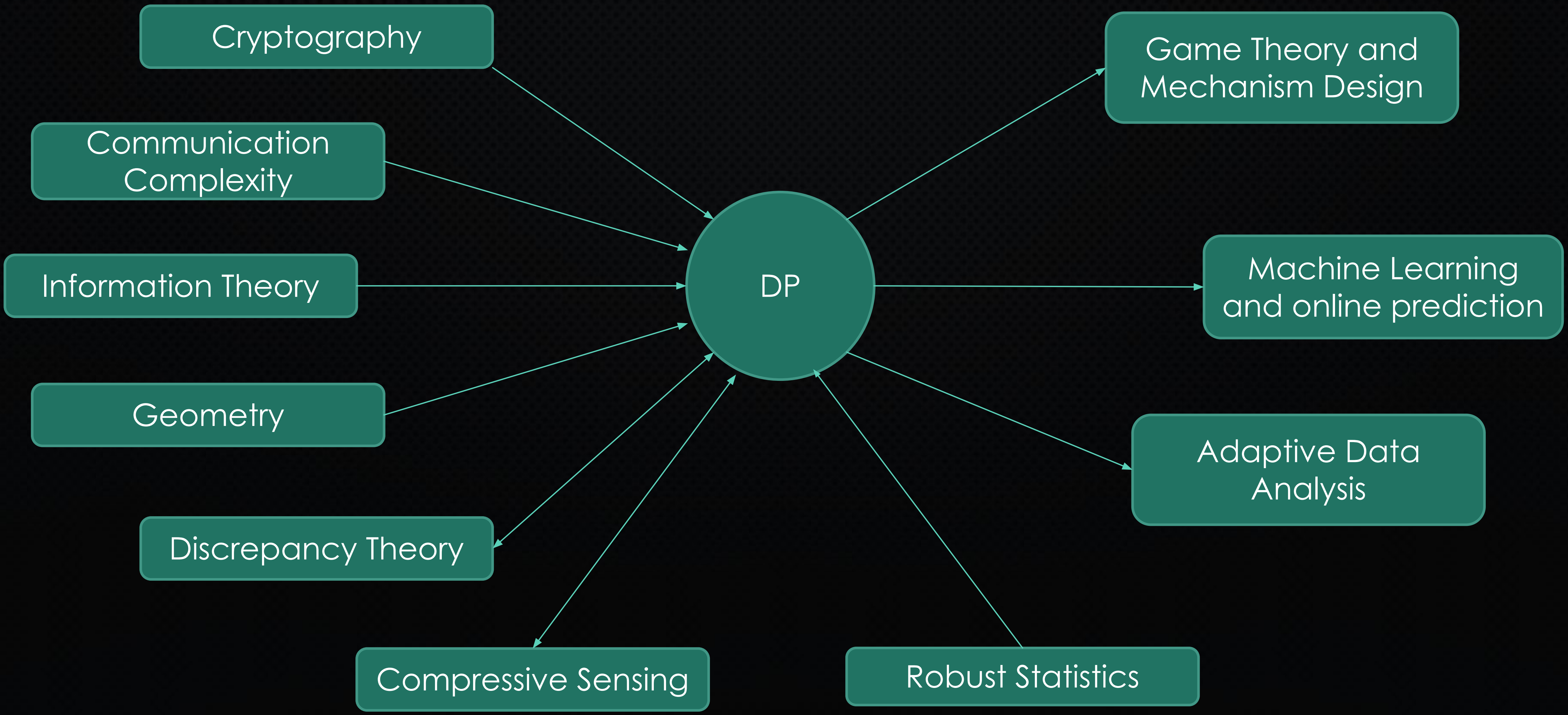
OTHER PARADIGMS

- Sparse Vector Technique
- Smoothed Sensitivity
- Aggregation
- Roll-up-your-sleeves

- Local Differential Privacy
- Multiparty Differential Privacy
- Privacy under Continual Observation
- Weaker privacy models

SUMMARY

- A sampling of primitives and paradigms
- A lot of tasks can be done with little loss in utility
- Noise addition does not work $\neq$ DP does not work
- Deep connections to other fields of study


Cryptography
Communication Complexity
Information Theory
Geometry
Discrepancy Theory
Compressive Sensing
DP
Game Theory and Mechanism Design
Machine Learning and online prediction
Adaptive Data Analysis
Robust Statistics