Efficient Private Mean Estimation

Gautam Kamath
University of Waterloo

Georgia Tech AI4OPT Seminar Series

February 2, 2023

STOC 2022

With Samuel B. Hopkins (MIT) and Mahbod Majid (University of Waterloo MMath → CMU PhD)

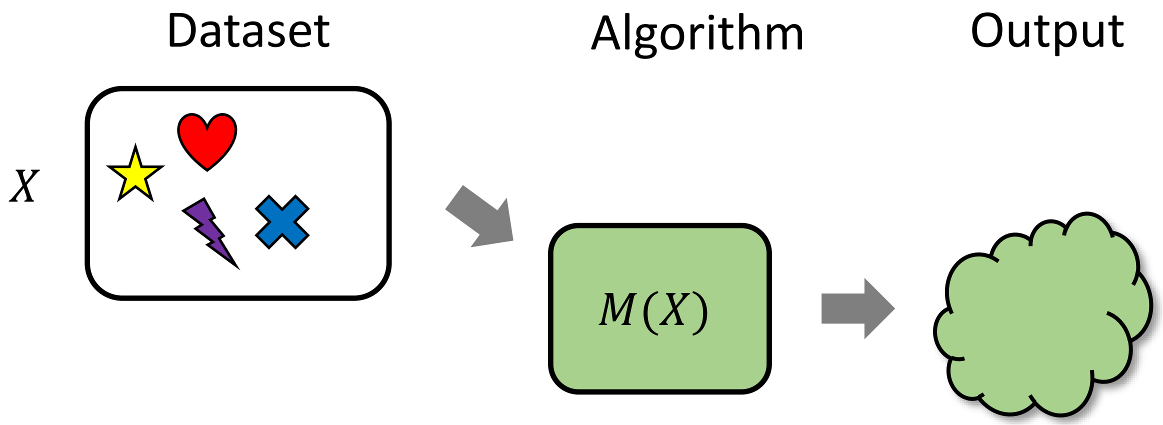
A Sneak Preview

- An all-in-one private mean estimator!
- Efficient, private, with near-optimal sample complexity
- But also robust with sub-Gaussian rates
- More broadly:
  - Addresses a fundamental deficiency in our understanding of DP estimation
  - Algorithmic high-dimensional statistics strikes again!
- A new area to explore for multiple communities
  - Robust Statistics
  - Sum-of-Squares

Privacy?

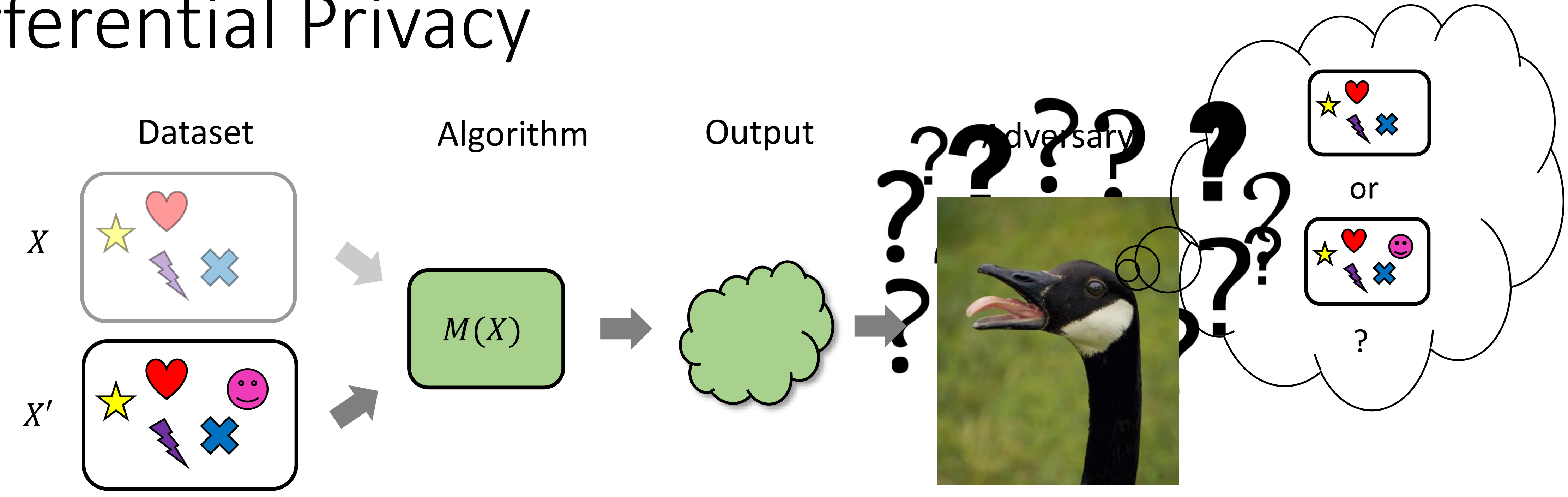
How do we ensure statistics don’t leak information about individual datapoints?

Differential Privacy



[Dwork, McSherry, Nissim, Smith], 2006

Differential Privacy

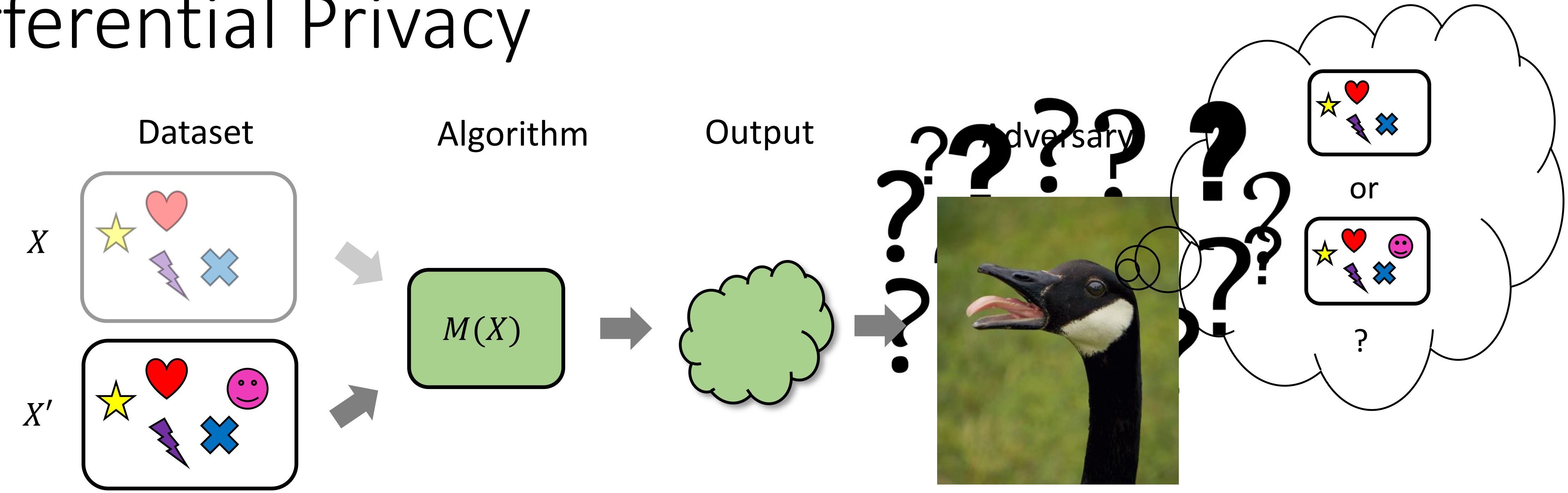


- $M: D^n \to R$ is (ε, δ)-DP if for all inputs X, X' which differ on one entry:

$$\forall S \subseteq R \qquad \Pr[M(X) \in S] \approx_{\varepsilon,\delta} \Pr[M(X') \in S]$$

[Dwork, McSherry, Nissim, Smith], 2006

Differential Privacy



- $M: D^n \to R$ is (ε, δ)-DP if for all inputs X, X' which differ on one entry:

$$\forall S \subseteq R \qquad \Pr[M(X) \in S] \leq e^{\varepsilon} \Pr[M(X') \in S] + \delta$$

[Dwork, McSherry, Nissim, Smith], 2006

(ε, δ)-Differential Privacy

- "Privacy loss random variable" is bounded by ε with probability $1 - \delta$ for all datasets X and X' which differ in a single entry
- Pure differential privacy: $\delta = 0$
- Approximate differential privacy: $\delta > 0$
- Qualitatively different notions
  - Much easier to design algorithms for approx. DP
- Today: Goal is pure DP algorithms

Non-Private Mean Estimation

Given d-dimensional $X_1, \ldots, X_n \sim p$, where p has covariance $\|\Sigma\|_2 \leq 1$, output $\hat{\mu}$ such that $\|\hat{\mu} - E[p]\|_2 \leq \alpha$ with prob. 99%

- Empirical mean: $\hat{\mu} = \frac{1}{n}\sum X_i$
  - $n = O(d/\alpha^2)$ samples suffice non-privately (rate: $\alpha \leq O(\sqrt{d/n})$
- Fancier techniques: … w.p. $\geq 1 - \beta$ using $n = O\left(\frac{d+\log(1/\beta)}{\alpha^2}\right)$ samples
  - Rate: $\alpha \leq O(\sqrt{(d + \log 1/\beta)/n})$

Non-Private Mean Estimation

Given d-dimensional $X_1, \ldots, X_n \sim p$, where p has covariance $\|\Sigma\|_2 \leq 1$, output $\hat{\mu}$ such that $\|\hat{\mu} - E[p]\|_2 \leq \alpha$ with prob. 99%

- Empirical mean: $\hat{\mu} = \frac{1}{n}\sum X_i$
  - $n = O(d/\alpha^2)$ samples suffice non-privately (rate: $\alpha \leq O(\sqrt{d/n})$)
- Fancier techniques: … w.p. $\geq 1 - \beta$ using $n = O\left(\frac{d+\log(1/\beta)}{\alpha^2}\right)$ samples
  - Rate: $\alpha \leq O(\sqrt{(d + \log 1/\beta)/n})$
  - "Sub-Gaussian rates" [Lugosi, Mendelson], 2019, [Hopkins], 2020, …
  - Not really our focus, but we will eventually get it for free
- Today: pretend we start with a "coarse estimate" of $E[p]$
  - i.e., by recentering, $\|E[p]\|_2 \leq O(\sqrt{d})$

Algorithms for Private Mean Estimation

	Sample Complexity	Privacy Notion	Running Time
Empirical Mean	$O(d)$	Not private	Polynomial

The plan from here…

1. Three Deficient Private Algorithms
2. Fixing the Exponential Mechanism
3. The Algorithm
4. The Bigger Picture

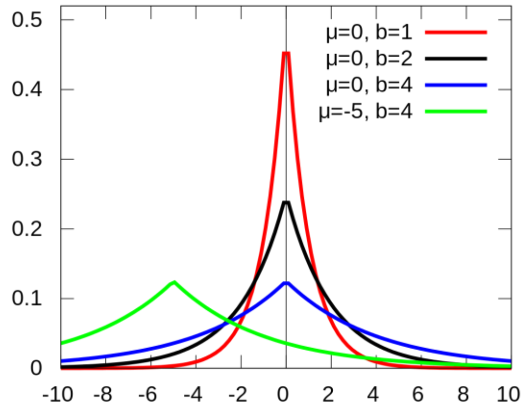
The plan from here…

1. Three Deficient Private Algorithms
2. Fixing the Exponential Mechanism
3. The Algorithm
4. The Bigger Picture

Three Deficient Algorithms

Every flawed algorithm is flawed in its own way…

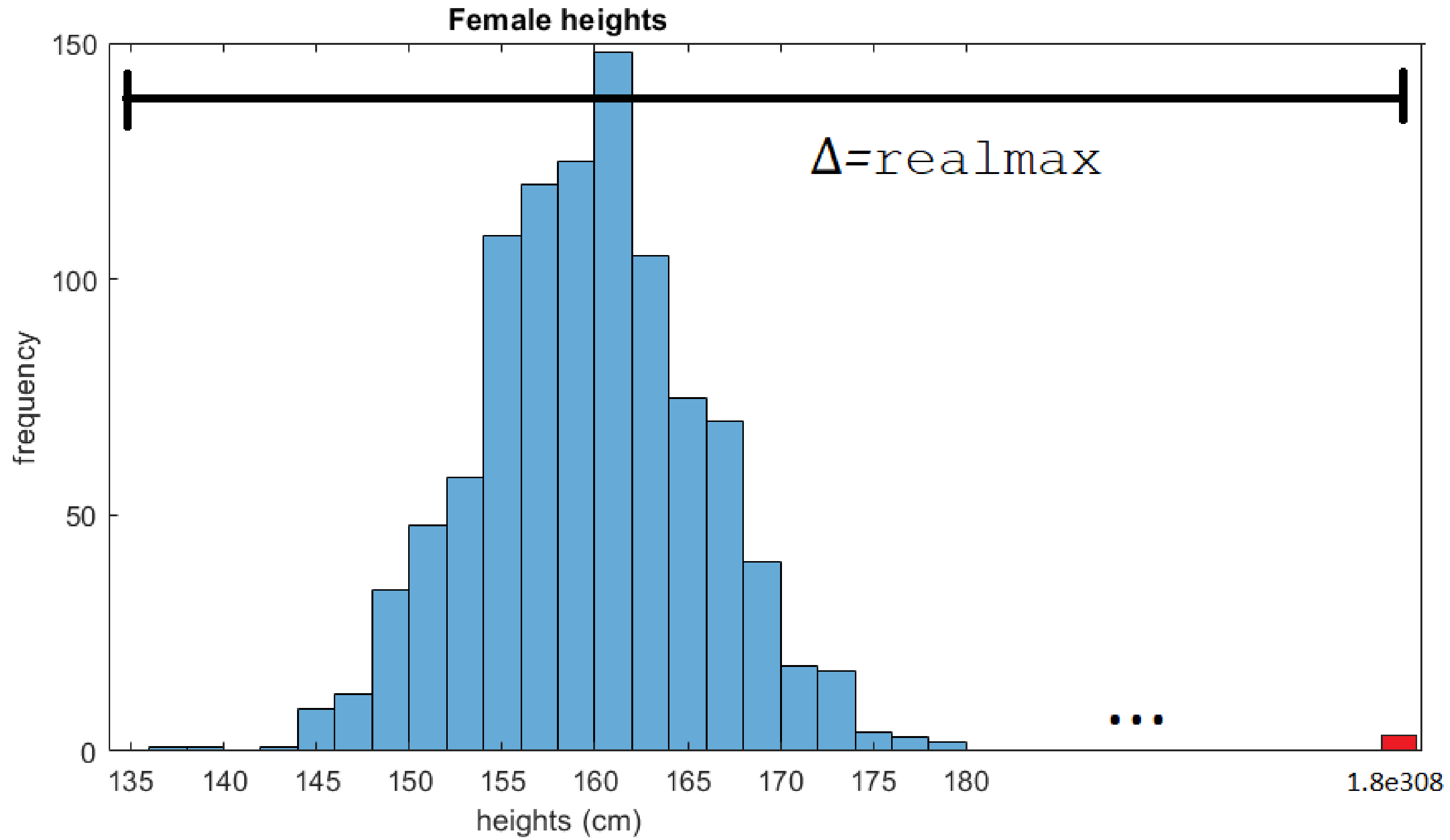
Differential Privacy 101



- Let $f : D^n \to \mathbf{R}^d$ be a vector-valued function of interest
    - e.g., some non-private mean estimation algorithm
- ℓ_1-sensitivity of f: $\Delta_1^f = \max_{X,X':d_H(X,X')=1} \|f(X) - f(X')\|_1$
    - "How much can the function change by modifying one datapoint?"
- The Laplace Mechanism: $f(X) + \mathrm{Lap}\left(\Delta_1^f/\varepsilon\right)^{\otimes d}$ is $(\varepsilon, 0)$-DP
    - "Add Laplace noise to each coordinate, proportional to the ℓ_1-sensitivity"

Laplace Mechanism

- Empirical mean: $f(X) = \frac{1}{n}\sum X_i$
- Sensitivity is infinite



Laplace Mechanism

- Clipped empirical mean: $f(X) = \frac{1}{n}\sum \text{clip}(X_i)$
  - Limit sensitivity by clipping to an ℓ_2-ball of radius $O\left(\sqrt{d}\right)$
  - Biases the statistic, but we can control the bias [K., Singhal, Ullman], 2020
- Resulting ℓ_1-sensitivity: $\Delta_1^f \leq d/n$
- Output $\hat{\mu} = \frac{1}{n}\sum \text{clip}(X_i) + \text{Lap}(d/\varepsilon n)^{\otimes d}$
- ℓ_2 error due to noise $\approx d^{1.5}/\varepsilon n$
- Resulting sample complexity: $O(d^{1.5}/\varepsilon)$

Algorithms for Private Mean Estimation

	Sample Complexity	Privacy Notion	Running Time
Empirical Mean	$O(d)$	Not private	Polynomial
Laplace Mechanism	$O(d^{1.5})$	Pure DP	Polynomial

Gaussian Mechanism (diffs)

- Add Gaussian noise instead of Laplace
- Scaled to ℓ_2-sensitivity instead of ℓ_1-sensitivity
- Resulting sample complexity: $O(d/\varepsilon)$
- Only gives approximate DP $(\delta > 0)$ instead of pure DP $(\delta = 0)$

Algorithms for Private Mean Estimation

	Sample Complexity	Privacy Notion	Running Time
Empirical Mean	$O(d)$	Not private	Polynomial
Laplace Mechanism	$O(d^{1.5})$	Pure DP	Polynomial
Gaussian Mechanism	$O(d)$	Approximate DP	Polynomial

Differential Privacy 102

- Privately select an object from a set based on a “score”
- Given: Sensitive dataset $X = X_1, \ldots, X_n$

  Set of objects Q

  Score function $f: D^n \times Q \to \mathbf{R}$
- Output: $q \in Q$ which (approximately) maximizes $f(X, q)$
- Exponential mechanism: Sample q with probability $\propto \exp(\varepsilon \cdot f(X, q))$
- $(\varepsilon, 0)$-differentially private

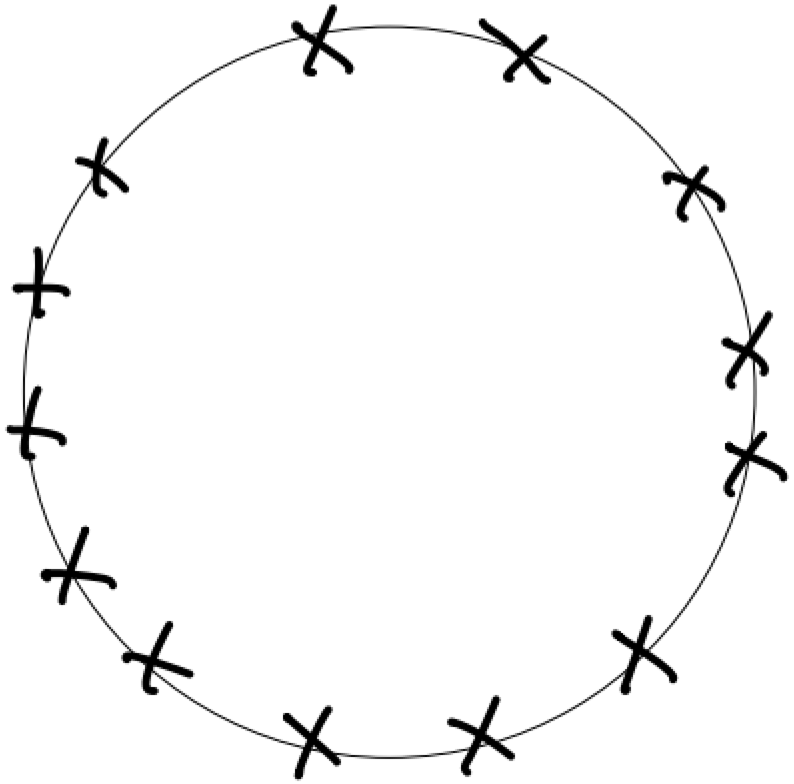
Exponential Mechanism Example

- Running an election
  - Set of objects: election candidates
  - Sensitive dataset: votes
  - Score function: number of votes for each candidate
- Non-privately: pick the highest score
- Privately: sample winner $\propto \exp(\varepsilon \cdot \text{Score})$
- Assign scores, use to noisily pick winner

	15 votes	19 votes	20 votes
$\varepsilon = 0.1$	24% chance	36% chance	40% chance

Exponential Mechanism

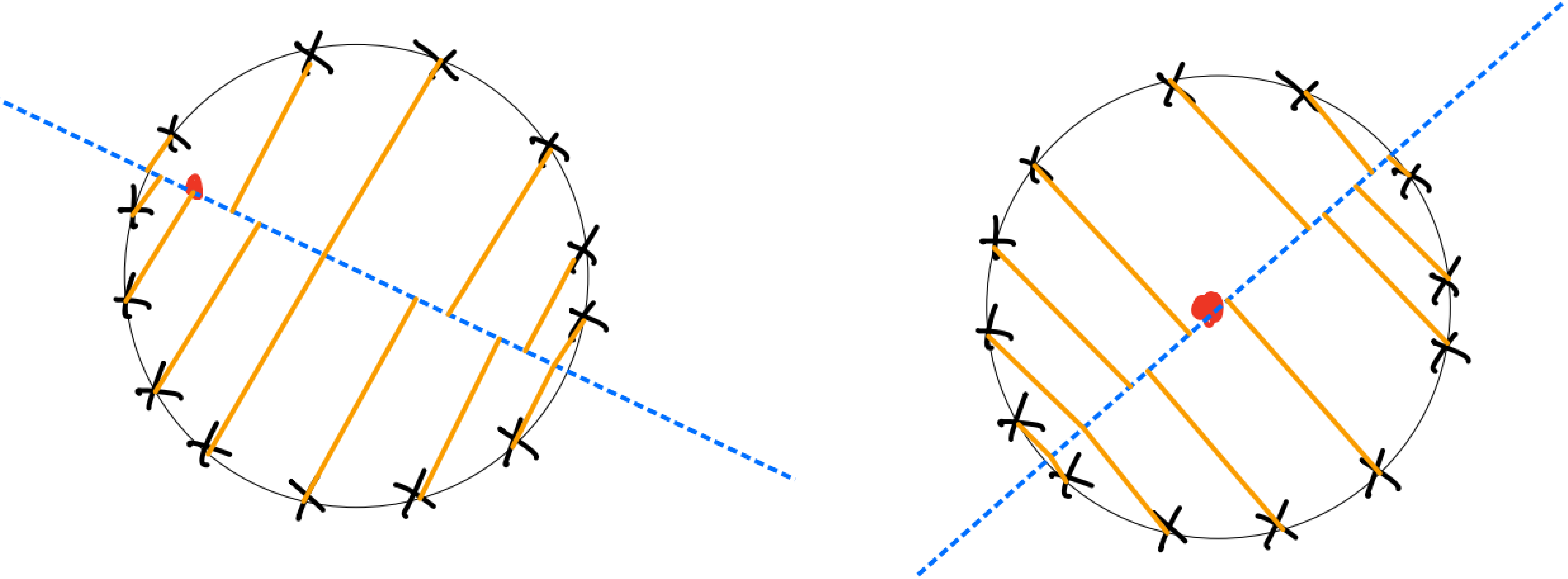
- Intuition: Empirical mean should be close to true mean in every 1D projection



[Bun, K., Steinke, Wu], NeurIPS ’19, [K., Singhal, Ullman], COLT ‘20

Exponential Mechanism

- Intuition: Empirical mean should be close to true mean in every 1D projection

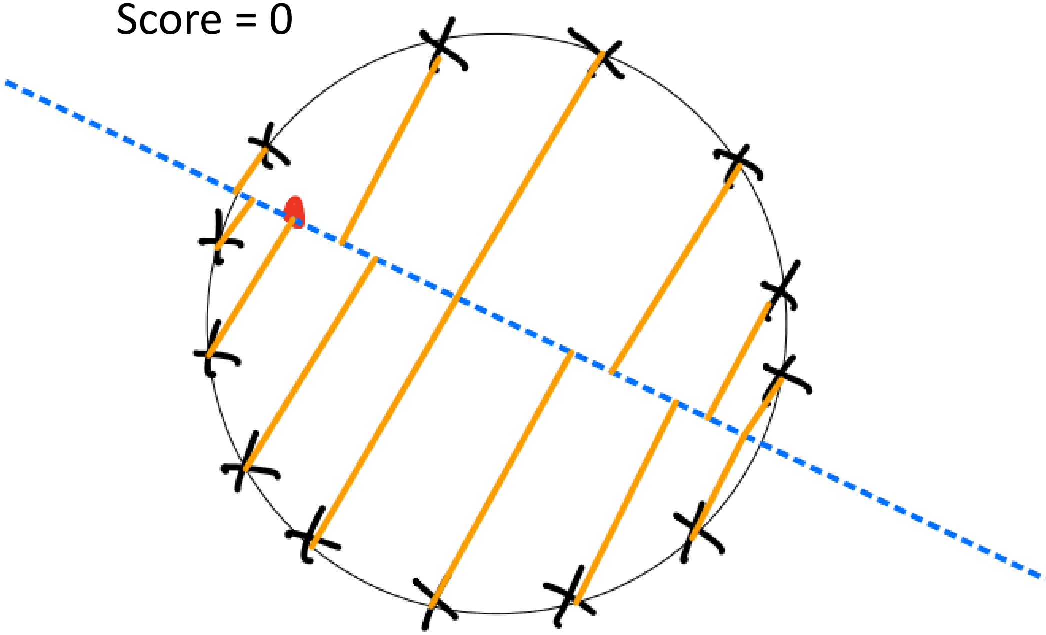


[Bun, K., Steinke, Wu], NeurIPS ’19, [K., Singhal, Ullman], COLT ‘20

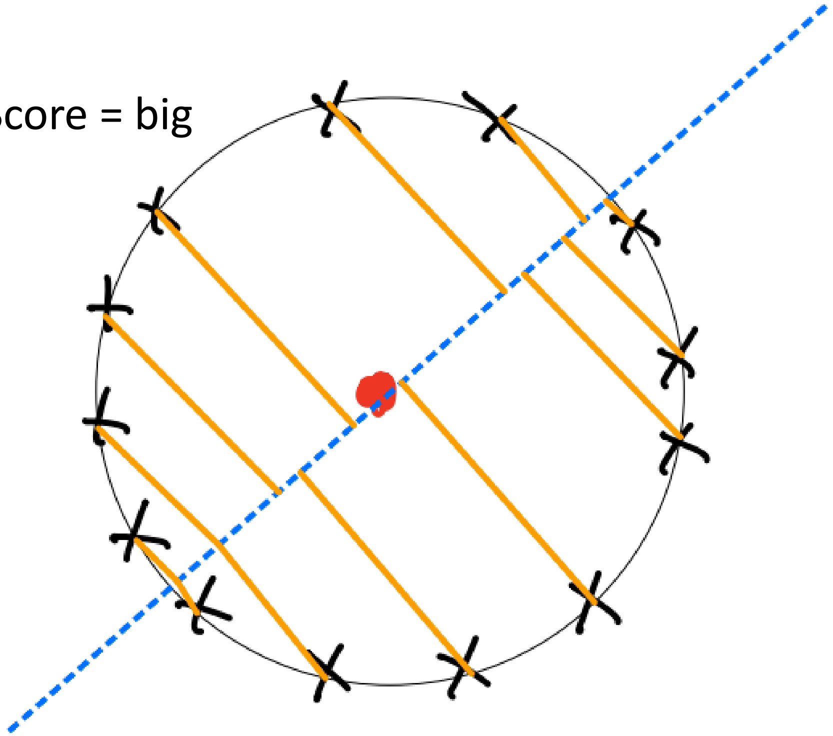
Exponential Mechanism

- Intuition: Empirical mean should be close to true mean in every 1D projection
- Score function at q: "How many points must be changed to have q be far from empirical (clipped) mean in some projection?"

Score = 0



Score = big



[Bun, K., Steinke, Wu], NeurIPS '19, [K., Singhal, Ullman], COLT '20

Exponential Mechanism

- Intuition: Empirical mean should be close to true mean in every 1D projection
- Score function at q: "How many points must be changed to have q be far from empirical (clipped) mean in some projection?"
  - Have to look at all projections to compute…
- Set of objects: Cover of set of possible means $\left(2^{\tilde{O}(d)}\right)$
  - Exponentially large…
- Standard analysis gives that $n = \tilde{O}(d)$ samples suffice
- Related: private hypothesis selection via Scheffé's method

[Bun, K., Steinke, Wu], NeurIPS '19, [K., Singhal, Ullman], COLT '20

Algorithms for Private Mean Estimation

	Sample Complexity	Privacy Notion	Running Time
Empirical Mean	$O(d)$	Not private	Polynomial
Laplace Mechanism	$O(d^{1.5})$	Pure DP	Polynomial
Gaussian Mechanism	$O(d)$	Approximate DP	Polynomial
Exponential Mechanism	$\tilde{O}(d)$	Pure DP	Exponential

Algorithms for Private Mean Estimation

	Sample Complexity	Privacy Notion	Running Time
Empirical Mean	$O(d)$	Not private	Polynomial
Laplace Mechanism	$O(d^{1.5})$	Pure DP	Polynomial
Gaussian Mechanism	$O(d)$	Approximate DP	Polynomial
Exponential Mechanism	$\tilde{O}(d)$	Pure DP	Exponential
Our Algorithm [HKM22]	$\tilde{O}(d)$	Pure DP	Polynomial

Theorem: Given $X_1, \dots, X_n \sim p$ where p has covariance $\|\Sigma\|_2 \leq 1$ and $\|E[p]\|_2 \leq O(\sqrt{d})$, there exists a computationally efficient $(\varepsilon, 0)$-DP algorithm which outputs $\hat{\mu}$ such that $\|\hat{\mu} - E[p]\|_2 \leq \alpha$ with probability $1 - \beta$. It requires $n = \tilde{O}\left(\frac{d+\log(1/\beta)}{\alpha^2 \varepsilon}\right)$ samples.

[Hopkins, K., Majid], STOC 2022

The plan from here…

1. Three Deficient Private Algorithms
2. Fixing the Exponential Mechanism
3. The Algorithm
4. The Bigger Picture

Why was the Exponential Mechanism slow?

Two problems to address:

1. Computing the score function for a single candidate is slow
   - Solution: Efficient robust + high-dimensional statistics
2. Have to compute the score function for exponentially many candidates
   - Solution: Efficient log-concave sampling from $\propto \exp\left(\varepsilon \cdot f(X, q)\right)$

1. Efficiently computing the score function

Recent line of work on robust multivariate statistics excels at *efficiently* finding “interesting directions”

- Robust estimation (η-fraction of data is corrupted)
  - [Diakonikolas, K., Kane, Li, Moitra, Stewart], 2016, [Lai, Rao, Vempala], 2016, …
- Sub-Gaussian rates for heavy-tailed estimation (…$+\log(1/\beta)$)
  - [Lugosi, Mendelson], 2019 made efficient by [Hopkins], 2020, …
- Privacy
  - [Hopkins, K., Majid], 2022, [Kothari, Manurangsi, Velingker], 2022, …?

Where do we go now? Finding a direction

- Score function over **directions** (not candidate means)
  - Ideas from [Hopkins], 2020, [Cherapanamjeri, Flammarion, Bartlett], 2019

Definition 6.3 (quadratic optimization problem [Hop20, CFB19]). *Let $Z_1, \ldots, Z_k, \tilde{\mu} \in \mathbb{R}^d$, $r > 0$. Let $\mathrm{QUAD}(\tilde{\mu}, r, Z)$ be the following quadratic program.*

$$\mathrm{QUAD}(\tilde{\mu}, r, Z) :=$$

Z_i: datapoints
$\tilde{\mu}$: current estimate
r: "radius of interest"
v: direction
b_i: 0/1 indicators on points

"Direction v's score: How many points are 'far' in direction v?"

b_i: either 0 or 1

v: unit vector

Either $b_i = 0$
OR
Z_i is far from current estimate $\tilde{\mu}$ in direction v

Where do we go now? Finding a direction

- Score function over **directions** (not candidate means)
  - Ideas from [Hopkins], 2020, [Cherapanamjeri, Flammarion, Bartlett], 2019
- Use SDP relaxation of the quadratic program

Definition 6.4 (SDP Relaxation [Hop20, CFB19]). *Let* $Z_1, \ldots, Z_k, \widetilde{\mu} \in \mathbb{R}^d$, $r > 0$. *Let* $\mathrm{SDP}(\widetilde{\mu}, r, Z)$ *be the following semi-definite program.*

$$\mathrm{SDP}(\widetilde{\mu}, r, Z) := \max_{v,b,B,U,W} \quad \mathrm{Tr}\,(B)$$

$$s.t. \quad \begin{bmatrix} 1 & b^\top & v^\top \\ b & B & W \\ v & W^\top & V \end{bmatrix} \succcurlyeq 0$$

$$B_{ii} = b_i \quad \forall i$$

$$\mathrm{Tr}(V) = 1 \quad \forall i$$

$$B_{ii} \cdot r \leq \langle Z_i - \widetilde{\mu}, W_i \rangle \quad \forall i$$

"Rounding scheme" = just use v

2. Running the Exponential Mechanism efficiently

- Sampling q with probability $\propto \exp\big(\varepsilon \cdot f(X, q)\big)$
  - Naively requires computing $f(X, q)$ for every q in an exponentially-sized cover
- Idea: if $f(X, q)$ is *concave*, then resulting distribution is *log-concave*
- Efficient private samplers for log-concave distributions exist
  - Need multiplicative approximation versus usual total variation guarantee
  - [Bassily, Smith, Thakurta], FOCS 2014, [Mangoubi, Vishnoi], NeurIPS 2022
- Must use continuous version of exponential mechanism
- Additionally need a Lipschitz property

Efficient Log-Concave Sampling

Consider
$$
\begin{aligned}
\text{SDP-VAL}(y;\widetilde{\mu},r,Z) := \max_{v,b,B,W,V} \quad & \operatorname{Tr}(B) \\
s.t. \quad & \begin{bmatrix} 1 & b^\top & v^\top \\ b & B & W \\ v & W^\top & V \end{bmatrix} \succcurlyeq 0 \\
& v_i = y_i \quad \forall i \\
& B_{ii} = b_i \quad \forall i \\
& \operatorname{Tr}(V) = 1 \\
& B_{ii} \cdot r \le \langle Z_i - \widetilde{\mu}, W_i \rangle \quad \forall i
\end{aligned}
$$

Claim: SDP-VAL is both Lipschitz and concave.

Therefore we can sample $\propto \exp(\varepsilon \cdot \mathrm{SDPVAL}(q))$ efficiently.

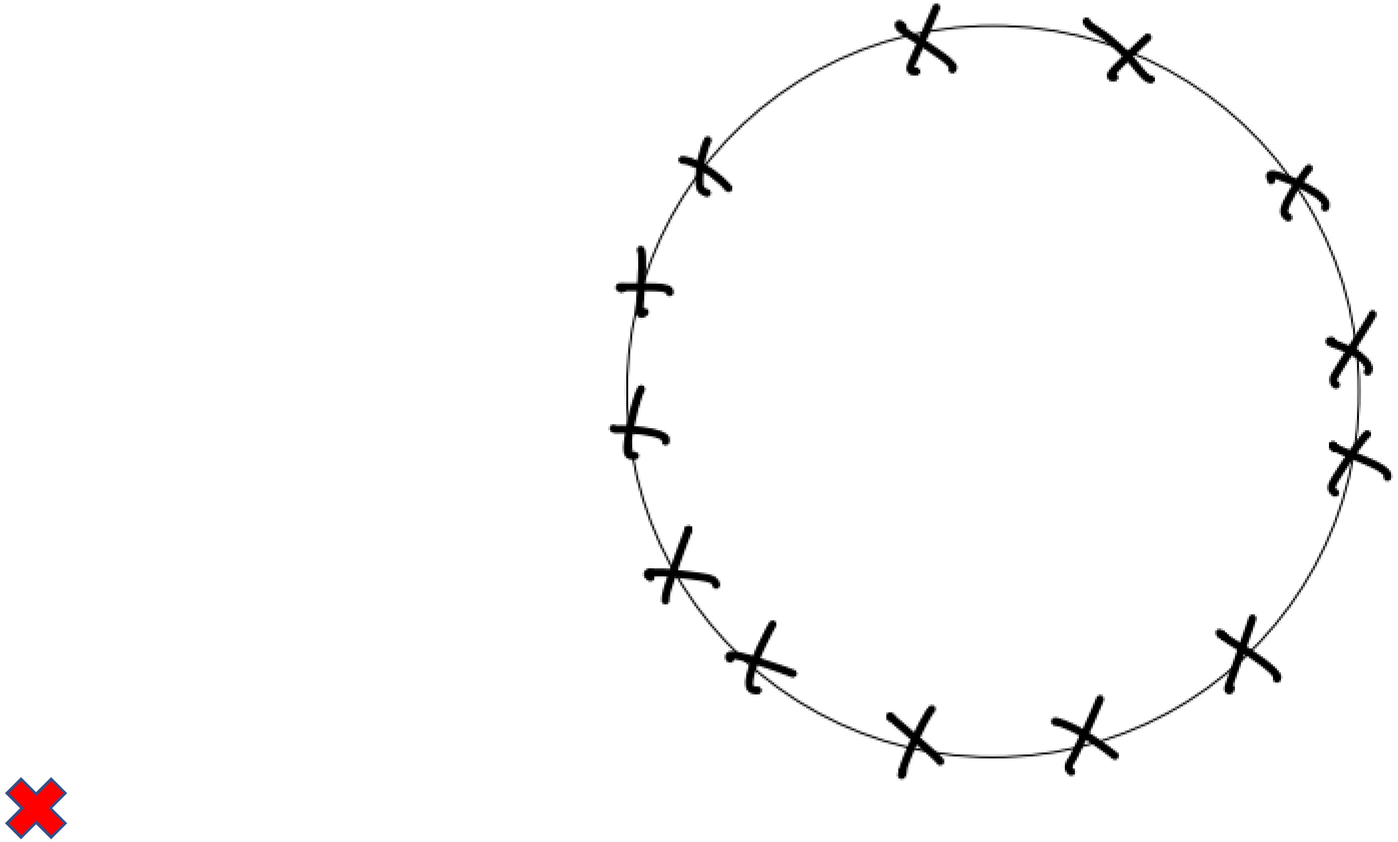
The plan from here…

1. Three Deficient Private Algorithms
2. Fixing the Exponential Mechanism
3. The Algorithm
4. The Bigger Picture

Overall algorithm

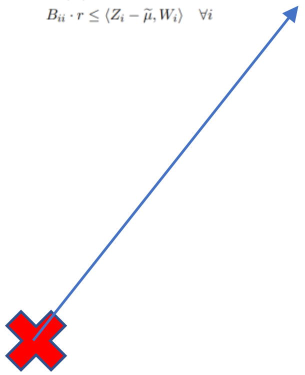
1. Find a direction v
   - Use the exponential mechanism with efficient sampling to pick v
     - Via random walk over directions that computes SDP score function at each step
2. Step in that direction
   - Use exponential mechanism to privately estimate the step size
3. Repeat

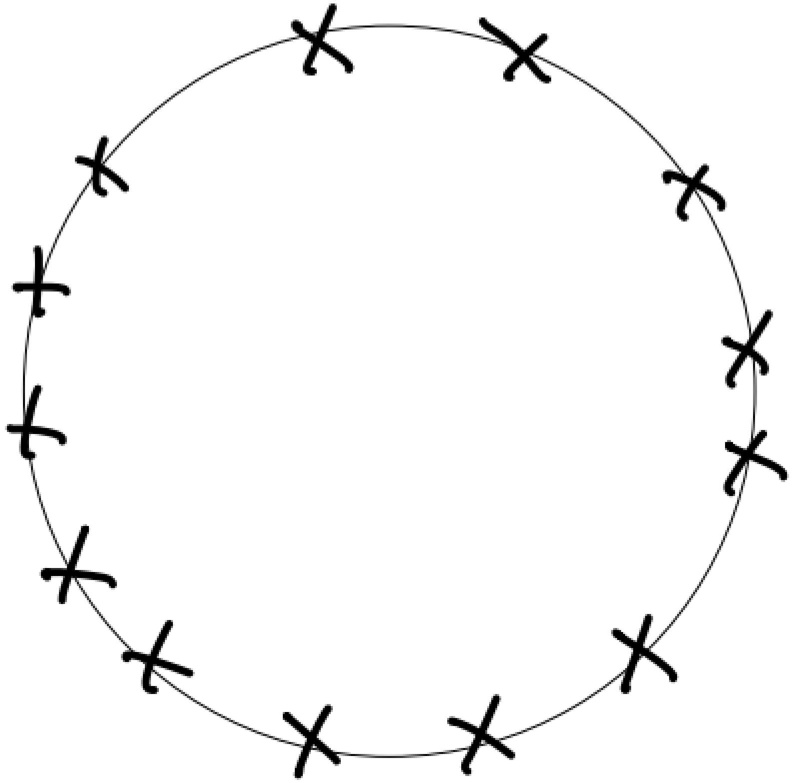
Run exponential mechanism!



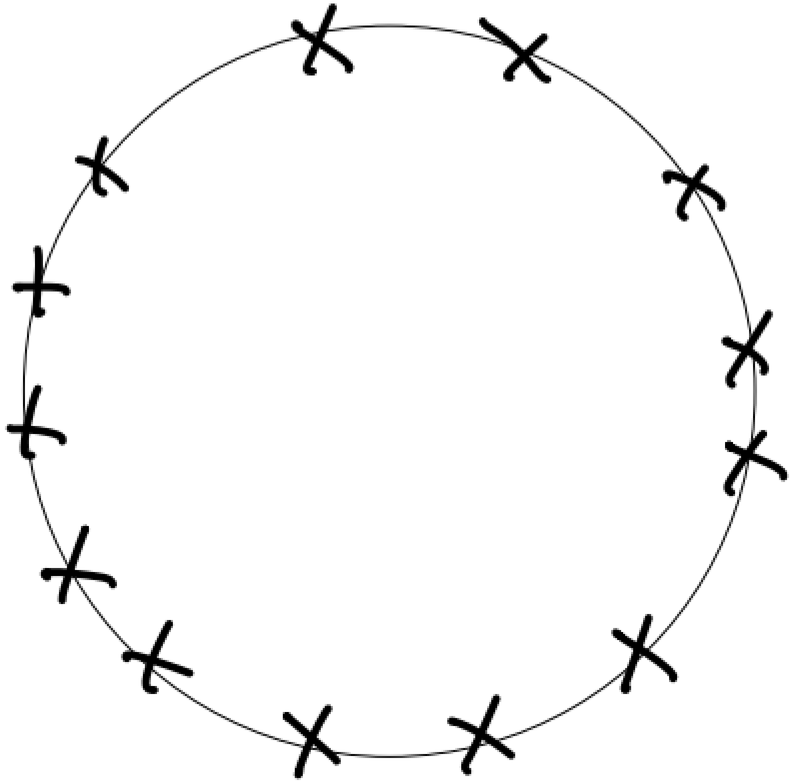
Run exponential mechanism!
Running efficient sampler…

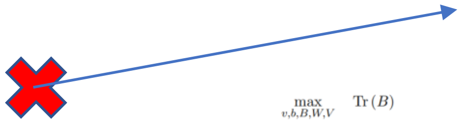
$$
\begin{aligned}
\max_{v,b,B,W,V} \quad & \mathrm{Tr}(B) \\
s.t. \quad & \begin{bmatrix} 1 & b^\intercal & v^\intercal \\ b & B & W \\ v & W^\intercal & V \end{bmatrix} \succcurlyeq 0 \\
& v_i = y_i \quad \forall i \\
& B_{ii} = b_i \quad \forall i \\
& \mathrm{Tr}(V) = 1 \\
& B_{ii} \cdot r \leq \langle Z_i - \widetilde{\mu}, W_i \rangle \quad \forall i
\end{aligned}
$$





Run exponential mechanism!
Running efficient sampler…

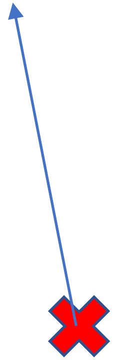


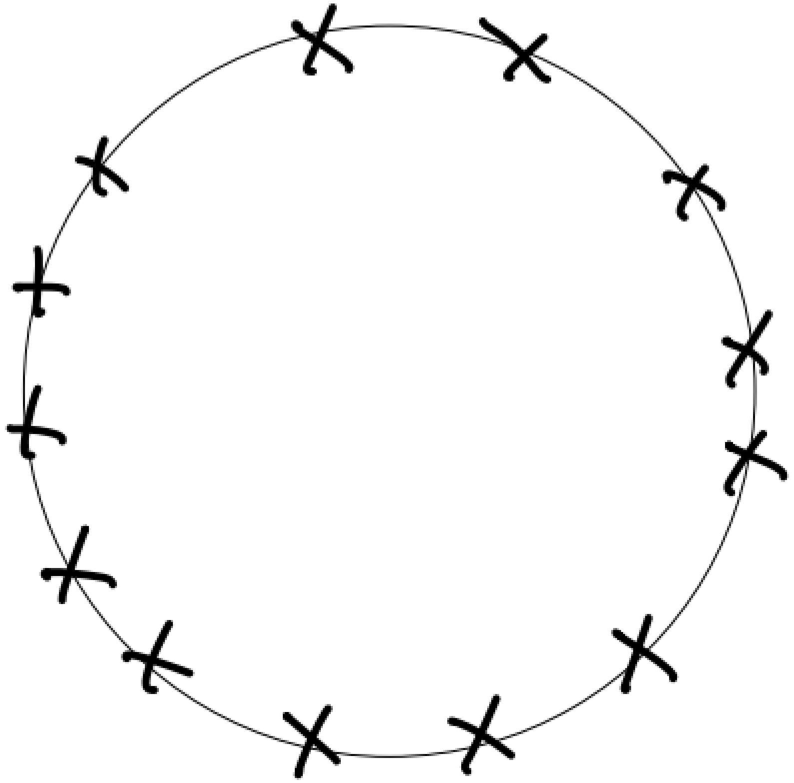


$$
\begin{aligned}
\max_{v,b,B,W,V} \quad & \mathrm{Tr}\,(B) \\
s.t. \quad & \begin{bmatrix} 1 & b^{\mathsf{T}} & v^{\mathsf{T}} \\ b & B & W \\ v & W^{\mathsf{T}} & V \end{bmatrix} \succcurlyeq 0 \\
& v_i = y_i \quad \forall i \\
& B_{ii} = b_i \quad \forall i \\
& \mathrm{Tr}(V) = 1 \\
& B_{ii} \cdot r \leq \langle Z_i - \widetilde{\mu}, W_i \rangle \quad \forall i
\end{aligned}
$$

Run exponential mechanism!
Running efficient sampler…

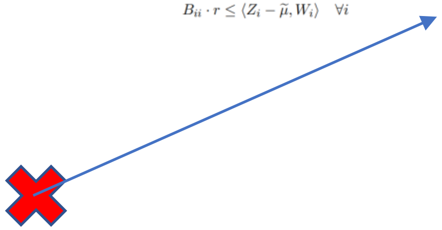
$$
\begin{aligned}
\max_{v,b,B,W,V} \quad & \mathrm{Tr}\,(B) \\
s.t. \quad & \begin{bmatrix} 1 & b^\intercal & v^\intercal \\ b & B & W \\ v & W^\intercal & V \end{bmatrix} \succcurlyeq 0 \\
& v_i = y_i \quad \forall i \\
& B_{ii} = b_i \quad \forall i \\
& \mathrm{Tr}(V) = 1 \\
& B_{ii} \cdot r \leq \langle Z_i - \widetilde{\mu}, W_i \rangle \quad \forall i
\end{aligned}
$$

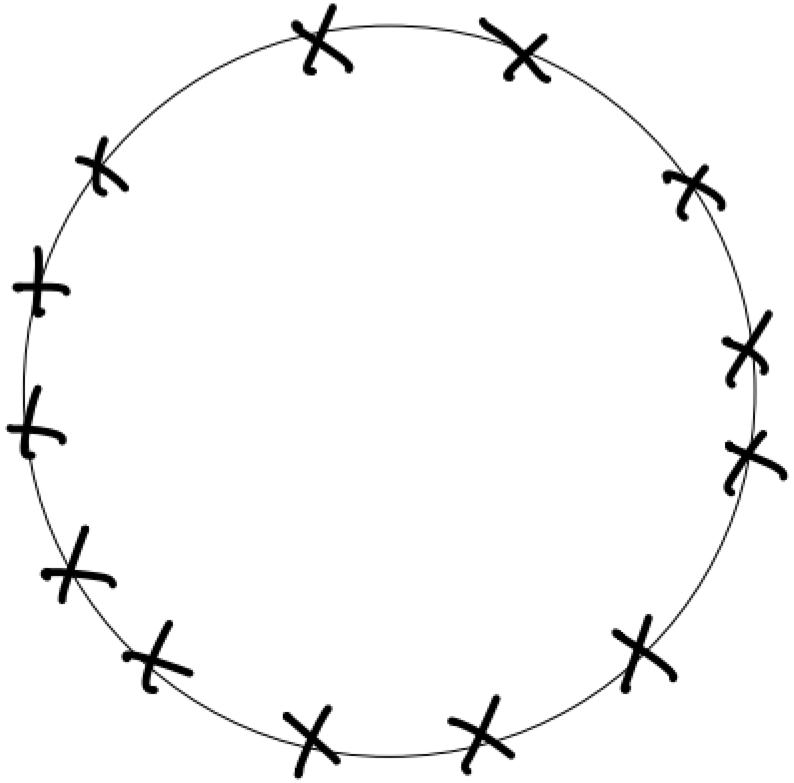




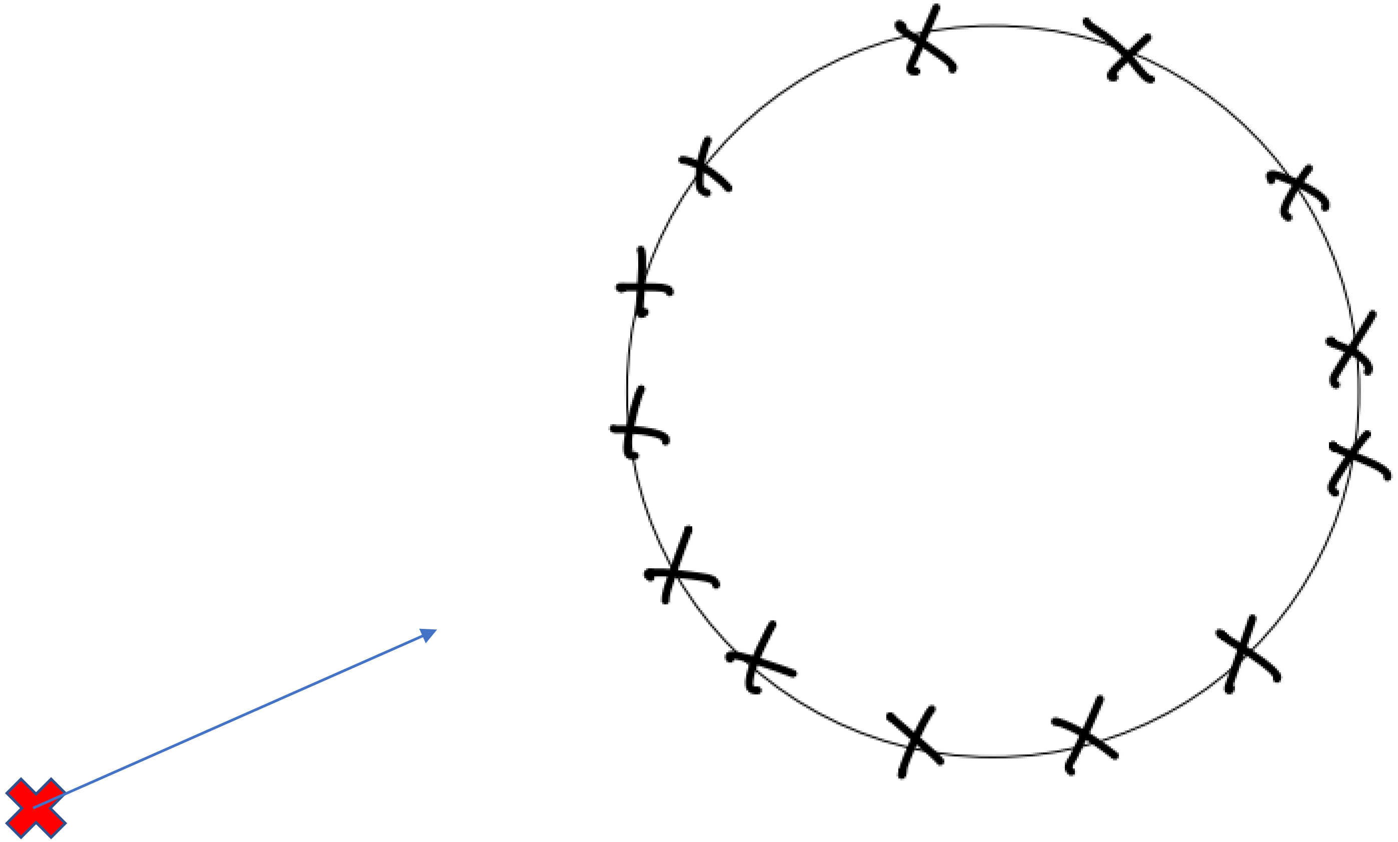
Run exponential mechanism!
Running efficient sampler…

$$
\begin{aligned}
\max_{v,b,B,W,V} \quad & \mathrm{Tr}\,(B) \\
s.t. \quad & \begin{bmatrix} 1 & b^\intercal & v^\intercal \\ b & B & W \\ v & W^\intercal & V \end{bmatrix} \succcurlyeq 0 \\
& v_i = y_i \quad \forall i \\
& B_{ii} = b_i \quad \forall i \\
& \mathrm{Tr}(V) = 1 \\
& B_{ii} \cdot r \leq \langle Z_i - \widetilde{\mu}, W_i \rangle \quad \forall i
\end{aligned}
$$

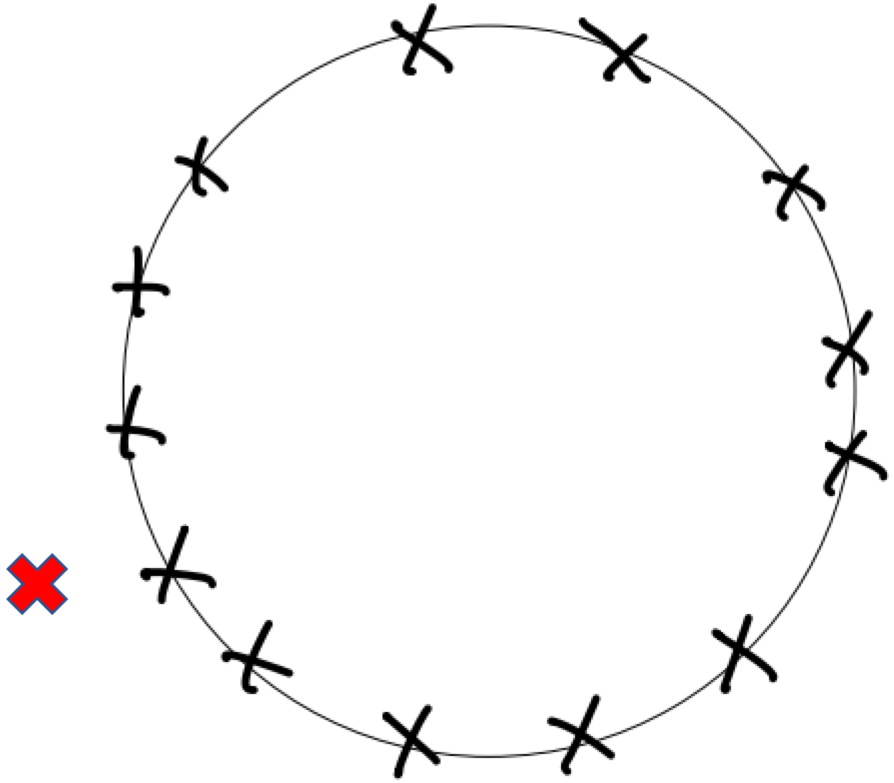




Take a step

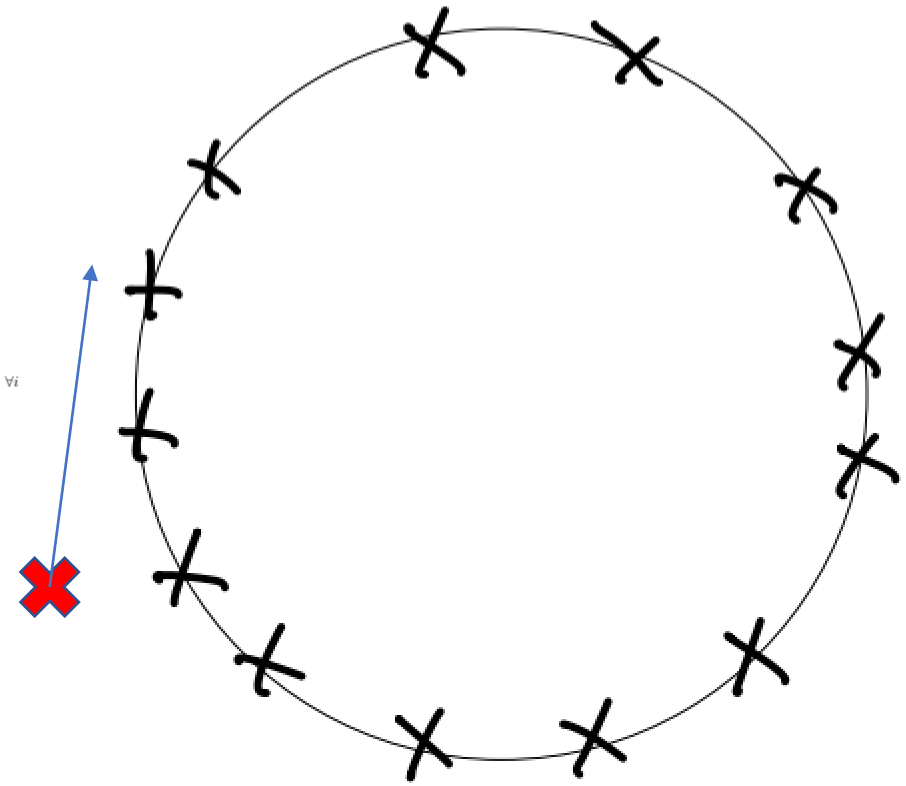


Run exponential mechanism!



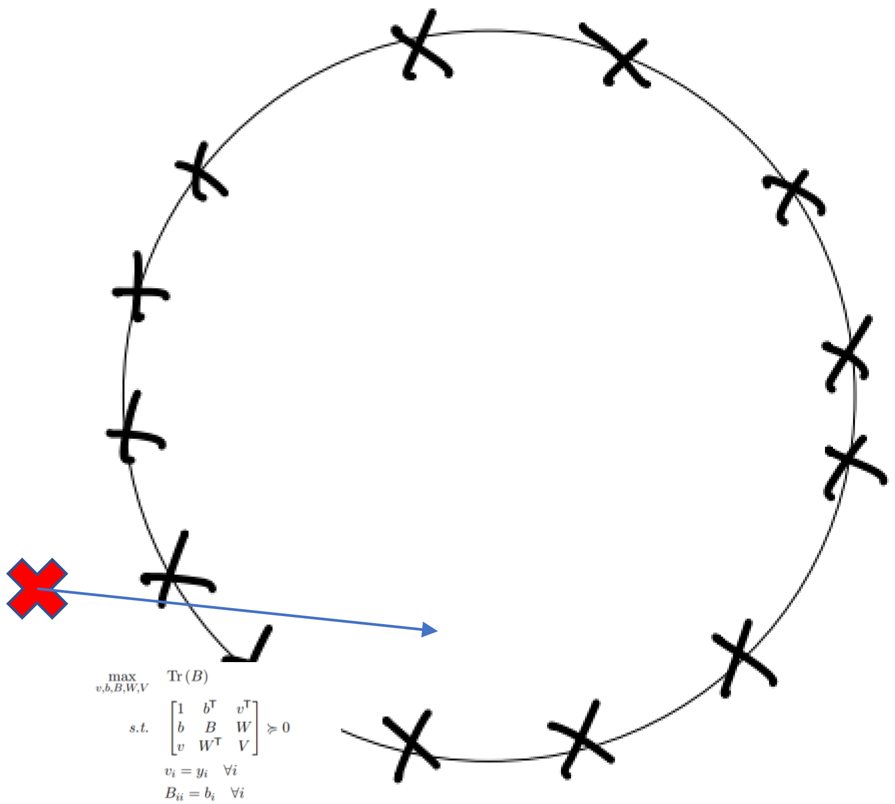
Run exponential mechanism!
Running efficient sampler…

$$
\begin{aligned}
\max_{v,b,B,W,V} \quad & \mathrm{Tr}\,(B) \\
s.t. \quad & \begin{bmatrix} 1 & b^{\mathsf{T}} & v^{\mathsf{T}} \\ b & B & W \\ v & W^{\mathsf{T}} & V \end{bmatrix} \succcurlyeq 0 \\
& v_i = y_i \quad \forall i \\
& B_{ii} = b_i \quad \forall i \\
& \mathrm{Tr}(V) = 1 \\
& B_{ii} \cdot r \leq \langle Z_i - \widetilde{\mu}, W_i \rangle \quad \forall i
\end{aligned}
$$



Run exponential mechanism!

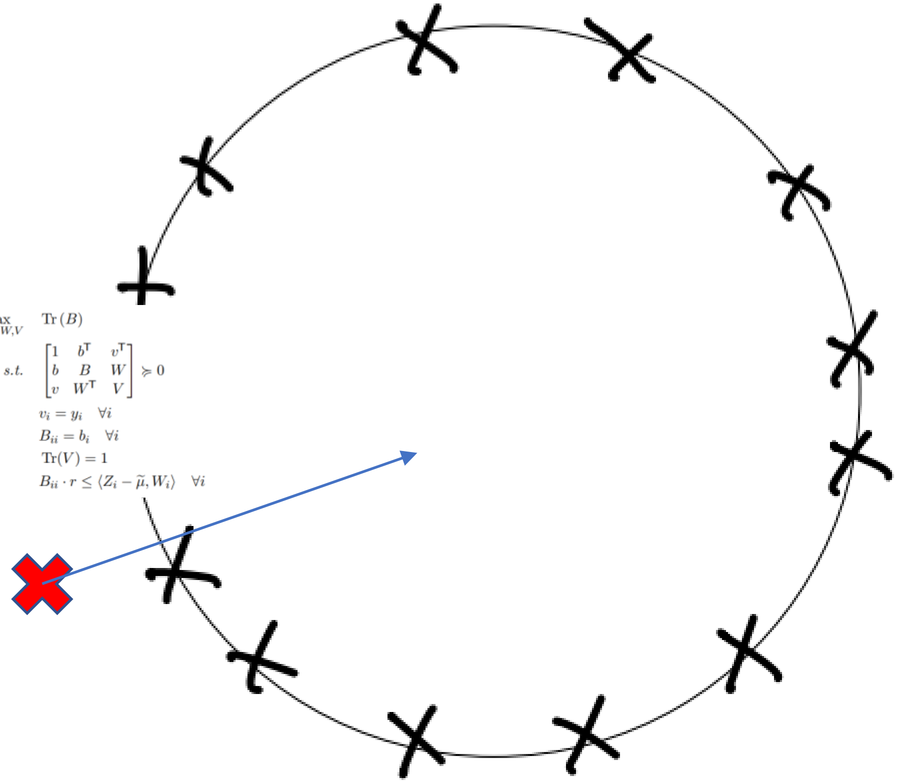
Running efficient sampler…



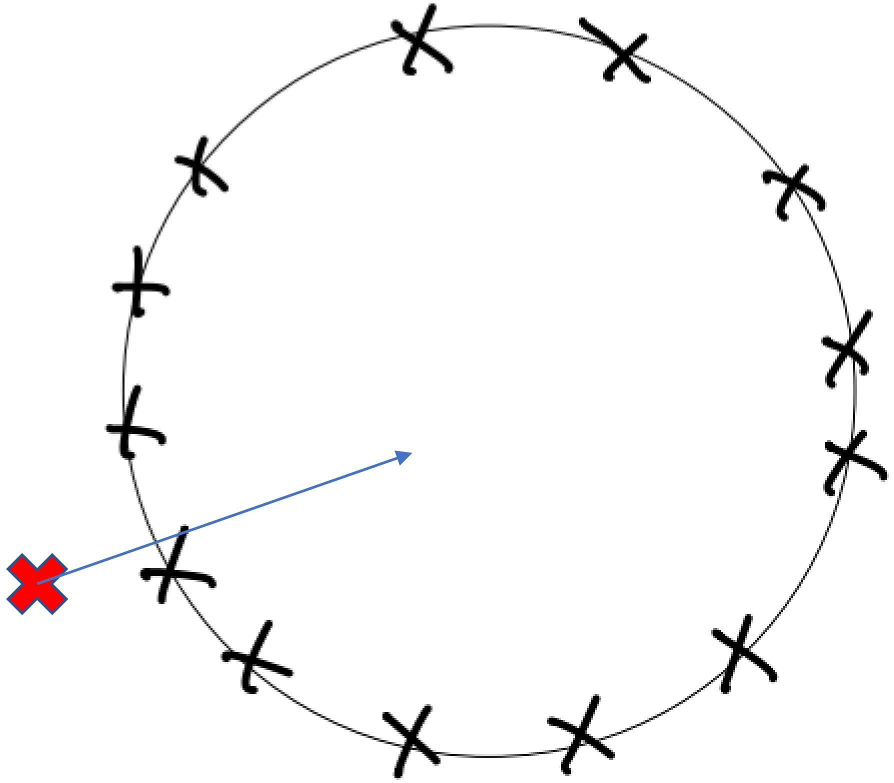
$$
\begin{aligned}
\max_{v,b,B,W,V} \quad & \mathrm{Tr}\,(B) \\
s.t. \quad & \begin{bmatrix} 1 & b^{\mathsf{T}} & v^{\mathsf{T}} \\ b & B & W \\ v & W^{\mathsf{T}} & V \end{bmatrix} \succcurlyeq 0 \\
& v_i = y_i \quad \forall i \\
& B_{ii} = b_i \quad \forall i \\
& \mathrm{Tr}(V) = 1 \\
& B_{ii} \cdot r \leq \langle Z_i - \widetilde{\mu}, W_i \rangle \quad \forall i
\end{aligned}
$$

Run exponential mechanism!
Running efficient sampler…

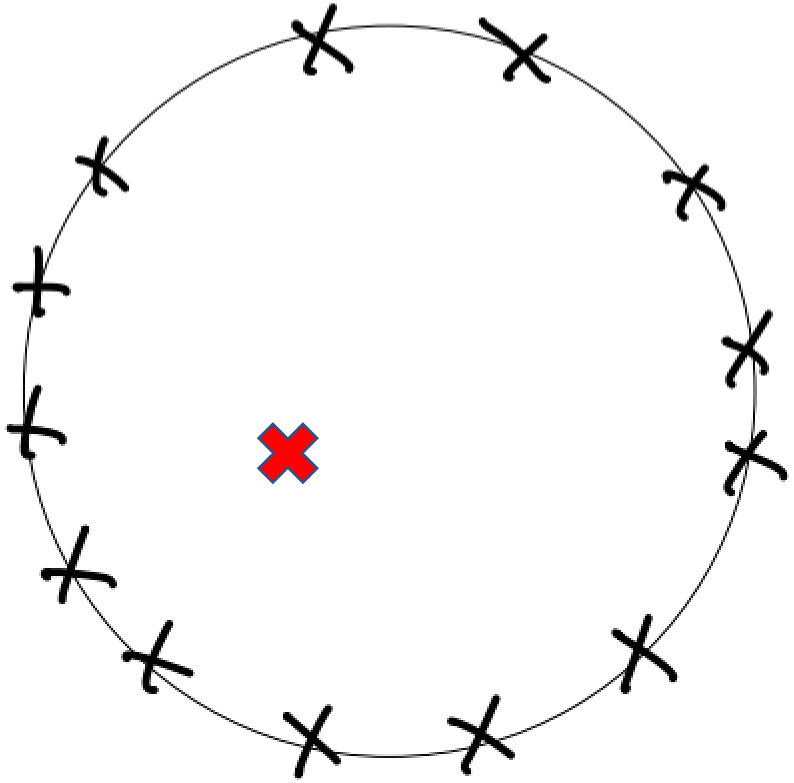
$$
\begin{aligned}
\max_{v,b,B,W,V} \quad & \mathrm{Tr}\,(B) \\
s.t. \quad & \begin{bmatrix} 1 & b^\intercal & v^\intercal \\ b & B & W \\ v & W^\intercal & V \end{bmatrix} \succcurlyeq 0 \\
& v_i = y_i \quad \forall i \\
& B_{ii} = b_i \quad \forall i \\
& \mathrm{Tr}(V) = 1 \\
& B_{ii} \cdot r \leq \langle Z_i - \tilde{\mu}, W_i \rangle \quad \forall i
\end{aligned}
$$



Take a step



Etc.



A Meta-Theorem

Suppose we have a s … llowing properties:

1. High score funct
2. Bounded sensit
3. Lipschitz wrt do
4. Concave wrt do
5. Large volume of … main of candidates

Then exp. mech. wil … y point efficiently.

We give a specific m … score functions

Framework also works for "coarse estimation" (not mentioned today)

Theorem 4.5 (Meta-Theorem on SoS Exponential Mechanism). *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a compact, convex set and $\mathcal{X}$ a universe of possible datasets, equipped with a "neighbors" relation. Suppose that for every dataset X there exists an Archimedean and η-robustly satisfiable system of polynomial inequalities $\mathcal{P}^X(x,y) = \{p_1^X(x,y) \geq 0, \ldots, p_N^X(x,y) \geq 0\}$ and a polynomial $p^X(x,y)$, all of degree at most D, in indeterminates $x_1, \ldots, x_n, y_1, \ldots, y_N$ such that for every neighboring dataset X' there is a linear function $y'(y)$ such that bounded sensitivity has an SoS proof:*

$$\forall j,\ \mathcal{P}^X(x,y) \vdash_{\deg(p_j^{X'})} p_j^{X'}(x, y'(y)) \text{ and } \mathcal{P}^X(x,y) \vdash_D p^X(x,y) - p^{X'}(x,y') \leq 1\,.$$

Suppose also that for every X, there are SoS proofs $\mathcal{P}^X(x,y) \vdash_D p(x,y) \leq 1/\eta$ and $\mathcal{P}^X(x,y) \vdash_D -p(x,y) \leq 1/\eta$. Furthermore, suppose that the polynomials $\mathcal{P}^X$ and p^X, and the polynomials used in the above SoS proofs, all have coefficients expressible in at most B bits.

Then for every $\varepsilon > 0$ and $D \in \mathbb{N}$ there exists an ε-differentially private algorithm which takes as input the polynomials $p^X, p_1^X, \ldots, p_m^X$ and $B, \eta > 0$, with the following guarantees:

Utility: *For every X, if there is an SoS proof of utility for X which is degree-$\deg(p)$ with respect to p, i.e.,*

$$\mathcal{P}^X(x,y) \cup \{p(x,y) \geq 0\} \vdash_{D, \deg_p = \deg(p)} \|x - x^*(X)\|^2 \leq \alpha^2$$

for some vector $x^*(X) \in \mathbb{R}^n$ and $\alpha > 0$, where the coefficients of all polynomials involved in the proof are expressible with B bits, and if

$$\frac{\mathrm{vol}(\mathcal{C})}{\mathrm{vol}(\{x \in \mathcal{C} : \exists y \text{ s.t. } \mathcal{P}^X(x,y) \text{ and } p^X(x,y) \geq t\})} \leq r\,,$$

then the algorithm outputs x such that $\|x - x^*(X)\| \leq \alpha + 2^{-B}$ with probability at least $1 - r\exp(-\Omega(\varepsilon t))$.

Running time: *The algorithm runs in time*

$$\mathrm{poly}\left(n^D, N^D, m^D, \frac{1}{\varepsilon}, \frac{1}{\eta}, \mathrm{diam}(\mathcal{C}), B\right),$$

making at most this many calls to membership and projection oracles for $\mathcal{C}$.

Overall Theorem

Given an η-corrupted set of samples $X_1, \ldots, X_n \sim p$ where p has covariance $\|\Sigma\|_2 \leq 1$ and $\|E[p]\|_2 \leq R$, there exists a computationally efficient $(\varepsilon, 0)$-DP algorithm which outputs $\hat{\mu}$ such that $\|\hat{\mu} - E[p]\|_2 \leq \alpha + O\left(\sqrt{\eta}\right)$ with probability $1 - \beta$. It requires

$$n = \tilde{O}\left(\frac{d+\log(1/\beta)}{\alpha^2 \varepsilon} + \frac{d \log R + \min(d, \log R) \log(1/\beta)}{\varepsilon}\right) \text{ samples.}$$

No algorithm can succeed with fewer than

$$n = \Omega\left(\frac{d+\log(1/\beta)}{\alpha^2 \varepsilon} + \frac{d \log R + \log(1/\beta)}{\varepsilon}\right) \text{ samples.}$$

[Hopkins, K., Majid], STOC 2022

Overall Theorem

Given an η-corrupted set of samples $X_1, \ldots, X_n \sim p$ where p has covariance $\|\Sigma\|_2 \leq 1$ and $\|E[p]\|_2 \leq R$, there exists a computationally efficient $(\varepsilon, 0)$-DP algorithm which outputs $\hat{\mu}$ such that $\|\hat{\mu} - E[p]\|_2 \leq \alpha + O(\sqrt{\eta})$ with probability $1 - \beta$. It requires

$$n = \tilde{O}\left(\frac{d+\log(1/\beta)}{\alpha^2\varepsilon} + \frac{d\log R + \min(d, \log R)\log(1/\beta)}{\varepsilon}\right) \text{ samples.}$$

No algorithm can succeed with fewer than

$$n = \Omega\left(\frac{d+\log(1/\beta)}{\alpha^2\varepsilon} + \frac{d\log R + \log(1/\beta)}{\varepsilon}\right) \text{ samples.}$$

[Hopkins, K., Majid], STOC 2022

Overall Theorem

Given an η-corrupted set of samples $X_1, \ldots, X_n \sim p$ where p has covariance $\|\Sigma\|_2 \leq 1$ and $\|E[p]\|_2 \leq R$, there exists a computationally efficient $(\varepsilon, 0)$-DP algorithm which outputs $\hat{\mu}$ such that $\|\hat{\mu} - E[p]\|_2 \leq \alpha + O(\sqrt{\eta})$ with probability $1 - \beta$. It requires

$$n = \tilde{O}\left(\frac{d + \log(1/\beta)}{\alpha^2 \varepsilon} + \frac{d \log R + \log(1/\beta)}{\varepsilon}\right) \text{ samples.}$$

No algorithm can succeed with fewer than

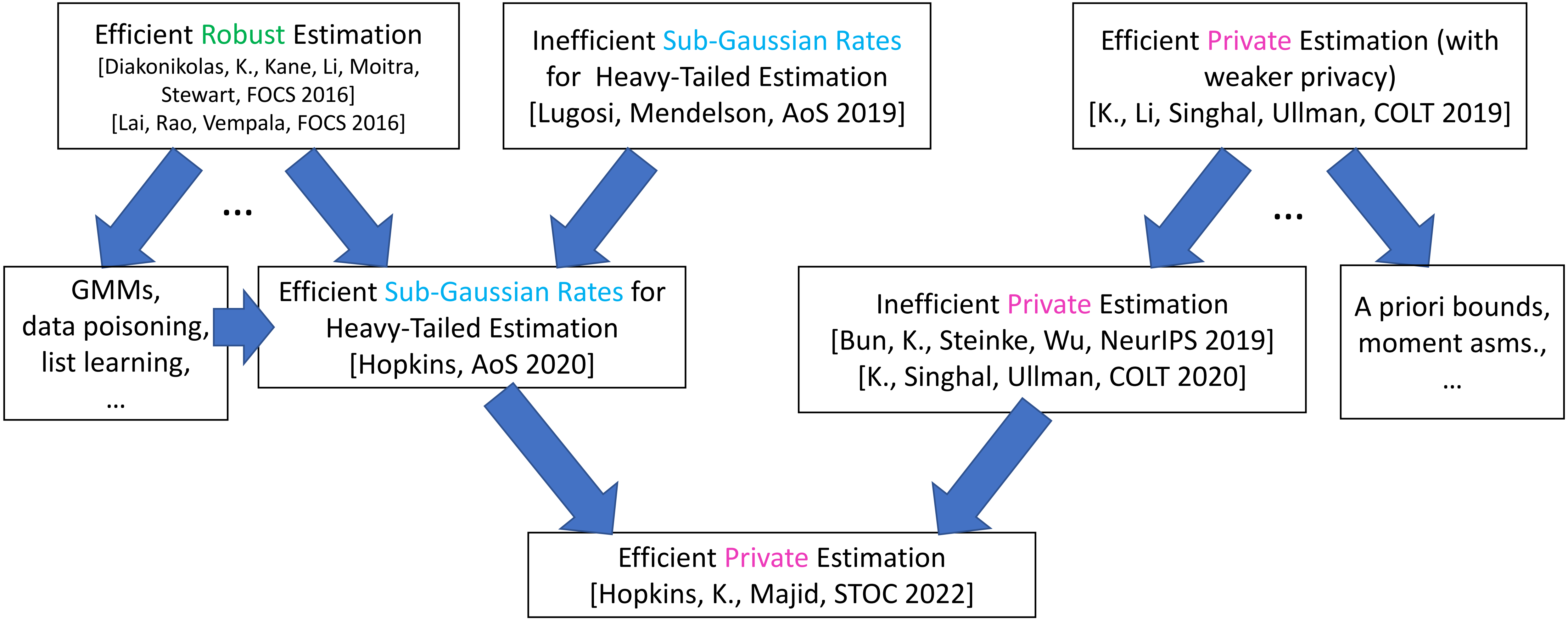
$$n = \Omega\left(\frac{d + \log(1/\beta)}{\alpha^2 \varepsilon} + \frac{d \log R + \log(1/\beta)}{\varepsilon}\right) \text{ samples.}$$

More importantly: new results for Gaussians (including covariance)

[Hopkins, K., Majid, Narayanan], 2022

The plan from here…

1. Three Deficient Private Algorithms
2. Fixing the Exponential Mechanism
3. The Algorithm
4. The Bigger Picture



Robustness, sub-Gaussian rates, and privacy:

All connected by the same technical ideas!

Are Robust and Private Estimation Equivalent?

- Privacy implies robustness…
  - As long as the private algorithm has a very high success probability
  - [Georgiev, Hopkins, NeurIPS 2022]
- Robustness implies privacy…
  - Assuming the “good” solutions have a large enough volume
  - [Hopkins, K., Majid, Narayanan, 2022]
- Close… but still some (significant) gaps

Conclusion

- First efficient pure DP algorithm with $\tilde{O}(d)$ sample complexity for mean estimation
- Also gets robustness and sub-Gaussian tails for free!

Open directions:

- More connections between robust and private estimation?
- Where else can the powerful SoS framework be used for DP estimation?