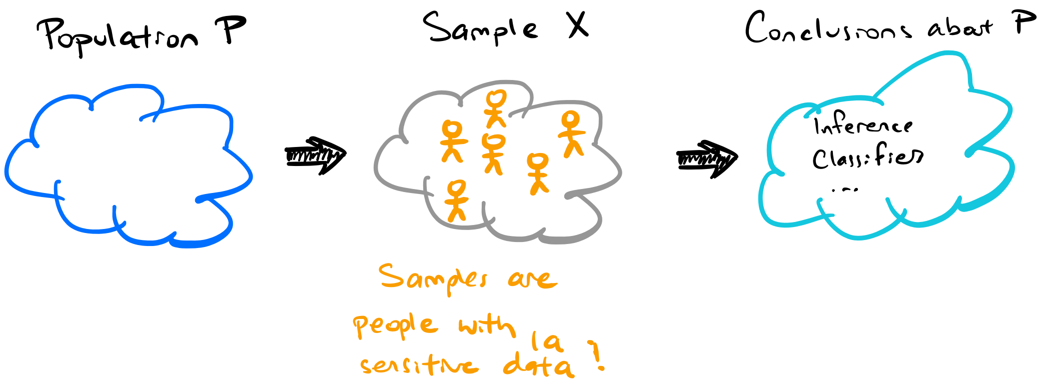
Privacy and the Complexity of Simple Queries

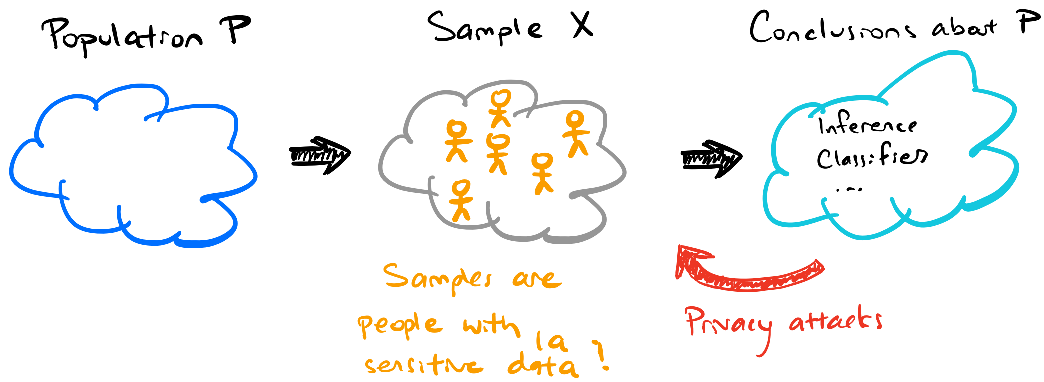
Jonathan Ullman
Northeastern University

Privacy in ML and Statistics



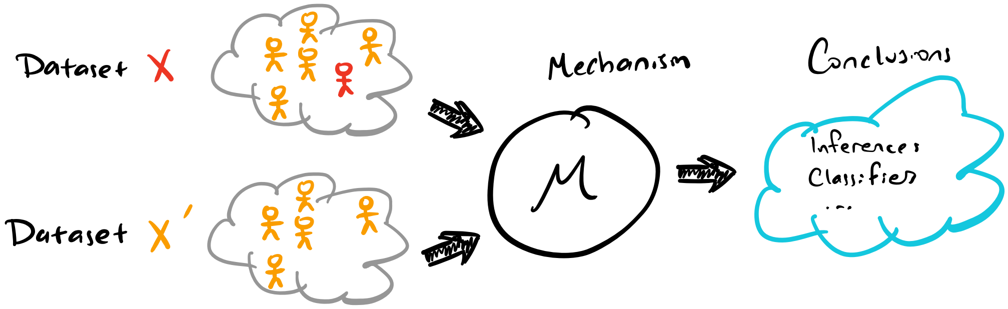
- Goal is to learn about the population while respecting the privacy of the sample

Privacy in ML and Statistics



- Many natural statistical methods reveal a lot of information about the individuals in the sample
  - Reconstruction attacks (DN03)
  - Membership-inference (Homer+08, DSSUV15, SSSS16)
  - Extracting memorized data (CLEKS19)

Differential Privacy (DMNS06)



No attacker can infer much about you because you were sampled

"Distribution stability"

<u>(ε,δ)-Differential Privacy</u>: For every $X \sim X'$ and every $T \subseteq \mathrm{Range}(M)$

$$\mathbb{P}(M(X)\in T) \le e^{\varepsilon}\,\mathbb{P}(M(X')\in T)+\delta$$

$\varepsilon \approx \frac{1}{10}$ $\qquad$ $\delta \approx 2^{-60}$

Private Statistical Queries

$\approx$ SQ Model (K93)

- distribution P over universe $\mathcal{U} = \{\pm 1\}^d$
- queries $q_1 \dots q_k : \{\pm 1\}^d \to \{\pm 1\}$

  $q_j(P) = \mathbb{E}_{x \sim P}(q_j(x))$ $\quad Q(P) = (q_1(P), \dots, q_k(P))$
- dataset $X = (X_1 \dots X_n)$ drawn iid from P

Goal: Given queries $q_1, \dots, q_k$ design an (ε, δ)-dp estimator M such that for all distributions P over $\mathcal{U}$

$$\mathbb{E}_{X \sim P^n}\left(\|Q(P) - M(X)\|_\infty \right) \leq \frac{1}{100}$$

Private Statistical Queries: Algorithms

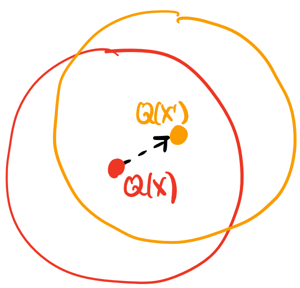
<u>Non-private baseline:</u>

$M(X) = Q(X)$ "Empirical Mean"

$n = O(1)$ samples suffice

<u>Private baseline (Gaussian Mechanism):</u> (DN03, DN04, BDMN05, DMNS06, DKMMN06)

$M(X) = Q(X) + \mathcal{N}(0, \sigma^2 \mathbb{I}_{k \times k})$ "Noisy empirical mean"



$n = \tilde{O}(k^{1/2})$ samples suffice

Sensitivity:

$\sigma \approx \max_{X \sim X'} \| Q(X) - Q(X') \|_2 = \frac{k^{1/2}}{n}$

Private Statistical Queries: Algorithms

Wave 1:

Thm: (DN03, DN04, BDMN05, DMNS06, DKMMN06)

For any set of queries $q_1 \cdots q_k : \{\pm 1\}^d \to \{\pm 1\}$, there is a differentially private algorithm with

sample complexity $n = \tilde{O}(k^{1/2})$ running time $\mathrm{poly}(k)$ (*)

Wave 2:

Thm: (BLR08, DNRRV09, DRV10, RR10, HR10, GRU12, HLM12, NTZ13)

For any set of queries $q_1 \cdots q_k : \{\pm 1\}^d \to \{\pm 1\}$, there is a differentially private algorithm with

sample complexity $n = \tilde{O}(d^{1/2})$ running time $\mathrm{poly}(k, 2^d)$ (*)

Private Statistical Queries

Worst-Case SQs $\qquad k \geq d$

Time (*)	SC Upper Bound	Lower Bound
poly(k)	$\tilde{O}(k^{1/2})$	$\tilde{\Omega}(k^{1/7})^{(*)}$ KMUZ16
poly($k, 2^d$)	$\tilde{O}(d^{1/2})$	$\Omega(d^{1/2})$ BUV14

(*) Assuming program obfuscation

Research Thrust 1: When can we improve the sample complexity for simple families of queries?

Research Thrust 2: When can we improve the computational complexity for simple families of queries?

Private Statistical Queries: Marginals

m-way marginals (aka moments/parities)

$\{q_S\}_{S \subseteq [d], |S| \leq m}$ $\quad q_S(P) = \mathbb{E}_{x \sim P}\left(\prod_{i \in S} x_i\right)$ $\approx d^m$ queries

Very common in applications (e.g. Decennial Census)

Gaussian Mechanism: $\tilde{O}(d^{m/2})$ samples $\approx d^m$ time

Advanced Mechanisms: $\tilde{O}(d^{1/2})$ samples $\approx 2^d$ time

Private Statistical Queries

m-way marginals

constant $m \geq 2$

Time (*)	SC Upper Bound	Lower Bound
poly(k)	$\tilde{O}(d^{m/2})$	—
poly(k, 2^d)	$\tilde{O}(d^{1/2})$	$\Omega(d^{1/2})$ BUV14

<u>Research Thrust 3:</u> Find computationally efficient algorithms for privately estimating marginals or give evidence of computational hardness

I will take you out for AYCE sushi!

Learning-Based Mechanisms (GHRU11, HRS12, TUV12, ...)

$$f_P : Q \to [-1,1] \qquad f_P(q) = q(P)$$

Try to learn f from examples

error proportional to $|Q'|$

1: Choose random $Q' \subseteq Q$

2: Use Gaussian mechanism to obtain $\boxed{\approx f_P(q) \text{ for } q \in Q'}$

3: $\boxed{\text{Fit a function } \hat{f} \in \mathcal{H}}$ so that $\hat{f}(q) \approx f_P(q)$ for $q \in Q$

sample and computational complexity depends on $\mathcal{H}$

Non-trivial algorithms: E.g. $n = d^{O(m^{1/2})}$ samples (TUV12)

Learning-Based Mechanisms (GHRU11, HRS12, TUV12, ...)

For m-way marginals, the learning problem is <u>noisy tensor completion</u>

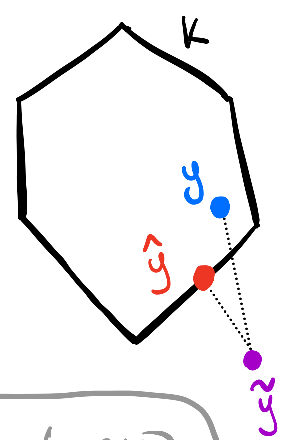
$$f_p(i_1, \ldots, i_m) = \underbrace{\mathbb{E}_{x \sim P}\left(\prod_{\ell=1}^{m} x_{i_\ell}\right)}_{\substack{\text{convex combination} \\ \text{of rank-1 } m\text{-tensors}}}$$

<u>Thm</u>: $\tilde{O}(d)$ entries are enough for tensor completion

<u>Thm</u> (BM15): Assuming hardness of refuting 3-XOR, any efficient algorithm for 3-tensor completion requires $\Omega(d^{3/2})$ entries.

$\Longrightarrow$ Efficient private algs in this framework need $\Omega(d^{3/4})$ samples

The Projection Mechanism (NTZ13)



$$K = \left\{ a \in \mathbb{R}^k : \begin{matrix} a = Q(P) \text{ for} \\ \text{some } P \text{ over } \{\pm 1\}^d \end{matrix} \right\}$$

polytope w/ 2^d vertices

1: $y = Q(x)$ is the sample mean

2: $\tilde{y} = Q(x) + \mathcal{N}(0, \sigma^2 \mathbb{I}_{k \times k})$

3: $\hat{y} = \operatorname{argmin}_{p \in K} \|p - \tilde{y}\|_2$

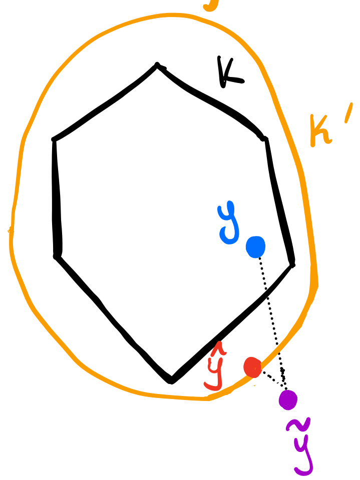
requires 2^d time

Thm (NTZ13):

$\|\hat{y} - y\|_2 \propto$ "Gaussian width of K"

$$w(K) = \mathbb{E}_{g \sim \mathcal{N}(0, \mathbb{I})} \left(\max_{z \in K} |\langle g, z \rangle| \right)$$

The Projection Mechanism (NT Z13)



$K = \left\{ a \in \mathbb{R}^k : \begin{array}{c} a = Q(P) \text{ for} \\ \text{some } P \text{ over } \{\pm 1\}^d \end{array} \right\}$

__Approach:__ Convex Relaxations

K' is a "nice" convex set

① $K \subseteq K'$

② K' admits efficient projection

③ K' has small Gaussian width

__Thm__ (DNT13):

There is a poly-time SDP algorithm for m-way marginals using $\tilde{O}(d^{\lceil m/2 \rceil / 2})$ samples

(optimal for $m = 2$)

Private Marginals

m-way marginals

constant $m \geq 2$

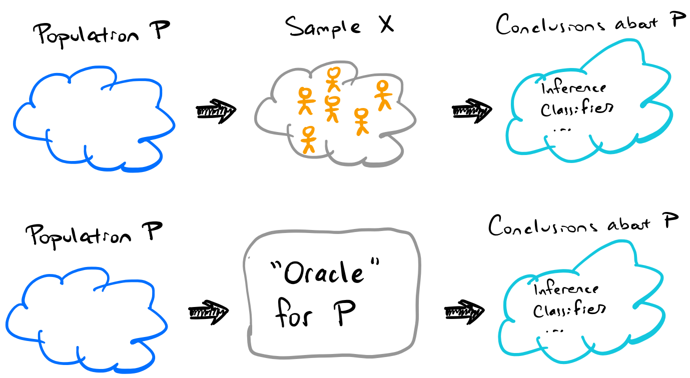
Time (*)	SC Upper Bound	Lower Bound
poly(k)	$\tilde{O}(d^{Cm^{1/2}}) / \tilde{O}(d^{\lceil m/2 \rceil/2})$	—
poly(k, 2^d)	$\tilde{O}(d^{1/2})$	$\Omega(d^{1/2})$ BUV14

<u>Research Thrust 3:</u> Find computationally efficient algorithms for privately estimating marginals or give evidence of computational hardness

Why Are We Stuck?

Answer 1: We used up all the cryptography

Answer 2: We dont really understand how privacy restricts algorithm design

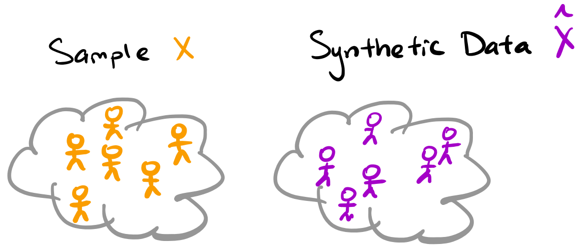


e.g. a statistical query oracle

What I Haven't Talked About

Synthetic Data:

Output a new sample $\hat{X}$ of "fake" data such that $\| Q(X) - Q(\hat{X}) \|_\infty \leq \frac{1}{100}$

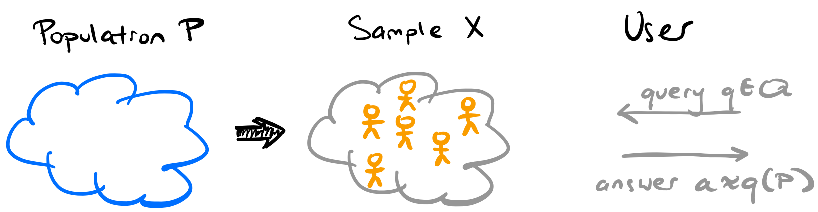


Thm (DNRRV09, UV11):
No polynomial time algorithm with polynomial samples can output private synthetic data even for 2-way marginals!

What I Haven't Talked About

Interactive Algorithms:

What if we have the user tell us which queries they need answered?



Thm (U13):

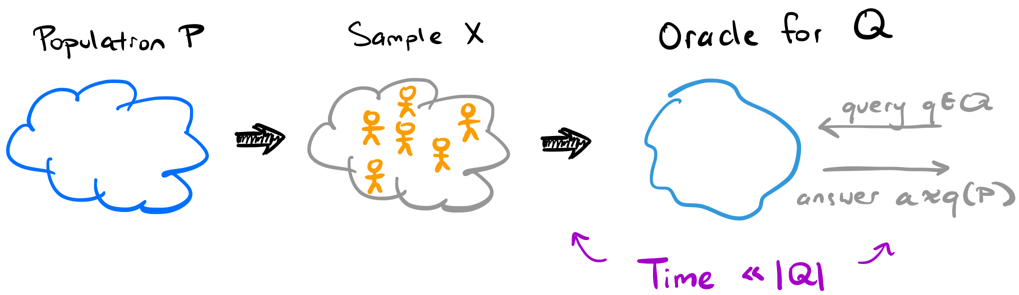
Any polynomial time private algorithm that answers k arbitrary queries needs $\tilde{\Omega}(k^{1/2})$ samples

What I Haven't Talked About

Sublinear-Time Algorithms:

Can we generate private summaries in time poly(|X|)?

(e.g. all m-way marginals in poly(d, m) time)



What I Haven't Talked About

Connections to Robust Statistics:

(DL09) et seq.

(ε,δ)-Differential Privacy: For every $X \sim X'$ and every $T \subseteq \text{Range}(M)$

$$\mathbb{P}(M(X) \in T) \le e^{\varepsilon}\,\mathbb{P}(M(X') \in T) + \delta$$

Differential Privacy $\Longrightarrow$ "Strong Robustness"

But we're often in a regime where "standard robustness" is tractable

Thanks !!!