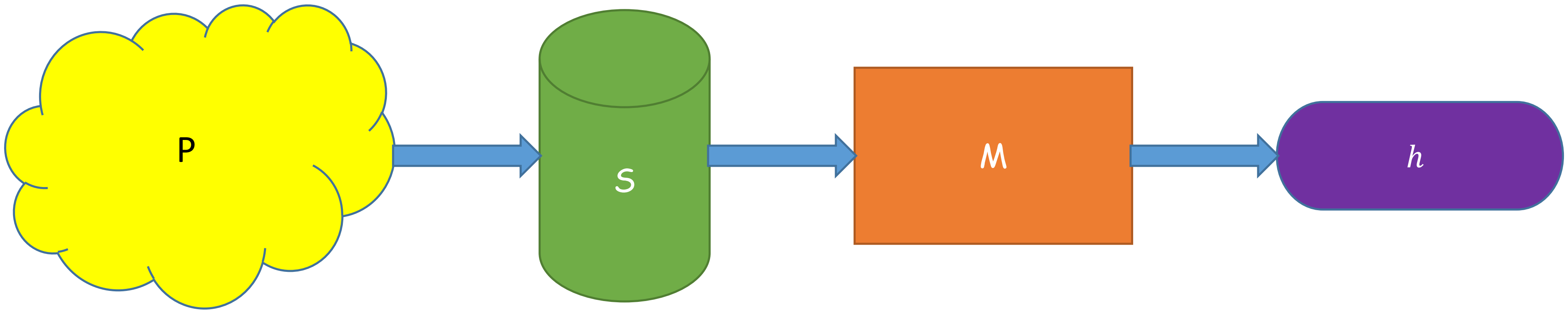
Private Learning - 2



Kobbi Nissim, Georgetown University

Bar-Ilan Winter School on Differential Privacy
February 2017

Pure-Privately Learning Points

x	**1**	**2**	**…**	***i*-1**	***i***	***i*+1**	**…**	
$c_i(x)$	0	0	…	0	1	0	…	0

- Concept class: $C = POINT_d = \{c_1, \ldots, c_{2^d}\}$
- Recall: Proper point learner with O(1) samples
- Generic construction of private learners results in O(log |C|) = O(d) samples
  - Is the gap essential?
- Thm 1 [BKN 10]: Proper pure-private PAC learner of Points must use $\Omega\left(\frac{d}{\epsilon}\right)$ samples.
- Proof:
  - PAC learning: On database

x	i	i	…	i
c(x)	1	1	…	1

learner must return c_i w.p. > 1/2

  - Pure Differential Privacy: By group privacy, learner must return c_i with probability ≥ $e^{-\epsilon n}$ on database

x	1	1	…	1
c(x)	1	1	…	1

  - There are 2^d-1 options for i≠1, hence need $(2^d - 1) \cdot e^{-\epsilon n} < 1/2$. Hence, $n = \Omega\left(\frac{d}{\epsilon}\right)$.
- Can we do better?

An improper private learner for POINT_d [BKN10, BNS14]

- Choose a family of $m = O(\frac{1}{\alpha})$ hypotheses H as follows:
  - Construct h_i by setting $h_i(x) = 1$ with probability $\frac{\alpha}{4}$ and $h_i(x) = 0$ otherwise.
  - Let $H = \{h_i\}$
    - Efficiency: enough if entries of h_i are pairwise independent
- Use exponential mechanism to choose $h \in H$ with small error
  - This would work well if H contains a hypothesis with error $\frac{\alpha}{2}$

Note: We apply generic construction on H instead of POINTd

- Fix $c_j \in POINT_d$ and a distribution P on $\{0,1\}^d$
- Claim: w.p. $\geq \frac{1}{2}$ H contains h_i s.t. $error_P(c_j, h_i) \leq \frac{\alpha}{2}$.
- Proof:
  - $E\left[error_P(c_j, h_i) \middle| h_i(j) = 1\right] \leq \frac{\alpha}{4}$.
  - By Markov's inequality $\Pr\left[error_P(c_j, h_i) > \frac{\alpha}{2} \middle| h_i(j) = 1\right] \leq \frac{1}{2}$.
  - $\Pr\left[error_P(c_j, h_i) \leq \frac{\alpha}{2}\right] \geq \Pr[h_i(j) = 1] \Pr\left[error(h_i) \leq \frac{\alpha}{2} \middle| h_i(j) = 1\right] \geq \frac{\alpha}{8}$.
  - H fails to contain h_i s.t. $error_P(c_j, h_i) \leq \frac{\alpha}{2}$ w.p. $\leq \left(1 - \frac{\alpha}{8}\right)^m \leq \frac{1}{2}$ if $m > O(\frac{1}{\alpha})$

Representation of concept classes [BNS13]

- **Probabilistic Representation for class $\mathcal{C}$:** a list of hypothesis classes $H_1, \dots, H_r$ s.t.
  - for every $c \in \mathcal{C}$ and distribution P over examples,
  - w.p. ¾ , a randomly chosen H_i contains a hypothesis h s.t. $error_P(c, h) \leq$ ¼.
  - The size of Rep is defined as $\max_i \log |H_i|$
- $RepDim(\mathcal{C})$: the size of C's minimal probabilistic representation
- Theorem [BNS13]: $\Theta(RepDim(\mathcal{C}))$ samples are necessary and sufficient for pure-privately learning $\mathcal{C}$ (improperly)

Concept class	learner	Sample complexity (pure DP)	Sample complexity (approx DP)
C	Proper	$O(\log\|C\|)$ [KLNRS'08]	
	Improper	$\Theta(RepDim(C))$ [BNS'13]	
points	Proper	$\Theta(\log(T))$ [KLNRS'08, BKN'10]	$\Theta(1)$ [BNS'14]
	Improper	$\Theta(1)$ [BKN'10, BNS'13]	
thresholds	Proper	$\Theta(\log(T))$ [KLNRS'08, BKN'10]	$2^{O(\log^* T)}$ [BNS'14], $\Omega(\log^* T)$ [BNSV'15]
	Improper	$\Theta(\log(T))$ [FX'13]	

Back to Example 0: Learning points with approx. differential privacy

A_{dist} by Smith & Thakurtha:

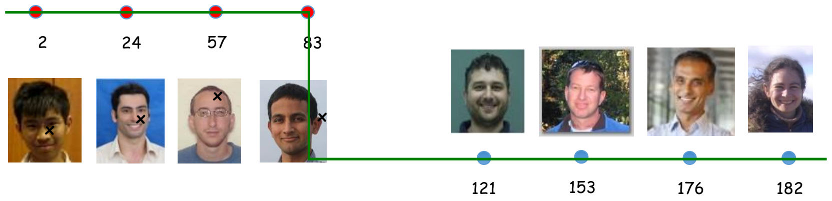
- Inputs:
  - A set of possible solutions $\boldsymbol{F}$
  - Database $\boldsymbol{S} \in \boldsymbol{X}^*$
  - Sensitivity-1 quality function $\boldsymbol{q}: \boldsymbol{X}^* \times \boldsymbol{F} \to \mathbb{R}$
- Algorithm:
  1) Let $\boldsymbol{f_1} \neq \boldsymbol{f_2}$ be two highest score solutions in $\boldsymbol{F}$, where $\boldsymbol{q}(\boldsymbol{S}, \boldsymbol{f_1}) \geq \boldsymbol{q}(\boldsymbol{S}, \boldsymbol{f_2})$
  2) Compute $\boldsymbol{gap}(\boldsymbol{S}) = \boldsymbol{q}(\boldsymbol{S}, \boldsymbol{f_1}) - \boldsymbol{q}(\boldsymbol{S}, \boldsymbol{f_2})$ and $\boldsymbol{gap}^* = \boldsymbol{gap}(\boldsymbol{S}) + \boldsymbol{Lap}\left(\frac{1}{\epsilon}\right)$
  3) If $\boldsymbol{gap}^* < \frac{1}{\epsilon}\log\left(\frac{1}{\delta}\right)$ then output $\perp$ and halt. Otherwise, output $\boldsymbol{f_1}$

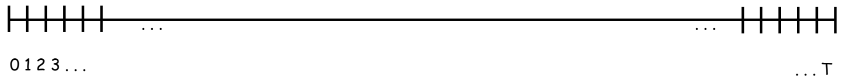
Back to Example 0: Learning points with approx. differential privacy

- Given a labeled sample $\boldsymbol{S} = (\boldsymbol{x_i}, \boldsymbol{y_i})_{\boldsymbol{i=1}}^{\boldsymbol{m}}$, define the quality of a domain element $\boldsymbol{z} \in \boldsymbol{X}$ as:
  - $\boldsymbol{q(S, z) = |\{i \ : \ (x_i = z\ )\ and\ (y_i = 1)\}|}$

- Learner for Points [BNS'14]
  - Execute $\boldsymbol{A_{dist}}$ on $\boldsymbol{S}$ and $\boldsymbol{q}$.
  - If returned $\perp$, output a random $\boldsymbol{h} \in \mathbf{POINT}_{\boldsymbol{d}}$.
  - Else, if a domain element $\boldsymbol{j}$ was returned, then return $\boldsymbol{h = c_j}$.

Back to Learning Thresholds

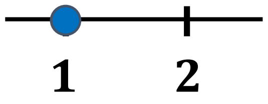




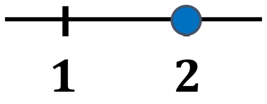
- Why?
  - Seems fundamental and simple
    - "should not be too hard", disturbing difference between private and non-private setting
  - [BNSV'15] Equivalent under differential privacy to:
    - Distribution learning:
      - D – unknown distrib over X with cumulative F_D
      - Goal: Given oracle access to D, find F: X → [0,1] with small $|F(x)-F_D(x)|$ for all $x \in X$
    - Query release:
      - Given points $(x_1,\dots,x_n) \in X$, output data structure approximating $|\{i : x_i < z\}|/n$ for all $z \in X$
    - (Approximate) Median:
      - Given points $(x_1,\dots,x_n) \in X$, output z such that (approx.) half the points are smaller/greater than z
    - Interior point:
      - Given points $(x_1,\dots,x_n) \in X$, output z between min and max points

Solving Interior Point with Approx DP Requires Ω(log*T) Samples [BNVS'15]

- Observe: Impossible to have $\boldsymbol{n} = \mathbf{1}$ when T ≥ 2



Output 1 w.p. $\geq \frac{3}{4}$

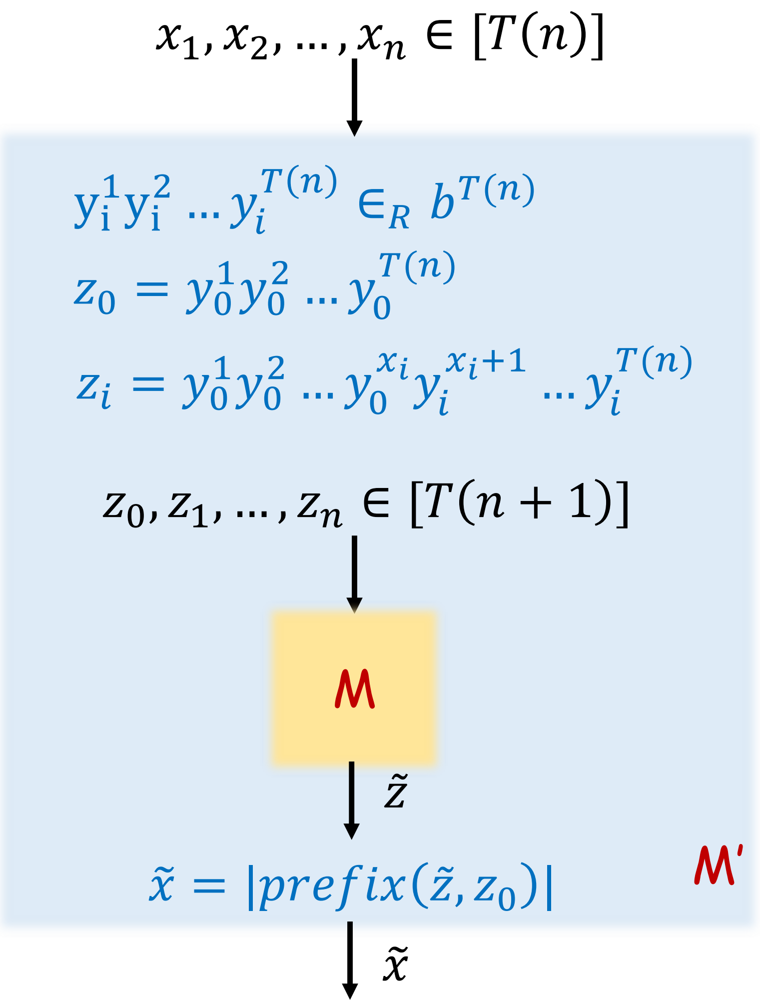


Output 1 w.p. $\geq \frac{\frac{3}{4}-\delta}{e^{\epsilon}} > \frac{1}{4}$

- Strategy: Induction
  - Approx. dp mechanism M for solving IP over T(n+1) w/ n+1 samples

  → Approx. dp mechanism M' for solving IP over T(n) w/ n samples
  - Where $T(n+1) = b(n)^{T(n)}$

Solving Interior Point with Approx DP Requires Ω(log*T) Samples[BNVS'15]



- If M approx private so is M'

- Suppose M succeeds
  - $z_0, z_1, \dots, z_n$ all share a prefix of length $\min(x_1, \dots, x_n)$ and hence $\tilde{z}$ also shares this prefix with z_0
  - Hence, $\tilde{x} \geq \min(x_1, \dots, x_n)$

- Let $w = \max(x_1, \dots, x_n)$
  - If $\tilde{x} > w$ then $\tilde{z}$ reveals y_0^{w+1}
  - By approx. privacy, this can happen with probability at most $\frac{e^{\epsilon}}{b} + \delta$

Privacy used for claiming correctness

Variations on a the PPAC learning model

- Pure-privacy, proper
  - Generic construction, but exhibits higher sample complexity than in non-private learning
- Pure-privacy, improper
  - Characterization of sample complexity in terms of randomized representation
  - Limited gain in sample complexity (POINTS but not THREHOLD)
- Approximate privacy, proper
  - Mostly not well understood
  - Improved sample complexity (POINTS and THRESHOLD, but cannot learn THRESHOLD over the reals)
- Label privacy
  - Significantly weaker notion of privacy (label protected but not sample)
  - Characterization of sample complexity in terms of VC dimension
- Semi-Supervised learning
  - Some examples labeled, most are not
  - Characterization of labeled sample complexity in terms of VC dimension

Semi-Supervised Learning [BNS'15]

- **Input:** batches of labeled and unlabeled samples
- **Generic construction:**
  Every finite concept class C can be learned privately using $O(\mathrm{VC}(C))$ labeled examples.
  - The construction uses $O(\log |C|)$ unlabeled examples

- **Boosting the labeled sample complexity:**
  Given a private learner for a concept class C, it is possible to reduce its labeled sample complexity to $O(\mathrm{VC}(C))$.
  - While maintaining the unlabeled sample complexity

Reducing the labeled sample complexity of a given learner $\mathcal{A}$

- Base learner $\mathcal{A}$ with sample complexity n.
- Input: Database S of size n, partially labeled

1. Let H be the set of all dichotomies over S realized by the target concept class C
2. Choose $h \in H$ using the exponential mechanism with the **labeled portion** of S
3. Relabel S using h,
4. Execute $\mathcal{A}$

x_1 ,	y_1
x_2 ,	$\cancel{y_2}$ $\hat{y}_2$
x_3 ,	y_3
x_4 ,	y_4
$\vdots$	
x_t ,	y_t
x_{t+1} ,	$\cancel{?}$ $\hat{y}_{t+1}$
x_{t+2} ,	$\cancel{?}$ $\hat{y}_{t+2}$
x_{t+3} ,	$\cancel{?}$ $\hat{y}_{t+3}$
$\vdots$	
x_n ,	$\cancel{?}$ $\hat{y}_n$

Reducing the labeled sample complexity of a given learner $\mathcal{A}$

- Base learner $\mathcal{A}$ with sample complexity $\boldsymbol{n}$.
- Input: Database $\boldsymbol{S}$ of size $\boldsymbol{n}$, partially labeled

$\exists \mathbf{f} \in \mathbf{H}$ s.t. $\mathbf{error_S(f)} = \mathbf{0}$

1. Let $\boldsymbol{H}$ be the set of all dichotomies over $\boldsymbol{S}$ realized by the target concept class $\boldsymbol{C}$

2. Choose $\boldsymbol{h} \in \boldsymbol{H}$ using the exponential mechanism with the **labeled portion** of $\boldsymbol{S}$

If S contains $\approx \mathbf{VC(C)\log|}S\mathbf{|}$ labeled examples then $\mathbf{h}$ is close to the target concept

3. Relabel $\boldsymbol{S}$ using $\boldsymbol{h}$,

$\mathcal{A}$ returns a hypothesis that is close to $\mathbf{h}$

4. Execute $\mathcal{A}$

Reducing the labeled sample complexity of a given learner $\mathcal{A}$

- Base learner $\mathcal{A}$ with sample complexity $\boldsymbol{n}$.
- Input: Database $\boldsymbol{S}$ of size $\boldsymbol{n}$, partially labeled

1. Let $\boldsymbol{H}$ be the set of all dichotomies over $\boldsymbol{S}$ realized by the target concept class $\boldsymbol{C}$

Difficulty:
H depends on S!
Outputting h would breach privacy!

2. Choose $\boldsymbol{h} \in \boldsymbol{H}$ using the exponential mechanism with the **labeled portion** of $\boldsymbol{S}$

Solution:
use h to relabel sample, analyze distribution of relabeled databases

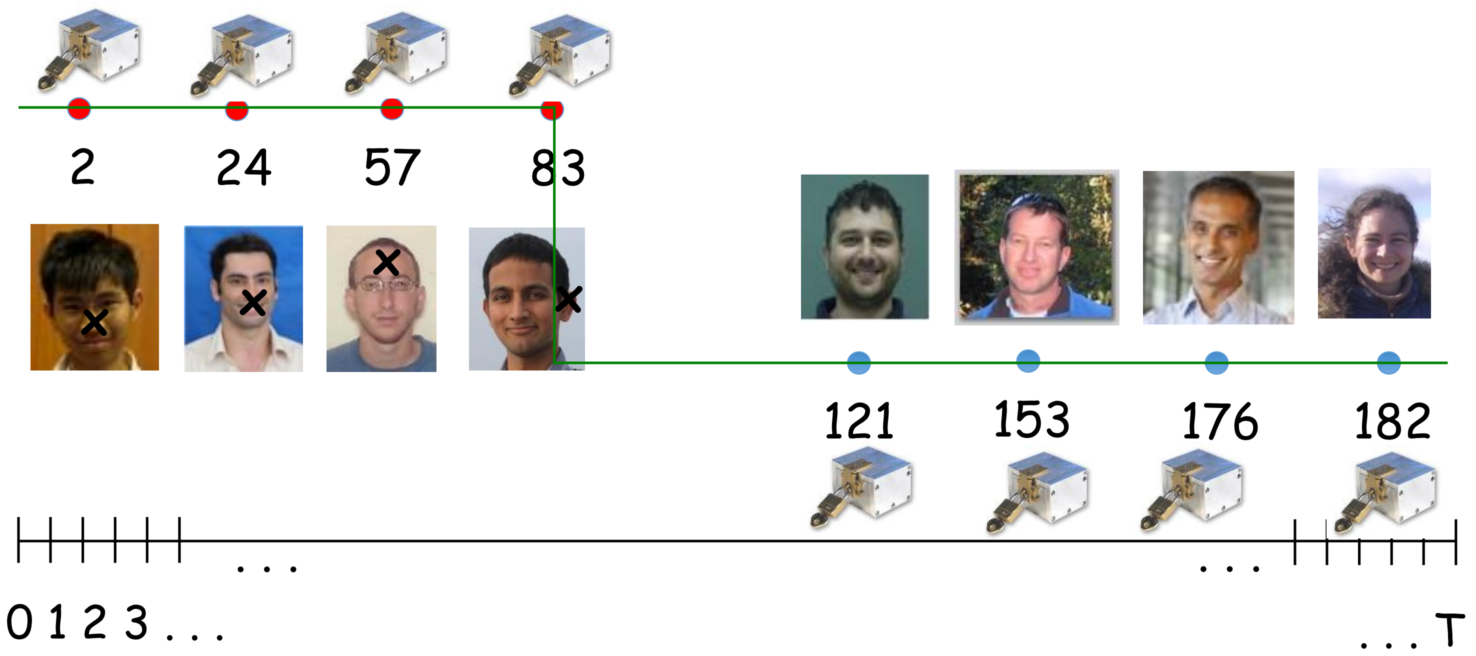
3. Relabel $\boldsymbol{S}$ using $\boldsymbol{h}$

4. Execute $\mathcal{A}$

Thresholds and Computational Complexity [Bun-Zhandry'15]

- Order Revealing Encryption:
- Learn  {<, >, =}  but nothing else

- Efficiently learnable, but not efficiently privately learnable



Thanks: Mark Bun

Epilog: why study private learning?

- Real-world implementations of learning algorithms can lead to loss of privacy
  - Recall Vitaly Shmatikov's talks!
  - Carefully distinguish privacy breaches from doing science
  - McSherry's "secrets about you" vs. "your secrets"
- Learning: a basic task that abstracts many of the computations performed on collections of private individual data
  - Hence, important to understand to what extent it can be done under the restriction of differential privacy
- A test bed for many ideas (e.g., how to circumvent sample complexity bounds)
- Learning intimately related with differential privacy
  - Learning theory tools useful for privacy [BLR'08, HR'10]
  - Differential privacy implies generalization [McSherry, DFHPRR'15, BNSSU'15]
    - Useful even when privacy is not the goal!

What have we Learned?

- PAC learning exhibits a lot of complexity under differential privacy
  - Even for simple complexity classes like points and thresholds
  - A variety of applicable strategies, quite a full picture
    - Still open: improper learning and characterization of sample complexity under approx. privacy
  - Crypto used for showing hardness (fingerprinting codes, order-revealing encryption)
    - But can it be used positively?

Some more references

- Synthetic data
  - Avrim Blum, Katrina Ligett, Aaron Roth: A learning theory approach to non-interactive database privacy. STOC 2008
- Boosting
  - Cynthia Dwork, Guy N. Rothblum, Salil P. Vadhan: **Boosting and Differential Privacy.** FOCS 2010
- Continuous domains:
  - Kamalika Chaudhuri, Daniel J. Hsu: Sample Complexity Bounds for Differentially Private Learning. COLT 2011
- Other machine learning
  - See Adam's talk