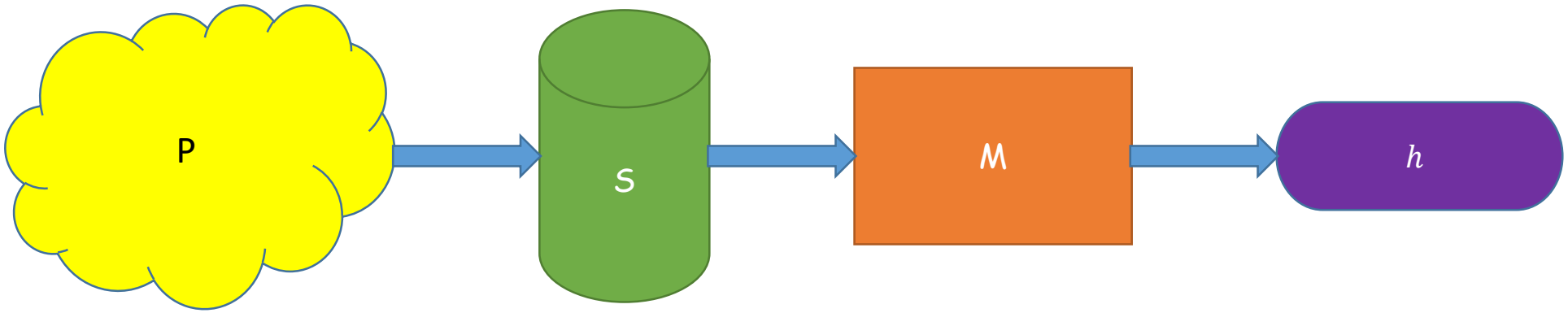
Private Learning - 1

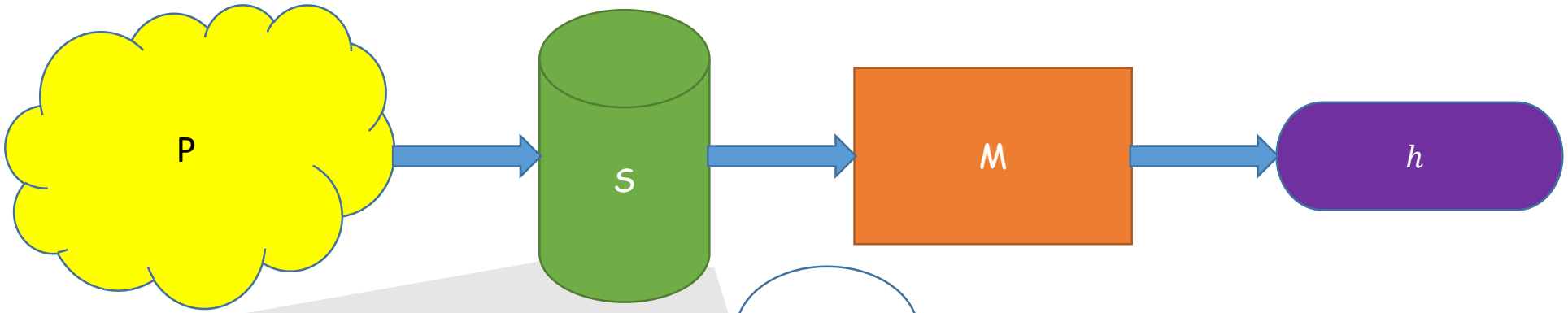


Kobbi Nissim, Georgetown University

Bar-Ilan Winter School on Differential Privacy

February 2017

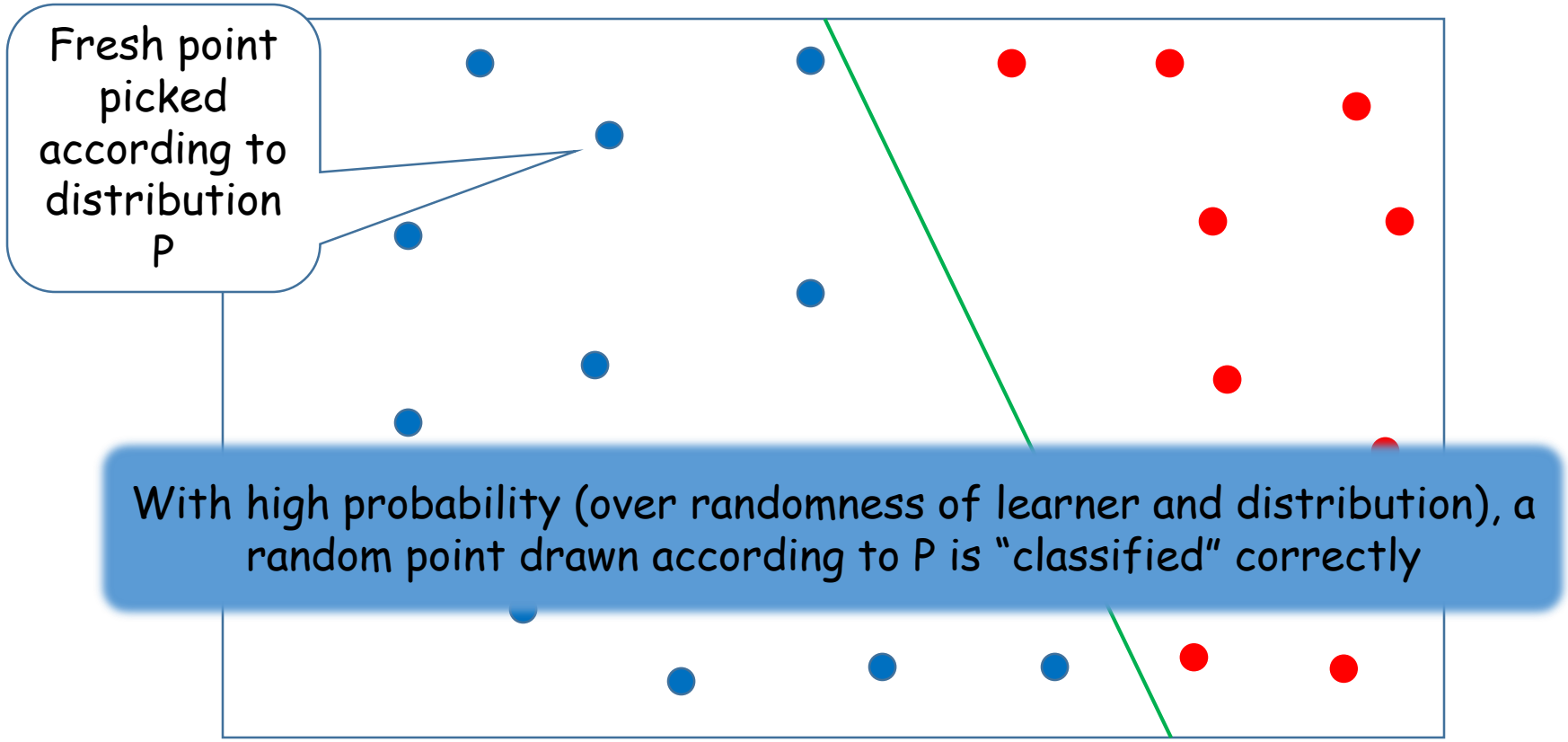
What is learning?



% down	High Debt	Other accts	Mmp /inc	Good Risk?
10	No	Yes	0.32	**Yes**
10	No	No	0.25	**Yes**
5	Yes	No	0.30	**No**
20	No	Yes	0.31	**Yes**
10	No	No	0.25	**Yes**

- Given this data, some reasonable rules might be
  - Predict YES if …
    - 100*(Mmp/inc) – (% down) < 25
    - (!High Debt) AND (% down > 5)
    - …

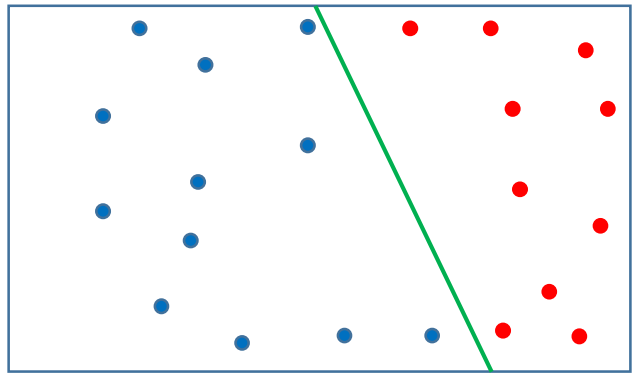
What is learning?



A distribution P on X. Each point in X labeled 0/1

Samples drawn according to P

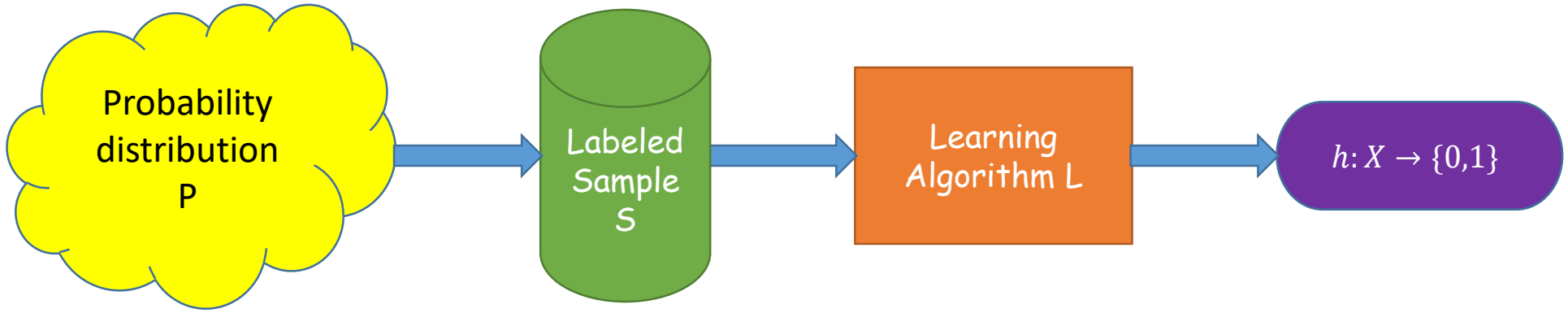
Notation



A distribution P on X.
Each point in X labeled 0/1
Samples drawn according to P

- X : set of all possible (unlabeled) instances
- A concept is a predicate over X: c: $X \rightarrow \{0,1\}$
- A concept class C is a set of concepts
  - We will assume that instances are labeled by a target concept $c \in C$, the label of instance x is $c(x)$
- P is an unknown distribution over X
- L is a learning algorithm
- Goal of learning algorithm: output a hypothesis h that "approximates" the target concept c on the distribution P
  - If $h \notin C$ is allowed we call the learner improper

PAC learning [Valiant 84]



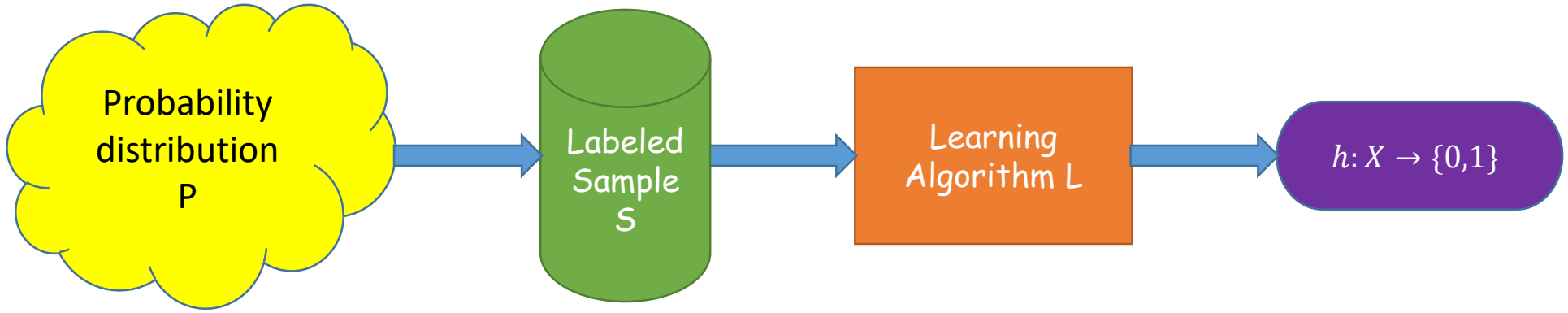
- For a probability distribution P, concept c and hypothesis h define

$$error_P(c,h) = \Pr_{x \sim P}[c(x) \neq h(x)]$$

- Ideally, we would like our learner to produce a hypothesis h with zero error
- Error is inevitable:
  - There is a chance that the labeled sample S does not contain enough information to produce an error-less hypothesis
- Relax requirements:
  - Typically, h approximately correct
  - With small probability not even that

PAC : Probably Approximately Correct

PAC learning [Valiant 84]

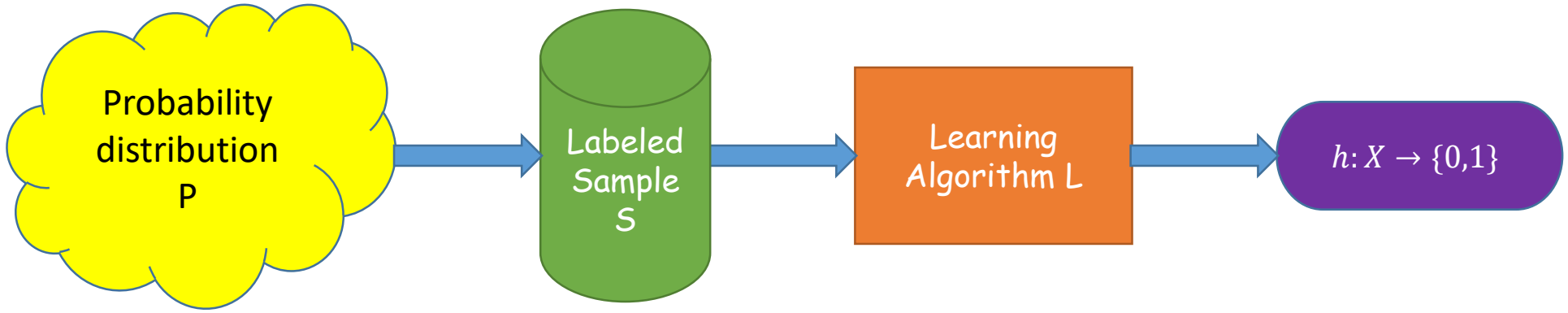


- Recall: $error_P(c,h) = \Pr_{x \sim P}[c(x) \neq h(x)]$

- L is a (α, β)- PAC learner for concept class C if
  - For all distributions P and target concepts $c \in C$
  - After receiving a polynomial number of examples sampled i.i.d. from P and labeled according to c it outputs a hypothesis h such that

$$\Pr[error_P(c,h) < \alpha] \geq 1 - \beta$$

Learning over sample vs. over distribution



- Suppose that given $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ the learner outputs a hypothesis h such that

$$error_S(c, h) = \Pr_{i \in_R [n]}[h(x_i) \neq y_i] \leq \alpha$$

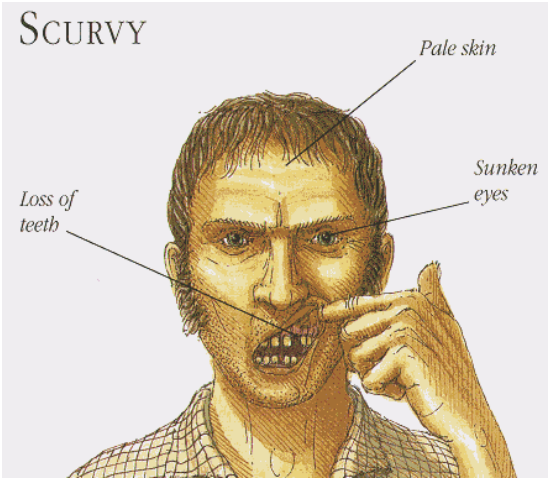
  - Is $error_P(c, h)$ small?
    - Not necessarily as h may not generalize
    - E.g., choose h s.t. $h(x_i) = y_i$ for all i and let $h(x)$ be random for $x \notin S$

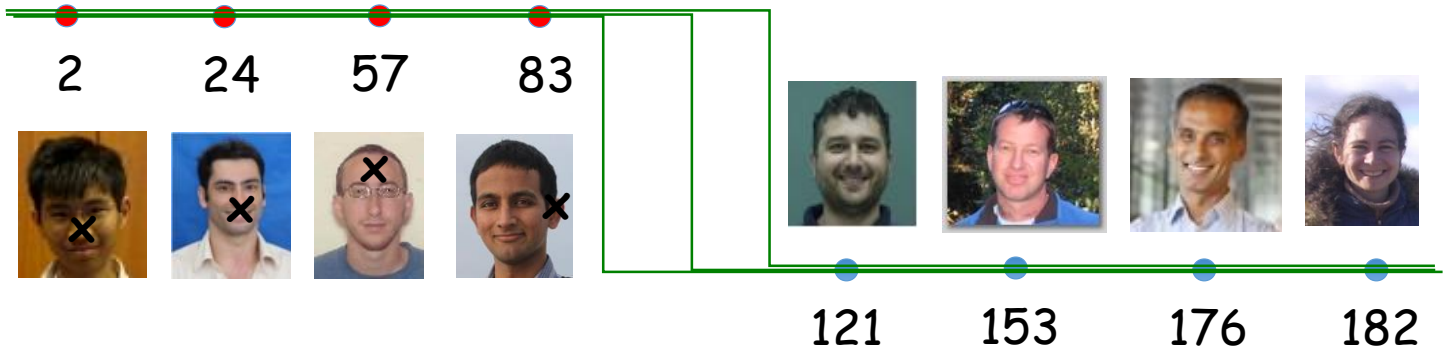
Questions asked about PAC learning

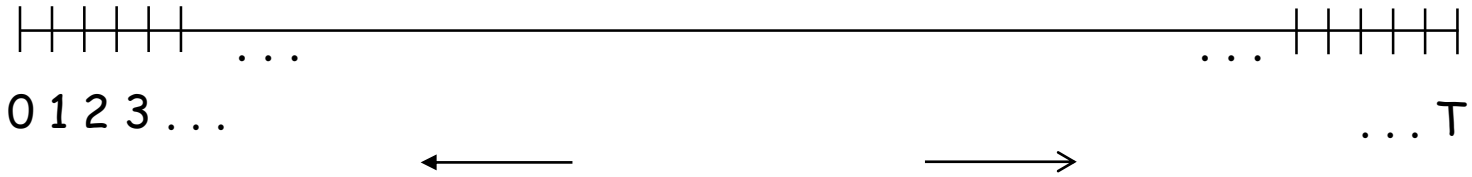
- Sample complexity: Lower and upper bounds on the number of samples needed to learn a concept class C
- Time complexity: Time complexity of learning a concept class C
- Is there a benefit in improper learning (i.e., allowing $h \notin C$)?
  - Yes! Sometimes proper learning intractable but improper learning tractable!

Let’s do some science!

- Scurvy: a problem throughout human history
- Caused by vitamin C deficiency
- How much vitamin C is enough?



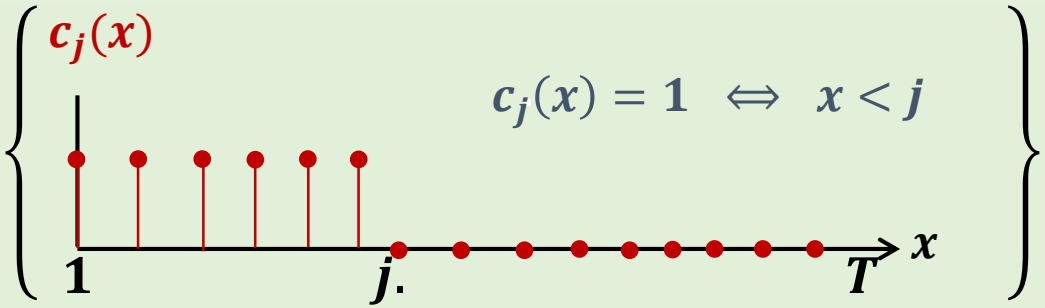




Thanks: Mark Bun

Example 1: learning thresholds

- Concept Class: $\mathcal{C} = \mathbf{THRESHOLD_d} = \{c_0, \ldots, c_{2^d}\}$

$\mathbf{THRESHOLD}_d = \left\{ \right.$ 

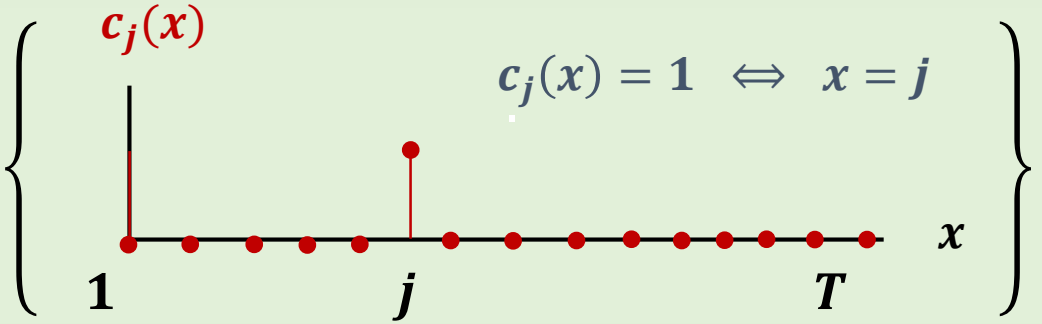
$\left. \right\}$

- Learning over sample is easy: output any consistent hypothesis from $\mathrm{THRESHOD_d}$
  - Is this good?

Let's consider something even simpler…

Example 0: learning points

- Concept Class: $\mathcal{C} = \mathbf{POINT_d} = \{c_1, \ldots, c_{2^d}\}$

$POINT_d =$ 

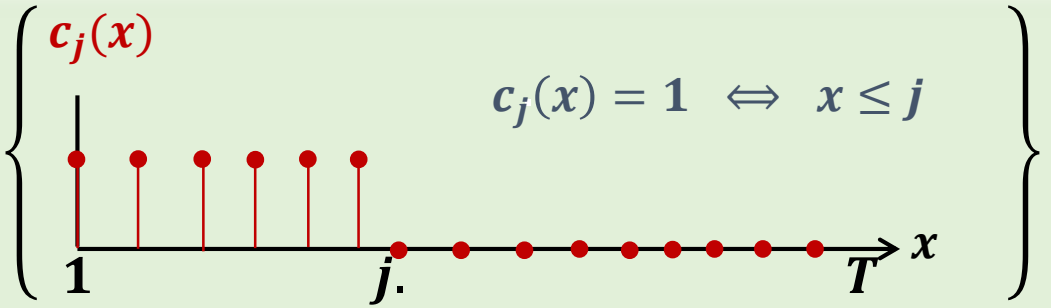
- A PAC learner for $\mathrm{POINT_d}$ with $O(1)$ samples:
  - If there exists i s.t. $(i, 1)$ in the sample, return $h = c_i$.
  - Otherwise (all labels are zero), return $h \equiv 0$ (or a random $h \in C$}
- $error_P(c_j, h) > \alpha$ only if $= \Pr_{x \sim P}[x = j] > \alpha$ but sample contains no example $(j, 1)$
- Happens w.p. $\leq (1 - \alpha)^n$, to get (α, β)-PAC take $n > O(\frac{\log \frac{1}{\beta}}{\alpha})$

PAC learning finite concept classes (Occam's Razor)

- Let C be a finite concept class
- $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$: a sample of n i.i.d. examples sampled from P and labeled according to c, i.e., $y_i = c(x_i)$
- We say that a hypothesis $h \in C$ is consistent with S if $h(x_i) = y_i$ for all i
- Learner:
  - Input: $\mathrm{n} \geq \frac{\log |C| + \log \frac{1}{\beta}}{\alpha}$ labeled samples
  - Output: a consistent hypothesis
- Proof of learner's correctness:
  - For any hypothesis h with $error_P(c, h) > \alpha$, $\Pr[h\ consistent] \leq (1-\alpha)^n \leq e^{-\alpha n}$
  - Hence, probability algorithm outputs h with $error_P(c, h) > \alpha$ is at most $|C|e^{-\alpha n}$
  - Taking $\mathrm{n} \geq \frac{\log |C| + \log \frac{1}{\beta}}{\alpha}$ we get that the probability of such event is at most β

Example 1: learning thresholds

- Concept Class: $\mathcal{C} = \mathbf{THRESHOLD_d} = \{c_0, \ldots, c_{2^d}\}$

$\mathbf{THRESHOLD}_d = \{$ 

$\}$

- A PAC learner for $\mathrm{THRESHOD_d}$ with $O(1)$ samples:
  - Let i be largest s.t. $(i, 1)$ in the sample, return $h = c_i$.
  - Otherwise (all labels are zero), return $h \equiv 0$ (or a random $h \in C$}

VC dimension

- Given a collection of distinct points $S = (x_1, \dots, x_n)$ and a concept c, define the dichotomy $c(S) = (c(x_1), \dots, c(x_n))$ (i.e., a string of n bits)
- Given $S = (x_1, \dots, x_n)$ and a concept class C define $C(S) = \{c(S): c \in C\}$
- If $|C(S)| = 2^n$ we say that S is shattered by C

- Example: Let $X = \mathbb{R}$ and $C = \{c_{a,b}: a, b \in \mathbb{R}\}$ where $c_{a,b}(x) = 1_{a \le x \le b}$
  - Let $S = (2, 7)$
    - $c_{0,1}(S) = (0,0); c_{1,3}(S) = (1,0); c_{3,10}(S) = (0,1); c_{0,10}(S) = (1,1)$. Hence, S is shattered by C
    - No S containing three distinct points is shattered by C: dichotomy $(1,0,1)$ is impossible

$$VC(C) = 2$$

- The Vapnik Chervonenkis dimension of concept class C, denoted $VC(C)$, is the size of the largest collection of points that is shattered by C

VC dimension and PAC learning

- C: concept class, let $d = VC(C)$
- $S = (x_1, \ldots, x_n)$: a collection of n distinct points
- Recall: $C(S) = \{c(S): c \in C\}$
- By definition, if $n \leq d$ then $|C(S)| \leq 2^d$ (with equality for at least one set S)
- Theorem: if $n > d$ then $|C(S)| \leq \left(\frac{en}{d}\right)^d$
- PAC Learner for classes with finite VC dimension:
  - Input: $\mathrm{n} \geq O\left(\frac{VC(C)\log\frac{1}{\alpha}+\log\frac{1}{\beta}}{\alpha}\right)$ labeled samples
  - Output: a consistent hypothesis from $H = C$
- An almost matching lowerbound:
  - $\Omega\left(\frac{VC(C)+\log\frac{1}{\beta}}{\alpha}\right)$ labeled samples required

Example 2: Learning parity functions

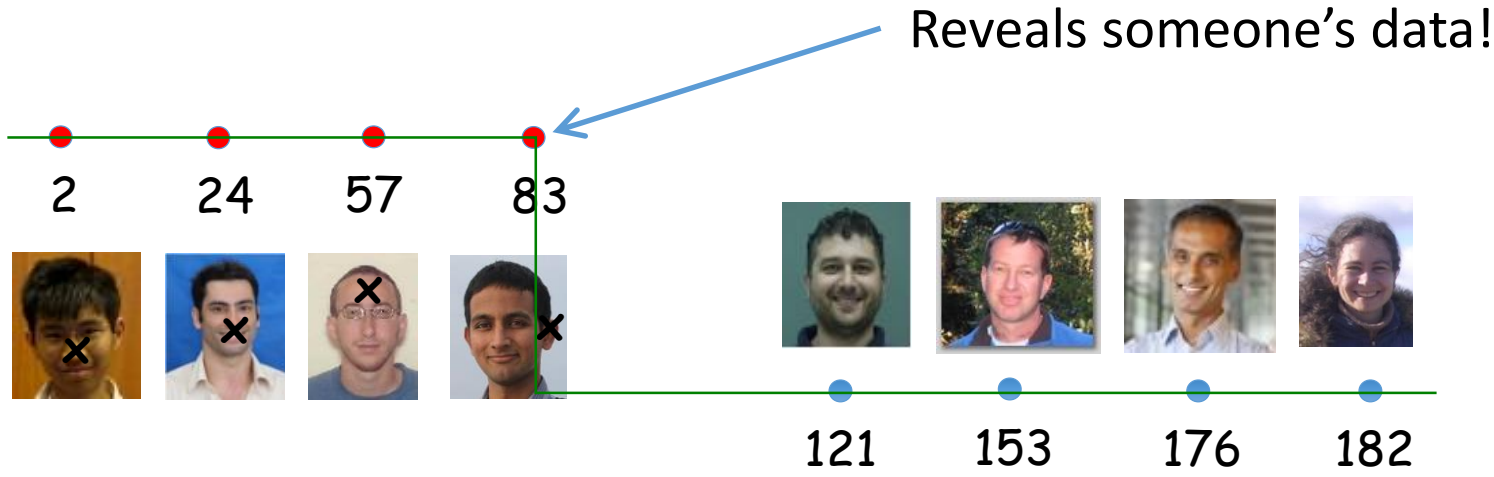
- **Concept Class:** $\mathcal{C} = \mathbf{PARITY_d} = \{c_r\}_{r \in \{0,1\}^d}$
- $c_r(x) =< r, x > mod\ 2$
- $VC(C) = d$

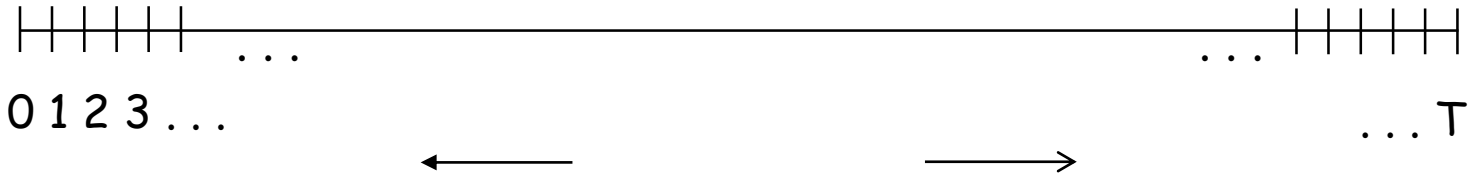
Let $S = (e_1, \ldots, e_d)$
For any $r = \{0,1\}^d, c_r(S) = r$
Hence $C(S) = \{0,1\}^d$

- **An efficient PAC learner for $PARITY_d$ with $O(d)$ samples:**
  - Each example (x_i,$c_r(y_i)$) makes a linear constraint
    - E.g., sample (1101, 1) translates to $r_1 + r_2 + r_4 \pmod 2 = 1$
  - Find a consistent r' by solving the set of linear equations over GF(2) imposed by input x

PAC learning – where's the privacy problem?

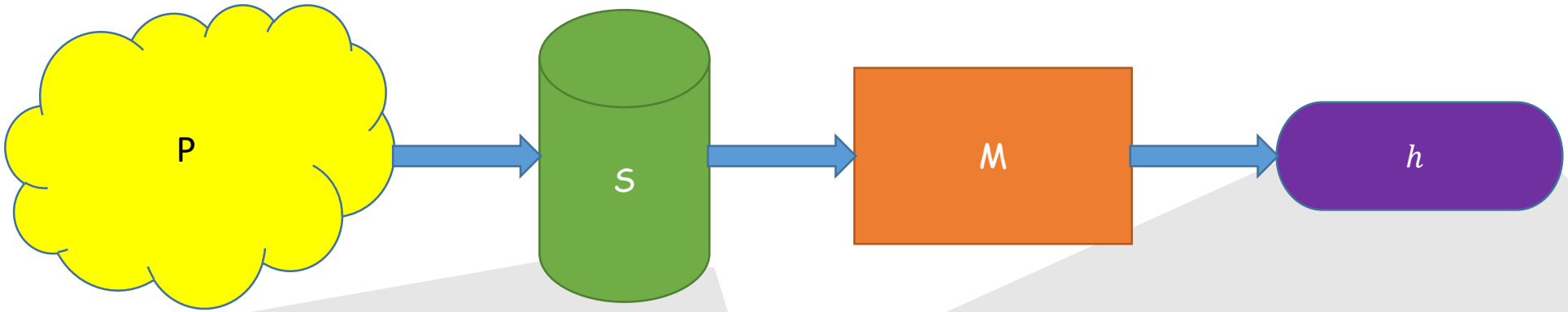
- Learner: returns a consistent threshold function
  - E.g., transition on largest point labeled "1"
  - (differential) privacy not preserved





Thanks: Mark Bun

What is *private* learning?



% down	High Debt	Other accts	Mmp /inc	Good Risk?
10	No	Yes	0.32	**Yes**
10	No	No	0.25	**Yes**
15	No	Yes	0.32	Yes
20	No	Yes	0.31	**Yes**
10	No	No	0.25	**Yes**

Predict YES if 100*(Mmp/inc) – (% down) < 25

- Private Learner:
  - Satisfies standard definition of PAC learning
  - Is differentially private

Average-case guarantee

Worst-case guarantee

Private PAC (PPAC)

- Definition: L is a $(\alpha, \beta, \epsilon, \delta)$- PPAC learner for concept class C if
  - Utility: L is a (α, β)- PAC learner for concept class C:
    - For all distributions P and target concepts $c \in C$, L outputs a hypothesis h such that

$$\Pr[error_P(c, h) < \alpha] \geq 1 - \beta$$

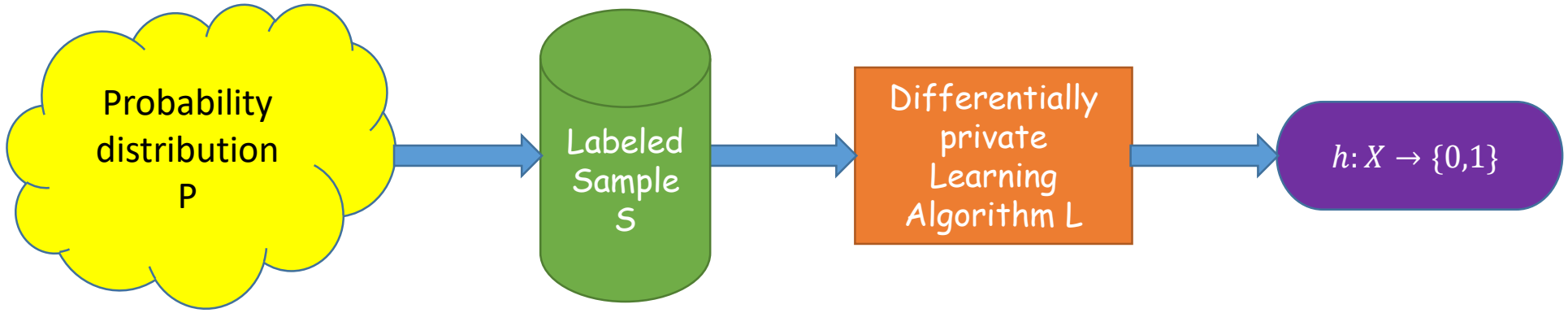
Probability over sample and algorithm

  - Privacy: L is a (ϵ, δ)-differentially private:
    - for all neighboring sample sets S, S' and for all sets of hypothesis T

$$\Pr[L(S) \in T] \leq e^{\epsilon} \cdot \Pr[L(S') \in T] + \delta$$

Probability over ~~sample and~~ algorithm

Private learning over sample vs. over distribution



- Suppose that given $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ the differentially private learner outputs a hypothesis h such that

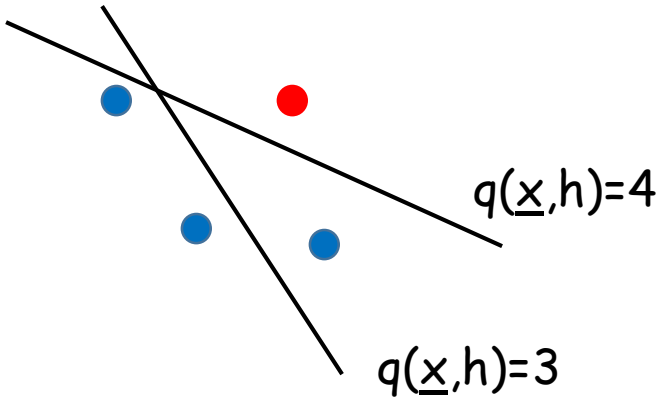
$$error_S(c, h) = \Pr_{i \in_R [n[}[h(x_i) \neq y_i] \leq \alpha$$

  - Is $error_P(c, h)$ small?
  - Yes!
  - Generalization of differential privacy implies that $error_P(c, h) \leq \alpha + \epsilon$ if

$$n \geq O\left(\frac{\ln\frac{1}{\delta}}{\epsilon^2}\right)$$

PPAC learning of finite concept classes [KLNRS 08]

- Theorem: every finite concept class can be learned privately, using a polynomial number of samples
- Let C be a finite concept class
- $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$: a sample of n i.i.d. examples sampled from P and labeled according to c, i.e., $y_i = c(x_i)$
- Learner: (based on the Exponential Mechanism [MT07])
  - Define $q(S, h) = |\{i : h(x_i) = y_i\}|$
  - Output hypothesis $h \in C$ w.p. proportional to $e^{\frac{\epsilon}{2} q(S,h)}$
- Using properties of the exponential mechanism:
  - Learner is $(\epsilon, 0)$-differentially private
  - Proper PAC learner if $n \geq O\left(\left(\log|C| + \log\frac{1}{\beta}\right) \cdot \max\left(\frac{1}{\epsilon^2}, \frac{1}{\alpha\epsilon}\right)\right)$
- Can be extended to agnostic learning
- Running time may be exponential



Some PPAC learners

- $\mathbf{POINT_d} = \{c_1, \ldots, c_{2^d}\}; c_j(x) = 1 \iff x = j$
  - Generic PPAC construction:
    - Polynomial time
    - Requires $O(d)$ samples
  - But PAC learner with O(1) samples
- $\mathbf{THRESHOLD_d} = \{c_0, \ldots, c_{2^d}\}; c_j(x) = 1 \iff x < j$
  - Generic PPAC construction:
    - Polynomial time
    - Requires $O(d)$ samples
  - But PAC learner with O(1) samples
- $\mathcal{C} = \mathbf{PARITY_d} = \{c_r\}_{r \in \{0,1\}^d}$; $c_r(x) = < r, x > mod\ 2$
  - Efficient construction [KLNRS08]:
    - Polynomial time
    - Requires $O(d)$ samples
  - PAC learner also requires O(d) samples

Efficient PPAC learner for Parity

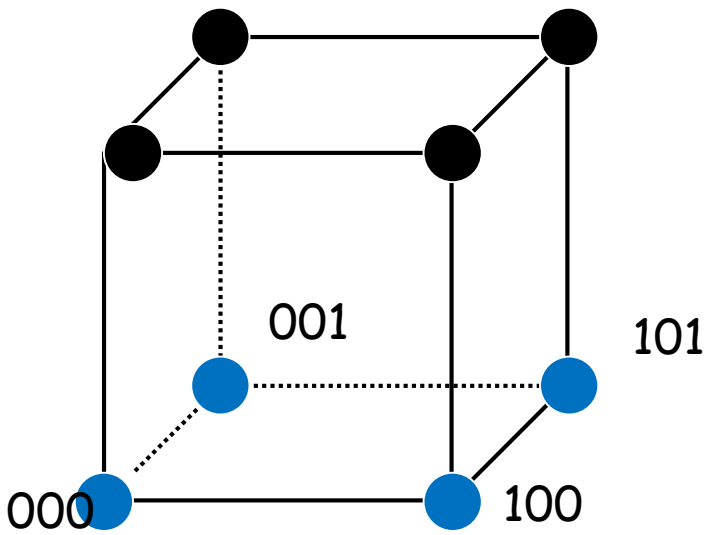
- Parity: $c_r(x) = \langle r, x \rangle \pmod 2$
- Input: $\underline{x}=((y_1,c_r(y_1)),\ldots..,(y_n,c_r(y_n)))$
- Recall Non-private learning algorithm:
  - Solving the set of linear equations over GF(2) imposed by input x to recover a consistent r'
  - Is this privacy preserving?
- The Effect of a Single Example:
- Let S_i be space of feasible solutions for the set of equations imposed by $\underline{x}_i$
  - Add a fresh example $(y_{i+1},c_r(y_{i+1}))$ to $\underline{x}_i$ and let S_{i+1} be the new solution space
  - Then,
    1. $|S_{i+1}| \geq |S_i|/2$, or
    2. $|S_{i+1}| = 0$

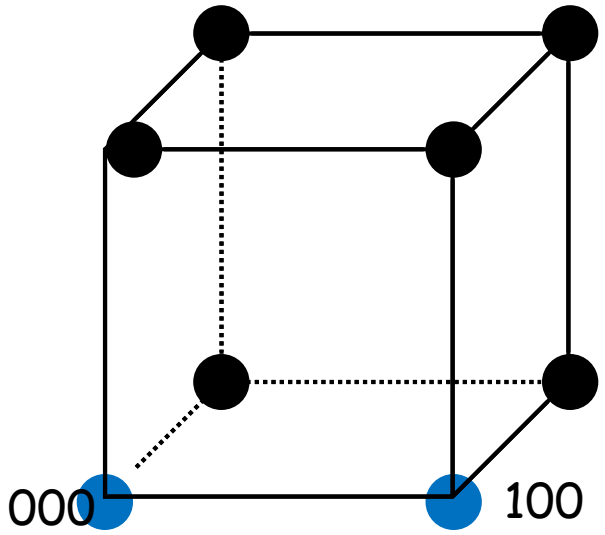
Size of solution space reduces by 1/2

System becomes inconsistent

The Effect of a Single Example

● = Valid solution

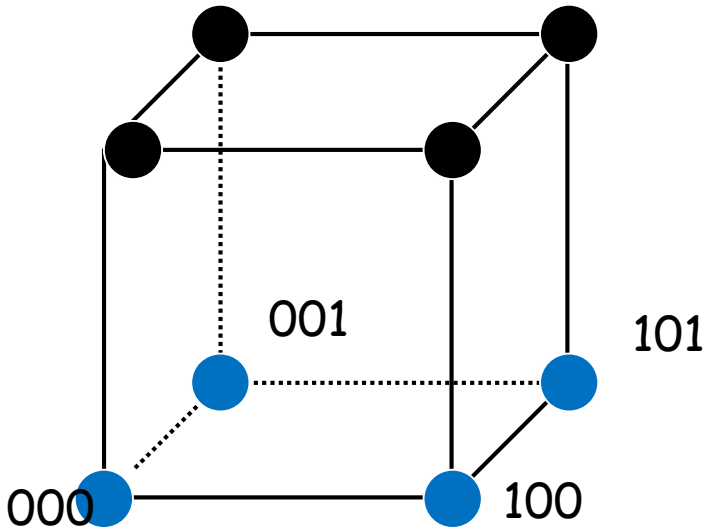


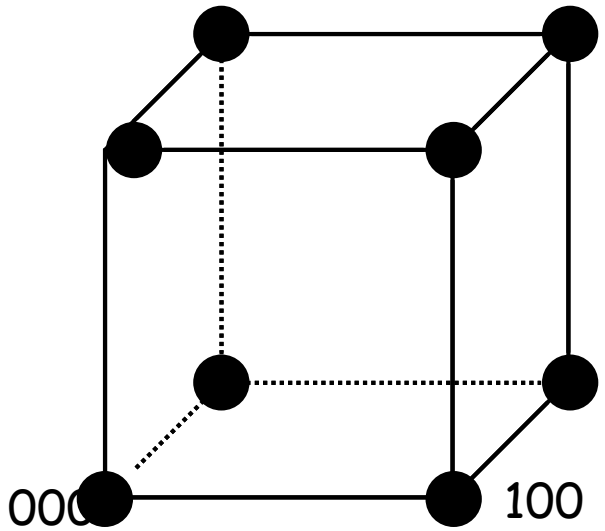


new constraint: third coordinate is 0

The Effect of a Single Example

● = Valid solution





new constraint: second coordinate is 1

Solution space changes drastically only when algorithm fails

PRIVATE LEARNER FOR PARITY

1. With probability ½ output "fail"

Smoothes out extreme jumps in S

2. $\underline{x}_S \leftarrow$ pick each example from $\underline{x}$ with probability $\varepsilon/4$

3. Use Gaussian elimination to solve the system of equations imposed by examples in $\underline{x}_S$.
   - Let S be the set of feasible solutions

4. If S = Ø, output "fail".
   Otherwise, output a random vector in S

Private Learner for Parity

- ε-differential privacy preserved:
  - E.g. $\underline{x}$, $\underline{x}'$ neighboring:
    - $\underline{x}$ consistent with some solutions S.
    - $\underline{x}'$ inconsistent.
    - Pr[Fail] changes from ½ to ½+ $\varepsilon/4$
- Learning:
  - Confidence can be amplifies by repeating log $1/\beta$ runs (decreasing ε accordingly)
  - Accuracy α, confidence β privacy ε:

$$n=O((d \log 1/\beta + \log^2 1/\beta)/\alpha\varepsilon) \text{ examples suffice}$$

dimension

PAC learning vs. PPAC learning

- PAC learning:
  - Occam's razor : sample complexity $\sim \log |C|$
  - Generally: sample complexity $\sim VC(C)$
    - $VC(C) \leq \log |C|$
    - Finite for some infinite concept classes
- PPAC learning:
  - 'Private Occam's razor': sample complexity $\sim \log |C|$
    - Can we close gap with non-private learning?
  - What about infinite concept classes, is there an analog of the VC dimension? Does PAC learnability imply PPAC learnability?
  - Efficient PPAC parity learner
    - Is every efficiently PAC learnable concept class also efficiently PPAC learnable?

References

- Michael J. Kearns and Umesh Vazirani: **An Introduction to Computational Learning Theory.** MIT Press, 1994
- Shai Shalev-Shwartz and Shai Ben-David: **Understanding Machine Learning – From Theory to Algorithms**, Cambridge University Press, 2014
- Avrim Blum, Cynthia Dwork, Frank McSherry, Kobbi Nissim: **Practical privacy: the SuLQ framework.** PODS 2005
- Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, Adam D. Smith: **What Can We Learn Privately?** FOCS 2008
- Cynthia Dwork, Guy N. Rothblum, Salil P. Vadhan: **Boosting and Differential Privacy.** FOCS 2010