Statistical Algorithms and the Planted Clique Problem

Lev Reyzin
ARC @ Georgia Tech → Math @ UIC

with Vitaly Feldman, Elena Grigorescu, Santosh Vempala, Ying Xiao

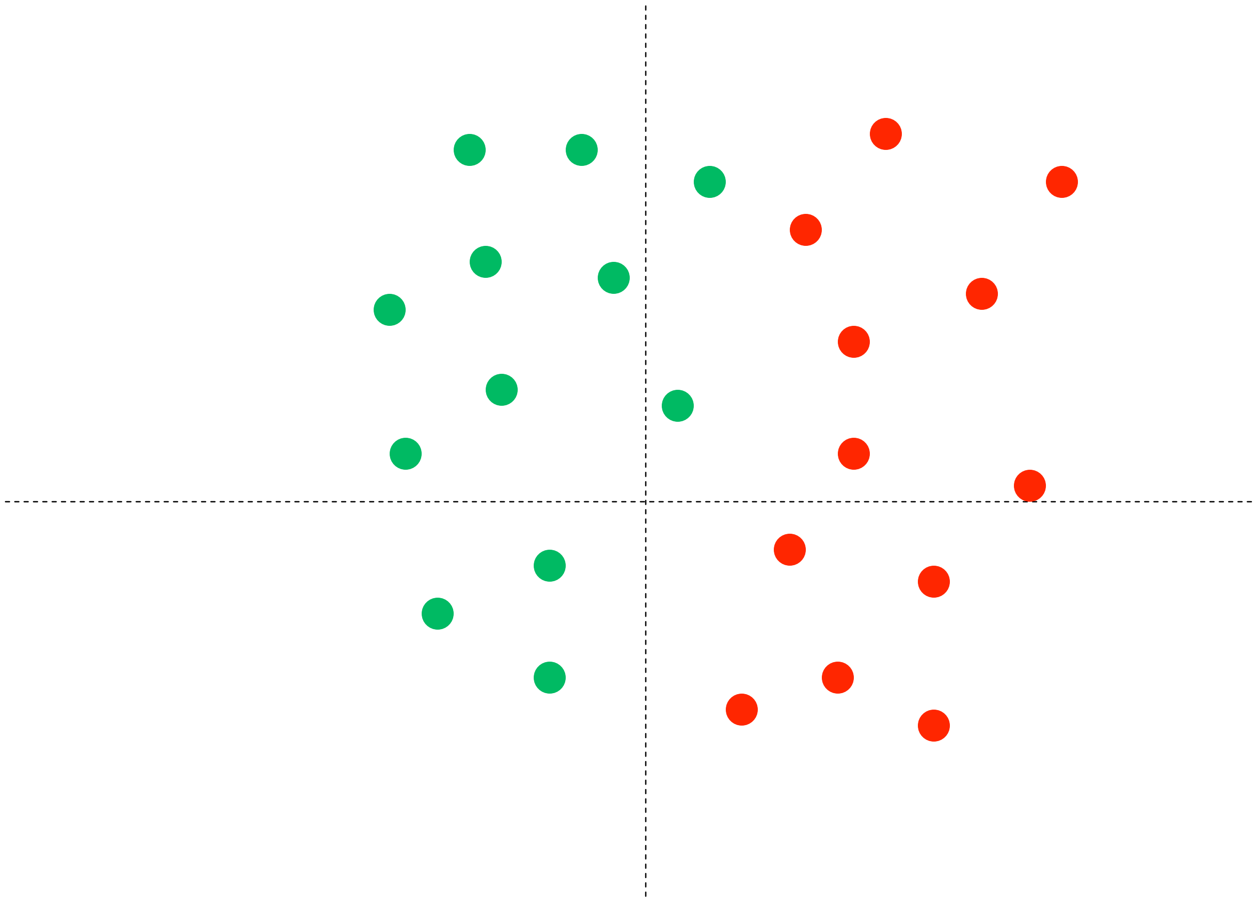
Outline

1. PAC Learning and the noisy parity problem.
2. Statistical algorithms and a general theorem.
3. A lower bound for planted clique
4. New hardness results for other problems

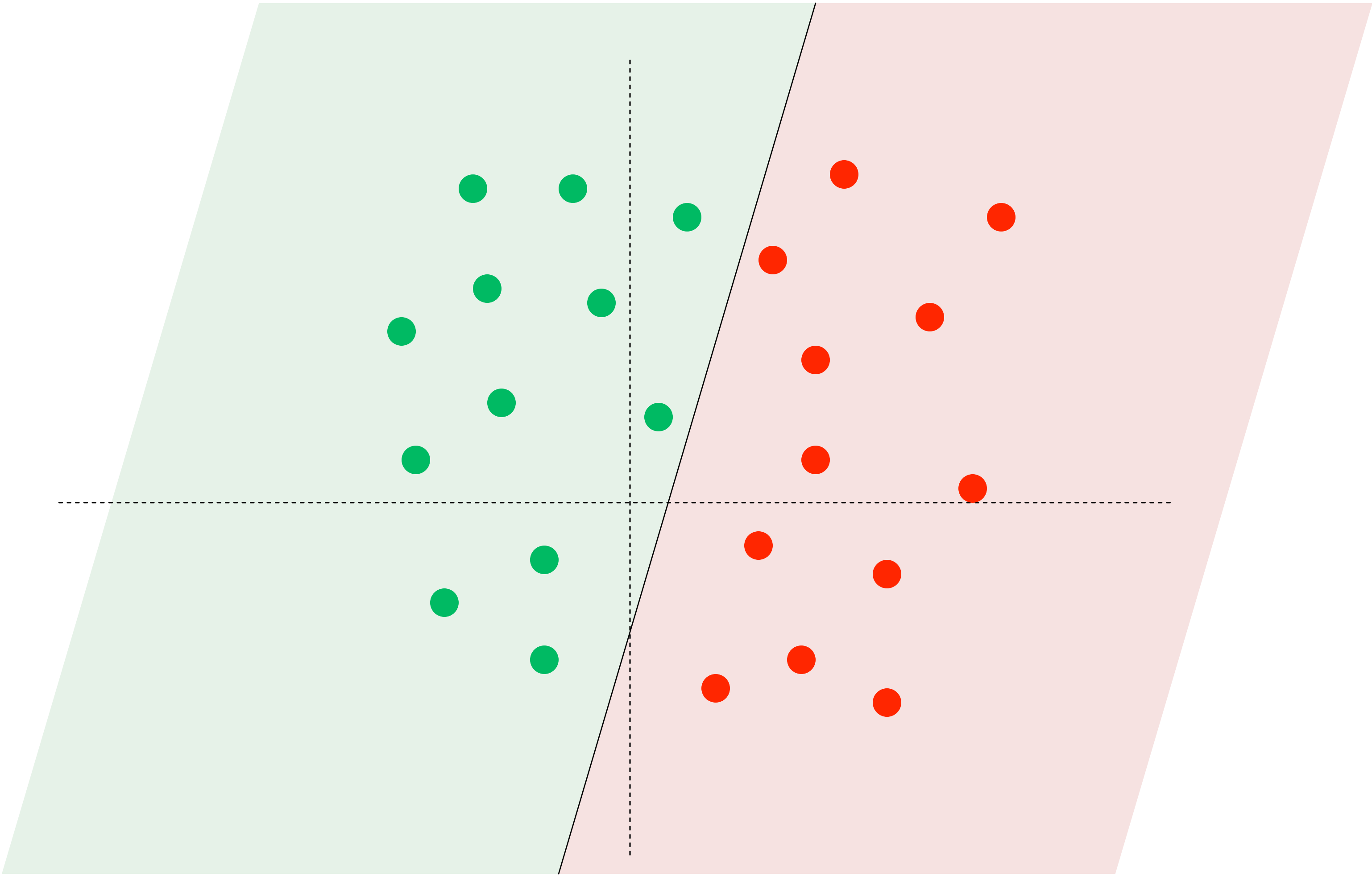
A Brief Introduction to Learning

some context

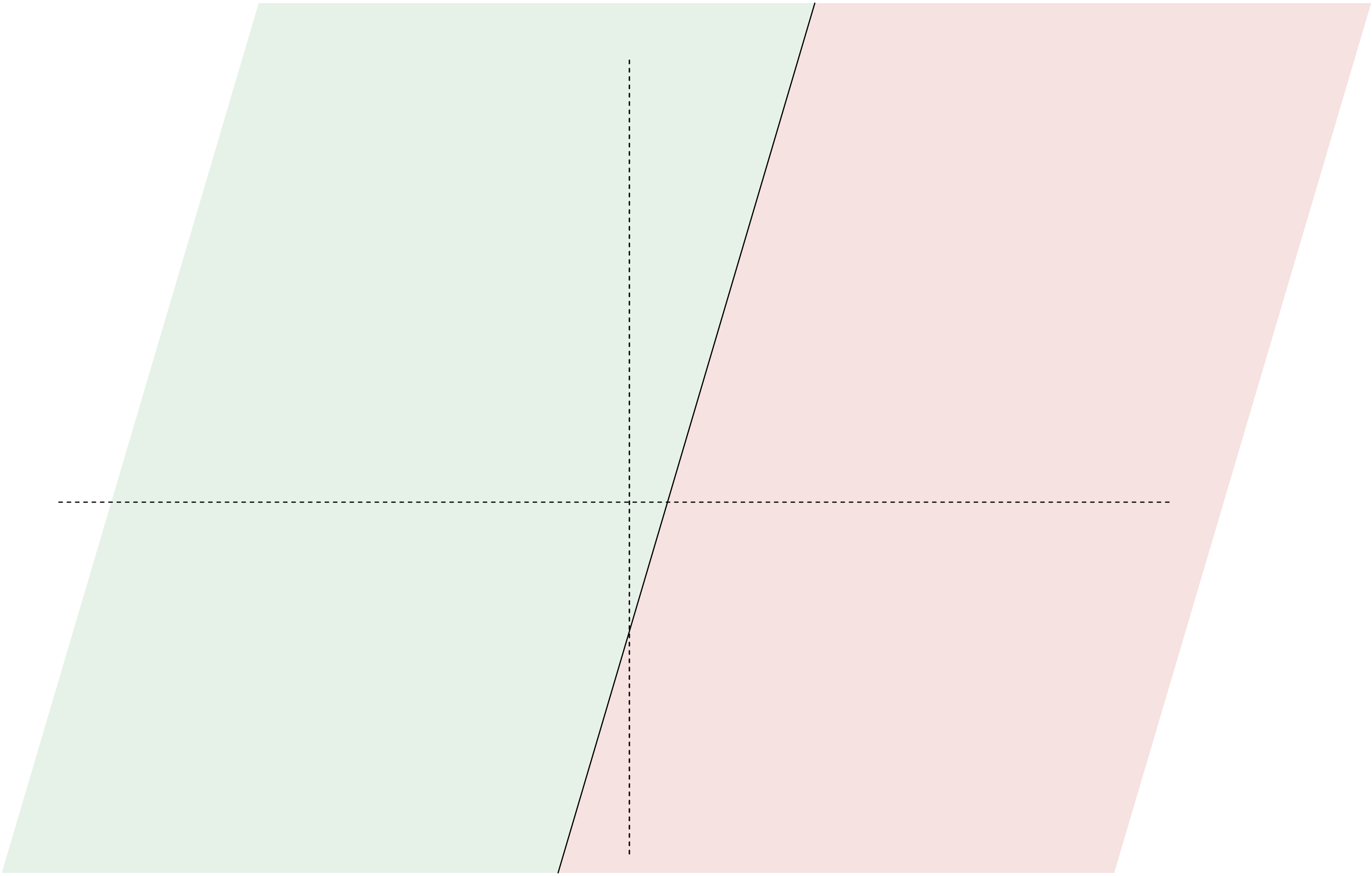
Learning Half-Planes



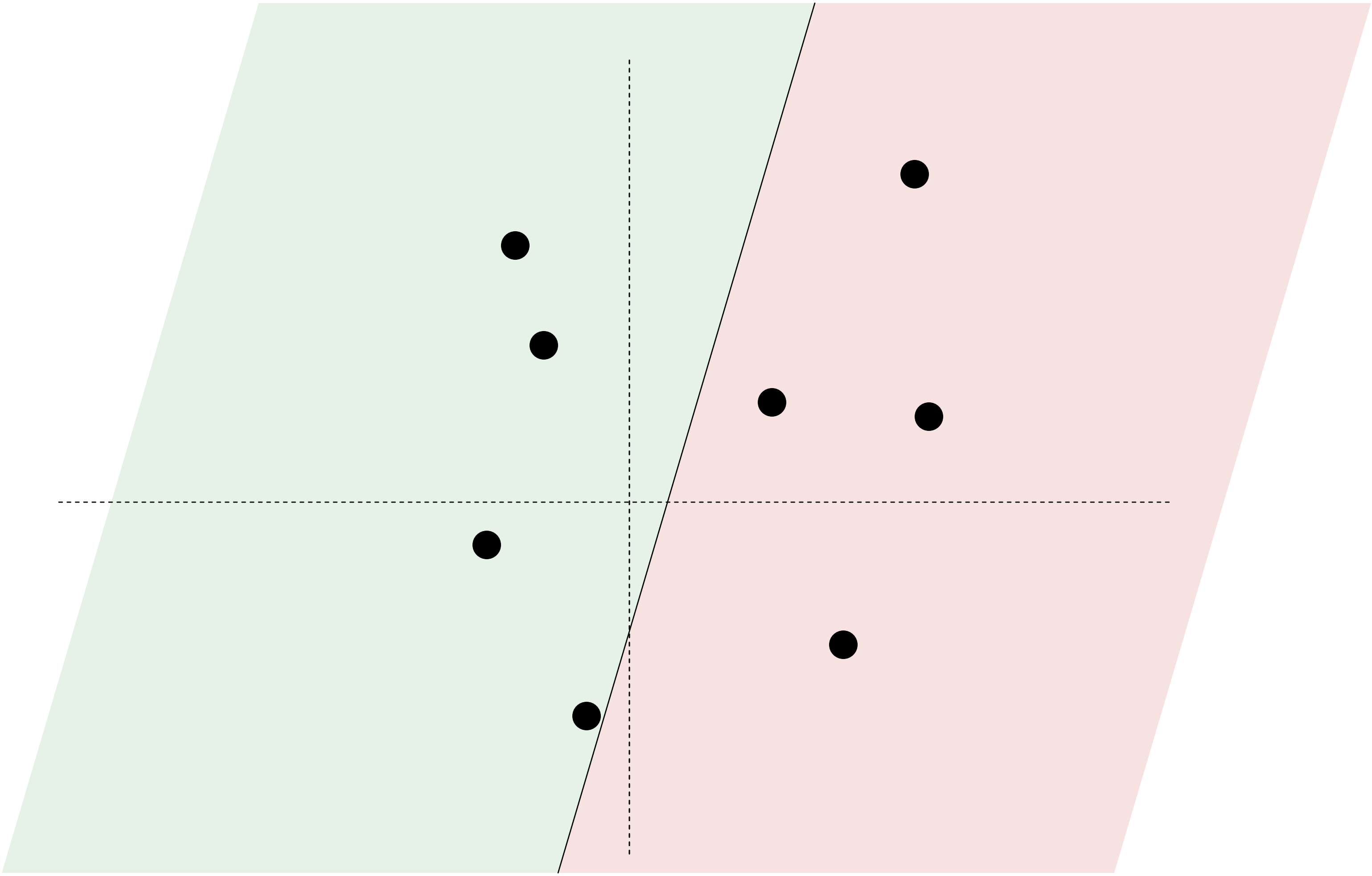
Learning Half-Planes



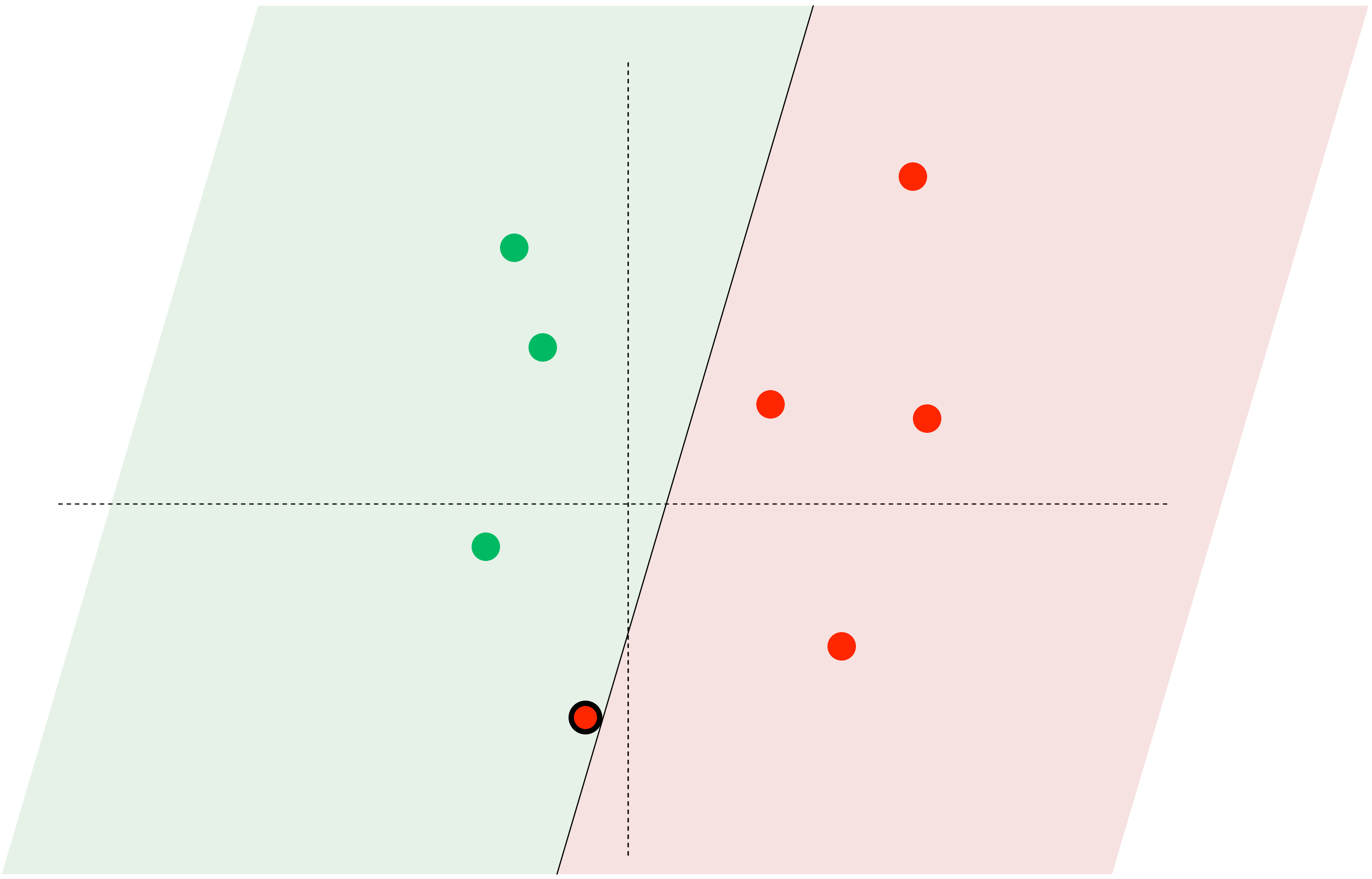
Learning Half-Planes



Learning Half-Planes



Learning Half-Planes



PAC Learning [Valiant '84]

- X is domain.
- D a probability distribution over X.
- Let c: $X \rightarrow \{-1,1\}$ be a target "concept" and C be the set of possible targets c.

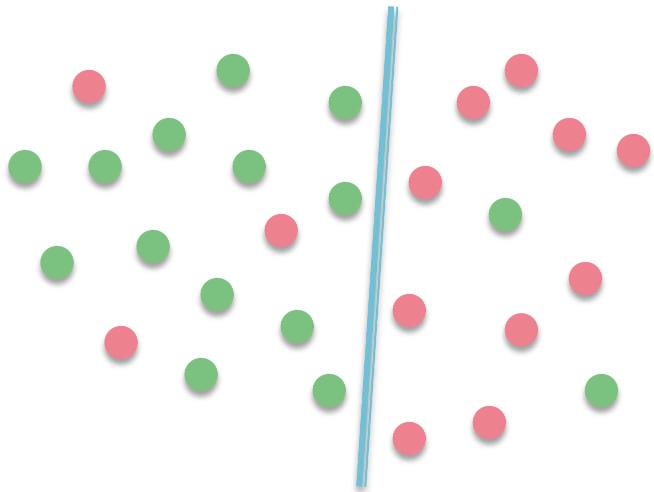
- Class C is learnable if $\forall$ c $\in$ C, D, $\varepsilon>0$, $\delta>0$, a learner can receive a set S of m "labeled examples" from D: $\{(x_1,c(x_1)), \ldots, (x_m,c(x_m))\}$ and produce a hypothesis $h_S: X \rightarrow \{-1,1\}$ such that:

$$\Pr_{S\sim D}[\Pr_{x\sim D}[h_s(x) \neq c(x)] > \varepsilon] < \delta.$$

(ideally want m to be "small")

An Overview of PAC Learning

- In trying to understand which PAC algorithms can handle noise, "statistical queries" (SQ) were invented [Kearns '93].
  - SQ algorithms are noise-tolerant.
  - Turns out that most learning algs fall into this category.

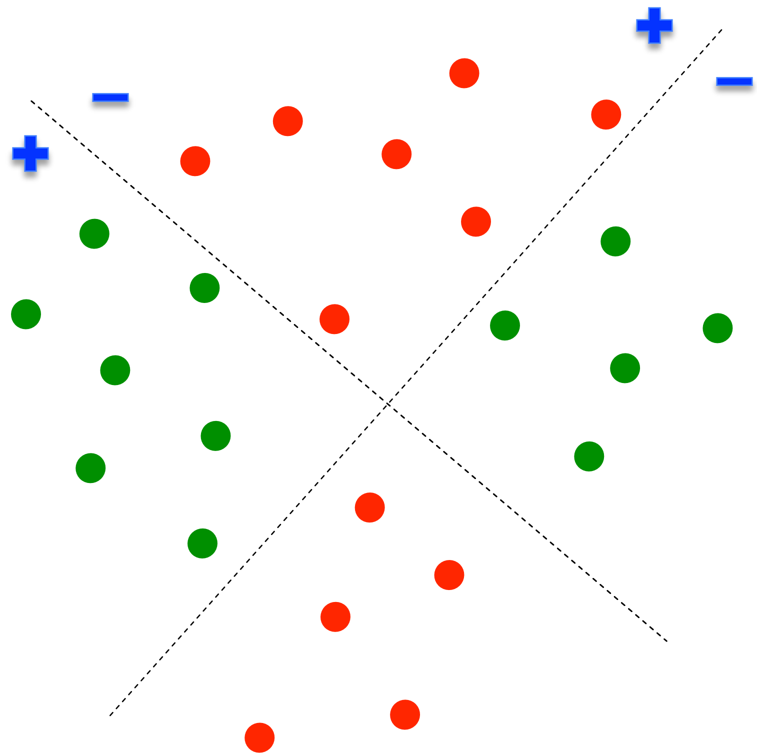


An Overview of PAC Learning

- In trying to understand which PAC algorithms can handle noise, "statistical queries" (SQ) were invented [Kearns '93].
  - SQ algorithms are noise-tolerant.
  - Turns out that most learning algs fall into this category.
- Unfortunately, it is also known that SQ algorithms have serious limitations.
  - Notably, SQ algorithms cannot learn parities, among other classes of functions [Blum et al '93].

SQ Dimension

- A class is hard to learn with SQ algorithms if it has high SQ dimension – the maximum number of "orthogonal" hypotheses. [Blum et al '93]



PAC Learning Parities

- [***Def.***] For $x \in \{0,1\}^n$ and $c \in \{0,1\}^n$, let $\chi_c(x)$ take the value 1 if $c \cdot x$ is odd and -1 otherwise.
  - If c has 1's only in r positions, we call c an r-parity.
- For an unknown target c, the learner sees labeled examples $(x, \chi_c(x))$ from some distribution, e.g.
  (00110101,1), (10011010,1), (00101111,-1), …
- Learner needs to determine c (or more generally predict labels of future examples).
- Learning parities turns out to be hard for SQ algorithms *even over the uniform distribution on* $\{0,1\}^n$.

Parities

- Therefore, we can prove lower bounds on SQ learning by showing certain classes encode parities.

- There has been little progress on noisy parity:
  - Best progress: $O(2^{n/\lg n})$ [Blum Kalai Wasserman '00]

- **Noisy parity is widely believed to be hard, and the SQ lower bound is the concrete reason.**

- **Our goal is to extend these ideas outside learning.**

Search and Optimization

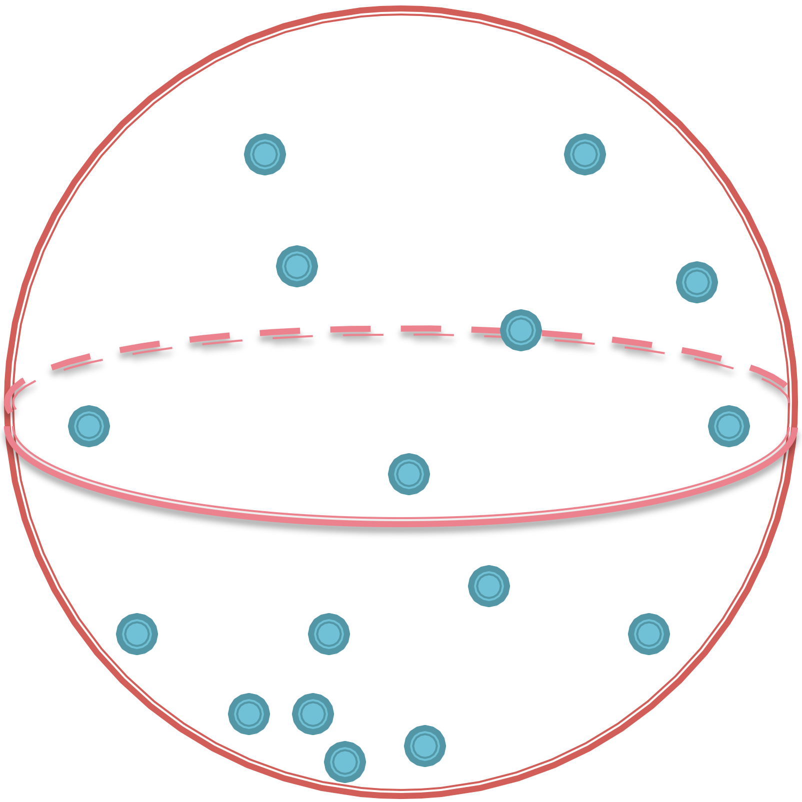
Motivating Example

problem: moment maximization

Let D be a distribution over points in $[-1,1]^n$ and let $r \in \mathcal{Z}^+$. The goal is to find a unit vector u* that **approximately maximizes the expected r'th moment** of the projection to u of a random point x chosen from D.

i.e. find

$$u^* \approx \arg\max_{u \in \mathfrak{R}^n:\, \|u\|=1} \mathop{\mathrm{E}}_{x \sim D}[(u \cdot x)^r].$$



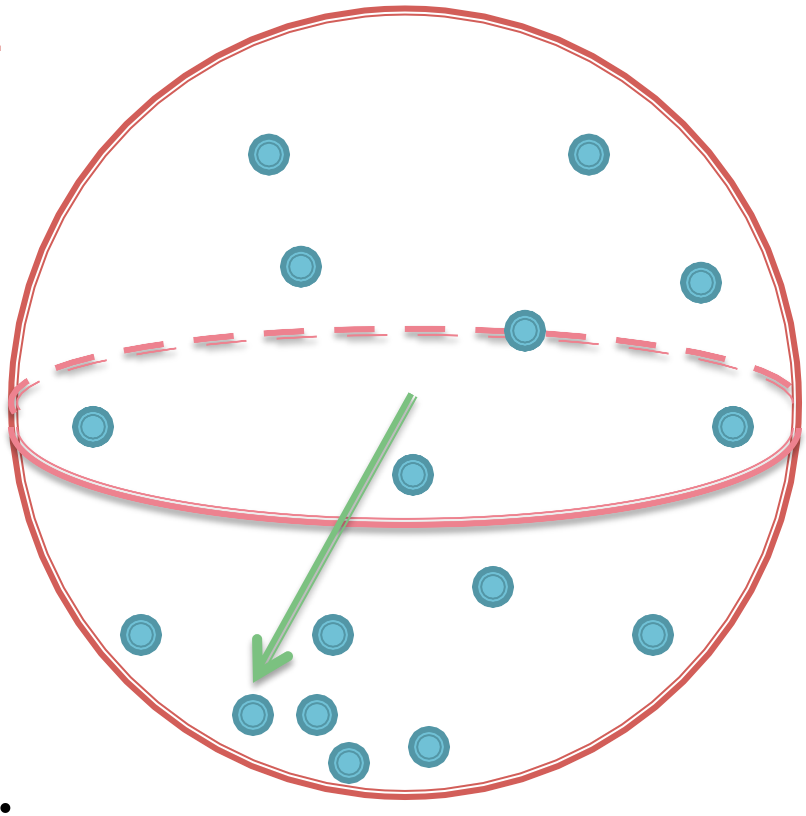
Motivating Example

problem: moment maximization

Let D be a distribution over points in $[-1,1]^n$ and let $r \in \mathcal{Z}^+$. The goal is to find a unit vector u* that **approximately maximizes the expected r'th moment** of the projection to u of a random point x chosen from D.

i.e. find

$$u^* \approx \arg\max_{u \in \Re^n:\ \|u\|=1} \mathop{\mathrm{E}}_{x \sim D}[(u \cdot x)^r].$$



Possible Approaches for Large r

- **Idea 1 (Gradient descent):** Start with some unit vector u. Estimate the gradient (via samples), and move in that direction. Repeat until local maximum is found.
  - Many local maxima. Can we avoid this by taking new samples with each estimate?

- **Idea 2** [Kannan] **(Markov chains)**: Consider a Markov chain that attempts to sample u with density proportional to $e^{E[(x \cdot u)^r]}$. Implement via Metropolis filter. At each step we only need to estimate $E[(x \cdot u)^r]$.
  - Does this Markov chain mix rapidly?

Possible Approaches for Large r

- **Idea 1 (Gradient descent):** Start with some unit vector u. Estimate the gradient (via samples), and move in that direction. Repeat until local maximum is found.
  - Many local maxima. Can we avoid this by taking new samples with each estimate? Our work shows that **NO!**

- **Idea 2** [Kannan] **(Markov chains)**: Consider a Markov chain that attempts to sample u with density proportional to $e^{E[(x\cdot u)^r]}$. Implement via Metropolis filter. At each step we only need to estimate $E[(x\cdot u)^r]$.
  - Does this Markov chain mix rapidly? Our work shows that **NO!**

Possible Approaches for Large r

- **Idea 1 (Gradient descent):** Start with some unit vector u. Estimate the gradient (via samples), and move in that direction. Repeat until local maximum is found.
  - Many local maxima. Can we avoid this by taking new samples with each estimate? Our work shows that **NO!**

- **Idea 2** [Kannan] **(Markov chains)**: Consider a Markov chain that attempts to sample u with density proportional to $e^{E[(x \cdot u)^r]}$. Implement via Metropolis filter. At each step we only need to estimate $E[(x \cdot u)^r]$.
  - Does this Markov chain mix rapidly? Our work shows that **NO!**

(Disclaimer: moment maximization is NP-hard for r > 3 [Brubaker '09])

Statistical Algorithms

- Both approaches fall under a class of **statistical algorithms**.
- In this talk, we will show that for many optimization problems over distributions, statistical algorithms **unconditionally** have complexity **exponential** in their input parameters.
- Our lower bounds use only a single parameter of the optimization problem we call **statistical dimension**.
  - Inspired by the statistical query model in learning theory.

General Optimization

- **Optimization problems over distributions.** Let $\mathcal{D}$ be the set of input distributions over a domain X and $\mathcal{F}$ be a set of functions $X \rightarrow \mathcal{R}$ over which we want to optimize. An optimization problem **P($\mathcal{F}$,$\mathcal{D}$)** over an input distribution $D \in \mathcal{D}$ has a solution function $f^* \in \mathcal{F}$ such that $f^* = \text{argmax}_{f \in \mathcal{F}} E_{x\sim D}[f(x)]$.

- For a function $g \in \mathcal{F}$, distribution D, and $\varepsilon > 0$, we say that g is **ε-optimal** for D if

$$E_{x\sim D}[g(x)] \geq E_{x\sim D}[f^*(x)] - \varepsilon.$$

The objective is to ε-optimize over $\mathcal{F}$ w.r.t. D, i.e. to find an ε-optimal $g \in \mathcal{F}$.

Statistical Algorithms and Statistical Dimension

the definitions

Statistical Algorithms Defined

We say an algorithm is **statistical** if it interacts with the target distribution via an oracle, **$SAMPLE_D$**, which takes as inputs a **query function** $h \in H: X \rightarrow \{-1,1\}$ and a **sample size** $t > 0$. $SAMPLE_D(h,t)$ draws $x_1 \ldots x_t$ independently from D and returns

$$\frac{1}{t}\sum_{i=0}^{t} h(x_i)$$

The **sample complexity** of an algorithm is the sum of sample sizes sent to the oracle over the run of the algorithm.

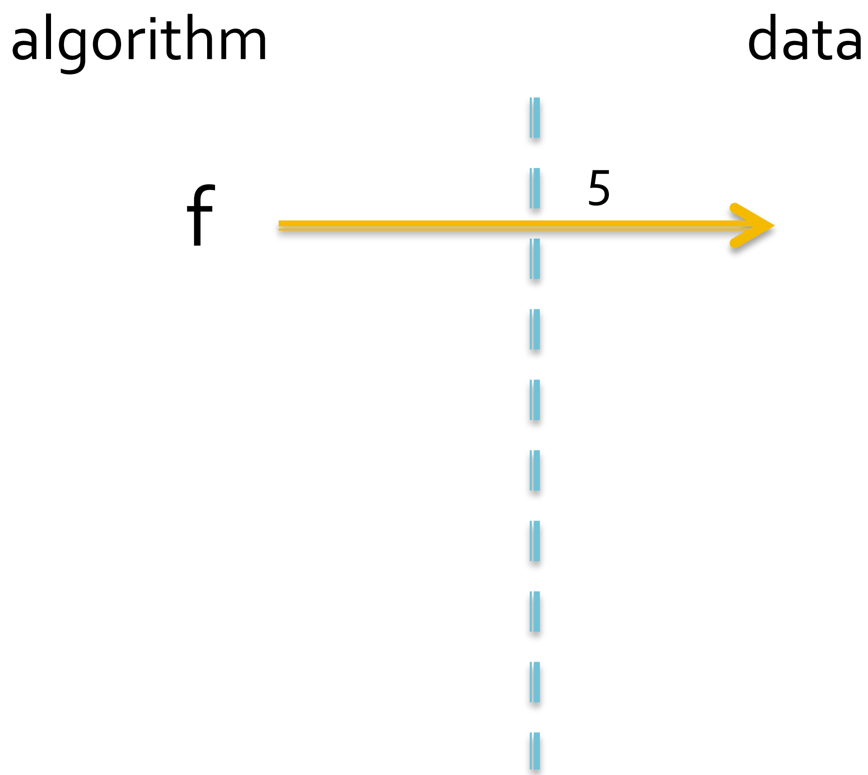
*The closest model in learning is "honest statistical queries" [Ke Yang '02]

Statistical Algorithms

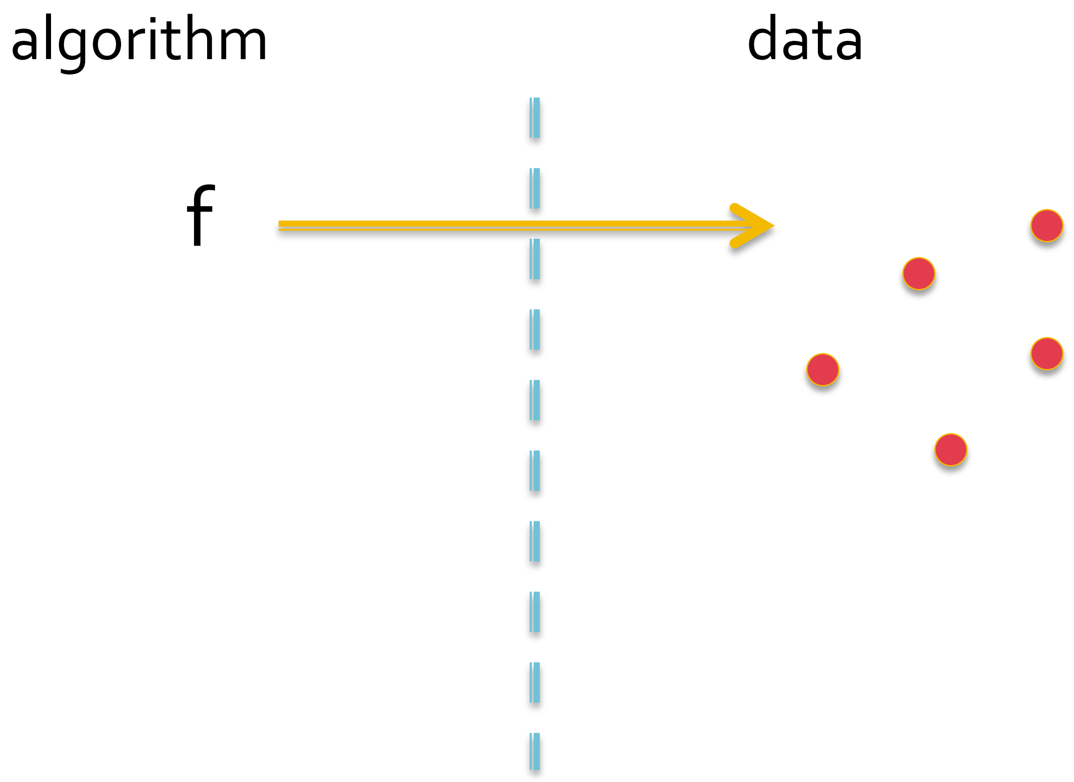
algorithm

data

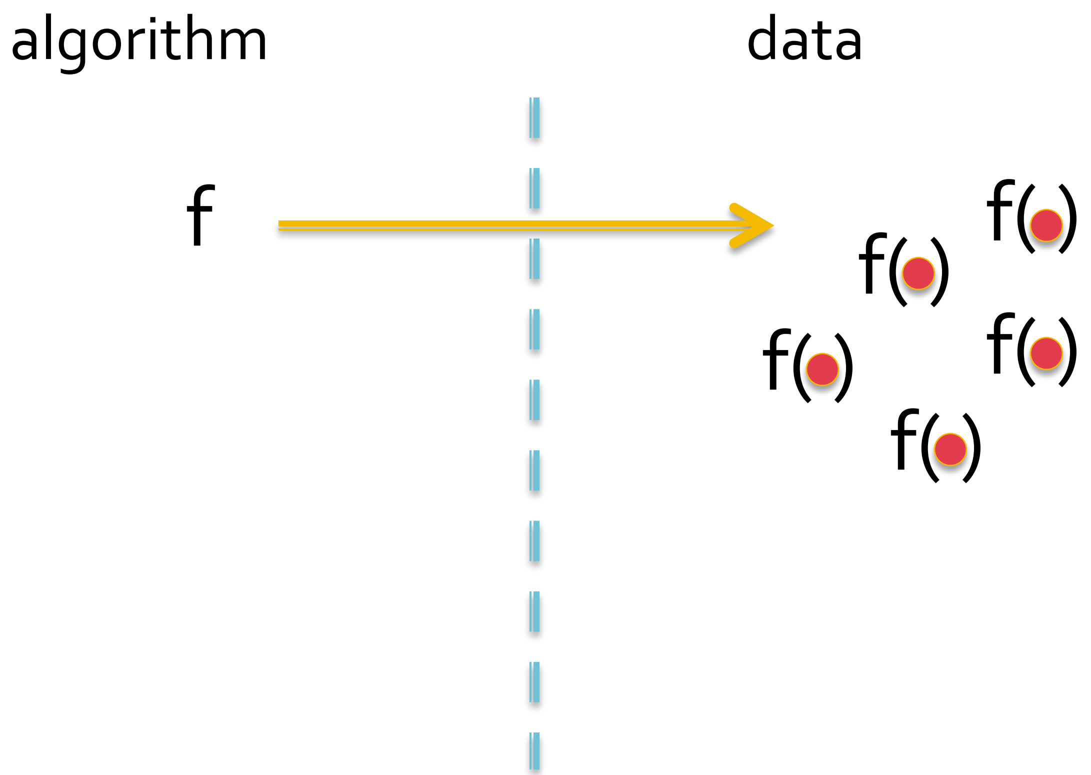
Statistical Algorithms



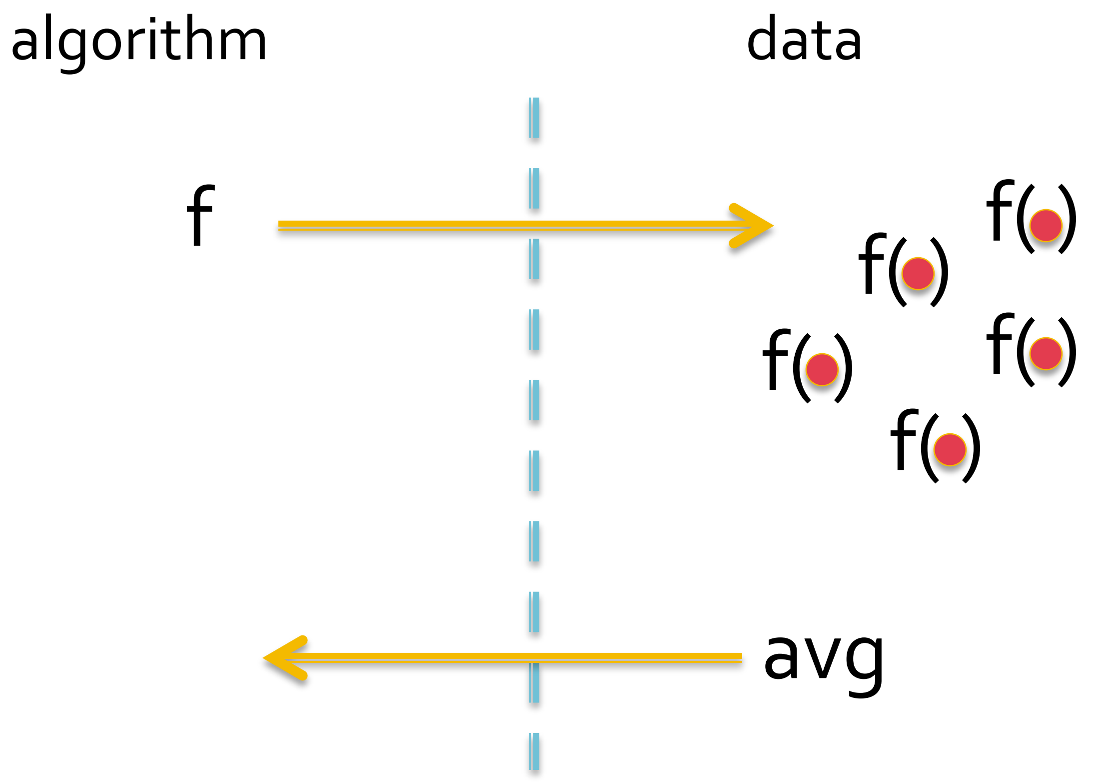
Statistical Algorithms



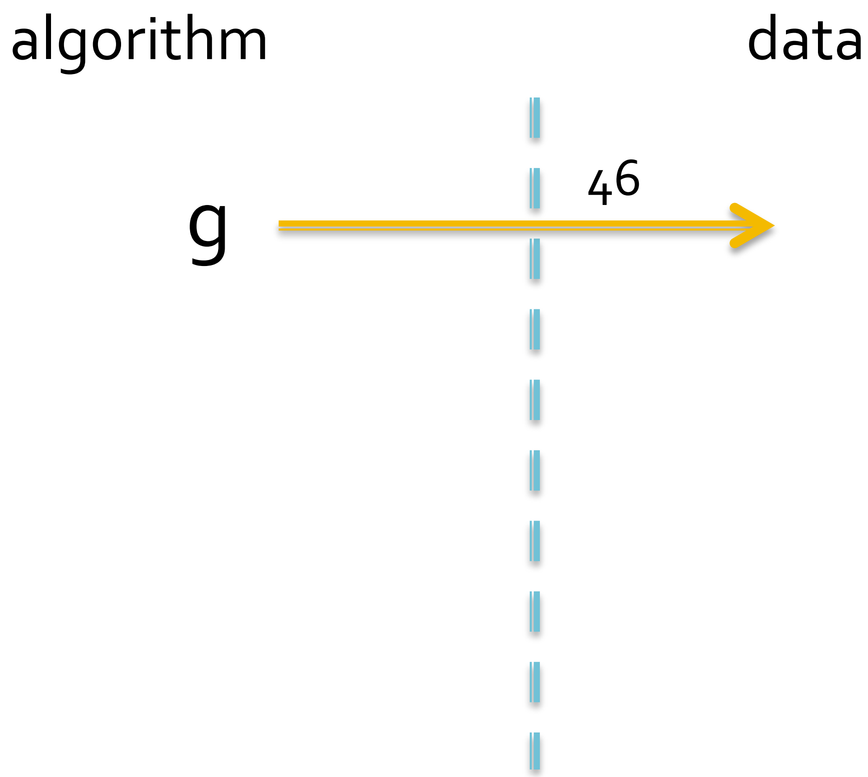
Statistical Algorithms



Statistical Algorithms



Statistical Algorithms



Examples of Statistical Algorithms

- What optimization algorithms can be implemented via statistical estimates?
  - local search
  - k-means
  - simulated annealing
  - EM
  - MCMC
  - gradient descent
  - convex optimization
  - almost anything practical has a statistical variant…

A Note on Optimization vs Learning

- For **optimization** we just introduced a definition for *statistical* algorithms. It is inspired by the concept of *statistical query* algorithms [Kearns 1993] from **learning** theory.
- In **learning**, examples come from some distribution and are labeled by an unknown concept. Therefore, there can exist hard distributions. In **optimization**, for any fixed input distribution, there is a fixed answer. Hence, we can't have a "hard distribution."
- In **learning**, for many classes, the uniform distribution a hard distribution. In **optimization**, the uniform distribution is usually trivial (consider moment maximization).
- In **learning**, it is sometimes reasonable to wish to learn the target *exactly*. In **optimization**, usually we're interested in *approximating* the optimum (in our case additive).

Statistical Dimension

- **Learning** a class with statistical queries is hard if there is a distribution under which the class contains many (nearly) **pairwise uncorrelated functions** [Blum et al. '94].

- For **optimization/search**, we will want something similar, but for distributions instead of labeling functions.
  - We will want there to be **many possible "uncorrelated" input distributions**, such that eliminating one distribution as the real input will not help in eliminating others.
  - Our notion will strictly generalize the notion of "SQ dimension" in learning.

Statistical Dimension

- For γ, $\beta > 0$, domain X, class of functions $\mathcal{F}$, and a class of distributions $\mathcal{D}$ over X, let m be the maximum s.t. there exists **a reference distribution D** over X s.t. for every $f \in \mathcal{F}$ there exists a **set of m distributions D_f** $= \{D_1 \ldots D_m\} \in \mathcal{D}$ satisfying:
  1. f is not valid/ε-optimal for any D_i for $i \in \{1\ldots m\}$
  2. $$\left\langle \frac{D_i}{D} - 1, \frac{D_j}{D} - 1 \right\rangle_D \leq \begin{cases} \beta \text{ for } i = j \in [m] \\ \gamma \text{ for } i \neq j \in [m] \end{cases}$$

- We define the **statistical dimension** of ε-optimizing (or searching) over F, denoted **SD($\mathcal{F}$,$\mathcal{D}$,γ,β)**, to be **m**.

Main Theorem

Theorem: If for a class of functions $\mathcal{F}$, class of distributions $\mathcal{D}$, and γ, $\beta > 0$, SD($\mathcal{F}$,$\mathcal{D}$,γ,β) = m, then **the sample complexity** of optimizing/searching over $\mathcal{F}$ and $\mathcal{D}$ is at least

$$\min\{1/\gamma, (m/\beta)^{1/2}\}.$$

proof technique

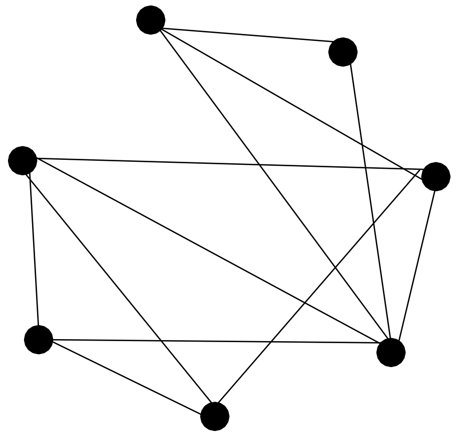
- Adversary picks target randomly among the m distributions.
- We can assume all queries are asked with 1 sample.
- Learning begins knowing all targets are equally likely.
- With each query, learner receives limited information about the target.
- Learner must ask many queries before being sure of the target.

Planted Clique

the first concrete evidence of its hardness

Erdős–Rényi Random Graphs

G(n,0.5) is a random graph on n vertices, with each edge equiprobably present or absent.

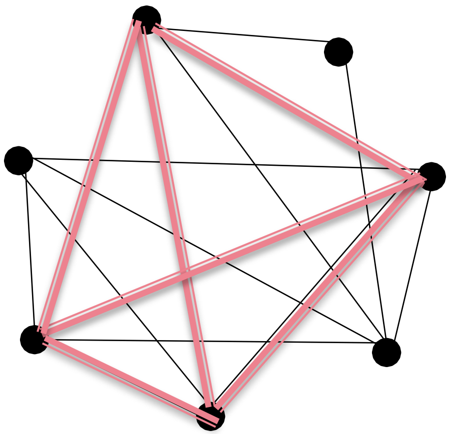


As n gets large, G(n,0.5) almost surely has a clique of size 2 log n .
Conjectured hard to find a clique of size (1+ε) log n. [Karp '76]

still open

Plants in Erdős–Rényi Random Graphs

What if a clique of size k is planted in a random graph? Can we find the planted clique? [Jerrum '92]

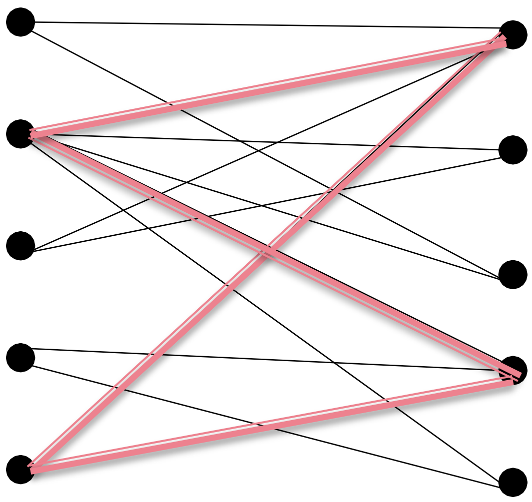


For $k = \Omega(n^{1/2})$ we have efficient algorithms. [Kucera '95, Alon et al. '98]

But after 20 years, we still don't know how to find plants of size $o(n^{1/2})$.

Bipartite Plants

What if a clique of size k is planted in a random graph? Can we find the planted clique? [Jerrum ’92]



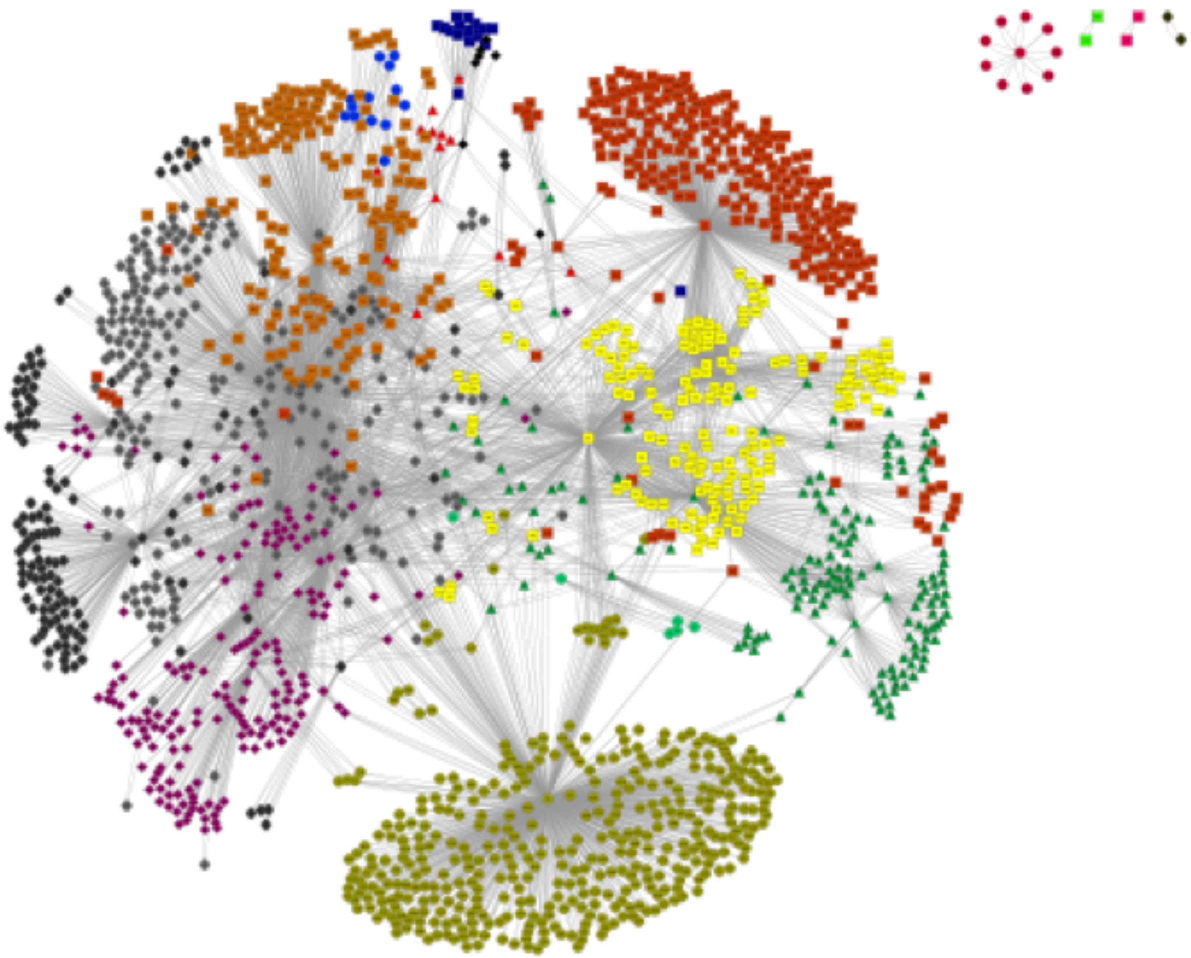
For k= $\Omega(n^{1/2})$ we have efficient algorithms. [Kucera ’95, Alon et al. ’98]

But after 20 years, we still don’t know how to find plants of size $o(n^{1/2})$.

The Planted Clique Problem

- Jerrum ['92] showed a specific Markov chain fails for small k.
- Used for cryptographic primitives [Juels and Peinado '00].
- If planted clique is hard, so is finding approximate Nash equilibria in some games [Hazan Krauthgamer 2011, Minder Vilenchik 2009].
- A version of the bipartite problem also used for cryptography [Applebaum Barak Wigderson '10].

Example Application: Community Detection



Our Result

Result: no statistical algorithm can find a bipartite planted clique of size $k = O(n^{1/2-\delta})$, for $\delta > 0$.

The reason to believe that finding planted cliques is hard is now similar to the reason to believe that the notorious noisy parity problem is hard. Namely, both problems are hard for statistical algorithms.

The Proof Sketch

First, we formulate an equivalent statistical problem

Let D be a distribution over $\{0,1\}^n$. When x is chosen from D,

- w.p. (n-k)/n, x is a uniform random vector.
- w.p. k/n, x has a fixed k coordinates set to 1 and the rest uniformly at random.

If only n such vectors are chosen, this can be seen as the adjacency matrix of a bipartite planted clique instance.

The Proof Sketch

Let D be the uniform distribution over $\{0,1\}^n$ and let D_S be the planted clique distribution on a k-variable (vertex) subset S.

Lemma [Babai and Frankl '92]: If all cliques overlap on at most λ vertices, there are $m \geq n^{\lambda/2}$ such D_S's.

$\gamma = 2^{\lambda} k^2/n^2$ and $\beta = 2^{k} k^2/n^2$ when computing statistical dimension.

Optimizing for λ, we get a sample complexity lower bound of

$$n^2/(2^{k/\log n} k^2) .$$

Assuming we only have n samples, for $k = o(\log^2 n)$, no statistical algorithm can solve planted clique.

Planted Clique

- To get this all the way up to $k = O(n^{1/2-\delta})$, we need to extend our general theorems and do a lot more work…
- In the end, we can prove that ~ n^2/k^2 samples are necessary and sufficient to find planted cliques.
- With only n samples (an actual instance), a statistical algorithm cannot succeed when $k < n^{1/2-\delta}$.
- This bound is also almost tight for the (harder) planted densest subgraph.

Other Applications

to MAX-XOR-SAT, k-clique, and moment maximization

MAX-XOR-SAT

Problem: Let D be a distribution over XOR clauses $c \in \{0,1\}^n$ ($c_i=1$ means variable i appears in c). The problem is to find an assignment $x \in \{0,1\}^n$ that **maximizes the expected number of satisfied clauses**.

- Clause c is **satisfied** by assignment x if $\chi_c(x)=1$.
- Similar to the parity problem in learning, but the **distribution is over *clauses***

Clause	Prob.
$c1 \oplus c3 \oplus c4$	1/2
$c1 \oplus c2$	1/8
$c4$	1/4
$c1 \oplus c2 \oplus c4$	1/16
$c2 \oplus c4$	1/16

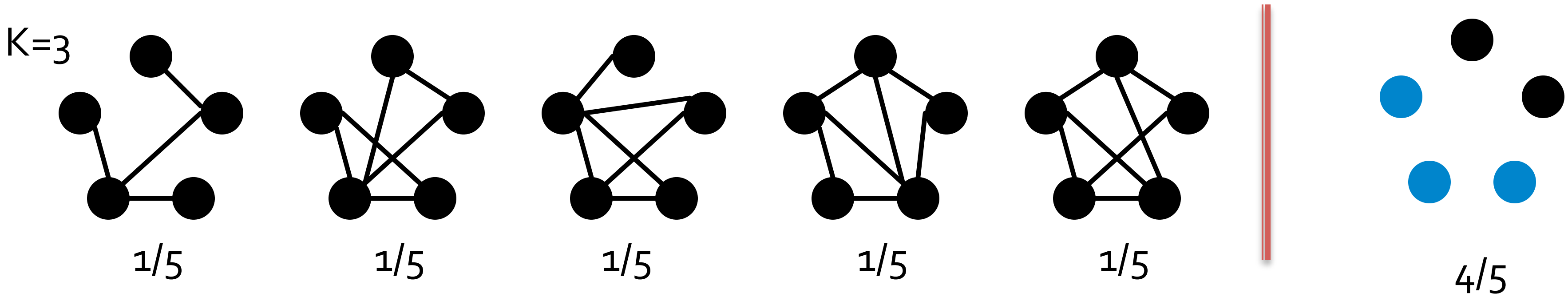
The assignment **x1 = 1**; **x2 = 0**; **x3=1**; **x4 = 1** has probability 15/16 of satisfying a clause.

Result: For r odd, any statistical algorithm for MAX-XOR-SAT **requires $2^{n/2}$** samples to approximate the probability of satisfying a random clause to within ½.

k-Clique

Note: C(n,k) is “n choose k”

Problem: Let D be a distribution over $X = \{0,1\}^{C(n,2)}$, corresponding to graphs G on n vertices. Let **$I_S(G) = 1$ if S induces a clique on G and $I_S(G) = 0$ otherwise**. The problem is to find a subset $S \subseteq V$ that maximizes $E_{G\sim D}[I_s(G)]$.



Result: For r odd, any statistical algorithm for distributional k-Clique **requires $n^{k/2}$** samples to approximate the probability of hitting a clique to within 2^{-k^2}.

Moment Maximization

Problem: Let D be a distribution over $\{-1,1\}^n$ and $r \in Z^+$. The goal is to find a vector u* that **maximizes the expected r'th moment** of the projection to u of a random point x from D. i.e.

$$u^* = \arg\max_{u \in \Re : ||u|| = 1} \mathrm{E}_{x \sim D}[(u \cdot x)^r].$$

Result: For r odd, any statistical algorithm for moment maximization **requires $n^{r/2}$** samples to approximate the rth moment **to within ~ $(r/e)^{r/2}$**.

Discussion

Presented a powerful framework for proving unconditional lower bounds for a wide class of algorithms.

Inspired by and generalizes the SQ model in learning theory.

Gave first concrete hardness result for planted clique, as well as new hardness results for some other problems.

Fundamentally new, non-statistical, algorithms are needed to make progress on these and a variety of other problems!

Thank You!

Questions?