

# LOWER BOUNDS ON LEARNING RANDOM STRUCTURES WITH STATISTICAL QUERIES

1

**Lev Reyzin**

**Georgia Institute of Technology**

**Dana Angluin**

Yale University

**David Eisenstat**

Brown University

**Leonid Kontarovich**

Ben Gurion University

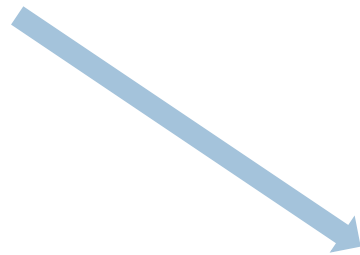
ALT 2010

October 2010

# A Quick Look

2

## Lower Bounds on Learning Random Structures with Statistical Queries



(monotone/not) DNF, Decision Trees, and Finite Automata

# What Are Statistical Queries?

3

## □ PAC Learning [Valiant '84]

- Learner gets hypothesis class  $C$ ,  $m$  instances  $x \in X$  of length  $n \sim D$ , and parameters  $0 < \epsilon, \delta < 1$ .
- To PAC learn  $C$ , learner must produce hypothesis  $h$  such that for all  $c$  in  $C$ , for all  $D$ , and  $m$  examples  $x \sim D$  labeled by  $c$ ,  $\Pr_{x \sim D}(h(x) \neq c(x) > \epsilon) < \delta$ .
  - want  $m$ , running time to be  $\text{poly}(n, |C|, 1/\epsilon, 1/\delta)$

# What Are Statistical Queries?

4

- PAC w/ **Classification Noise** [Angluin and Laird '88]
  - Learner gets hypothesis class  $C$ ,  $m$  instances  $x \in X$  of length  $n \sim D$ , and parameters  $0 < \epsilon, \delta < 1$ ,  $\eta < 1/2$ .
  - To PAC learn  $C$ , learner must produce hypothesis  $h$  such that for all  $c$  in  $C$ , for all  $D$ , and  $m$  examples  $x \sim D$  labeled by  $c$  **w.p.  $(1 - \eta)$** ,  $\Pr_{x \sim D}(h(x) \neq c(x) > \epsilon) < \delta$ .
  - want  $m$ , running time to be  $\text{poly}(n, |C|, 1/\epsilon, 1/\delta)$

# What Are Statistical Queries?

5

- Statistical Query (SQ) learning [Kearns '93] is another way to model PAC learning under noise.
- In PAC learning, you get to see individual examples, but in SQ learning, you **query an SQ oracle** without seeing individual examples.
- The SQ oracle returns approximate answers.
- It turns out that SQ learning is (strictly) harder than PAC learning.

# SQ Learning

6

- Let  $D$  be a distribution over  $X = \{0,1\}$ . Learner gets class  $C$ ,  $n$ ,  $0 < \varepsilon < 1$ .
- SQ Oracle:
  - Learner presents  $(\chi, \tau)$ 
    - $\chi$  maps  $X \times \{0,1\}$  to  $\{0,1\}$ , comp in  $\text{poly}(n, 1/\varepsilon)$
    - $\tau \in [0,1]$ , and is  $\text{poly}(n, 1/\varepsilon)$
  - Oracle returns  $v$  s.t.
    - $E_{x \sim D}[\chi(x, c(x)) - v] \leq \tau$
- An Algorithm SQ learns  $C$  if for all  $c \in C$  for all  $D$ , it produces  $h$  such that  $P_D(h(x) \neq c(x)) < \varepsilon$  and runs in time  $\text{poly}(n, 1/\varepsilon)$

# SQ Oracle

7

- A statistical query abstracts the process of drawing a sample of labeled examples  $(x, c(x))$  and estimating the probability they satisfy  $\chi$ .
- One can simulate SQ learning in the PAC model, so if  $C$  is SQ learnable, it is also PAC learnable.
- Many PAC learning algorithms are actually SQ algorithms (ie we can turn them into SQ algorithms)
  - ▣ Boosting [Aslam & Decatur '93]
  - ▣ Algorithms using statistical estimates of various parameters

# Our Result

8

- Our result:
  - ▣ Random instances of monotone DNF, Decision Trees, and DFA are not (weakly) learnable with Statistical Queries.
  
- Why is this interesting?
  - ▣ Under the uniform distribution, we know that random monotone DNF [Sellie '09] and Decision Trees [Jackson and Servedio '03] are PAC learnable.
    - These results use Fourier techniques (which are SQ)
  - ▣ It shows that distributional assumptions are necessary of this type algorithm.

# How Did We Prove It?

9

- Learning Parities -- the hypothesis class is parity functions on a subset (of, say,  $k$ ) of the  $n$  variables.
  - ▣ Parities are easy to PAC learn.
    - By seeing individual examples, it's not hard to tell what the relevant variables are. [Gauss 1810]
  - ▣ SQ cannot learn Parities [Kearns '93, Blum et al. '94]
    - Unless the “guess” to the oracle is the correct parity, the statistical query basically returns no information (ie. 50% agreement).

# Status of Learning Parity

10

PAC	PAC w/ noise “noisy parity”	Statistical Queries
Yes [Gauss]	$O(2^{n/\log n})$ [BKW] maybe better?	No!

# How Did We Prove It?

11

- It is easy to make DNF, Decision Trees, and DFAs that encode arbitrary parity functions.
- It turns out that even random monotone DNF, Decision Trees, and DFAs encode the parity function on  $\omega(1)$  variables.
  - ▣ We have to “drive” the distribution to the proper place.
- We will start by giving a proof sketch of the DNF result, which is the simplest case.

# [Sellie '09] Model of Random DNF

12

- Let  $V = \{v_1, v_2, \dots, v_n\}$  be the variables
- Each term is a conjunction of a random subset of  $(c \log n)$  variables, with each variable is negated with probability  $1/2$ .
- The target DNF is a disjunction of selected terms.
- [Sellie '09] gives a poly time algorithm for learning a random DNF with at most  $(n^c \log \log n)$  terms.

# A Read-Once DNF

13

- Remember: [Sellie '09] learns a DNF of  $n$  variables, ( $c \log n$ ) variables per term, and  $(n^c \log \log n)$  terms.
- Let  $\phi$  be a random DNF with  $\sim n$  variables,  $n^{1/3}$  terms, and  $\frac{1}{3} \log(n)$  variables per term.
- **Claim:** with probability  $1 - o(1)$   $\phi$  is read-once
  - ▣ In a read-once formula, each variable occurs only once.
- For ease of explanation, we'll assume  $\phi$  is monotone (no negated variables)
  - ▣ this assumption doesn't change the main ideas.

# Using the Distribution

14

- say we wish to compute parity on  $\{x_{33}, x_{57}, x_{108}\}$ 
  - Can write this as a DNF  $\psi$
  - $\psi = (x_{33}, x_{57}, x_{108}) \vee (x_{33}, x'_{57}, x'_{108}) \vee (x'_{33}, x_{57}, x'_{108}) \vee (x'_{33}, x'_{57}, x_{108})$
  
- Now we shall show how to use the distribution  $D$  to make our random read-once formula  $\phi$  compute this parity.
  
- say  $\phi = (v_{14}v_{133}v_{170}) \vee (v_{22}v_{101}v_{337}) \vee (v_{55}v_{266}v_{413}) \vee (v_{10}v_{332}v_{507})$ 
  - $D$  can make  $x_{33}$  be represented by  $v_{14}$ ,  $v_{22}$ ,  $v_{55}$ , and  $v_{10}$
  - To be consistent with  $\psi$ ,  $D$  will set  $v_{14} = v_{22} = x_{33}$  and  $v_{55} = v_{10} = x'_{33}$

# The Equi-Grouping

15

- $\psi = (x_{33}, x_{57}, x_{108}) \vee (x_{33}, x'_{57}, x'_{108}) \vee (x'_{33}, x_{57}, x'_{108}) \vee (x'_{33}, x'_{57}, x_{108})$
- $\phi = (v_{14}, v_{133}, v_{170}) \vee (v_{22}, v_{101}, v_{337}) \vee (v_{55}, v_{266}, v_{413}) \vee (v_{10}, v_{332}, v_{507})$
- The rest of the  $n$  variables will similarly be grouped into “equi-grouping” to fool the learner

# Finishing Up the DNF Lower Bound

- In general, if  $\psi$  represents parity on  $\frac{1}{3} \log n$  variables then  $\phi$  will have terms of size  $\frac{1}{3} \log n$  and have  $\sim n^{1/3}$  terms to cover all settings.
- This procedure will work as long as  $\phi$  is read-once, which we've shown will occur almost-certainly (as  $n$  gets large).
- Learning parity on  $\frac{1}{3} \log n$  variables out of a possible  $n^{2/3}$  variables cannot be done in poly time with SQ queries.

# Decision Trees and DFA

17

- The reductions for Decision Trees and DFA are more complicated, but also follow similar ideas of having the distribution “guide” the parity.
- Takeaway: neither are learnable with SQs.

# Status of Problems

18

	Random DNF	Random DTs	Random DFA
SQ	No	No	No
PAC/SQ (uniform)	Yes	Yes	open
PAC	open	open	open

# Conclusions

- Our work explains the distributional assumptions in the previous work on DTs and DNF.
- It is believed that random DTs, DNFs, and DFAs are PAC learnable (w.o. assumptions on  $D$ ), but to do this algorithms will have to look at individual examples.
  - ▣ The end goal (to many) in this line of research is to PAC learn arbitrary DTs and DNFs (though not properly).
  - ▣ Almost no hope for PAC learning arbitrary DFAs (even under uniform) as they can encode RSA in the worst case.

# Related Open Problems

20

- Open problem: understand the complexity of “noisy parity”.
  - ▣ Our work shows that these random structures are probably not PAC learnable under classification noise.
- Open problem: SQ (or PAC) learn random DFA under the uniform distribution
- Open problem: extend the positive SQ results for DTs and DNF to other distributions (ie. product distributions).