Algebraic Methods in Computer Science 7/1/2014

Lecture: 11

Lecturer: *Amir Shpilka* Scribe: *Efraim Gelman*

1 Computational learning theory

Given $F \subseteq \{g : \{0,1\}^n \to \{0,1\}\}$ and $f \in F$ and we can send x and get $f(x)$. We want to find $h : \{0,1\}^n \to \{0,1\}$ s.t. $Pr_x[f \neq h] \leqslant \varepsilon$. This is called **learning with membership queries.** We allow probabilistic algorithm , so we would actually like to have

$$\Pr_{quries}[\Pr_x[f \neq h] > \varepsilon] \leqslant \delta.$$

PAC learning: probably (δ) almost (ε) correct. Question: What kind of functions can we learn efficiently this way, when efficient means polynomial in $(\log 1/\delta, 1/\varepsilon, n)$.
Example 1: F is the set of linear functions i.e. $f \in F \Rightarrow \exists a, b \in \{0,1\}^n$ s.t. $f(x) =< x, a > +b$ and obviously we can find a, b by query $f(0)$ to find b and $f(e_1), \ldots, f(e_n)$ to find $a = (a_1, \ldots, a_n)$. So we found f within $n+1$ queries.
Example 2: F is the set of t-sparse functions, i.e. functions that at most t of its fourier coefficients are different from zero.
Lets say you know that $a_1, \ldots, a_t$ are the fourier coefficients that are different from zero, can we learn f ? We will try to approximate the fourier coefficients $\hat{f}(a_i)$ with $\hat{h}(a_i)$ for $i = 1, \ldots, t$ and define

$$h(x) = \sum_{i=1}^{t} \hat{h}(a_i)\chi_{a_i}(x)$$

and answer $\mathbf{sgn}(\mathbf{h})$. (we assume for convenience that $f : \{0,1\}^n \to \pm 1$). Assume that $\forall i |\hat{h}(a_i) - \hat{f}(a_i)| < \varepsilon$ and we will estimate $\Pr[sgn(h) \neq f]$.

$$\Pr[sgn(h) \neq f] \leqslant E[(f-h)^2] = \sum_a ((f \hat{-} h)(a))^2 =$$

$$\sum_a (\hat{f}(a) - \hat{h}(a))^2 = \sum_{i=1}^{t} (\hat{f}(a_i) - \hat{h}(a_i))^2 \leqslant t\varepsilon^2$$

The first inequality since for every x in the left side s.t. $\mathbf{sgn}h(x) \neq f(x)$ the difference between $f(x)$ and $\mathbf{h}(\mathbf{x})$ is at least 1.
The equality after is by Parseval.

Corollary 1.1. *If for every i we have that $|\hat{f}(a_i) - \hat{h}(a_i)| \leqslant \frac{\sqrt{\varepsilon}}{\sqrt{t}}$ then*

$$\Pr[sgn(h) \neq f] \leqslant \varepsilon$$

How can we approximate the fourier coefficients of f assuming we know $a_1, \ldots, a_t$?

$$\hat{f}(a) = \langle f, \chi_a \rangle = \frac{1}{2^n} \sum_x f(x)\chi_a(x) = E[f(x)\chi_a(x)]$$

We notice that $|f(x)\chi_a(x)| = 1$. For every a_i we choose randomly and independently $x_1, \ldots, x_m$ and estimate

$$\hat{h}(a_i) = \frac{1}{m} \sum_{j=1}^{m} f(x_j)\chi_{a_i}(x_j)$$

A bad event is that we got "bad" estimate in at least on of the a_i, but

$$\Pr[|\frac{1}{m} \sum_{j=1}^{m} f(x_j)\chi_{a_i}(x_j) - E[f(x)\chi_{a_i}(x)] > \frac{\sqrt{\varepsilon}}{\sqrt{t}}] \quad \leqslant \quad exp(-(\frac{\sqrt{\varepsilon}}{\sqrt{t}})^2 m) < \frac{\delta}{t}$$

when we take

$$m = O(\frac{t}{\varepsilon} \log \frac{t}{\delta})$$

Corollary 1.2. *The algorithm that randomly asks m queries as described, will give us with probability greater than $1 - \delta$ a function h s.t. the distance from f is less than ε.*

We now look for conditions that ensures f is close to $t - sparse$ function.

Definition 1.1. *The spectral norm of f is*

$$||\hat{f}||_1 = \sum_a |\hat{f}(a)|$$

For $t - sparse$ function f we know by Cauchy-Swartz that $||\hat{f}||_1 \leqslant \sqrt{t}$.

Proposition 1.3. *If $||\hat{f}||_1 \leqslant A$ then f is ε-close to the **sgn** of a $(t = \frac{A^2}{\varepsilon}) - sparse$ function*

Proof. Assume $\hat{f}(\alpha_1), \ldots, \hat{f}(\alpha_t)$ for $t = \frac{A^2}{\varepsilon}$ are the t largest coefficients of f in absolute value. In particular, for every $\beta \notin \{\alpha_1, \ldots, \alpha_t\}$ we have that $|\hat{f}(\beta)| \leqslant \frac{A}{t}$. Let

$$g(x) = \sum_{i=1}^{t} \hat{f}(\alpha_i)\chi_{\alpha_i}(x)$$

So,

$$\Pr[sgn(g) \neq f] \leqslant E[(f-g)^2] = \sum_{\alpha}((f \hat{-} g)(\alpha))^2 =$$

$$\sum_{\alpha}(\hat{f}(\alpha) - \hat{g}(\alpha))^2 = \sum_{\beta \notin \{\alpha_1,\ldots,\alpha_t\}} (\hat{f}(\beta))^2 \leqslant \frac{A}{t} \sum_{\beta \notin \{\alpha_1,\ldots,\alpha_t\}} \hat{f}(\beta) \leqslant \frac{A}{t}A = \frac{A^2}{t}$$

□

Corollary 1.4. *If $||\hat{f}||_1 \leqslant A$ then so with $Poly(A, \frac{1}{\varepsilon}, \log 1\delta)$ queries, one can find $Poly(\frac{A}{\delta}) - sparse$ function that her **sgn** approximates f (if one knows which ar the large fourier coefficients) .*

We now would like to find the Fourier coefficients that are larger then $\theta = \frac{\varepsilon}{A}$:

2 Kushilevitz-Mansour Algorithm for finding the large fourier coefficients

The idea:
We think of $v \in \{0,1\}^n$ as subset $S \subset [n]$ (α as characteristic vector of S). We will go over a binary tree, the root is the empty set, and in layer i we want to determine wether $i \in S$ or not. In each vertex we'll try to estimate the sum of weights in the subtree under it while we are not interested in subtrees with low weight. In each layer we have at most $\frac{A}{\theta}$ interesting subtrees, this way we will be able to build S.

Our goal now is the following:
Given prefix $\alpha \in \{0,1\}^k$, to estimate

$$\sum_{\beta \in \{0,1\}^{n-k}} \hat{f}(\alpha \circ \beta)^2,$$

where we use square instead of absolute value for technical reasons. Now

$$\sum_{\beta \in \{0,1\}^{n-k}} \hat{f}(\alpha \circ \beta)^2 = E_y(\sum_{\beta \in \{0,1\}^{n-k}} \hat{f}(\alpha \circ \beta)\chi_\beta(y))^2$$

So we define $f_\alpha(y) = \sum_{\beta \in \{0,1\}^{n-k}} \hat{f}(\alpha \circ \beta)\chi_\beta(y)$, *hence*

$$\sum_{\beta \in \{0,1\}^{n-k}} \hat{f}(\alpha \circ \beta)^2 = E_y(f_\alpha(y))^2$$

Now, we can't get queries from f_α, *so how can we estimate* $f_\alpha(y)$ *using queries from* f*? Well,*

$$\begin{aligned} f_\alpha(y) &= \sum_{\beta \in \{0,1\}^{n-k}} \hat{f}(\alpha \circ \beta)\chi_\beta(y) = \sum_{\beta \in \{0,1\}^{n-k}} (E_x f(x)\chi_{\alpha \circ \beta}(x))\chi_\beta(y) \\ &= \sum_{\beta \in \{0,1\}^{n-k}} (E_{x_1,x_2} f(x_1 \circ x_2)\chi_\alpha(x_1)\chi_\beta(x_2))\chi_\beta(y) \\ &= \frac{1}{2^k}\sum_{x_1,x_2} f(x_1 \circ x_2)\chi_\alpha(x_1)\frac{1}{2^{n-k}}\sum_{\beta \in \{0,1\}^{n-k}} \chi_\beta(x_2))\chi_\beta(y) \\ &= \frac{1}{2^k}\sum_{x_1} f(x_1 \circ y)\chi_\alpha(x_1) = E_{x_1}[f(x_1 \circ y)\chi_\alpha(x_1)] \end{aligned}$$

Conclusions: 1. For every y *we have that* $|f_\alpha(y)| \leqslant 1$
2. Given $y \in \{0,1\}^{n-k}$, *if we sample* $Poly(1/\varepsilon, \log(1/\delta) = m$ x_1*'s and calculate* $\frac{1}{m}\sum_{j=1}^{m} f(x_j \circ y)\chi_\alpha(x_j)$ *we'll get with probability at least* $1-\delta$, *an* ε *close to* $f_\alpha(y)$.

We now have an algorithm to approximate $\sum_{\beta \in \{0,1\}^{n-k}} (\hat{f}(\alpha \circ \beta))^2$ *and we want to find the heavy Fourier coefficients. We'll start at the* 0 *layer at the vertex* $\emptyset$. *In step k, for every* $\alpha \in \{0,1\}^k$ *that is "alive" we'll approximate* $\sum(\hat{f}(\alpha \circ 0))^2$ *and* $\sum(\hat{f}(\alpha \circ 1))^2$ *and set "alive"* $\alpha \circ b$, $b \in \{0,1\}$ *if*

$$\sum(\hat{f}(\alpha \circ b))^2 > (\theta - \varepsilon)^2$$

3 Decision tree

The vertices are labeled with variables, the leaves with 0 *or* 1. *The edges form a vertex labeled* x_i *are labeled* $x_i = 0$ *and* $x_i = 1$. *The value of input* x *calculated*

by reading the variable in the root an going according to the tree. The size of a DT is the number of its leaves. Given a function f we denote $DT(f)$ the minimal size DT that computes it.
Examples: 1. DT(parity)=2^n
2. DT(or)=$n-1$

We'll show connection between DT(f) and $||\hat{f}||_1$.

Claim 3.0.1. $||\hat{f}||_1 \leqslant 3DT(f)$

Proof. Look at the optimal DT for f. It has m leaves. Each leaf l has function of the form

$$f_l(x) = (x_{i_1}^{\varepsilon_1} \wedge \ldots \wedge x_{i_{d_l}}^{\varepsilon_{d_l}})$$

such that when the input x gets the leaf l iff $f_l(x) = 1$. So $f(x) = \sum_l f_l(x)$. this sum has at most m summands, we'll see

$$||\hat{f}_l||_1 \leqslant 3$$

Indeed let $g = x_1^{\varepsilon_1} \wedge \ldots \wedge x_k^{\varepsilon_k}$ then it can be written as $\frac{1}{2^k} \prod_{i=1}^{k} (1 \pm y_i)$ depending on the ε_i, where the $y_i \in \{1, -1\}$.

4 Two source extractors

We have two independent sources X, Y with a lot of Entropy and we would like to extract one bit close to uniform, i.e.

$$E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$$

s.t. if $H_\infty(X), H_\infty(Y)$ are large then

$$E(X,Y) \cong U_1.$$

Chor-Goldreich construction: $E(X,Y) = (-1)^{<X,Y>}$

Claim 4.0.2. *If* $X, Y \in \{0,1\}^n$ *s.t.* $|x| \cdot |y| \geqslant \frac{2^n}{\varepsilon^2}$ *then*

$$E_{x \in X, y \in Y} (-1)^{<x,y>} \leqslant \varepsilon$$

Proof.

$$|\frac{1}{|X|}\frac{1}{|Y|}\sum_{x\in X,y\in Y}(-1)^{<x,y>}| \quad = \quad |\frac{1}{|X|}\frac{1}{|Y|}\sum_{x\in\{0,1\}^n,y\in Y}1_x(x)(-1)^{<x,y>}| =$$

$$|\frac{2^n}{|X|}\frac{1}{|Y|}\sum_{y\in Y}\hat{1_x}(y)| \quad \leqslant_{cauchy-swartz} \quad \frac{2^n}{|X|}\frac{1}{|Y|}\sqrt{|Y|}\sqrt{\sum_{y\in Y}(\hat{1_x}(y))^2}$$

But $\sum_{y\in Y}(\hat{1_x}(y))^2 \leqslant \sum_{y\in\{0,1\}^n}(\hat{1_x}(y))^2 = E_x[1_x^2] = E[1_x] = \frac{|X|}{2^n}$, hence plugging into the term above we get $\sqrt{\frac{2^n}{|X||Y|}} < \varepsilon$ □