Scribe: Omid Etesami

Notes for Lecture 9

In this lecture, we will see an application of the Goldreich-Levin theorem to learning decision trees with queries under uniform distribution.

In a general learning problem, we are given a function $f : \{0,1\}^n \to \{1,-1\}$ from a known class of functions $\mathcal{F}$. A learning algorithm tries to guess the function f by querying the value of f at different points, and finally outputs a candidate function $h : \{0,1\}^n \to \{1,-1\}$ which is close to f with high probability. The boolean function h can be specified by a boolean circuit computing it.

Definition 1 (Learnability) *A class $\mathcal{F}$ of functions is* learnable with queries under uniform distribution *if there exists a learning algorithm A such that for every function $f \in F$, given oracle access to f and parameters $\epsilon, \delta > 0$, the algorithm outputs, with probability $\geq 1 - \delta$ over the randomness of the algorithm, a function $h = A^f(\epsilon, \delta)$ which is ϵ-close to f. Moreover, the running time of algorithm A should be poly$(n, 1/\epsilon, 1/\delta)$.*

As an example, it is easy to see that the class of all linear functions, $\{\chi_S : \chi_S(x_1, \ldots, x_n) = (-1)^{\sum_{i \in S} x_i},\ S \subseteq [n]\}$, is learnable.

1 Kushilevitz-Mansour General Learning Algorithm

We describe a general learning algorithm that uses the Goldreich-Levin theorem.

Theorem 2 (Goldreich-Levin) *There exists a probabilistic algorithm GL that given paramter τ and oracle access to function $f : \{0,1\}^n \to \{1,-1\}$, outputs in time poly$(n, 1/\tau)$ a list $L = \mathrm{GL}^f(\tau)$ of subsets of coordinates that with high probability contains every $S \subseteq [n]$ such that $|\hat{f}(S)| \geq \tau$.*

PROOF: There are two small differences between the above statement of the theorem and the statement we proved in the last lecture:

- We have replaced the condition $\mathbf{Pr}_x[f(x) = \chi_S(x)] \geq 1/2 + \epsilon$ by $|\hat{f}(S)| \geq \tau$. If $|\hat{f}(S)| \geq \tau$ then $\mathbf{Pr}_x[f(x) = \chi_S(x)] \geq 1/2 + \tau/2$ or $\mathbf{Pr}_x[\neg f(x) = \chi_S(x)] \geq 1/2 + \tau/2$. Hence, if $\hat{f}(S) \geq \tau$, by running the Goldreich-Levin algorithm once for f and once for $\neg f$ (with parameter $\epsilon = \delta/2$), we find S with high probability.
- We require that with high probability *every* S such that $\hat{f}(S) \geq \tau$ is included in list L. By Parseval inequality, the number of such S is at most $1/\epsilon^2$. Therefore, after $O(\log \epsilon)$ iterations of the Goldreich-Levin algorithm, we find all such S with high probability.

$\square$

Lemma 3 (Estimating Fourier Coefficients) *By random sampling, we can get an estimate $\bar{f}(S)$ for the Fourier coefficient $\hat{f}(S) = 2\,\mathbf{Pr}_x[f(x) = \chi_S] - 1$; using $O(k/\delta^2)$ samples, we have $\mathbf{Pr}[|\bar{f}(S) - \hat{f}(S)| > \delta] \leq \exp(-k)$.*

Here is a general learning scheme:

> Kushilevitz-Mansour-Learn (f)
> let $L = GL^f(\tau)$
> **for** all $S \in L$, let $\bar{f}(S)$ be an estimation of $\hat{f}(S)$
> let $g(x) = \sum_{S \in L} \bar{f}(S)\chi_S(x)$
> **return** $h(x) = \text{sign}(g(x))$

We have $\mathbf{Pr}_x[f(x) \neq h(x)] \leq \mathbb{E}_x(f(x) - g(x))^2$, because if $f(x) \neq h(x)$ then $|f(x) - g(x)| \geq 1$. Moreover with high probability, $\{S : |\hat{f}(S)| \geq \tau\} \subseteq L$ and $|\bar{f}(S) - \hat{f}(S)| \leq \delta$ for every $S \in L$. Thus,

$$\begin{aligned}
\mathbf{Pr}_x[f(x) \neq h(x)] &\leq \mathop{\mathbb{E}}_x(f(x) - g(x))^2 \\
&= \sum_S (\hat{f}(S) - \hat{g}(S))^2 \\
&\leq \sum_{S \in L} \delta^2 + \sum_{S \notin L} \hat{f}^2(S) \\
&\leq \delta^2|L| + \sum_{S \notin L} \hat{f}^2(S).
\end{aligned}$$

2 Learning Decision Trees

If f is a decision tree tree with m leaves, then $\sum_S |\hat{f}(S)| \leq m$, and we have

$$\sum_{S \notin L} \hat{f}^2(S) \leq \tau \sum_{S \notin L} |\hat{f}(S)| \leq \tau m.$$

Thus, we have

$$\mathop{\mathbf{Pr}}_x[f(x) \neq h(x)] \leq \delta^2|L| + \tau m \leq \epsilon/2 + \epsilon/2 \leq \epsilon$$

for $\tau = \epsilon/2m$ and $\delta = (\epsilon/2|L|)^{1/2} = \text{poly}(m, 1/\epsilon)$.

Corollary 4 *Polynomial size decision trees are learnable under uniform distribution.*

3 Learning Constant-Depth Circuits

We will now sketch an analysis of the learnability of the class $\mathcal{F}(s,d)$ of functions computable by a circuit of size $\leq s$ and depth $\leq d$ consisting of AND and OR gates of arbitrary fan-in and NOT gates (with the usual assumption that NOT gates only appear on the first level and are not counted in the size and depth of the circuit). Notice that $\mathcal{F}(s,2)$ is the class of CNF and DNF formulas with $s-1$ clauses.

Remark 1 *There are functions $f \in \mathcal{F}(O(n), 2)$ such that $\sum_S |\hat{f}(S)| \geq 2^{\Omega(n)}$. (Prove as an exercise.) Therefore, for analyzing learnability of $\mathcal{F}(s, d)$, we cannot use the same argument that we gave for decision trees.*

We will use a corollary of Hastad's switching lemma:

Theorem 5 (Hastad, Linial-Mansour-Nisan) *For all $f \in \mathcal{F}(s, d)$, we have $\sum_{|S|>t} \hat{f}^2(S) \leq \alpha$ for $t = 28(14 \log(2s/\alpha))^{d-1}$.*

By the above theorem, for all $f \in \mathcal{F}(s, d)$, we have

$$\begin{aligned}
\mathop{\mathbf{Pr}}_x[f(x) \neq h(x)] &\leq \delta^2 |L| + \sum_{S \notin L, |S|>t} \hat{f}^2(S) + \sum_{S \in L, |S|>t} \hat{f}^2(S) \\
&\leq \delta^2 |L| + \alpha + \tau \sum_{S \in L} |\hat{f}(S)| \\
&\leq \delta^2 |L| + \alpha + \tau \binom{n}{t} \leq \epsilon,
\end{aligned}$$

for $\alpha = \epsilon/4$, $\tau = \epsilon/4n^t$, $\delta = (\epsilon/2|L|)^{1/2}$, and $|L| = \mathrm{poly}(n/\tau)$.

Corollary 6 *The class $\mathcal{F}(s, d)$ is learnable in time $n^{O(\log(s/\epsilon))^{d-1}}$ with accuracy ϵ.*

Finally, we note that for the case $d = 2$ (and assuming $s = n^{O(1)}$ and constant ϵ), Mansour showed the above learning algorithm learns $\mathcal{F}(s, 2)$ in quasi-polynomial time $n^{O(\log \log n)}$. It is an open question whether the above learning algorithm can learn in polynomial time; however, Jackson showed using a generalization of the above algorithm that $\mathcal{F}(s, 2)$ is learnable in polynonial time.