6.895 Randomness and Computation	February 27, 2008
Lecture 07: Learning Halfspaces	
Lecturer: Ronitt Rubinfeld	*Scribe: Ning Xie*

1 Review of Last Lecture

Last time we said that a function $f : \{-1,1\}^n \to \mathbb{R}$ has $\alpha(\epsilon, n)$-Fourier concentration if

$$\sum_{S \subseteq [n], |S| > \alpha(\epsilon,n)} \hat{f}(S)^2 \leq \epsilon$$

for all $0 < \epsilon < 1$. For functions that have $d = \alpha(\epsilon, n)$-Fourier concentration, we showed the *Low Degree Algorithm* for learning such functions: estimate all the low-degree Fourier coefficients (that is, $\hat{f}(S)$ for all $|S| \leq d$) and output the sign of the estimated low-degree polynomial (output hypothesis $\mathrm{sign}(\sum_{S:|S|\leq d} C_S \chi_S(x))$, where C_S is the estimated Fourier coefficients). Today we are going to see further applications of the Low Degree Algorithm in learning theory.

2 Noise Sensitivity

Definition 1 (Linear Threshold Function) *A Boolean function $h(x)$ is called a Halfspace Function (or Linear Threshold Function) if h can be written as $h(x) = \mathrm{sign}(\sum_{i=1}^n w_i x_i - \theta)$, where w_i are real numbers called weights and $\mathrm{sign}(x)$ is 1 if $x \geq 0$ and -1 otherwise.*

We are going to see an algorithm that learns halfspaces (under the uniform distribution) with sample complexity $n^{O(1/\epsilon^2)}$. There are other learning algorithms with better sample complexity. The advantage of the algorithm we study is that it can be easily generalized to learn any function that depends on a constant number of halfspaces. The main tool we are going to use is the Low Degree Algorithm but combined with a key new idea: noise sensitivity.

Definition 2 (Noise Operator) *For any $0 < \epsilon < 1/2$, define the noise operator $N_\epsilon : \{-1,1\}^n \to \{-1,1\}^n$ such that each bit of $N_\epsilon(x)$ is obtained by randomly flipping each bit of x independently with probability ϵ. That is, independently for each $1 \leq i \leq n$, $\Pr[N_\epsilon(x)_i = -x_i] = \epsilon$.*

Definition 3 (Noise Sensitivity) *For any Boolean function f, define its noise sensitivity, denoted by $\mathrm{NS}_\epsilon(f)$, to be*

$$\mathrm{NS}_\epsilon(f) = \Pr_{x,\text{random noise}}[f(x) \neq f(N_\epsilon(x))]$$

Note that the notion of noise operator is similar to the δ-biased distribution we saw in Håstad's test. One may think Håstad's dictator testing algorithm tests both linearity and noise sensitivity at the same time. An easy fact is, if x is uniform over $\{-1,1\}^n$ then so is $N_\epsilon(x)$.

We next see the noise sensitivities of some functions.

Fact 4 (Dictator Function) *If $f(x) = x_i$, then $\mathrm{NS}_\epsilon(f) = \epsilon$.*

Fact 5 (AND Function) *If $f(x) = x_1 \bigwedge \cdots \bigwedge x_k$, then $\mathrm{NS}_\epsilon(f) = \frac{1}{2^{k-1}}(1-(1-\epsilon)^k)$. This is because*

$$\begin{aligned}
\mathrm{NS}_\epsilon(f) &= \Pr[f(x) = -1 \textit{ and } f(N_\epsilon(x)) = 1] + \Pr[f(x) = 1 \textit{ and } f(N_\epsilon(x)) = -1] \\
&= 2\Pr[f(x) = 1 \textit{ and } f(N_\epsilon(x)) = -1] \\
&= 2\frac{1}{2^k}(1-(1-\epsilon)^k).
\end{aligned}$$

Note that for $k \ll \frac{1}{\epsilon}$, $\mathrm{NS}_\epsilon(f) \approx \frac{k\epsilon}{2^{k-1}}$. If $k \gg \frac{1}{\epsilon}$, then $\mathrm{NS}_\epsilon(f) \approx \frac{1-e^{-k\epsilon}}{2^{k-1}}$.

Fact 6 (Majority Function) *If* $f(x) = \mathrm{MAJ}(x_1, \ldots, x_n) = \mathrm{sign}(x_1 + \cdots + x_n)$*, then* $\mathrm{NS}_\epsilon(f) = O(\sqrt{\epsilon})$*.*

Sketch of Proof Here we only give a rough outline of the proof. One may think of computing the majority of x as a random walk on the real line. The random walk starts from origin and at step i it flips a fair coin to determine the value of of x_i and moves left or right accordingly. After n steps, it stops and outputs 1 if it ends at some position $z \geq 0$ and outputs -1 otherwise. A well-known fact is that the expected distance from the origin after n unbiased coin-flips is $\Theta(\sqrt{n})$. In fact, if c is a sufficiently small constant, then the probability that the random walk ends at distance from origin $\geq c\sqrt{n}$ is pretty high. One way of seeing this fact is to consider the weight distribution of vectors in the Boolean cube. Although $\sum_i x_i = 0$ is the most likely configuration, but there are only $\Theta(\frac{2^n}{\sqrt{n}})$ vectors at this point. In fact, almost all vectors are distributed between $\sum_i x_i = -\sqrt{n}$ and $\sum_i x_i = \sqrt{n}$.

Now we consider $N_\epsilon(x)$ as a second random walk starting from the endpoint of the previous walk (that is, starts from $\sum_i x_i$). This time there are only ϵn coin-flips and each coin-flip outputs 1 and -1 equally likely. Note that since we are "correcting" the previous noiseless random walk, so the step size of the second walk is 2 and consequently the expected displacement is $2\sqrt{\epsilon n}$. Suppose the first random walk ends at $c\sqrt{n}$ for some small constant c. Then by Markov inequality,

$$\begin{aligned}
&\Pr[\text{2nd walk leaves us on the other side of origin}] \\
\leq\ &\Pr[\text{the displacement of the second walk is larger than } c\sqrt{n}] \\
\leq\ &\frac{2\sqrt{\epsilon n}}{c\sqrt{n}} = O(\sqrt{\epsilon}).
\end{aligned}$$

■

In fact, it is known that this bound on the noise sensitivity of Majority functions is tight (up to a constant factor). That is, $\mathrm{NS}_\epsilon(\mathrm{MAJ}) = \Theta(\sqrt{\epsilon})$.

Fact 7 (Linear Threshold Function [Peres]) *For any linear threshold function LTF,*

$$\mathrm{NS}_\epsilon(\mathrm{LTF}) \leq 8.8\sqrt{\epsilon}.$$

Fact 8 (Parity Function) *If* $f(x) = \chi_S(x)$ *for some* $S \subseteq [n]$*, then*

$$\mathrm{NS}_\epsilon(f) = \frac{1 - (1-2\epsilon)^{|S|}}{2}.$$

This fact is a special case of the theorem we are going to prove next.

Theorem 9 *For any Boolean function* $f : \{-1,1\}^n \to \{-1,1\}$*,*

$$\mathrm{NS}_\epsilon(f) = \frac{1}{2} - \frac{1}{2}\sum_{S \subseteq [n]} (1-2\epsilon)^{|S|} \hat{f}(S)^2.$$

Proof By the definition of noise sensitivity, we have

$$\begin{aligned}
\mathrm{NS}_\epsilon(f) &= \Pr_{x, y = N_\epsilon(x)}[f(x) \neq f(y)] \\
&= \mathbb{E}[\mathbf{1}_{f(x) \neq f(y)}] \\
&= \mathbb{E}[\frac{(f(x) - f(y))^2}{4}] \quad \text{(since } f \text{ is a Boolean-valued function)} \\
&= \mathbb{E}[\frac{2 - 2f(x)f(y)}{4}]
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2} - \frac{1}{2}\mathbb{E}_{x,y}[f(x)f(y)] \\
&= \frac{1}{2} - \frac{1}{2}\sum_{S,T\subseteq[n]} \hat{f}(S)\hat{f}(T)\mathbb{E}_{x,y}[\chi_S(x)\chi_T(y)] \\
&= \frac{1}{2} - \frac{1}{2}\sum_{S\subseteq[n]} \hat{f}(S)^2\mathbb{E}_{x,y}[\chi_S(x)\chi_S(y)].
\end{aligned}
$$

Note that since $\chi_S(x)$ and $\chi_S(x)$ take values in $\{-1,1\}$, so if we let e_{x_i} (resp. e_{y_i}) denote the unit vector that has value x_i (resp. y_i) at position i and 1 at all other places, then

$$
\begin{aligned}
\mathbb{E}_{x,y}[\chi_S(x)\chi_S(y)] &= \mathbb{E}_{x,y}[\prod_{i=1}^{n} \chi_S(e_{x_i})\chi_S(e_{y_i})] \\
&= \mathbb{E}_{x,y}[\prod_{i\in S} \chi_S(e_{x_i})\chi_S(e_{y_i})] \\
&= \prod_{i\in S} \mathbb{E}_{x,y}[\chi_S(e_{x_i})\chi_S(e_{y_i})] \\
&= \prod_{i\in S} (\Pr[\chi_S(e_{x_i}) = \chi_S(e_{y_i})] - \Pr[\chi_S(e_{x_i}) \neq \chi_S(e_{y_i})]) \\
&= \prod_{i\in S} (\Pr[x_i = y_i] - \Pr[x_i \neq y_i]) \\
&= \prod_{i\in S} (1 - 2\,\mathrm{NS}_\epsilon(x_i)) \\
&= (1-2\epsilon)^{|S|}.
\end{aligned}
$$

This completes the proof of the theorem. ■

3 Noise Sensitivity vs. Fourier Concentration

The main reason that we study noise sensitivity is the following connection between noise sensitivity and Fourier concentration for Boolean functions.

Theorem 10 *Let $f : \{-1,1\}^n \to \{-1,1\}$ be a Boolean function and let $0 < \gamma < 1/2$. Then*

$$
\sum_{|S|\geq 1/\gamma} \hat{f}(S)^2 < 2.32\,\mathrm{NS}_\gamma(f).
$$

Proof

$$
\begin{aligned}
2\,\mathrm{NS}_\gamma(f) &= 1 - \sum_{S\subseteq[n]} (1-2\gamma)^{|S|}\hat{f}(S)^2 \\
&= \sum_{S\subseteq[n]} \hat{f}(S)^2 - \sum_{S\subseteq[n]} (1-2\gamma)^{|S|}\hat{f}(S)^2 \\
&= \sum_{S\subseteq[n]} (1-(1-2\gamma)^{|S|})\hat{f}(S)^2 \\
&\geq \sum_{|S|\geq 1/\gamma} (1-(1-2\gamma)^{1/\gamma})\hat{f}(S)^2 \\
&\geq \sum_{|S|\geq 1/\gamma} (1-e^{-2})\hat{f}(S)^2.
\end{aligned}
$$

Finally by numerical calculation, $\frac{2}{1-e^{-2}} < 2.32$. ■

The following is a simple corollary of Theorem 10 which says that a Boolean function f has small Fourier concentration if there is a good upper bound on the noise sensitivity of f.

Corollary 11 *Let $f : \{-1,1\}^n \to \{-1,1\}$ be a Boolean function and $\beta : [0,1/2] \to [0,1/2]$ be a real-valued function such that $\mathrm{NS}_\gamma(f) \le \beta(\gamma)$, then*

$$\sum_{|S| \ge \left(\beta^{-1}\left(\frac{\epsilon}{2.32}\right)\right)^{-1}} \hat{f}(S)^2 \le \epsilon,$$

where β^{-1} is the inverse function for function β.

4 Application: Learning Halfspaces and Intersections of Halfspaces

Now it is easy to see the following corollary by combining Fact 7 and Corollary 11:

Corollary 12 *If $f : \{-1,1\}^n \to \{-1,1\}$ is a halfspace function, then*

$$\sum_{|S| \ge O(\frac{1}{\epsilon^2})} \hat{f}(S)^2 \le \epsilon.$$

Therefore, by applying the Low Degree Algorithm to f, we see that halfspace functions can be learned with $n^{O(\frac{1}{\epsilon^2})}$ samples under the uniform distribution.

Note that the Fourier concentration bound of halfspace functions in Corollary 12 can be easily generalized to arbitrary functions that depend on k halfspace functions by upper bound the noise sensitivity of such functions. Let $h_1, \ldots, h_k$ by k arbitrary halfspace functions. Let $g : \{-1,1\}^k \to \{-1,1\}$ be any Boolean functions defined on k variables. Define $f(x) = g(h_1(x), \ldots, h_k(x))$. Then we have the following upper bound on the noise sensitivity of f.

Theorem 13

$$\mathrm{NS}_\epsilon(f) \le 8.8k\sqrt{\epsilon}.$$

Proof

$$\begin{aligned}
\mathrm{NS}_\epsilon(f) &= \Pr[g(h_1(x), \ldots, h_k(x)) \neq g(h_1(N_\epsilon(x)), \ldots, h_k(N_\epsilon(x)))] \\
&\le \sum_{i=1}^{k} \Pr[h_i(x) \neq h_i(N_\epsilon(x))] \quad \text{(By union bound)} \\
&\le k \cdot 8.8\sqrt{\epsilon}. \quad \text{(By Fact 7)}
\end{aligned}$$

■

Applying the Low Degree Algorithm again, we conclude that any function that depends on k halfspace functions can be learned with $n^{O(\frac{k^2}{\epsilon^2})}$ samples under the uniform distribution.