COMS 6998-2 Advanced Topics in Computational Learning Theory — Spring 2005

# Lecture 10 : March 31, 2005

*Lecturer: Rocco A. Servedio* — *Scribe: Hila Becker*

## 1 Lecture Overview

In this lecture we cover the following

- Finish proof of the LMN theorem by proving Facts 2 and 3 from lecture 9
- Learning from Membership Queries under the Uniform Distribution

## 2 Proof of LMN - continued

Recall that in order to complete the proof of the LMN theorem, we must show that Lemma 2(which we now rename Lemma 1) from the previous lecture holds.

**Lemma 1** *Let $f$ be any boolean function, $f : \{-1,1\}^n \rightarrow \{-1,1\}$ , and $\rho$ be a random restriction with parameter $p$. For any $t$ such that $pt \geq 8$*

$$\sum_{|S|>t} \hat{f}(S)^2 \leq 2 \Pr_{\rho}\left[\exists A : live(A) \geq \frac{pt}{2} \text{ and } \hat{f}_{\rho}(A) \neq 0\right]$$

The proof of this lemma follows from 3 facts.

**Fact 1**

$$\sum_{A:|A|>t} \hat{f}(A)^2 \leq 2E_S\left[\sum_{A:|A\cap S|>(pt)/2} \hat{f}(A)^2\right],$$

The proof of this fact was given in lecture 9.

**Fact 2** *Let $S$ be chosen from $[n]$ by independently including each $i \in [n]$ with probability $p$. Let $f$ be a boolean function and let $\rho$ be a $p$-biased restriction.*

$$E\left[\sum_{A:|A\cap S|\geq \frac{pt}{2}} \hat{f}^2(A)\right] = E_\rho\left[\sum_{A:live(A)\geq \frac{pt}{2}} \hat{f}_\rho^2(A)\right]$$

**Fact 3** *Let $f : \{-1,1\}^n \to \{-1,1\}$ be a boolean function, let $\rho$ be a random restriction with parameter $p$.*

$$E_\rho\left[\sum_{A:live(A)\geq \frac{pt}{2}} \hat{f}_\rho^2(A)\right] \leq \Pr_\rho\left[\exists A : live(A) \geq \frac{pt}{2} \text{ and } \hat{f}(A) \neq 0\right]$$

We first present a proof of Fact 3:

**Proof:** Given a restriction $\rho$ , define the indicator random variable

$$Ind(\rho) = \begin{cases} 1 & \exists A : live(A) \geq \frac{pt}{2}, \hat{f}(A) \neq 0 \\ 0 & otherwise \end{cases}$$

We can therefore rewrite the probability as

$$\Pr_\rho\left[\exists A : live(A) \geq \frac{pt}{2} \text{ and } \hat{f}(A) \neq 0\right] = \sum_\rho \Pr[\rho] \cdot Ind(\rho)$$

Fix any $\rho$. If $\rho$ is such that $Ind(\rho) = 1$, then we have

$$0 \leq \sum_{live(A)\geq \frac{pt}{2}} \hat{f}_\rho^2(A) \leq 1,$$

therefore,

$$Ind(\rho) = 1 = \sum_A \hat{f}^2(A) \geq \sum_{live(A)\geq \frac{pt}{2}} \hat{f}_\rho^2(A).$$

If $\rho$ is such that $Ind(\rho) = 0$ we have $\sum_{live(A)\geq \frac{pt}{2}} \hat{f}_\rho^2(A) = 0$ since by definition, $\nexists A$ such that $live(A) \geq \frac{pt}{2}$ and $\hat{f}(A) \neq 0$.
Since we showed that for any restriction $\rho$

$$Ind(\rho) \geq \sum_{live(A)\geq \frac{pt}{2}} \hat{f}_\rho^2(A),$$

the fact follows. ■

To complete the proof of the LMN theorem we now only need to prove Fact 2.

**Definition 1** *Let $f : \{-1,1\}^n \to \{-1,1\}$. Fix $0 \le K \le n$ and let $S \subseteq [K]$. Define $f_{K,S_1} : \{-1,1\}^{n-K} \to \mathbb{R}$ as*

$$f_{K,S_1}(x) = \sum_{T_2 \subseteq \{K+1,\ldots,n\}} \hat{f}(S_1 \cup T_2)\chi_{T_2}(x),$$

**Example 1**

- *If $K = 0$ then $f_{0,\emptyset} = \sum_{T_2 \subseteq [n]} \hat{f}(T_2)\chi_{T_2}(x) = f(x)$*
- $f_{1,\emptyset}(x) = \sum_{T \subseteq [n], 1 \notin T} \hat{f}(T)\chi_T(x)$
- $f_{1,\{1\}}(x) = \sum_{T \subseteq [n], 1 \in T} \hat{f}(T)\chi_T(x)$
- $f_{1,\emptyset}(x) + f_{1,\{1\}}(x) = f(x)$
- $f_{n,S_1}(x) = \hat{f}(S_1)$

*Note how this function creates a partition of the fourier spectrum.*

**Lemma 2** *Let $f : \{-1,1\}^n \to \{-1,1\}$. Fix a value $k \in \{0,1,\ldots,n\}$ and fix $S_1 \in \{1,\ldots,k\}$. For any $x = x_{k+1}\ldots x_n \in \{-1,1\}^{n-k}$ we have*

$$f_{k,S_1}(x) = E_{y \in \{-1,1\}^k}\left[f(yx)\chi_{S_1}(y)\right]$$

**Proof:** We have that $f(xy) = \sum_T \hat{f}(T)\chi_T(yx)$. Note that if $T = T_1 \cup T_2$ with $T_1 = \{1,\ldots,K\}$ and $T_2 = \{K+1,\ldots,n\}$, then $\chi_T(yx) = \chi_{T_1}(y) \cdot \chi_{T_2}(x)$.

Therefore,

$$E_{y \in \{-1,1\}^k}\left[f(yx)\chi_{S_1}(y)\right] = E_y\left[\left(\sum_{T_1}\sum_{T_2} \hat{f}(T_1 \cup T_2)\chi_{T_1(y)}\chi_{T_2}(x)\right)\chi_{S_1}(y)\right]$$

$$= \sum_{T_1}\sum_{T_2} \hat{f}(T_1 \cup T_2)\chi_{T_2}(x) E_y\left[\chi_{T_1}(y)\chi_{S_1}(y)\right].$$

We can write $E_y\left[\chi_{T_1(y)}\chi_{S_1}(y)\right] = E_y\left[\chi_{S_1\Delta T_1}(y)\right].$

By the orthonormality of the basis, it follows that

$$E_y\left[\chi_{T_1(y)}\chi_{S_1}(y)\right] = \begin{cases} 1 & if \quad S_1 = T_1 \\ 0 & if \quad S_1 \neq T_1 \end{cases}$$

so only the terms with $S_1 = T_1$ contribute to the sum.

So

$$\sum_{T_1}\sum_{T_2}\hat{f}(T_1 \cup T_2)\chi_{T_2}(x)E_y\left[\chi_{T_1}(y)\chi_{S_1}(y)\right] = \sum_{T_2}\hat{f}(S_1 \cup T_2)\chi_{T_2}(x) = f_{K,S_1}$$

■

Note that $\forall K, S_1, x : f_{K,S_1}(x) \in [-1,1]$

**Definition 2** *Given $S \subseteq [n]$ and an assignment $x \in \{-1,1\}^n$,*
*let $f_{\bar{S}\leftarrow x}$ denote the function $f_\rho$ where $\rho$ is a restriction with live$(\rho) = S$ and the other variables in $\bar{S}$ are assigned a value according to x.*

**Example 2** $n = 5, S = \{1,2,3\}, x = (-1,1,-1,1,-1)$

$f_{\bar{S}\leftarrow x}(y) = f(y_1, y_2, y_3, 1, -1)$

**Claim 1** *Fix $S \subseteq [n]$, $S_1 \subseteq S$. Then*

$$\sum_{T_2\subseteq\bar{S}}\hat{f}^2(S_1 \cup T_2) = E_x\left[\hat{f}^2_{\bar{S}\leftarrow x}(S_1)\right]$$

**Proof:** WLOG, we may assume $S = \{1,\ldots,k\}$. Let $f_{k,S_1} = \sum_{T_2\in\bar{S}}\hat{f}(S_1 \cup T_2)\chi_{T_2(x)}$. Using Parseval's identity we get

$$E_x\left[f_{k,S_1}(x)^2\right] = \sum_{T_2\subseteq\bar{S}}\hat{f}^2(S_1 \cup T_2).$$

From Lemma 2 we have that $f_{k,S_1}(x) = E_y\left[f(yx)\chi_{S_1}(y)\right]$. so

$$\sum_{T_2\subseteq\bar{S}}\hat{f}^2(S_1 \cup T_2) = E_x\left[\left(E_y\left[f(yx)\chi_{S_1}(y)\right]\right)^2\right]$$

We only need to show that

$$E_y\left[f(yx)\chi_{S_1}(y)\right] = \hat{f}_{\bar{S}\leftarrow x}(S_1)$$

This holds since we know that $f_{\bar{S}\leftarrow x}$ has variables $y_1,\ldots,y_k$ and the last $n-1$ bits are set according to $x$. Therefore, $\hat{g}(S_1) = E\left[g(y)\chi_{S_1}(y)\right]$.

■

We now present the proof of Fact 2.

**Proof:** Fix $S, A \subseteq S$. We have that

$$\sum_{T_2\subseteq\bar{S}} \hat{f}^2(A\cup T_2) = E_x\left[\hat{f}^2_{\bar{S}\leftarrow x}(A)\right].$$

Summing over all $A \subseteq S$ with $|A| \geq l$ we get

$$\sum_{A\subseteq S,|A|\geq l}\sum_{T_2\subseteq\bar{S}} \hat{f}^2(A\cup T_2) = \sum_{A\subseteq S,|A|\geq l} E_x\left[\hat{f}^2_{\bar{S}\leftarrow x}(A)\right]$$

Since this is true for all $S$, we can let $S$ be random so averaging over S maintains the identity

$$E_S\left[\sum_{A\subseteq S,|A|\geq l}\sum_{T_2\subseteq\bar{S}} \hat{f}^2(A\cup T_2)\right] = E_S\left[\sum_{A\subseteq S,|A|\geq l} E_x\left[\hat{f}^2_{\bar{S}\leftarrow x}(A)\right]\right]$$

Now we have that

$$\sum_{A\subseteq S,|A|\geq l}\sum_{T_2\subseteq\bar{S}} \hat{f}^2(A\cup T_2) = E_S\left[\sum_{A':|A\cap S|\geq l} \hat{f}(A')\right] \tag{1}$$

and since $E_S E_x = E_\rho$,

$$\sum_{A\subseteq S,|A|\geq l} E_x\left[\hat{f}^2_{\bar{S}\leftarrow x}(A)\right] = E_S\left[E_x\left[\hat{f}^2_{\bar{S}\leftarrow x}(A)\right]\right] = E_\rho\left[\sum_{live(A)\geq l} \hat{f}^2(A)\right] \tag{2}$$

Setting $l = \frac{pt}{2}$, the claim follows from Equations 1 and 2 .

■

- (LMN) We can learn $poly(n)$-size, depth-d AND/OR/NOT circuits in $n^{(logn)^d}$
- (Kharitonov) If factoring integers is ”very hard” ($2^{n^{\varepsilon}} - hard$ for some fixed $\varepsilon > 0$), then any algorithm for learning $poly(n)$-size, depth-d circuits must take time $n^{logn^{\Omega(d)}}$
- (Richer) Circuit classes are also learnable: it is known that circuits of $poly(n)$-size, $O(1)$-depth with AND/OR/NOT and $polylog(n)$-many MAJ gates (single gate that computes the majority function) are also learnable in $n^{polylog(n)}$ time.

# 3 Learning with Membership Queries under the Uniform Distribution

We present a new learning model which uses Membership Queries for learning under the uniform distribution. Given a hypothesis $h : \{-1,1\}^n \to \{-1,1\}$, define the error of h, $error(h, f) = \Pr_{x \in \mathcal{U}}[h(x) \neq f(x)]$. Our goal is to find a hypothesis which is $\varepsilon-$accurate with probability $1-\delta$. In this learning model we get a black-box access to a membership query oracle $MQ(f)$. Given $x \in \{-1,1\}^n$, $MQ(f)$ outputs f(x). $MQ(f)$ can simulate $EX(f)$ by picking x uniformly at random and feeding it into $MQ(f)$. Therefore, this new learning model is no-less powerful than the model using a random example oracle.

**Fact 4** *If one-way functions exist, there are concept classes that are poly-n time learnable with $MQ(f)$ but not with $Ex(f)$*

Concept classes such as DNF and Decision Trees are known to be polynomial time learnable with $MQ(f)$ and not with $EX(f)$.

## 3.1 Learning Juntas

**Definition 3** *A boolean function $f : \{-1,1\}^n \to \{-1,1\}$ is said to be a $r-junta$ if f only depends on some particular set of r variables.*

**Example 3** $f(x_1 \dots x_n) = x_{917} \vee (x_3 \wedge \bar{x_5})$ *is a 3-junta*

Let $C_r$ = $\{$all r-juntas$\}$. $|C_r| \leq 2^{2^r}\binom{n}{r}$. This is not so bad when $r = log(n)$, since any $log(n)$-junta has an n-leaf Decision Tree. We can learn $r$-juntas in time $\approx \binom{n}{r}2^r \approx n^r$ by trying all sets of r variables.

**Open Questions**

- Are there $EX(f)$ algorithms to learn r-juntas in time substantially better than $n^r$? We know that we can learn in time $n^{.704}$.
- Can we learn $r = log(n) - juntas$ in $poly(n)$ time? ($1000 prize offered by Avrim Blum).

**Theorem 1** *Any $r - junta$ is learnable in $poly(2^r, n) - time$ with $MQ(f)$.*

In fact, we can learn any $r - junta$ exactly, with probability $1 - \delta$ output the relevant variables and truth table for $f$.

**Sketch of Proof** Draw $poly(2^r)$ random examples. Note that half of the time we will output r due to the uniform distribution.

If every label is 1, output 1. If every label is -1, output -1. Otherwise, we have two examples $(x, 1)$ and $(y, -1)$. Convert x to y by changing one bit per step where x and way disagree, and giving the new assignment to $MQ(f)$. If the value of $f$ changes after changing the assignment to the $k$'s bit, we know that the $k$'th variable is a relevant variable for the junta. When a relevant variable is found, recurse to learn each of the two induced $(r - 1) - juntas$ by fixing the relevant variable in both possible ways. In the worst case, we have to solve at most $2^r$ sub-problems, where each sub-problem is solvable in $poly(2^r, n) - time$.

■