Analysis of Boolean Functions **(CMU 18-859S, Spring 2007)**

Lecture 9: Learning Decision Trees and DNFs

Feb. 18, 2007

Lecturer: Ryan O'Donnell *Scribe: Suresh Purini*

1 Two Important Learning Algorithms

We recall the following definition and two important learning algorithms discussed in previous lecture.

Definition 1.1 *Given a collection $\mathcal{S}$ of subsets of $[n]$, we say $f : \{-1,1\}^n \to \mathbb{R}$ has ϵ-concentration on $\mathcal{S}$, if*

$$\sum_{S \notin \mathcal{S}} \hat{f}(S)^2 \leq \epsilon.$$

Theorem 1.2 *Let $\mathcal{C}$ be a class of n-bit functions, such that $\forall f \in \mathcal{C}$, f is ϵ-concentrated on $\mathcal{S} = \{S \subseteq [n] | \, |S| \leq d\}$, then the function class $\mathcal{C}$ is learnable under the uniform distribution to an accuracy of $O(\epsilon)$, with a probability of at least $1-\delta$, in time $\mathrm{poly}(|\mathcal{S}|, 1/\epsilon)\mathrm{poly}(n)\log(1/\delta)$ using random examples only.*

This algorithm is called Low Degree algorithm and was proposed by Linial, Mansour and Nisan in [3]. Refer theorem 5.4 in lecture notes 8.

Theorem 1.3 *Let $\mathcal{C}$ be a class of n-bit functions, such that $\forall f \in \mathcal{C}$, f is ϵ-concentrated on some collection $\mathcal{S}$. Then the function class $\mathcal{C}$ is learnable using membership queries (Goldreich-Levin Algorithm) in $\mathrm{poly}(|\mathcal{S}|, 1/\epsilon)\mathrm{poly}(n)\log(1/\delta)$ time.*

This algorithm is called Kushilevitz-Mansour algorithm [2]. Refer corollary 5.5 in lecture notes 8.

2 Learning Decision Trees

A decision tree is a binary tree in which the internal nodes are labeled with variables and the leafs are labeled with either -1 or $+1$. And the left and right edges corresponding to any internal node is labeled -1 and $+1$ respectively. We can think of the decision tree as defining a boolean function in the natural obvious way. For example, the decision tree in the figure 1 defines a boolean function whose DNF formula is $x_1x_2x_3 + x_1\bar{x_2}x_4 + \bar{x_1}x_2$.

Note that, given any boolean function we can come up with a corresponding decision tree.

Let P be a path in the decision tree. An example of a path in the figure 1 is $P = (x_1 = -1, x_2 = +1, x_4 = -1)$.

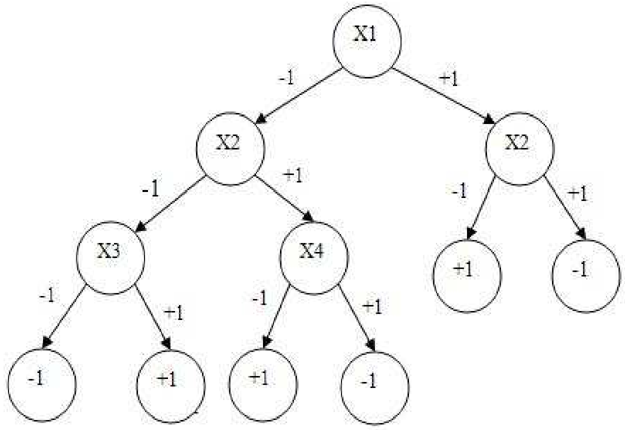

Figure 1:

Let $\mathbf{1}_P : \{-1,1\}^n \rightarrow \{0,1\}$ be an indicator function for path P. For example,

$$\mathbf{1}_P = \begin{cases} 1 & \text{if } x_1 = -1, x_2 = +1, x_4 = -1 \\ 0 & \text{else} \end{cases}$$

Observation 2.1 *A boolean function f can be expressed in terms of path functions $\mathbf{1}_P$'s, corresponding to various paths in the decision tree of the function f as follows*

$$f(x) = \sum_{Paths\ P} \mathbf{1}_P(x) f(P)$$

where $f(P)$ is the label on the leaf when the function f takes the path P in its decision tree.

Observation 2.2 *Let V be the set of variables occurring in a path function $\mathbf{1}_P$ and d be the cardinality of the set V. Then the Fourier expansion of $\mathbf{1}_P$ looks like*

$$\sum_{S \subseteq V} \pm 2^{-d} X_S.$$

It is easy to see the proof of the above observation by noting that the Fourier expansion for the path function $\mathbf{1}_P$, when $P = (x_1 = -1, x_2 = +1, x_4 = -1)$, is $\mathbf{1}_P = x_1 \bar{x_2} x_4 = (\frac{1}{2} - \frac{1}{2}x_1)(\frac{1}{2} + \frac{1}{2}x_2)(\frac{1}{2} - \frac{1}{2}x_4)$.

Proposition 2.3 *If $f : \{-1,1\}^n \rightarrow \{-1,1\}$ is computable by a depth-d decision tree then*

1. *Fourier expansion of f has degree at most d i.e., $\sum_{|S|>d} \hat{f}(S)^2 = 0$.*

2. *All Fourier coefficients are integer multiples of 2^{-d}.*

3. *The number of nonzero Fourier coefficients is at most 4^d.*

Proof:(1) follows from observation 2.1. We can observe that all the Fourier coefficients look like $k2^{-d'}$ for some $d' \leq d$, which can be written as $k2^{d+d'}2^{-d}$. This proves (2). A depth-d decision tree has at most 2^d leaves and hence we have at most $2^d \cdot 2^d = 4^d$ Fourier coefficients, which proves (3). □

Corollary 2.4 *Depth-d decision trees are exactly learnable with random examples in time* $\mathrm{poly}(4^d)\mathrm{poly}(n)\log(1/\delta)$.

Proof: Use Kushilevitz-Mansour algorithm, with $\epsilon = \frac{2^{-d}}{4}$ and round each Fourier coefficient estimate to the nearest multiple of 2^{-d}. □

Remark 2.5 $\log(n)$*-depth decision trees are exactly learnable in polynomial time. This algorithm can be derandomized.*

Observation 2.6 *Size-*s *decision trees are* ϵ*-close to a depth* $\log(s/\epsilon)$ *decision trees.*

Proof: Let T be decision tree of size s corresponding to boolean function f. Consider the decision T' obtained from T by chopping all paths whose depth is greater than $\log(\frac{s}{\epsilon})$ to $\log(\frac{s}{\epsilon})$. The decision tree T' gives an incorrect value for $f(X)$ only when X takes a path of length greater than $\log(\frac{s}{\epsilon})$ in T. When we pick X at random, this happens with probability $2^{-\log(\frac{s}{\epsilon})} = \frac{\epsilon}{s}$. Therefore by union bound, we get that $\mathbf{Pr}_{\mathbf{x}\in\{-1,1\}^n}[T(\mathbf{x}) \neq T'(\mathbf{x})] \leq \epsilon$.

□

Corollary 2.7 *Size-*s *decision trees are* $O(\epsilon)$*-concentrated on a collection of size size* $4^{\log(s/\epsilon)} = (s/\epsilon)^2$.

Definition 2.8 *Given a function* $f : \{-1,1\}^n \to \mathbb{R}$*, the spectral norm or* L_1*-Fourier norm of* f *is*

$$||\hat{f}||_1 = \sum_{S\subseteq[n]} |\hat{f}(S)|$$

Observation 2.9 *If a function* f *is an AND of literals, then* $||\hat{f}||_1 = 1$*. Refer observation 2.2 for the proof idea.*

The following observation follows from the fact $\forall a, b \in \mathbb{R}$, $|a+b| \leq |a| + |b|$ and $|ab| = |a||b|$.

Observation 2.10

1. $||\widehat{f+g}||_1 \leq ||\hat{f}||_1 + ||\hat{g}||_1$
2. $||\widehat{cf}||_1 = |c|||\hat{f}||_1$

Proposition 2.11 *If f has a decision tree of size s, $||\hat{f}||_1 \leq s$.*

Proof:

$$
\begin{aligned}
||\hat{f}||_1 &\leq \sum_{Paths\ P} \widehat{\mathbf{1}_P f(P)} \\
&\leq \sum_{Paths\ P} \hat{\mathbf{1}}_P \\
&\leq s
\end{aligned}
$$

$\square$

Proposition 2.12 *Given any function f with $||f||_2{}^2 \leq 1$ and $\epsilon > 0$, $\mathcal{S} = \{S \subseteq [n] || \hat{f}(S)| \geq \frac{\epsilon}{||\hat{f}||_1}\}$, then f is ϵ-concentrated on $\mathcal{S}$. Note that $|\mathcal{S}| \leq \left(\frac{||\hat{f}||_1}{\epsilon}\right)^2$.*

Proof:

$$
\begin{aligned}
\sum_{S \notin \mathcal{S}} \hat{f}(S)^2 &\leq max_{S \notin \mathcal{S}} |\hat{f}(S)| \left[\sum_{S \notin \mathcal{S}} |\hat{f}(S)|\right] \\
&\leq max_{S \notin \mathcal{S}} |\hat{f}(S)| \left[\sum_{S \notin \mathcal{S}} |\hat{f}(S)| + \sum_{S \in \mathcal{S}} |\hat{f}(S)|\right] \\
&\leq \frac{\epsilon}{||\hat{f}||_1} \cdot ||\hat{f}||_1 \\
&\leq \epsilon
\end{aligned}
$$

$\square$

Corollary 2.13 *Any class of functions $\mathcal{C} = \{f | \ \ ||f||_2{}^2 \leq 1$ and $||\hat{f}||_1 \leq s\}$ is learnable with random examples in time $\mathrm{poly}(s, \frac{1}{\epsilon})$.*

Let us now consider functions which are computable by decision trees where nodes branch on arbitrary parities of variables. Figure 2 contains an example of a function computable by decision tree on the parity of the various subsets of variables. Another example is parity function which is computable by a depth-1 parity decision tree.

Proposition 2.14 *If a function $f : \{-1,1\}^n \rightarrow \{-1,1\}$ is expressible as a size-s decision tree on parities, then $||\hat{f}||_1 \leq s$.*

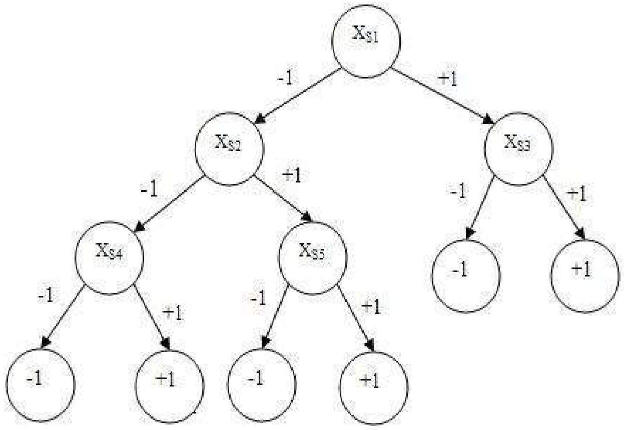

Figure 2:

Proof: Let $\mathbf{1}_P$ be an $\{0,1\}$-indicator function for a path P in the decision tree. Let the path $P = (X_{S_1} = b_1, \cdots, X_{S_d} = b_d)$, i.e., we get the path P by taking the edges labeled $b_1, \cdots, b_d \in \{-1,1\}$ starting from the root node. We have

$$\mathbf{1}_P = (\frac{1}{2} + \frac{1}{2}b_1 X_{S_1}) \cdots (\frac{1}{2} + \frac{1}{2}b_d X_{S_d})$$

It can be seen that $||\widehat{\mathbf{1}_P}||_1 = 1$. Since $f(x) = \sum_{Paths\ P} \mathbf{1}_P(x) f(P)$, we have $||\hat{f}||_1 \leq s$. $\square$

Definition 2.15 *An AND of parities is called a coset.*

Remark 2.16 *If a function $f : \{-1,1\}^n \rightarrow \{-1,1\}$ is expressible as $\sum_{i=1}^{s} \pm \mathbf{1}_{P_i}$, where P_i's are cosets then $||\hat{f}||_1 \leq s$.*

Remark 2.17 *Proposition 2.14 implies that we can learn all parity functions in $\mathrm{poly}(\frac{1}{\epsilon})$ time. Observe that we cannot see this result straightforward from the usual decision trees on parity functions.*

Theorem 2.18 *[1] If a function $f : \{-1,1\}^n \rightarrow \{-1,1\}$ with $||\hat{f}||_1 \leq s$, then*

$$f = \sum_{i=1}^{2^{2^{O(s^4)}}} \pm \mathbf{1}_{P_i}$$

where P_i's are cosets.

3 Learning DNFs

Proposition 3.1 *If f has a size-s DNF formula, it is ϵ-close to a width-$\log(\frac{s}{\epsilon})$ DNF.*

Proof: Let the function $f : \{-1,1\}^n \to \{-1,1\}$ has a size-s DNF. Drop all the terms whose width is larger than $\log(\frac{s}{\epsilon})$ from the DNF of f and let the new DNF represents the function f'. If we look at a particular term in the DNF of f whose width is greater than $\log(\frac{s}{\epsilon})$, then the probability that a randomly chosen $x \in \{-1,1\}$ sets it to -1(or 1 if we look at f as boolean function from $\{0,1\}^n$ to $\{0,1\}$) is at most $2^{-\log(\frac{s}{\epsilon})} = \frac{\epsilon}{s}$. Since there are at most s terms in the DNF, we have that $\mathbf{Pr}_{\mathbf{x}}[f(\mathbf{x}) \neq f'(\mathbf{x})] \leq \epsilon$ by union bound. $\square$

Proposition 3.2 *If a function $f : \{-1,1\}^n \to \{-1,1\}$ has a width w DNF, then $\mathbb{I}(f) \leq 2w$.*

Proof: Left as an exercise. $\square$

Corollary 3.3 *If a function $f : \{-1,1\}^n \to \{-1,1\}$ has a width w DNF, then f is ϵ-concentrated on a $\mathcal{S} = \{S \mid |S| \leq \frac{2w}{\epsilon}\}$. Thus the function f can be learnable in $n^{O(\frac{w}{\epsilon})}$.*

In the rest of the class, we shall prove the following theorem making use of Hastad's switching lemma.

Theorem 3.4 *DNF's of width w are ϵ-concentrated on degree up to $O(w\log(\frac{1}{\epsilon}))$.*

Remark 3.5 *Observe that we are replacing the $\frac{1}{\epsilon}$-factor with $\log(\frac{1}{\epsilon})$-factor on the maximum degree of the Fourier coefficients.*

Definition 3.6 *A random restriction with $*$-probability ρ on $[n]$ is a random pair $(\mathbf{I},\mathbf{X})$ where $\mathbf{I}$ is a random subset of $[n]$ chosen by including each coordinate with probability ρ independently and $\mathbf{X}$ is a random string from $\{-1,1\}^{|\bar{\mathbf{I}}|}$.*

Given a function $f : \{-1,1\}^n \to \{-1,1\}$, we shall write $f_{\mathbf{X}\to\bar{\mathbf{I}}} : \{-1,1\}^{|\mathbf{I}|} \to \mathbb{R}$ for a restriction of f. If the function f is computable by a width w DNF, then after a random restriction with $*$-probability $\rho = \frac{1}{10w}$, with very high probability, $f_{\mathbf{X}\to\bar{\mathbf{I}}} : \{-1,1\}^{|\mathbf{I}|} \to \mathbb{R}$ has a $O(1)$-depth decision tree. The reason for this is intuitively that in each term of the DNF, $\frac{1}{10}$ variables survive the random restriction on an average. Thus resulting in a a constant depth decision tree. This intuition is formalized in the following lemma due to Hastad.

Theorem 3.7 *(Hastad's Switching Lemma) Let $f : \{-1,1\}^n \to \{-1,1\}$ be a width w computable DNF. When we apply a random restriction on the function f with $*$-probability ρ, then*

$$\mathop{\mathbf{Pr}}_{(\mathbf{I},\mathbf{X})}[\textit{DT-depth}(f_{\mathbf{X}\to\bar{\mathbf{I}}}) > d] \leq (5\rho w)^d$$

Theorem 3.8 *Let f be computable by a width-w DNF. Then $\forall d \geq 5$,*

$$\sum_{|U|\geq 20dw} \hat{f}(u)^2 \leq 2^{-d+1}.$$

Proof: Let $(\mathbf{I}, \mathbf{X})$ be a random restriction with $\rho = \frac{1}{10w}$. We know from Hastad's switching lemma $f_{\mathbf{X}\to\bar{\mathbf{I}}}$ has a depth greater than d with a probability less than 2^{-d}. Hence the following sum is nonzero (and less than 1) with a probability less than 2^{-d}.

$$\sum_{S\subseteq I, |S|>d} \hat{f}_{\mathbf{X}\to\bar{\mathbf{I}}}(S)^2$$

Therefore, we have

$$\begin{aligned}
2^{-d} &\geq \mathop{\mathbf{E}}_{(\mathbf{X},\mathbf{I})}\left[\sum_{\substack{S\subseteq I\\ |S|>d}} \hat{f}_{\mathbf{X}\to\bar{\mathbf{I}}}(S)^2\right] \\
&= \mathop{\mathbf{E}}_{\mathbf{I}}\left[\mathop{\mathbf{E}}_{\mathbf{X}\in\{-1,1\}^{|\bar{\mathbf{I}}|}}\left[\sum_{\substack{S\subseteq \mathbf{I}\\ |S|>d}} \hat{f}_{\mathbf{X}\to\bar{\mathbf{I}}}(S)^2\right]\right] \\
&= \mathop{\mathbf{E}}_{\mathbf{I}}\left[\sum_{\substack{S\subseteq \mathbf{I}\\ |S|>d}} \mathop{\mathbf{E}}_{\mathbf{X}\in\{-1,1\}^{|\bar{\mathbf{I}}|}}\left[F_{S\subseteq\mathbf{I}}(\mathbf{X})^2\right]\right] \text{(Recall } F_{S\subseteq I}(x) = \hat{f}_x(S)) \\
&= \mathop{\mathbf{E}}_{\mathbf{I}}\left[\sum_{\substack{S\subseteq \mathbf{I}\\ |S|>d}} \sum_{T\subseteq\bar{\mathbf{I}}} \widehat{F}_{S\subseteq\mathbf{I}}(T)^2\right] \\
&= \mathop{\mathbf{E}}_{\mathbf{I}}\left[\sum_{\substack{S\subseteq \mathbf{I}\\ |S|>d}} \sum_{T\subseteq\bar{\mathbf{I}}} \hat{f}(S\cup T)^2\right] \\
&= \sum_U \hat{f}(U)^2 \mathop{\mathbf{Pr}}_{\mathbf{I}}\left[|U\cap\mathbf{I}| > d\right]
\end{aligned}$$

Suppose $|U| \geq 20dw$, then $|U\cap\mathbf{I}|$ is binomially distributed with mean $20dw\rho = 2d$. Using Chernoff bound, we get that $\mathbf{Pr}_{\mathbf{I}}\left[|U\cap\mathbf{I}| > d\right] \leq \frac{1}{2}$, when $d \geq 5$. Therefore we have the

$$\begin{aligned}
\sum_U \hat{f}(U)^2 \mathop{\mathbf{Pr}}_{\mathbf{I}}\left[|U\cap\mathbf{I}| > d\right] &\leq 2^{-d} \\
\sum_{\substack{U\\ |U|\geq 20dw}} \hat{f}(U)^2\frac{1}{2} &\leq 2^{-d} \\
\sum_{\substack{U\\ |U|\geq 20dw}} \hat{f}(U)^2 &\leq 2^{-d+1}
\end{aligned}$$

□

Remark 3.9 *By putting* $dw = w\log\left(\frac{1}{\epsilon}\right)$*, we get the theorem 3.4*

Further References Yishay Mansour's survey paper[4] also contains some of the ideas in this lecture notes.

References

[1] B. Green and T. Sanders. A quantitative version of the idempotent theorem in harmonic analysis. *ArXiv Mathematics e-prints*, Nov. 2006.

[2] E. Kushilevitz and Y. Mansour. Learning decision trees using the fourier spectrum. In *STOC '91: Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 455–464, New York, NY, USA, 1991. ACM Press.

[3] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, fourier transform, and learnability. *J. ACM*, 40(3):607–620, 1993.

[4] Y. Mansour. Learning boolean functions via the fourier transform. In V. Roychowdhury, K.-Y. Siu, and A. Orlitsky, editors, *Theoretical Advances in Neural Computation and Learning*. Kluwer, 1994.