CS 880: Advanced Complexity Theory 2/13/2008

Lecture 9: Learning DNFs

Instructor: Dieter van Melkebeek Scribe: Matt Anderson

The previous two lectures dealt with learning concepts with respect to the uniform distribution using harmonic analysis. The idea has been to leverage harmonic analysis to learn concepts whose Fourier coefficients are concentrated on a small part of the spectrum. We have considered two different situations regarding the set of large Fourier coefficients $\mathcal{S}$:

- If $\mathcal{S}$ is known we can learn the concept using random samples.
- If $\mathcal{S}$ is unknown but bounded ($\mathcal{S} \leq M$) we can learn the concept using membership queries.

We have applied the above approaches to learning decision trees. Now we will apply them to DNFs and more generally to constant depth-d circuits. Table 1 below summarizes the best known times to learn various types of concepts using samples and membership queries.

	Samples	Queries
Decision Trees	$N^{O(\log N)}$	$\text{poly}(N)$
DNFs	$N^{O(\log N)}$	$\text{poly}(N)$
Depth-d Circuits	$N^{O(\log^{d-1} N)}$	$N^{O(\log^{d-1} N)}$

Table 1: Best known running times for learning with $N = (s + n)$, where s denotes the size and n the number of variables.

Note that every decision tree of size s can be written as a DNF of size ns (by forming disjunctions of paths to the leaves represented by conjunctions of variables along the path). DNFs form a larger class of concepts than decision trees. The query algorithm for DNFs described in this lecture will run in time $N^{O(\log \log N)}$ rather than $\text{poly}(N)$. The faster algorithm listed in the table is known as the harmonic sieve; it requires a technique from machine learning known as boosting in addition to harmonic analysis similar to the analysis performed in this lecture.

1 Bounding the Width

In our study of decision trees we first considered learning depth-d trees. Next, we saw how to approximate a size s tree by a depth-d tree while introducing an error of at most $s/2^d$. Combining these results we were able to learn a size s decision tree by applying our depth-d tree learning algorithm. We can use a similar paradigm for constant depth circuits.

Claim 1. *Depth-d circuits of size s can be approximated to within ϵ by a width-w depth-d circuit of size at most s, where $w = \log(s/\epsilon)$.*

Proof. Width-w circuits are circuits with bottom fan-in at most w. Transform a depth-d size s circuit C into a depth-d size $\leq s$ width-w circuit C' by ignoring all but the first w inputs to each

bottom gate. Consider a bottom OR gate. Ignore all but the first w inputs to this gate. Then the chance that that gate produces an incorrect result with respect to the uniform distribution is

$$\Pr_x[\text{Error}] \leq \frac{1}{2^w},$$

since all of the first w variables must be set to 0 for an error to occur. A similar argument gives an identical bound for AND gates. By a union bound the chance that C' errs is at most $\frac{s}{2^w}$. For an overall error of at most ϵ, set $w = \log(\frac{s}{\epsilon})$. □

Using an argument similar to those we have used in the past two lectures, it is sufficient to focus on learning circuits which have width-$\log(\frac{s}{\epsilon})$.

2 Random Restrictions

We can show that small circuits of small width have small influence using a result known has the switching lemma. From the previous lecture we know that we can learn classes of concepts with small influence. Together this gives us an algorithm for learning small width circuits.

The switching lemma makes use of random restrictions to simplify circuits. A *random restriction* R is obtained by leaving each variable independently with probability ρ and setting the other ones uniformly at random. The restriction can be specified as $R = (I, v)$, where $I \subseteq [n]$ and $v \in \{-1, 1\}^{|I|}$. I denotes the set of variables that are fixed by the assignment v. Each $i \in [n]$ is put in $\bar{I}$ with probability ρ independently and the assignment, v, to the fixed variables x_i, $i \in I$, is chosen uniformly at random. Let $f|_R$ be the result of applying the restriction R to a function f, which is now a function of only the variables in $\bar{I}$.

The following switching lemma allows us to derive a bound on the decision tree complexity of a randomly restricted function ($D(f|_R)$):

Lemma 1 (Switching Lemma)**.** *If f is a DNF of width w then for all Δ, $\Pr_R[D(f|_R) > \Delta] \leq (5\rho w)^{\Delta}$.*

We do not prove this lemma here; it follows from a tricky induction argument.

The Switching Lemma shows that random reductions reduce the decision tree complexity of f with high probability. This will concentrate the Fourier spectrum onto sets of size at most Δ:

$$\mathrm{E}_R\left[\sum_{S_1 \subseteq \bar{I}, |S_1| > \Delta} (\hat{f|_R}(S_1))^2\right] \leq 0 \cdot \Pr[D(f|_R) \leq \Delta] + 1 \cdot \Pr[D(f|_R) > \Delta] = \Pr[D(f|_R) > \Delta].$$

This is because a decision tree of depth at most Δ has no weight on sets larger than Δ.

Let x_1 be the part of x not restricted by R. Let $\chi_S(x_1, v)$ be the character resulting from

partitioning the input variables into $\bar{I}$ and I respectively. Then

$$\begin{aligned} f|_R(x_1) &= \sum_{S\subseteq[n]} \hat{f}(S)\chi_S(x_1, v) \\ &= \sum_{S_1\subseteq\bar{I}} \sum_{S_2\subseteq I} \hat{f}(S_1 \cup S_2)\chi_{S_1\cup S_2}(x_1, v) \\ &= \sum_{S_1\subseteq\bar{I}} \sum_{S_2\subseteq I} \hat{f}(S_1 \cup S_2)\chi_{S_1}(x_1)\chi_{S_2}(v) \\ &= \sum_{S_1\subseteq\bar{I}} \left(\sum_{S_2\subseteq I} \hat{f}(S_1 \cup S_2)\chi_{S_2}(v) \right) \chi_{S_1}(x_1). \end{aligned}$$

Thus,

$$\widehat{f|_R}(S_1) = \sum_{S_2\subseteq I} \hat{f}(S_1 \cup S_2)\chi_{S_2}(v). \tag{1}$$

Going back to the expected two-norm of the Fourier coefficients over sets larger than Δ and plugging in the expression for $\hat{f|_R}(S_1)$:

$$\begin{aligned} \mathrm{E}_R \left[\sum_{S_1\subseteq\bar{I}, |S_1|>\Delta} (\widehat{f|_R}(S_1))^2 \right] &= \mathrm{E}_{I,v} \left[\sum_{S_1\subseteq\bar{I}, |S_1|>\Delta} \sum_{S_2, S_2'\subseteq I} \hat{f}(S_1 \cup S_2)\hat{f}(S_1 \cup S_2')\chi_{S_2}(v)\chi_{S_2'}(v) \right] \\ &= \mathrm{E}_I \left[\sum_{S_1\subseteq\bar{I}, |S_1|>\Delta} \sum_{S_2\subseteq I} (\hat{f}(S_1 \cup S_2))^2 \right] \\ &= \mathrm{E}_I \left[\sum_{S\subseteq[n], |S\cap\bar{I}|>\Delta} (\hat{f}(S))^2 \right] \\ &= \sum_{S\subseteq[n]} (\hat{f}(S))^2 \cdot \Pr_I[|S \cap \bar{I}| > \Delta]. \end{aligned}$$

The first equality follows from (1), and the second one from pushing $\mathrm{E}_v[\cdot]$ inside to get $E_v[\chi_{S_2}(v)\chi_{S_2'}(v)] = \delta_{S_2,S_2'}$. The third equality follows from rearranging the terms, and the last one by linearity of expectation. Each of the variables is in $\bar{I}$ with probability ρ independently, therefore $\mathrm{E}_I[|S \cap \bar{I}|] = |S|\rho$. By setting $\Delta \leq \frac{|S|\rho}{2}$, the probability that $|S \cap \bar{I}|$ exceeds Δ is large; assuming $\Delta > 5$ we have

$$\Pr[|S \cap \bar{I}| > \Delta] \geq \frac{1}{2}.$$

Putting this into our previous equation gives us a bound on the weight of the high frequencies in the Fourier spectrum in terms of the original expectation:

$$\frac{1}{2} \sum_{S\subseteq[n], |S|\geq\frac{2\Delta}{\rho}} (\hat{f}(S))^2 \leq \sum_{S\subseteq[n]} (\hat{f}(S))^2 \cdot \Pr_I[|S \cap \bar{I}| > \Delta] = \mathrm{E}_R \left[\sum_{|S_1|>\Delta} (\hat{f|_R}(S_1))^2 \right] \leq \Pr[D(f|_R) > \Delta],$$

which gives us the following theorem.

Theorem 1. *For $\Delta > 5$, $\sum_{S\subseteq[n],|S|\geq\frac{2\Delta}{\rho}}(\hat{f}(S))^2 \leq 2\cdot\Pr[D(f|_R) > \Delta]$.*

Applying the Switching Lemma we immediately get the following corollary:

Corollary 1. *If f is a DNF of width w then for all $\Delta > 5$, $\sum_{S\subseteq[n],|S|\geq\frac{2\Delta}{\rho}}(\hat{f}(S))^2 \leq 2(5\rho w)^{\Delta}$.*

This corollary can be directly applied to construct a learning algorithm for DNFs with small width from random samples as it shows that that concept class has a Fourier spectrum concentrated on low frequencies.

3 Application to Learning DNFs

Applying Theorem 1 will give us two learning algorithms: one using random samples and a faster one using membership queries.

3.1 Learning DNFs From Random Samples

This application is fairly direct. Setting $\rho = \frac{1}{10w}$ and $\Delta = \log(\frac{2}{\epsilon})$ in Corollary 1 gives that

$$\sum_{|S|>20w\Delta}(\hat{f}(S))^2 \leq 2^{-\Delta+1} \leq \epsilon.$$

Now apply the generic learning algorithm from random samples to get an algorithm that runs in time $n^{O(w\log\frac{1}{\epsilon})}$, since sampling all sets up to size $O(w\log\frac{1}{\epsilon})$ dominates the running time. Recall we need to first transform the DNF to bounded width-w with $w = \log\frac{s}{\epsilon}$. This gives us an overall running time of $n^{O(\log(\frac{s}{\epsilon})\log\frac{1}{\epsilon})}$, which is $N^{O(\log N)}$ for constant ϵ, where $N = n + s$.

Theorem 2. *If $\mathcal{C}_{n,s}$ is the set of functions computed by DNFs of size at most s on n variables, then there exists an algorithm that learns $\mathcal{C}_{n,s}$ with respect to the uniform distribution using only random samples that runs in time* $\mathrm{poly}(n^{O(\log(\frac{s}{\epsilon})\log(\frac{1}{\epsilon})}, \log\frac{1}{\delta})$.

3.2 Learning DNFs From Queries

The efficiency of the learning algorithm we just obtained hinges on the fact that we only need to consider components of the Fourier decomposition that correspond to small sets S. We next show that those Fourier coefficients are fairly concentrated so we can further restrict our attention to the ones for which $|\hat{f}(S)|$ is above a relatively high threshold. This will allow us to improve the running time by effectively replacing the n in $n^{O(w\log\frac{1}{\epsilon})}$ by w, but it will require membership queries in order to locate those sets S. Just like in the previous lecture, we can guarantee the concentration property we need by showing that $\sum_S|\hat{f}(S)|$ is small, where the sum ranges over the small sets S we consider for the previous algorithm.

We use the following upper bound.

$$\mathrm{E}_R\left[\sum_{S_1\subseteq\bar{I}}|\widehat{f|_R}(S_1)|\right] = \sum_{\Delta=0}^{n}\Pr[D(f|_R)=\Delta]\cdot\mathrm{E}_R\left[\ \sum_{S_1\subseteq\bar{I}}|\widehat{f|_R}(S_1)|\ \middle|\ D(f|_R)=\Delta\ \right] \leq \sum_{\Delta=0}^{n}(5\rho w)^{\Delta}\cdot 2^{\Delta} \leq 2, \tag{2}$$

where the last inequality assumes $\rho \leq \frac{1}{20w}$. The first step in (2) follows from partitioning the sample space based on the decision tree depth Δ of the restriction. The first inequality follows from a direct application of the Switching Lemma and the fact that 1-norm of Fourier coefficients of decision trees is at most the size of the tree and that decision trees of depth Δ have size at most 2^{Δ}. The final inequality follows by setting $\rho = \frac{1}{20w}$, (this is half the value used in the random sample application).

On the other hand, we can lower bound the above expectation as follows:

$$\begin{aligned}
\mathrm{E}_R\left[\sum_{S_1 \subseteq \bar{I}} |\widehat{f|_R}(S_1)|\right] &= \mathrm{E}_{I,v}\left[\sum_{S_1 \subseteq \bar{I}} \left|\sum_{S_2 \subseteq I} \hat{f}(S_1 \cup S_2)\chi_{S_2}(v)\right|\right] \\
&\geq \mathrm{E}_I\left[\sum_{S_1 \subseteq \bar{I}} \left|\sum_{S_2 \subseteq I} \hat{f}(S_1 \cup S_2)\mathrm{E}_v[\chi_{S_2}(v)]\right|\right] \\
&= \mathrm{E}_I\left[\sum_{S_1 \subseteq \bar{I}} |\hat{f}(S_1)|\right] \\
&= \mathrm{E}_I\left[\sum_{S \subseteq [n], S \subseteq \bar{I}} |\hat{f}(S)|\right] \\
&= \sum_{S \subseteq [n]} |\hat{f}(S)| \cdot \Pr_I[S \subseteq \bar{I}] \\
&= \sum_{S \subseteq [n]} |\hat{f}(S)|\rho^{|S|}.
\end{aligned}$$

The first step follows by expanding $\hat{f}(S_1)$ as in Equation (1). The next step follows from linearity of expectation and that fact that $|\mathrm{E}[x]| \leq \mathrm{E}[|x|]$. The third step follows because $\mathrm{E}_v[\chi_{S_2}(v)] = \delta_{S_2,\emptyset}$ as v is uniformly distributed. The fourth step follows from rewriting, and the next one by linearity of expectation. The final steps follows because each $i \in [n]$ is in $\bar{I}$ independently with probability ρ. Combining the resuling lower bound with the upper bound (2) we have:

$$\sum_{S \subseteq [n]} |\hat{f}(S)|\rho^{|S|} \leq 2.$$

If we consider only sets of size up to $20w\Delta$, this gives a bound of:

$$\sum_{|S| \leq 20w\Delta} |\hat{f}(S)| \leq \left(\frac{1}{\rho}\right)^{20w\Delta} \cdot \sum_{|S| \leq 20w\Delta} |\hat{f}(S)|\rho^{|S|} \leq 2 \cdot \left(\frac{1}{\rho}\right)^{20w\Delta}. \tag{3}$$

Let $\mathcal{S} = \left\{S \text{ such that } |S| \leq 20w\Delta \text{ and } |\hat{f}(S)| \geq \tau\right\}$. We can bound the 2-norm of the Fourier coefficients not in $\mathcal{S}$ by:

$$\sum_{S \notin \mathcal{S}} (\hat{f}(S))^2 \leq \sum_{|S| > 20w\Delta} (\hat{f}(S))^2 + \sum_{|S| \leq 20w\Delta, |\hat{f}(S)| < \tau} (\hat{f}(S))^2.$$

The first summation is at most $2^{-\Delta+1}$ by an application of Corollary 1. The second summation is bounded by $\tau \cdot \sum_{|S|\leq 20w\Delta} |\hat{f}(S)| \leq \tau \cdot 2 \cdot \left(\frac{1}{\rho}\right)^{20w\Delta}$; this follows from Equation (3) since τ is the largest possible 1-norm of any set considered. Thus, we have

$$\sum_{S \notin \mathcal{S}} (\hat{f}(S))^2 \leq \left[2^{-\Delta} + \tau \left(\frac{1}{\rho}\right)^{20w\Delta}\right] \cdot 2.$$

By setting $\rho = \frac{1}{20w}$, $\Delta = \log(\frac{4}{\epsilon})$ and $\frac{1}{\tau} = 2 \cdot (20w)^{20w\Delta}/\epsilon$ we lose no more than ϵ weight of the Fourier spectrum.

This gives $\frac{1}{\tau} = (20w)^{O(w \log \frac{1}{\epsilon})}$. Our list decoding algorithm from last lecture ran in time $\text{poly}(\frac{1}{\tau})$, which dominated the time of the learning algorithm. This gives an overall time of $w^{O(w \log \frac{1}{\epsilon})} = 2^{O(w \log w \log \frac{1}{\epsilon})}$. With $w = \log \frac{s}{\epsilon}$ the running time becomes $(\frac{s}{\epsilon})^{O(\log\log(\frac{s}{\epsilon}) \cdot \log \frac{1}{\epsilon})}$, which is $N^{O(\log\log N)})$ for fixed ϵ, as promised.

Theorem 3. *If $\mathcal{C}_{n,s}$ is the set of functions computed by DNFs of size at most s on n variables, then there exists an algorithm using membership queries that learns $\mathcal{C}_{n,s}$ in time* $\text{poly}((\frac{s}{\epsilon})^{(\log\log(\frac{s}{\epsilon}) \cdot \log \frac{1}{\epsilon})}, \log \frac{1}{\delta})$.

4 Learning Constant Depth Circuits

The following corollary to the Switching Lemma allows us to apply Theorem 1 to depth-d circuits

Corollary 2. *If f is a depth-d circuit of size at most s and bottom fan-in at most w then $\Pr[D(f|_R) > w] \leq s2^{-w}$, where R denotes a random restriction with parameter $\rho = (\frac{1}{10w})^{d-1}$.*

Proof. (Sketch) We can view a random restriction with parameter ρ as the succession of $d-1$ random restrictions with parameter $\frac{1}{10w}$. The Switching Lemma applied to the bottom-most DNFs allows us to replace those with high probability by decision trees of depth at most w, and therefore by CNFs of width at most w. Merging the ANDs with the ANDs on the previous level then reduces the depth of the circuit by one without affecting the number of gates at higher levels. Bottom-most CNFs can be handled in a similar way. Each application of the Switching Lemma fails with probability no more than 2^{-w}. Since there are at most s applications, the result follows. See CS 810 for more details. □

Based on Theorem 1 we can then apply our generic approach for learning from random samples. Using a similar analysis as for DNFs, we obtain an algorithm for learning depth-d circuits of size at most s from random samples. For fixed ϵ, the algorithm runs in time $N^{O(\log^d N)}$, where $N = n + s$. Additional effort gives time $N^{O(\log^{d-1} N)}$, as claimed in the table at the beginning of the lecture. An improvement using membership queries as in Section 3.2 remains open.

5 Wrap-Up

This was our last lecture on learning using harmonic analysis, though there are still many interesting research problems still be to tackled in this area. For example: Can you prove that DNFs are concentrated on a set of polynomial size? This would give a simpler polynomial time algorithm for learning DNFs.