CS 395T Computational Learning Theory

Lecture 5: September 17, 2007 Lecturer: Adam Klivans

Scribe: Aparajit Raghavan

5.1 Learning using Polynomial Threshold Functions

5.1.1 Recap

Definition 1 *A function $f : \{0,1\}^n \rightarrow \{+,-\}$ is computed by a* Polynomial Threshold Function *or PTF, of degree d if there exists a real, multivariate polynomial p of total degree at most d such that $\forall x \in \{0,1\}^n$ $f(x) = SIGN(p(x) - \theta)$ for some θ. The function f is then said to have PTF degree d.*

Note that the *total degree* of a multivariate polynomial is the maximum degree of any monomial term, where the degree of a monomial term is computed as the sum of the exponents of the variables in that term.

We have seen how polynomial threshold functions of degree d can be learned in time and mistake bound $n^{O(d)}$. This leads to the following theorem:

Theorem 1 *If all $c \in \mathcal{C}$ have PTF degree d then $\mathcal{C}$ is learnable in the Mistake Bound model in time and mistake bound $n^{O(d)}$.*

5.1.2 PTFs for different boolean function families

For the following families of functions, we want the lowest degree PTF that can compute a function from that family.

- Decision Lists: PTFs of degree 1. (We previously showed that Decision lists can be converted to equivalent half-spaces)
- Decision Trees: PTFs of degree $O(\log n)$ (Based on the conversion of decision trees to equivalent decision lists using the rank of the tree)
- DNFs: PTFs of degree $\tilde{O}(n^{\frac{1}{3}})$ (Note that $\tilde{O}$ hides log factors). This will be shown in the subsequent sections.

5.1.3 Background - Chebyshev Polynomials

We will use certain properties of Chebyshev Polynomials to deduce the PTF degree of DNFs. For more information on Chebyshev Polynomials, please see either the MathWorld article on 'Chebyshev

Polynomials of the First Kind' or the Wikipedia article on Chebyshev Polynomials.

A Chebyshev Polynomial $\mathbb{C}_d(x)$ is a univariate polynomial of degree d with the following properties (among others):

$$\mathbb{C}_d(1) = 1$$

$$\forall x \in [-1, 1],\ |\mathbb{C}_d(x)| \le 1$$

$$\forall x \ge 1,\ \mathbb{C}_d'(x) \ge d^2$$

$$\mathbb{C}_{\sqrt{d}}(1 + \frac{1}{d}) \ge 2$$

The first and second properties indicate that the Chebyshev polynomials are contained in $[-1, 1]$ for $x \in [-1, 1]$. The actual behavior is unspecified, but that does not affect our analysis.

Also, the third property indicates that outside $[-1, 1]$ the growth of the function is explosive. This is also reflected in the fourth property which indicates how far from 1 we need to be to achieve a value of at least 2.

An plot of the first few Chebyshev polynomials illustrates this functional behavior (see Figure 5.1). Notice that all plotted functions are bounded within $[-1, 1]$ for domain values in $[-1, 1]$, with the value 1 at $x = 1$. The explosive growth of the polynomials outside of this range can also be observed.

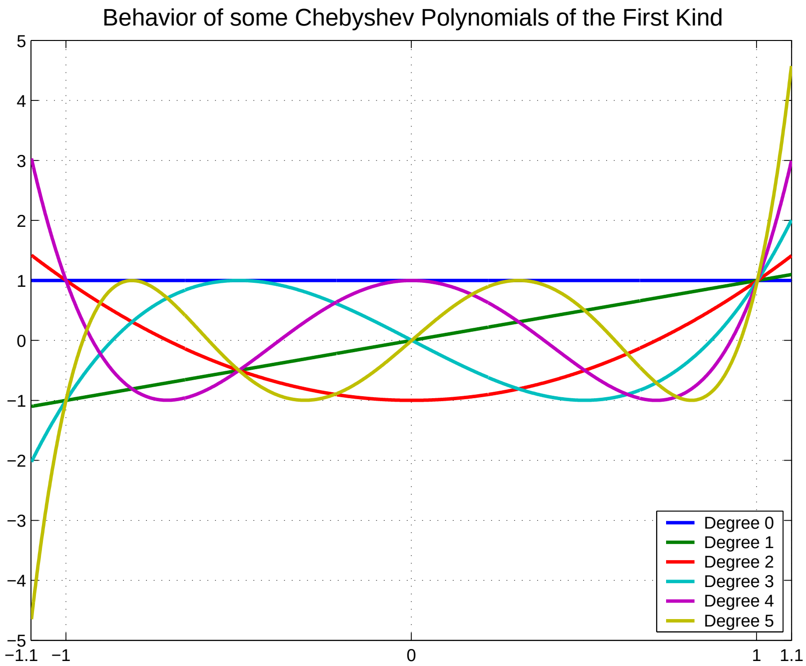

Figure 5.1: Behavior of the first few Chebyshev polynomials.

5.1.4 DNFs can be computed by PTFs

Lemma 1 *For any DNF formula on n variables with s terms of length at most t, there exists a PTF of degree $\sqrt{t}\log s$ that computes that DNF.*

Proof: Let the DNF formula be $D = T_1 \vee T_2 \vee \cdots \vee T_n$. For $1 \le i \le s$, let term $T_i = x_1 \wedge \cdots \wedge x_t$.

Let $\bar{x}$ denote the vector $[x_1, x_2, \cdots, x_n]$.Consider the function $a_i(\bar{x}) = \frac{x_1+\cdots+x_t}{t}$, and the Chebyshev polynomial $Q_i(\bar{x}) = \mathbb{C}_{\sqrt{t}}((1+\frac{1}{t})\cdot a_i(\bar{x}))$. As $a_i(\bar{x})$ is of degree 1, $Q_i(\bar{x})$ is of degree $\sqrt{t}$.

Now, if T_i is true on some vector $\bar{x}$, then $Q_i(\bar{x}) \ge 2$. This is because all x_j have to be 1, which implies that $a_i = 1$, and so $Q_i(\bar{x}) = \mathbb{C}_{\sqrt{t}}((1+\frac{1}{t})\cdot 1) \ge 2$ by the fourth property of Chebyshev polynomials listed above.

Similarly, if T_i is false on some vector $\bar{x}$, then $Q_i(\bar{x}) \le 1$. This is because at least one x_j is zero, implying that $a_i \le \frac{t-1}{t}$, and

$$\begin{aligned} Q_i(\bar{x}) &\le \mathbb{C}_{\sqrt{t}}((1-\frac{1}{t})(1+\frac{1}{t})) \\ &= \mathbb{C}_{\sqrt{t}}(1-\frac{1}{t^2}) \\ &< \mathbb{C}_{\sqrt{t}}(1) = 1 \end{aligned}$$

Now, using the above strategy, we can obtain a Q_i for each term in the original DNF. Consider the following threshold function:

$$Q_1^{\log 2s}(\bar{x}) + Q_2^{\log 2s}(\bar{x}) + \cdots + Q_s^{\log 2s}(\bar{x}) > s$$

Note that as each Q_i is a polynomial of degree $\sqrt{t}$, the above inequality represents a Polynomial Threshold Function of total degree $\sqrt{t}\log(2s)$.

For all $\bar{x}$ satisfying the original DNF, there exists (at least one) term T_i that is set to 1. Assume, without loss of generality, that the first term T_1 is set to 1. Then as $Q_1(\bar{x}) \ge 2$, and $Q_1^{\log 2s}(\bar{x}) \ge 2^{\log 2s} = 2s$, the term $Q_1^{\log 2s}(\bar{x})$ contributes at least $2s$ to the left side of the inequality. Also at worst all other terms in the DNF are not satisfied. For these terms, as $-1 \le Q_i(\bar{x}) \le 1$ and $Q_i^{\log 2s}(\bar{x}) \ge -1$. So,

$$\begin{aligned} Q_1^{\log 2s}(\bar{x}) + \cdots + Q_s^{\log 2s}(\bar{x}) &\ge 2s + (-1)\cdots + (-1) \\ &= 2s + \sum_{j=1}^{s-1}(-1) \\ &= 2s - s + 1 > s \end{aligned}$$

which satisfies the inequality.

For all $\bar{x}$ not satisfying the original DNF, every Q_i term is in $[-1, 1]$, which implies that $|Q_i^{\log 2s}(\bar{x})| \le$

1, and thus $\sum_{j=1}^{s} Q_i^{\log 2s}(\bar{x}) \leq \sum_{j=1}^{s} 1 \leq s$. The inequality is not satisfied. This means that the above threshold function is a PTF that computes the specified DNF.

■

The construction of a $\sqrt{t}\log(2s)$ PTF for DNF formulas immediately gives us an $n^{O(\sqrt{n}\log n)}$ time and mistake bound algorithm for learning DNFs, by exploiting the established algorithm for learning PTFs.

5.1.5 Improved bound for learning DNFs

We now attempt to improve the $n^{O(\sqrt{n}\log n)}$ algorithm using an alternate construction of the equivalent PTF. (Partly reproduced from *Klivans A. and Servedio R., "Learning DNF in Time* $2^{\tilde{O}(n^{1/3})}$*"*).

Sketch of Proof For this, we use the method of 'Augmented Decision Trees' that was previously employed to construct an equivalent Decision Tree from a given DNF formula. Recall that for any given DNF, we were able to construct a rank $\frac{2n}{t}\log s$ decision tree (all parameters are as before), that was augmented with DNFs of term length t at the leaves of the tree.

Choose $t = n^{\frac{2}{3}}$. Then for a given DNF, there exists a decision tree with rank $2\sqrt[3]{n}\log s$ with DNFs of term length $n^{\frac{2}{3}}$ at the leaves that is equivalent to that DNF.

Using the claim proved in previous lemma, this implies that for a given DNF, there exists a decision tree with rank $2\sqrt[3]{n}\log s$ with PTFs of degree $\sqrt[3]{n}\log s$ at the leaves of the tree that is equivalent that that DNF.

Now, using the method of converting decision trees to decision lists, we have that for any given DNF formula, there exists a decision list with conjuncts of length $2\sqrt[3]{n}\log s$, augmented with PTFs of degree $\sqrt[3]{n}\log s$ that computes that DNF. Call this decision list L.

Let $C_1, \ldots, C_R$ be the conjunctions contained in successive nodes of L and let $P_1(x), \ldots, P_R(x)$ be the corresponding polynomials for the associated polynomial threshold functions at the outputs, i.e. the polynomial threshold function corresponding to the j-th conjunction C_j computes the function "$P_j(x) \geq 0$". If $P_j(x) = 0$ for some $x \in \{0,1\}^n$ then we can replace $P_j(x)$ by $P_j(x) + \delta/2$, where $\delta = min\{-P_j(x) \ : \ x \in \{0,1\}^n \ and \ P_j(x) < 0\}$, without changing the function computed by the polynomial threshold function. Now by scaling each P_j by an appropriate multiplicative factor we can suppose without loss of generality that for each $j = 1, \ldots, R$ we have $min_{x\in\{0,1\}^n}|P_j(x)| \geq 1$.

Consider the polynomial

$$Q(x) = A_1\tilde{C}_1(x)P_1(x) + A_2\tilde{C}_2(x)P_2(x) + \cdots + A_R\tilde{C}_R(x)P_R(x)$$

Here $\tilde{C}_j$ is the zero/one valued polynomial which corresponds to the monomial C_j, for example if C_j is $x_3\bar{x}_4x_5$ then $\tilde{C}_j$ is $x_3(1-x_4)x_5$. Each value A_j is a positive constant chosen so as to satisfy the following conditions:

$$\begin{aligned} A_R &= 1 \\ A_{R-1} &> \max_{x\in\{0,1\}^n} |A_R\tilde{C}_R(x)P_R(x)|, \end{aligned}$$

$$
\begin{aligned}
&\vdots \\
A_j &> \max_{x \in \{0,1\}^n} |A_{j+1}\tilde{C}_{j+1}(x)P_{j+1}(x) + \cdots + A_R\tilde{C}_R(x)P_R(x)|, \\
&\vdots \\
A_1 &> \max_{x \in \{0,1\}^n} |A_2\tilde{C}_2(x)P_2(x) + \cdots + A_R\tilde{C}_R(x)P_R(x)|.
\end{aligned}
$$

Then the polynomial threshold function "$Q(x) \geq 0$" computes exactly the same function as the decision list L. To see this, fix an input $x \in \{0,1\}^n$. If j is the index of the first conjunction C_j which is satisfied by x, then $\tilde{C}_1(x) = \tilde{C}_2(x) = \cdots = \tilde{C}_{j-1}(x) = 0$, so the only terms of $Q(x)$ which make a nonzero contribution are $A_i\tilde{C}_i(x)P_i(x)$ for $i \geq j$. Since $\tilde{C}_j(x) = 1$ and $|P_j(x)| \geq 1$, the choice of A_j ensures that the sign of $Q(x)$ will be the same as that of $P_j(x)$.

The degree of the polynomial $Q(x)$ is at most $2\sqrt[3]{n}\log s + O(\sqrt[3]{n}\log s)$ which is $O(n^{1/3}\log s)$, giving us an improved bound for learning DNFs.

■