Advanced Course in Machine Learning Spring 2010

Kushilevitz-Mansour algorithm for learning decision trees

Handouts are jointly prepared by Shie Mannor and Shai Shalev-Shwartz

In this section we describe an algorithm, proposed by Eyal Kushilevitz and Yishay Mansour in 1992, that learns decision trees in polynomial time in the membership queries model, assuming uniform distribution over the instances (and realizability). In the membership queries model, the learner can query the correct label of each instance. This is similar to active learning, except that in active learning we assumed that the learner can only query the label of instances appearing in an unlabeled sample.

The class we consider is all decision trees over n Boolean variables with $m = \text{poly}(n)$ leaves. The calculation starts at the root node, and at each internal node we move to the left child iff $x_i = 1$ for some variable $i \in [n]$. [1]

The algorithm is derived based on the following ideas:

1. We can think on each Boolean function as a vector in a 2^d-dimensional space.
2. Applying Fourier transform on this vector space, it turns out that any decision tree function can be represented as a vector with small L_1 norm (i.e., the sum of the absolute values of the coefficients in the Fourier representation of the function is small). This is not true for any DNF formula but is true for decision trees, where the property we use is that in decision trees the number of satisfied clauses is either 0 or 1 (but no more than 1).
3. Any vector with low L_1 norm can be approximated by a sparse vector, such that all the non-zero coefficients of the sparse vector are relatively large.
4. Each large Fourier coefficient can be identified and approximated in polynomial time by using membership queries.

Combining the above, we obtain a polynomial time algorithm for learning decision trees. We now describe each of the steps in details.

1 Fourier transform of Boolean functions

Let $f : \{0,1\}^n \to \{-1,1\}$ be a Boolean function. We can think on f as a vector in $\{-1,1\}^{2^n}$ where elements are indices by strings $x \in \{0,1\}^n$. Define an inner product between two Boolean functions to be

$$\langle f, g \rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)g(x) = \mathop{\mathbb{E}}_{x \sim U(n)} [f(x)g(x)] \,.$$

Now, for each $z \in \{0,1\}^n$ define the function $\chi_z : \{0,1\}^n \to \{-1,1\}$ defined by

$$\chi_z(x) = (-1)^{\sum_i z_i x_i} \,.$$

That is, χ_z is the parity of those bits in x corresponding to the set of active bits in z. Observe that for any z, z', x, the value of $\chi_z(x)\chi_{z'}(x)$ will be 1 iff $\chi_z(x) = \chi_{z'}(x)$. This will happen iff $\chi_{z \oplus z'}(x) = 1$. Therefore,

$$\langle \chi_z, \chi_{z'} \rangle = \mathop{\mathbb{E}}_{x \sim U(n)} [\chi_z(x)\chi_{z'}(x)] = \mathop{\mathbb{E}}_{x \sim U(n)} [\chi_{z \oplus z'}(x)] = \mathbb{1}_{[z=z']} \,.$$

[1] In fact, the algorithm of Kushilevitz and Mansour can learn a more powerful form of decision trees, in which at each node a xor of several variables can determine the next node.

It follows that $\{\chi_{\mathbf{z}} : \mathbf{z} \in \{0,1\}^n\}$ is an orthonormal basis of $\mathbb{R}^{2^n}$. This is called the Boolean Fourier basis. The Fourier coefficients are defined by projections onto this basis, that is,

$$\hat{f}(z) = \langle f, \chi_z \rangle \ .$$

From the orthogonality of the basis it follows that $\langle f, g \rangle = \langle \hat{f}, \hat{g} \rangle$ and in particular $\|f\| = \|\hat{f}\|$, which is known as Parseval's theorem. In particular, if the range of f is $\{\pm 1\}$ then $1 = \|f\| = \|\hat{f}\|$.

We now show one of the main features of Fourier analysis that we will use in the next section. Suppose we are looking for the Fourier representation of an AND function. That is, $f : \{0,1\}^n \to \{0,1\}$ is defined by a set $I \subset [n]$ as $f(x) = \prod_{i \in I} x_i$. Then,

$$2^n \, \hat{f}(z) = \sum_x f(x)\chi_z(x) = \sum_{x_i : i \in I} f(x) \sum_{x_i : i \notin I} \chi_z(x) \ .$$

Now, if the support of z is not in I, it follows by symmetry that $\hat{f}(z) = 0$. Otherwise, $|\hat{f}(z)| = 2^{-\|z\|_1}$.

2 The L_1 norm of the spectrum of decision trees

We have shown that the L_2 norm of any Boolean function is 1, hence the L_2 norm of $\hat{f}$ is also 1. But, what about the L_1 norm of $\hat{f}$, namely, $\|\hat{f}\|_1 = \sum_z |\hat{f}(z)|$? In a general d-dimensional Euclidean space, the L_1 norm of a vector can be as large as $\sqrt{d}$ times the L_2 norm of the vector. In our case, $d = 2^n$ which is exponential in n. We now show that if f is associated with a decision tree with m leaves then $\|\hat{f}\|_1 \leq m$. In later section we shall see that it is possible to learn efficiently the class of functions with $\|\hat{f}\|_1 = \mathrm{poly}(n)$, which will imply that we can efficiently learn the class of decision trees with $\mathrm{poly}(n)$ leaves.

Lemma 1 *Let $f : \{0,1\}^n \to \{\pm 1\}$ be a function that can be implemented by a decision tree with m leaves. Than, $\|\hat{f}\|_1 \leq m$.*

Proof As mentioned in the proof, the key observation we make is that a decision tree can be written as a DNF formula, $(f(x)+1)/2 = T_1(x) + \ldots + T_k(x)$, where $k \leq m$ and each $T_i(x)$ is an AND function from $\{0,1\}^n$ to $\{0,1\}$. That is, for each T_i exists some set $S_i \subset [n]$ such that $T_i(x) = \prod_{j \in S_i} x_j$. This equality does not hold for any DNF and uses a special property of decision trees—for each x either $f(x) = -1$ and thus $T_i(x) = 0$ for all i or $f(x) = 1$ and then exactly one of the clauses is satisfied. The rest of the proof is left as an exercise. Hint: Analyze the L_1 norm of an AND function and show it is at most 1. Then, use the linearity of the Fourier transform. ■

3 From low L_1 norm vector to sparse vector

The following lemma shows that any function with low $\|\hat{f}\|_1$ can be approximated by a function with a sparse spectrum.

Lemma 2 *For any Boolean function f, there exists a Boolean function h such that $\|\hat{h}\|_0 \leq \|\hat{f}\|_1^2/\epsilon$ and $\mathbb{E}[(f-h)^2] \leq \epsilon$.*

Proof Consider the set $A = \{s : |\hat{f}(s)| \geq \epsilon/\|\hat{f}\|_1\}$. There are at most $\|\hat{f}\|_1/(\epsilon/\|\hat{f}\|_1) = \|\hat{f}\|_1^2/\epsilon$ elements in A. Let $h = \sum_{s\in A} \hat{f}(s)\chi_s(x)$, which is $\|\hat{f}\|_1^2/\epsilon$-sparse. Then,

$$\begin{aligned}\mathbb{E}[(f-h)^2] &= \sum_{s\notin A} \hat{f}(s)^2 \\ &\leq \max_{s\notin A} |\hat{f}(s)| \sum_s |\hat{f}(s)| \\ &\leq \frac{\epsilon}{\|\hat{f}\|_1} \|\hat{f}\|_1 = \epsilon.\end{aligned}$$

■

4 Identifying the large Fourier coefficients using membership queries