CS 7545: Machine Learning Theory **Fall 2021**

Lecture 13: Fourier Learning

Instructor: Santosh Vempala *Lecture date: 11/1/2021*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

It is an open problem to PAC learn DNF formulas or decision trees (we can learn decision lists). So we can consider special distributions on inputs, e.g.,

- uniform
- product
- Gaussian
- ...

and allow for membership queries, i.e., the learner is allowed to ask for the label of any input x it wants.

13.1 Fourier Expansion of Boolean functions

Assume $x \in \{-1,1\}^n$ and $f : \{-1,1\}^n \to \{-1,1\}$ is a Boolean function. Note that any such $f \in \{-1,1\}^{2^n}$.

Definition 13.1 (Inner product of Boolean functions)**.** *Define inner product of f and g with respect to distribution a D as*

$$\langle f, g\rangle_D = \sum_x D(x)f(x)g(x) = \mathbb{E}_D(f(x)g(x)).$$

With this definition, the norm of f is

$$\langle f, g\rangle_D = \|f\|_D^2 = 1$$

since $f^2(x) = 1$. Viewing f as a vector, the standard basis is $e_1, e_2, \ldots, e_{2^n}$. But we can use any basis and write $f(x) = \sum_v \langle f, v\rangle v$ where $\{v\}$ is an orthonormal basis.

13.1.1 Parity basis

For any $S \subseteq [n]$, we can define a parity function as $\chi_S(x) = \prod_{i\in S} x_i$. Note that there are 2^n such functions. For the uniform distribution D over $\{-1,1\}^n$,

$$\langle \chi_S, \chi_S\rangle_D = 1$$
$$\langle \chi_S, \chi_T\rangle_D = \mathbb{E}_D(\prod_{i\in S} x_i \prod_{j\in T} x_j) = 0 \text{ for } S \neq T.$$

Hence, $\{\chi_S\}$ is an orthogonal basis. So any f can be written as

$$f(x) = \sum_v \hat{f}_S \chi_S(x)$$

where $\hat{f}_S = \langle f, \chi_S\rangle_D$ are the discrete Fourier coefficients of f.

Theorem 13.2 (Parseval)**.**

$$\|f\|_D^2 = \langle f, f\rangle_D = \langle \hat{f}, \hat{f}\rangle.$$

Theorem 13.3 (Plancherel)**.**

$$\langle f, g\rangle_D = \langle \hat{f}, \hat{g}\rangle.$$

Proof.

$$\begin{aligned}
\langle f, g\rangle_D &= \mathbb{E}_D \left(\sum_S \hat{f}_S \chi_S(x)\right)\left(\sum_T \hat{g}_T \chi_T(x)\right) \\
&= \sum_{S,T} \hat{f}_S \hat{g}_T \mathbb{E}_D(\chi_S(x)\chi_T(x)) \\
&= \sum_S \hat{f}_S \hat{g}_S = \langle \hat{f}, \hat{g}\rangle.
\end{aligned}$$

□

13.2 Learning Decision Trees

A decision tree is a Boolean function f. We want to learn f by approximating all its significant Fourier coefficients $\hat{f}_S$. Suppose our approximation function is g (g need not map to $\{-1,1\}^n$). Note that

$$\Pr_D(f(x) \neq \text{sign}(g(x))) \leq \mathbb{E}_D\left((f(x) - g(x))^2\right) = \left\|\hat{f} - \hat{g}\right\|_D^2.$$

The equality above follows Theorem 13.3. Then our goal is to find g such that $\left\|\hat{f} - \hat{g}\right\|_D^2 \leq \epsilon$.

Lemma 13.4. *If we learn all* $\hat{f}_S \geq \frac{\epsilon}{\|\hat{f}\|_1}$*, then* $\left\|\hat{f} - \hat{g}\right\|_D^2 \leq \epsilon$.

Proof.

$$\left\|\hat{f} - \hat{g}\right\|^2 = \sum_{S: |\hat{f}_S| \leq \frac{\epsilon}{\|\hat{f}\|_1}} \hat{f}_S^2 \leq \sum_S |\hat{f}_S| \frac{\epsilon}{\|\hat{f}\|_1} = \epsilon.$$

□

Lemma 13.5 (DNF)**.** *If a decision tree has* m *leaves, then*

$$\left\|\hat{f}\right\|_1 = \sum_S |\hat{f}_S| \leq 2m + 1.$$

Proof. Consider a single conjunction T. Let

$$T(x) = \begin{cases} 1 & \text{if } x \text{ satisfies } T \\ 0 & \text{otherwise} \end{cases}$$

Then

$$\langle T, T\rangle_D = \mathbb{E}(T(x)^2) = \frac{1}{2^{|T|}}.$$

$$\begin{aligned}
\hat{T}_S &= \langle T, \chi_S\rangle_D \\
&= \Pr_D(T(x) = 1)\mathbb{E}_D(\chi_S(x) | T(x) = 1) \\
&= \begin{cases} 0 & \text{if } S \text{ contains } x_i \notin T \\ \frac{1}{2^{|T|}} & \text{otherwise} \end{cases}
\end{aligned}$$

This gives

$$\left\|\hat{T}\right\|_1 = \sum_S \hat{T}_S = \sum_{S \subseteq T} \frac{1}{2^{|T|}} = 1.$$

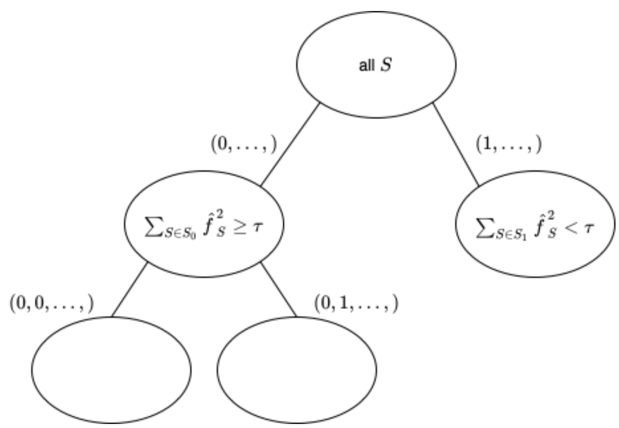
For a decision tree with m leaves, we can write it with conjunctions represented by its leaves

$$f(x) = 2(T_1(x) + \cdots + T_m(x)) - 1.$$

So $\left\|\hat{f}\right\|_1 \le 2\sum_{i=1}^m \left\|\hat{T}_i\right\|_1 + 1 \le 2m + 1.$ □

How to learn large Fourier coefficients? We will learn all $\hat{f}_S$ for which $\hat{f}_S \ge \tau$. By Lemma 13.4 and Lemma 13.5, we can set $\tau = \frac{\epsilon}{2m+1}$ for decision trees. Note $\sum_S \hat{f}_S^2 = 1$ and $|\hat{f}_S| \le 1$.
The algorithm is

1. Start with empty α.
2. At each node, estimate whether $\sum_{S:\text{prefix } \alpha} \hat{f}_S^2 \ge \tau$.
3. If $\sum_{S:\text{prefix } \alpha} \hat{f}_S^2 \ge \tau$, append 0/1 to α and iterate.



The width of the tree is at most $1/\tau$ since the sum of $\sum_{S_\alpha} \hat{f}_S^2$ for nodes at the same depth in the tree is at most 1 and the algorithm only explores nodes with $\sum_{S_\alpha} \hat{f}_S^2 \ge \tau$. The depth of the tree is at most n. So the number of nodes in the tree is at most n/τ.

How to estimate $\sum_{S \in S_\alpha} \hat{f}_S^2$?

Claim 13.6. *Suppose* $\alpha = (\underbrace{0, 0, \dots, 0}_{k})$.

$$\sum_{S_\alpha} \hat{f}_S^2 = \mathbb{E}_{\substack{x \sim \{0,1\}^{n-k} \\ y,z \sim \{0,1\}^k}} (f(yx)f(zx)).$$

Proof. Suppose f is a parity function. If f agrees with α, then $f(yx) = f(zx)$ so we get 1. Else $\Pr(f(yx) = f(zx)) = 1/2$ and we get 0. Any f can be written as a weighted sum of parities $f = \sum_U \hat{f}_U \chi_U$. So

$$\mathbb{E}(f(yx)f(zx)) = \mathbb{E}\left(\sum_U \hat{f}_U \chi_U(yx) \sum_V \hat{f}_V \chi_V(zx)\right)$$

$$
\begin{aligned}
&= \sum_{U,V} \hat{f}_U \hat{f}_V \underbrace{\mathbb{E}(\chi_U(yx)\chi_V(zx))}_{=0 \text{ if } U \neq V} \\
&= \sum_{U} \hat{f}_U^2 \underbrace{\mathbb{E}_D(\chi_U(yx)\chi_V(zx))}_{=0 \text{ if } U \text{ does not agree with } \alpha=(0,\ldots,0)} \\
&= \sum_{U \in S_\alpha} \hat{f}_U^2 .
\end{aligned}
$$

□

We can generalize the argument to any prefix α.

Lemma 13.7.

$$
\sum_{S_\alpha} \hat{f}_S^2 = \mathbb{E}_{\substack{x \sim \{0,1\}^{n-k} \\ y,z \sim \{0,1\}^k}} \left(f(yx) f(zx) \chi_\alpha(y) \chi_\alpha(z) \right).
$$