Boolean function complexity

Lecturer: *Nitin Saurabh* Meeting: 7
Scribe: *Nitin Saurabh* 05.06.2019

In this lecture we will prove lower bound on the number of leaves in a decision tree.

1 Decision tree size

Definition 1. *The* size *of a decision tree is defined to be the number of leaves in the tree. Let* $\mathsf{L}^{\mathsf{dt}}(f)$ *denote the minimum size of a deterministic decision tree computing* f*.*

Recall, nondeterministic decision tree complexity is the maximum certificate complexity over 1-inputs, and co-nondeterministic is the maximum certificate complexity over 0-inputs. Similarly, we can define the following nondeterministic version of size.

Definition 2. Nondeterministic decision tree size*,* $\mathsf{dnf}(f)$*, is defined to be the minimum number of terms in a DNF representing* f*.*

Similarly, co-nondeterministic decision tree size*,* $\mathsf{cnf}(f)$*, is defined to be the minimum number of clauses in a CNF representing* f*.*

The following proposition is easily seen.

Proposition 3. $\mathsf{cnf}(f) = \mathsf{dnf}(\neg f)$*.*

Consider the example of tribes function

$$\mathsf{Tribes}_{m,m}(x) = \bigvee_{i=1}^{m} \left(\bigwedge_{j=1}^{m} x_{ij} \right).$$

Clearly, nondet. decision tree size of tribes is small, $\mathsf{dnf}(\mathsf{Tribes}_{m,m}) = m$. However, $\mathsf{L}^{\mathsf{dt}}(f) \geq 2^m$, since size of every 1-certificate and 0-certificate is m. Therefore, we have $\mathsf{P} \neq \mathsf{NP}$ for decision tree size. Thus, a natural question to ask is what happens if both $\mathsf{dnf}(f)$ and $\mathsf{cnf}(f)$ are small, i.e., both f and $\neg f$ have small DNFs? In this lecture we will prove that even if both DNF and CNF sizes are small, the decision tree size could be quasi-polynomially larger. That is, $\mathsf{P} \neq \mathsf{NP} \cap$ co-NP for decision tree size.

Theorem 4. *There exist explicit Boolean function* f *such that* $\mathsf{dnf}(f) + \mathsf{dnf}(\neg f) = N$*, but* $\mathsf{L}^{\mathsf{dt}}(f) = 2^{\Omega(\log^2 N)}$*.*

To prove this we will use Fourier representation of Boolean functions, which we will introduce in the next section.

Open Problem 1.1. *Let* f *be a Boolean function such that* $\mathsf{dnf}(f) + \mathsf{dnf}(\neg f) = N$*. Then, is* $\mathsf{L}^{\mathsf{dt}}(f) \leq 2^{O(\log^2 N)}$*?*

2 Fourier representation of Boolean functions

Fourier analysis of Boolean function is a special case of general Harmonic analysis on finite abelian groups. In the Boolean function case the group is $\mathbb{F}_2^n$ with addition as group operation. However, we will mostly think of Fourier representation as real valued multilinear polynomials over $\{-1,1\}^n$.

In the Fourier representation we consider Boolean functions as functions from $\{-1,1\}^n$ to $\{-1,1\}$. In $\{0,1\}$ case, **True** $= 1$ and **False** $= 0$. In the Fourier representation, **True** $=$ -1 and **False** $= 1$. To convert from $\{0,1\}$ representation to $\{-1,1\}$ representation, we use the following map

$$x \mapsto 1 - 2x.$$

In particular, given a polynomial $p(x)$ representing a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, we can obtain the Fourier representation of f, $\tilde{p}\colon \{-1,1\}^n \to \{-1,1\}$, in the following way

$$\tilde{p}(x_1, \ldots, x_n) = 1 - 2 \cdot p\left(\frac{1-x_1}{2}, \frac{1-x_2}{2}, \ldots, \frac{1-x_n}{2}\right). \tag{1}$$

Since $x_i^2 = 1$, for $x_i \in \{-1,1\}$, we see that $\tilde{p}$ is again a multilinear polynomial. We now define the Fourier representation more formally.

Consider the space of all functions $f\colon \{-1,1\}^n \to \mathbb{R}$. It is a vector space over $\mathbb{R}$ of dimension 2^n. Define an inner product on this space as follows

$$\langle f, g \rangle := \frac{1}{2^n} \sum_{x \in \{-1,1\}^n} f(x)g(x) = \mathbb{E}_x[f(x) \cdot g(x)],$$

where the expectation is taken with respect to uniform distribution over $\{-1,1\}^n$. This naturally induces the L_2-norm on the vector space of these functions,

$$\|f\| := \sqrt{\langle f, f \rangle} = \sqrt{\mathbb{E}[f^2]}.$$

Now for each $S \subseteq [n]$, define a function $\chi_S\colon \{-1,1\}^n \to \{-1,1\}$,

$$\chi_S(x) := \prod_{i \in S} x_i.$$

By definition, we take $\chi_\emptyset$ to be the constant function 1. From the transformation given by Eq. (1) we see that every function $f\colon \{-1,1\}^n \to \mathbb{R}$ can be written as a linear combination of $\{\prod_{i \in S} x_i\}_{S \subseteq [n]}$. Therefore, the set $\{\chi_S\}_{S \subseteq [n]}$ of functions forms a basis for the space of all functions from $\{-1,1\}^n$ to $\mathbb{R}$. This is known as the *Fourier* basis.

Proposition 5. *$\mathbb{E}_x[\chi_\emptyset(x)] = 1$ and $\mathbb{E}_x[\chi_S(x)] = 0$ for $S \neq \emptyset$.*

Proof. Note that $\mathbb{E}_x[x_i] = 0$ for all x_i. □

Proposition 6. *$\{\chi_S\}_{S \subseteq [n]}$ forms an orthonormal basis.*

Proof. We already saw that it is a basis. To prove orthonormality, we need to show for $S, T \subseteq [n]$,

$$\mathbb{E}_x[\chi_S(x)\chi_T(x)] = \begin{cases} 1 & \text{if } S = T \\ 0 & \text{otherwise.} \end{cases}$$

We observe that $\mathbb{E}[\chi_S(x)\chi_T(x)] = \mathbb{E}[\prod_{i \in S} x_i \prod_{j \in T} x_j] = \mathbb{E}[\prod_{i \in S \triangle T} x_i]$. Now the assertion follows from the previous proposition. □

Thus, every function $f : \{-1, 1\}^n \to \mathbb{R}$ has the following Fourier representation

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S)\chi_S(x) = \sum_{S \subseteq [n]} \widehat{f}(S) \prod_{i \in S} x_i, \tag{2}$$

where $\widehat{f}(S)$ is called the Fourier coefficient of f at S.

From the orthonormality of basis we see that

$$\langle f, \chi_S \rangle = \mathbb{E}[f(x)\chi_S(x)] = \widehat{f}(S).$$

In the particular case when $f\colon \{-1, 1\}^n \to \{-1, 1\}$, we have that $|\widehat{f}(S)| \leq 1$.

Definition 7. *The* Fourier-degree *of a function is* $\max\{|S| \mid \widehat{f}(S) \neq 0\}$.

From the Eq. 1, the following proposition is easily seen.

Proposition 8. *For any Boolean function f, Fourier-degree of f = $\mathsf{deg}(f)$.*

Theorem 9 (Plancherel's Theorem)**.** *For any $f, g\colon \{-1, 1\}^n \to \mathbb{R}$,*

$$\mathbb{E}_x[f(x) \cdot g(x)] = \sum_{S \subseteq [n]} \widehat{f}(S) \cdot \widehat{g}(S).$$

Proof. Follows from the orthonormality of χ_S. □

In the particular case when f is Boolean we obtain the following theorem.

Theorem 10 (Parseval's Theorem)**.** *For any $f\colon \{-1, 1\}^n \to \{-1, 1\}$,*

$$\sum_{S \subseteq [n]} \widehat{f}(S)^2 = \mathbb{E}_x[f(x)^2] = 1.$$

The above theorem says that for a Boolean function the sum of the squared Fourier coefficients always add up to 1.

Example. *Consider the $\mathsf{AND}_n\colon \{-1, 1\}^n \to \{-1, 1\}$. The Fourier coefficients of AND_n are as follows:*

$$\widehat{\mathsf{AND}_n}(S) = \begin{cases} 1 - \frac{1}{2^{n-1}} & \textit{if } S = \emptyset \\ (-1)^{|S|-1} \frac{1}{2^{n-1}} & \textit{otherwise.} \end{cases}$$

Having developed the Fourier representation of Boolean function we now get back to proving a lower bound on the decision tree size.

3 Proof of Theorem 4

We will start with the following general lower bound on the size of a decision tree.

Theorem 11. *Let f be any Boolean function. Then,*

$$\sum_{S\subseteq[n]} |\widehat{f}(S)| \leq \mathsf{L}^{\mathsf{dt}}(f).$$

Before we prove the theorem, let us define the ℓ_1-norm of the Fourier transform of f to be, $\|\hat{f}\|_1 := \sum_{S\subseteq[n]} |\widehat{f}(S)|$.

Proposition 12. *For any function f and g,*

1. $\|\widehat{f+g}\|_1 \leq \|\hat{f}\|_1 + \|\hat{g}\|_1$.
2. $\|\widehat{f\cdot g}\|_1 \leq \|\hat{f}\|_1 \cdot \|\hat{g}\|_1$.

Proof of Theorem 11. Give a deterministic decision tree T computing f, it suffices to show that

$$\sum_{S\subseteq[n]} |\widehat{f}(S)| \leq \text{ size of } T.$$

Let us denote a root to leaf path in T by the leaf. Our goal is to write an polynomial expression for f using the tree T by summing over all leaves. Let ℓ be a leaf in T and consider the path from root to it. We now write indicator expression for this path that evaluates to 1 if the input follows that path in the tree else it evaluates to 0.

For example, suppose a path is given by the following set of edges $(x_1, 1), (x_3, -1), (x_7, -1)$ and $(x_5, 1)$, then the indicator polynomial is given as follows:

$$\left(\frac{1+x_1}{2}\right)\left(\frac{1-x_3}{2}\right)\left(\frac{1-x_7}{2}\right)\left(\frac{1+x_5}{2}\right).$$

Clearly the above expression evaluates to 1 iff $x_1 = 1$, $x_3 = -1$, $x_7 = -1$, and $x_5 = 1$. We will denote such an expression for a leaf ℓ by $\mathbb{1}_\ell$, and the value that f takes at that leaf by $f(\ell)$.

Therefore, we can write an polynomial expression for f as follows:

$$f(x) = \sum_{\ell\text{: leaves in } T} f(\ell)\cdot \mathbb{1}_\ell.$$

Now using Proposition 12, it follows that

$$\|\hat{f}\|_1 \leq \text{ \# leaves in } T.$$

□

We now give an explicit Boolean function whose ℓ_1-norm is large compared to its DNF/CNF size. The *NAND* function on two bits is given by $\neg(x \wedge y)$. Consider a binary tree T of depth h, where the internal nodes are labeled by *NAND* gates and leaves are labeled by distinct variables. Denote the function computed at the root node by G_h. This is known as the *iterated NAND function*.

Theorem 13. *Let* $N = \mathsf{dnf}(G_h) + \mathsf{dnf}(\neg G_h)$*. Then,*

$$\mathsf{L}^{\mathsf{dt}}(G_h) \geq N^{\Omega(\log N)}.$$

Proof. Choose $h = \log n$. Thus, the iterated NAND function is defined on n variables. We now claim the following.

Claim 3.1. $\mathsf{dnf}(G_h) \leq 2^{O(\sqrt{n})}$.

Claim 3.2. $\mathsf{dnf}(\neg G_h) \leq 2^{O(\sqrt{n})}$.

Claim 3.3. $\|\widehat{G_h}\|_1 \geq 2^{\Omega(n)}$.

It is easily seen, using Theorem 11, that the three claims above imply the theorem. We now prove the third claim. (The first two claims are left as an exercise.)

Proof of Claim 3.3: We know that $G_h = \text{NAND}(G^{(1)}_{h-1}, G^{(2)}_{h-1})$ where $G^{(1)}_{h-1}$ and $G^{(2)}_{h-1}$ are defined on disjoint sets of variables of size $n/2$, say x and y. Then,

$$G_h(x, y) = \frac{1}{2}\left(G^{(1)}_{h-1}(x)G^{(2)}_{h-1}(y) - G^{(1)}_{h-1}(x) - G^{(1)}_{h-1}(y) - 1\right).$$

Therefore, we have

$$\begin{aligned}
\widehat{G_h}(\emptyset) &= \frac{1}{2}\left(\widehat{G^{(1)}_{h-1}}(\emptyset)\widehat{G_h}(\emptyset) - \widehat{G^{(1)}_{h-1}}(\emptyset) - \widehat{G^{(2)}_{h-1}}(\emptyset) - 1\right), \\
\widehat{G_h}(S) &= \frac{1}{2}\left(\widehat{G^{(1)}_{h-1}}(S)\widehat{G^{(2)}_{h-1}}(\emptyset) - \widehat{G^{(1)}_{h-1}}(S)\right), \\
\widehat{G_h}(T) &= \frac{1}{2}\left(\widehat{G^{(1)}_{h-1}}(\emptyset)\widehat{G^{(2)}_{h-1}}(T) - \widehat{G^{(2)}_{h-1}}(T)\right), \\
\widehat{G_h}(S \cup T) &= \frac{1}{2}\ \widehat{G^{(1)}_{h-1}}(S) \cdot \widehat{G^{(2)}_{h-1}}(T),
\end{aligned}$$

where $S \neq \emptyset$ is a subset of x variables and $T \neq \emptyset$ is a subset of y variables. Hence we get that the following way to compute to the ℓ_1-norm of G_h.

$$\begin{aligned}
\|\widehat{G_h}\|_1 = &\frac{1}{2}\left(\|\widehat{G^{(1)}_{h-1}}\|_1 - |\widehat{G^{(1)}_{h-1}}(\emptyset)|\right) \cdot \left(\|\widehat{G^{(2)}_{h-1}}\|_1 - |\widehat{G^{(2)}_{h-1}}(\emptyset)|\right) \\
&+ \frac{1}{2}\left(1 - \widehat{G^{(2)}_{h-1}}(\emptyset)\right) \cdot \left(\|\widehat{G^{(1)}_{h-1}}\|_1 - |\widehat{G^{(1)}_{h-1}}(\emptyset)|\right) \\
&+ \frac{1}{2}\left(1 - \widehat{G^{(1)}_{h-1}}(\emptyset)\right) \cdot \left(\|\widehat{G^{(2)}_{h-1}}\|_1 - |\widehat{G^{(2)}_{h-1}}(\emptyset)|\right) \\
&+ |\widehat{G_h}(\emptyset)|.
\end{aligned}$$

Therefore, moving the empty coefficient of G_h on the left hand side we get

$$\begin{aligned}\|\widehat{G_h}\|_1 - |\widehat{G_h}(\emptyset)| = &\frac{1}{2}\left(\|\widehat{G^{(1)}_{h-1}}\|_1 - |\widehat{G^{(1)}_{h-1}}(\emptyset)|\right)\cdot\left(\|\widehat{G^{(2)}_{h-1}}\|_1 - |\widehat{G^{(2)}_{h-1}}(\emptyset)|\right)\\ &+\frac{1}{2}\left(1-\widehat{G^{(2)}_{h-1}}(\emptyset)\right)\cdot\left(\|\widehat{G^{(1)}_{h-1}}\|_1 - |\widehat{G^{(1)}_{h-1}}(\emptyset)|\right)\\ &+\frac{1}{2}\left(1-\widehat{G^{(1)}_{h-1}}(\emptyset)\right)\cdot\left(\|\widehat{G^{(2)}_{h-1}}\|_1 - |\widehat{G^{(2)}_{h-1}}(\emptyset)|\right) \qquad (3)\end{aligned}$$

Since the second and third summands on the right hand side is non-negative, we get the following recurrence for the growth of $\|\widehat{G_h}\|_1 - |\widehat{G_h}(\emptyset)|$,

$$T(h) \geq \frac{1}{2}\cdot T(h-1)^2.$$

The above recurrence solves to $T(h) \geq 2^{2^h}/2^h$. Putting $h = \log n$, we obtain the claim. □