Boolean function complexity

Lecturer: *Nitin Saurabh* Meeting: 6
Scribe: *Nitin Saurabh* 29.05.2019

In the last lecture we saw four different complexity measures associated with Boolean functions, namely decision tree complexity, certificate complexity, sensitivity, and block sensitivity. We further study relationships among them.

1 Relationships between different complexity measures

A sensitive block S for f at x is said to be *minimal* if for all $i \in S$, $S' := S \setminus \{i\}$ is not a sensitive block.

Lemma 1. *Let f be any Boolean function and S be a minimal sensitive block at x, then $|S| \leq \mathsf{s}(f, x^S)$.*

Proof. Since S is minimal at x, we have for all $i \in S$, $f(x^S) \neq f(x^{S'}) = f(x)$, where $S' = S \setminus \{i\}$. Thus, every bit of S is a sensitive bit for f at x^S. □

We now use the above observation to bound the certificate complexity in terms of the block sensitivity.

Theorem 2 (Nisan)**.** *For any f and any input x, $\mathsf{Cert}(f, x) \leq \mathsf{s}(f) \cdot \mathsf{bs}(f, x)$.*

Proof. Let $S_1, S_2, \ldots, S_t$ be a collection of disjoint minimal sensitive blocks at x such that $t = \mathsf{bs}(f, x)$. Consider the subset $T := S_1 \cup \cdots \cup S_t$ with the assignment α such that $\alpha|_{S_i} = x|_{S_i}$, for all $1 \leq i \leq t$. From Lemma 1, it follows that $|T| \leq \mathsf{s}(f) \cdot \mathsf{bs}(f, x)$.

We now show that, in fact, (T, α) is a certificate for x with respect to f. We prove by contradiction. Suppose not, then there exists an input y that is consistent with (T, α) but $f(y) \neq f(x)$. Define S_{t+1} to be the subset of variables where y and x differ. Clearly S_{t+1} is disjoint from all subsets S_j, $j \leq t$. But then we have found $t+1$ disjoint sensitive blocks for f at x, thereby contradicting the fact that $t = \mathsf{bs}(f, x)$. □

Recall in the last lecture we had seen $\mathsf{D}^{\mathsf{dt}}(f) \leq \mathsf{Cert}^0(f) \cdot \mathsf{Cert}^1(f)$. Now using Theorem 2, we obtain $\mathsf{D}^{\mathsf{dt}}(f) \leq \mathsf{s}(f) \cdot \mathsf{bs}(f) \cdot \mathsf{s}(f) \cdot \mathsf{bs}(f) = \mathsf{s}(f)^2 \cdot \mathsf{bs}(f)^2 \leq \mathsf{bs}(f)^4$. We now improve this relationship to $\mathsf{D}^{\mathsf{dt}}(f) \leq \mathsf{bs}(f)^3$.

Theorem 3 (Beals et al. 1998)**.** *For any Boolean function f,*

$$\mathsf{D}^{\mathsf{dt}}(f) \leq \mathsf{Cert}^1(f) \cdot \mathsf{bs}(f).$$

Proof. Given an x, we describe a decision tree algorithm to compute $f(x)$. It maintains a set $\mathcal{X} \subseteq \{0,1\}^n$ consisting of all inputs that are consistent with the replies to queries made so far. Initially $\mathcal{X} = \{0,1\}^n$.

1. Repeat the following $\mathsf{bs}(f)$ times:
   If the function is constant on $\mathcal{X}$, then return this value and stop. Otherwise, pick a 1-certificate consistent with the queries made so far, and query all variables in this certificate. If the queried values agree with the assignment given by the certificate then return 1 and stop. If not, then prune $\mathcal{X}$ to be the remaining set of inputs consistent with the answers to the queried variables.

2. Pick a $y \in \mathcal{X}$ and return $f(y)$.

It is clear that the algorithm queries at most $\mathsf{Cert}^1(f) \cdot \mathsf{bs}(f)$ variables, since the algorithm runs for $\mathsf{bs}(f)$ times and each time it queries at most $\mathsf{Cert}^1(f)$ variables.

It remains to show the correctness of the algorithm. Again it is easy to that if the algorithm returns in Stage 1 then it is correct. We now proceed to argue the same when it returns in Stage 2.

We prove by contradiction. Suppose not, i.e., there exists an input y consistent with all the queries made in Stage 1 but $f(y) \neq f(x)$. Wlog we assume that $f(y) = 1$, and so $f(x) = 0$. Let $t = \mathsf{bs}(f)$ and $(S_1, \alpha_1), \ldots, (S_t, \alpha_t)$ be t 1-certificates that were queried in Stage 1. Therefore, we know that $y|_{S_j} = x|_{S_j} \neq \alpha_j,\ 1 \leq j \leq t$. Let S_{t+1} be the set of indices where x and y differs. Then, S_{t+1} must be disjoint from S_j, for all $1 \leq j \leq t$. Clearly S_{t+1} is a sensitive block for f at x. We now find t sets $T_j \subseteq S_j$, $1 \leq j \leq t$ such that they are mutually disjoint and are sensitive for f at x. Assuming we find such sets, we would have $t+1$ mutually disjoint sets $T_1, \ldots, T_t, S_{t+1}$ that are sensitive for f at x, and thereby contradicting the fact that $t = \mathsf{bs}(f)$. Define T_j to be the set of indices where $x|_{S_j}$ differs from α_j. Clearly T_j is a sensitive block for f at x, since flipping the bits at T_j makes (S_j, α_j) a consistent 1-certificate. We now claim that in fact T_j's are disjoint too. This is because 1-certificates that are consistent with queries made so far are picked in Stage 1. Hence, α_2 must agree with x on $S_1 \cap S_2$, α_3 must agree with x on $(S_1 \cup S_2) \cap S_3$, and so on and so forth. Therefore, we obtain the contradiction. □

Using Theorem 2 with the above theorem, we obtain $\mathsf{D}^{\mathsf{dt}}(f) \leq \mathsf{Cert}^1(f) \cdot \mathsf{bs}(f) \leq \mathsf{s}(f) \cdot \mathsf{bs}(f) \cdot \mathsf{bs}(f) \leq \mathsf{bs}(f)^3$. The following question remains open.

Open Problem 1.1. *Is* $\mathsf{D}^{\mathsf{dt}}(f) = O(\mathsf{bs}^2(f))$ *?*

2 Polynomial representation of Boolean functions

Let $p(x_1, \ldots, x_n)$ be an n-variate polynomial over $\mathbb{R}$. We say that p represents a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ if for all $x \in \{0,1\}^n$, $p(x) = f(x)$. That is, on all Boolean inputs p agrees with f. We know, for $x_i \in \{0,1\}$, $x_i^2 = x_i$, and since we are only concerned with evaluation over $\{0,1\}$, we can assume wlog that p is a *multilinear* polynomial. The next lemma says that if we choose to represent a Boolean function by a multilinear polynomial, then there exists a unique such polynomial.

Lemma 4. *For every Boolean function f there exists a* unique *multilinear polynomial p representing f.*

Proof. Suppose not. Let p_1 and p_2 be two distinct multilinear polynomials that represent f. That is, $p_1 - p_2 \not\equiv 0$ and $p_1(y) = p_2(y) = f(y)$ for all $y \in \{0,1\}^n$.

Let M be a monomial of minimal degree in $p_1 - p_2$ and let $c_M \neq 0$ be its coefficient. We now set the variables to 1 iff it belongs to the monomial M. On such a setting, $p_1 - p_2$ evaluates to c_M. However, since its a Boolean input $p_1 - p_2$ must also be zero, and thereby we reach the contradiction. □

We now define an important algebraic complexity measure associated with Boolean functions.

Definition 5. *The* degree *of a Boolean function f, denoted $\mathsf{deg}(f)$, is the degree of the multilinear polynomial that represents f.*

In particular for any Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, $\mathsf{deg}(f) \leq n$. It is also easily seen that $\mathsf{deg}(\mathsf{AND}_n) = \mathsf{deg}(\mathsf{OR}_n) = n$.

Theorem 6. *For any Boolean function f, $\mathsf{deg}(f) \leq \mathsf{D}^{\mathsf{dt}}(f)$.*

Proof. Let T be a depth optimal decision tree for f. Consider a root to leaf path given by $(x_1, b_1), (x_2, b_2), \ldots, (x_d, b_d)$ where $b_j \in \{0,1\}$ is the value set to the variable x_j. We now define a $\{0,1\}$-indicator polynomial for this path: $\prod_{j:b_j=1} x_j \prod_{j:b_j=0}(1-x_j)$. This polynomial evaluates to 1 if the input follows the chosen path in the decision tree T, and otherwise it evaluates to 0. Also the degree of this indicator polynomial is at most d.

We now define $p = \sum_{\ell \text{ is a 1-leaf in T}} \mathbb{1}_\ell$, where $\mathbb{1}_\ell$ is the indicator polynomial for the root to ℓ path. It is now easily seen that $p(x) = f(x)$ for all $x \in \{0,1\}^n$, and $\mathsf{deg}(p) \leq \mathsf{D}^{\mathsf{dt}}(f)$. Thus, $\mathsf{deg}(f) \leq \mathsf{D}^{\mathsf{dt}}(f)$. □

Let us recall the Tribes function.

Example (Tribes)**.**

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

It is not hard to see that $\mathsf{s}(\mathsf{Tribes}) = \mathsf{bs}(\mathsf{Tribes}) = \mathsf{Cert}(\mathsf{Tribes}) = \sqrt{n}$*, while the* $\mathsf{deg}(\mathsf{Tribes}) = n$*.*

So the degree is quadratically larger than block sensitivity. Can it be lower? Nisan and Szegedy gave an example of a function with full sensitivity and but polynomially low degree.

Example (Nisan and Szegedy 1994)**.** *Consider the three variable Boolean function E_{12} defined as follows*

$$\mathsf{E}_{12}(x_1, x_2, x_3) = 1 \text{ iff } x_1 + x_2 + x_3 \in \{1, 2\}.$$

We have $\mathsf{deg}(\mathsf{E}_{12}) = 2$*, since*

$$\mathsf{E}_{12}(x_1, x_2, x_3) = x_1 + x_2 + x_3 - x_1x_2 - x_2x_3 - x_3x_1.$$

Now consider a ternary tree of depth k. Label the leaves of the ternary tree by distinct variables. The rest of the nodes of the tree are labeled by E_{12}. Denote the function computed at the root of this ternary tree by E_{12}^k. It is a function on 3^k variables. It is also easily seen that $\mathsf{deg}(\mathsf{E}_{12}^k) = 2^k$. Also observe that at all zero inputs, every bit is sensitive. That is, $\mathsf{s}(\mathsf{E}_{12}^k, 0^n) = n$ where $n := 3^k$. This follows from the definition of E_{12} and the ternary structure of the tree. Therefore, we have

$$\mathsf{s}(\mathsf{E}_{12}^k) = \mathsf{bs}(\mathsf{E}_{12}^k) = \mathsf{Cert}(\mathsf{E}_{12}^k) = n, \text{ whereas } \mathsf{deg}(\mathsf{E}_{12}^k) = n^{1/\log 3} = n^{0.631\ldots}.$$

They also showed that the degree can not be much smaller than this.

Theorem 7 (Nisan and Szegedy 1994)**.** *Let $f\colon \{0,1\}^n \to \{0,1\}$ be any Boolean function. Then,*

$$\mathsf{deg}(f) \geq \sqrt{\frac{\mathsf{bs}(f)}{2}}.$$

Proof. The proof strategy is to first define a Boolean function g on $\mathsf{bs}(f)$ many variables using f such that (i) $\mathsf{deg}(g) \leq \mathsf{deg}(f)$, and (ii) $\mathsf{s}(g, 0^{\mathsf{bs}(f)}) = \mathsf{bs}(f)$. Then, show that such a Boolean function has degree at least $\sqrt{\mathsf{bs}(f)/2}$.

Defining $g\colon \{0,1\}^{\mathsf{bs}(f)} \to \{0,1\}$. Let $a \in \{0,1\}^n$ be an input such that $\mathsf{bs}(f,a) = \mathsf{bs}(f) = t$, and $S_1, \ldots, S_t$ be the mutually disjoint sets of sensitive blocks at a. We assume wlog that $f(a) = 0$. We now define a Boolean function $g(y_1, y_2, \ldots, y_t)$ using the following mapping ρ between variables:

$$\rho(x_i) = \begin{cases} a_i & \text{if } i \notin S_1 \cup \cdots \cup S_t, \\ y_j & \text{if } i \in S_j \text{ and } a_i = 0, \\ 1 - y_j & \text{if } i \in S_j \text{ and } a_i = 1. \end{cases}$$

Then, $g(y_1, \ldots y_t) := f(\rho(x))$. Clearly $\mathsf{deg}(g) \leq \mathsf{deg}(f)$, since if p is a polynomial representing f then $p(\rho(x))$ is a polynomial representing g. We now observe the second property. Note that $g(0, \ldots, 0) = f(a) = 0$. Furthermore, for all unit vectors $e_i \in \{0,1\}^t$, $g(e_i) = f(a^{S_i}) = 1$. Hence, $g(e_i) \neq g(0)$ for all $i \in [t]$. This completes the definition of g.

We now show that $\mathsf{deg}(g) \geq \sqrt{t/2}$, and that will complete the proof. For this purpose we introduce a technique known as *symmetrization*, due to Minsky and Papert. Let $p\colon \mathbb{R}^n \to \mathbb{R}$ be a polynomial. Given a permutation $\sigma \in \mathcal{S}_n$ and an input $(x_1, x_2, \ldots, x_n)$, let $\sigma(x)$ denote the input $(x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n)})$. Then, the symmetrization p^{sym} of p is defined as follows

$$p^{\mathrm{sym}}(x) := \frac{1}{n!} \sum_{\sigma \in \mathcal{S}_n} p(\sigma(x)).$$

Note that $\deg(p^{\mathrm{sym}}) \leq \deg(p)$.

The usefulness of symmetrization follows from the following lemma.

Lemma 8. *Let $p\colon \mathbb{R}^n \to \mathbb{R}$ be a multilinear polynomial. Then, there exists a univariate polynomial $\tilde{p}$ of degree at most the degree of p such that*

$$\tilde{p}\left(\sum_i x_i\right) = p^{sym}(x_1, x_2, \ldots, x_n) \text{ for all } x \in \{0,1\}^n.$$

Let us continue with the proof of the theorem, and we shall prove the lemma afterwards. We also need the following technical theorem from approximation theory.

Theorem 9 (Ehlich and Zeller 1964, Rivlin and Cheney 1966)**.** *Let $p\colon \mathbb{R} \to \mathbb{R}$ be a univariate polynomial such that $b_1 \leq p(i) \leq b_2$ for every integer $0 \leq i \leq n$, and its derivative has $|p'(x)| \geq c$ for some real $0 \leq x \leq n$. Then,*

$$\deg(p) \geq \sqrt{\frac{cn}{b_2 - b_1 + c}}.$$

Let P_g be the polynomial representing the Boolean function g. Let $\tilde{P}_g$ be the univariate polynomial obtained from the application of Lemma 8 to P_g. Then we note that $0 \leq \tilde{P}_g(i) \leq 1$ for all integer $0 \leq i \leq t$. Also $\tilde{P}_g(0) = 0$ and $\tilde{P}_g(1) = 1$. Now from the mean value theorem it follows that there exists $x \in (0,1)$ such that $|\tilde{P}_g'(x)| \geq 1$. Now applying Theorem 9 to $\tilde{P}_g$, we obtain that $\deg(\tilde{P}_g) \geq \sqrt{t/2}$. Since $\deg(\tilde{P}_g) \leq \deg(P_g)$ which in turn is at most $\mathsf{deg}(f)$, we obtain the theorem. □

We now prove the lemma.

Proof of Lemma 8. Let d be the degree of p^{sym} which is at most the degree of p. Let $p^{=k}(x)$ denote the polynomial given by the sum of all $\binom{n}{k}$ products $\prod_{i \in S} x_i$ where $|S| = k$. Since p^{sym} is symmetrical, it can be shown using induction that

$$p^{\text{sym}}(x) = c_0 + c_1 p^{=1}(x) + c_2 p^{=2}(x) + \cdots + c_d p^{=d}(x),$$

where $c_i \in \mathbb{R}$. Let $z = \sum_i x_i$. Now note that for $x \in \{0,1\}^n$, $p^{=k}(x) = \binom{z}{k}$, which is a polynomial of degree k in the variable z. Therefore, the univariate polynomial $\tilde{p}$ is defined to be

$$\tilde{p}(z) = c_0 + c_1 \binom{z}{1} + c_2 \binom{z}{2} + \cdots + c_d \binom{z}{d}.$$

□