Lecture 7: Sensitivity based measures and relations between them

Rajat Mittal

IIT Kanpur

We will introduce a few other complexity measures on Boolean function related to the decision tree model: sensitivity, block sensitivity and certificate complexity. They are all lower bounds on deterministic query complexity, some of them are a lower bound on randomized query complexity too. All of these measures and other measures introduced before, with the exception of sensitivity, were known to be polynomially related from a long time.

In this lecture we will see these basic complexity measures and relationships between them.

1 Sensitivity and related measures

Once again, our central object of study is a Boolean function $f : \{0,1\}^n \to \{0,1\}$, where every input x is a string of n bits. A position in the input x can be indexed by an $i \in [n]$. The lower bounds we have seen on OR function (and sometimes Parity), do not depend upon the exact function, but the fact that there is an input where changing a bit changes the function. To formalize this, we define when an input is sensitive at an index.

Fix a Boolean function $f : \{0,1\}^n \to \{0,1\}$. For an index $i \in [n]$, define x^i to be the input where the i-th bit is flipped in x. An index i is called *sensitive* for input x if $f(x) \neq f(x^i)$. The local sensitivity $\mathrm{s}(f,x)$ at an input x is the number of sensitive indices in the input x. The *sensitivity* of the function, $\mathrm{s}(f)$, is the maximum possible sensitivity $\mathrm{s}(f,x)$ over all inputs x.

Exercise 1. Can you give an example of a function and an input where sensitivity is n (what about 1)?

Exercise 2. Show that sensitivity is a lower bound on deterministic query complexity.

The average sensitivity of the function f is,

$$\mathrm{Avg}-\mathrm{s}(f) = \frac{1}{2^n}\sum_x \mathrm{s}(f,x).$$

Exercise 3. What is the relation between average sensitivity and Influence?

Exercise 4. Can you think of a function whose sensitivity is $o(n)$? Can you think of a function and an input whose sensitivity is 0?

A very similar, but slightly complex, measure is called *block sensitivity*.

For a block of indices $B \subseteq [n]$, define x^B to be the input where *all* bits in block B are flipped. As before, a block B is called *sensitive* for input x if $f(x) \neq f(x^B)$. The local block sensitivity $\mathrm{bs}(f,x)$ at an input x is the maximum number of disjoint sensitive blocks possible in the input x respectively. As you might guess, the block sensitivity of a function is the maximum possible block sensitivity $\mathrm{bs}(f,x)$ over all inputs x.

Exercise 5. Show that block sensitivity is a lower bound on deterministic query complexity.

We moved from sensitivity to block sensitivity because we hope that it might allow us to give better lower bounds.

Exercise 6. Show that $\mathrm{bs}(f,x) \geq \mathrm{s}(f,x)$. Can you show a function and an input where this inequality is strict?

Can we improve the lower bound further?

Certificate complexity: Fix a Boolean function $f : \{0,1\}^n \to \{0,1\}$. For any input x, a *certificate* is a set of indices $C \subseteq [n]$ such that for any input $y \in \{0,1\}^n$, if $x_i = y_i \quad \forall i \in C$, then $f(x) = f(y)$. The size of the smallest certificate at input x is its local certificate complexity $C(f,x)$. The certificate complexity of f is the maximum $C(f,x)$ over all inputs x.

Exercise 7. What is the certificate complexity of OR and MAJORITY?

You have seen certificates before. For example, if you look at the path of any deterministic tree, it is actually a certificate (for any input on the leave).

Exercise 8. Show that certificate complexity is a lower bound on deterministic tree complexity? Is it a lower bound on randomized tree complexity by the same argument?

What is the relationship between certificate complexity and sensitivity/block sensitivity?

Exercise 9. Fix an input x, can you show that any sensitive bit on x should be part of any certificate on x?

Notice that the above exercise shows that the certificate complexity of a function is bigger than its sensitivity (prove it formally). Essentially the same argument works for block sensitivity.

Keep a function f and an input x in mind, suppose $C \subseteq [n]$ is a certificate and $B_1, B_2, \cdots B_{\mathrm{bs}(f,x)}$ be the blocks which give the maximum block sensitivity.

Exercise 10. What can you say about $C \cap B_i$?

Observe that the intersection can't be empty, proving $C(f,x) \geq \mathrm{bs}(f,x)$.

We have introduced many complexity measures with value between 0 and n. There are some trivial (and some not so trivial) dependencies known between them. Figure 1 summarizes them.

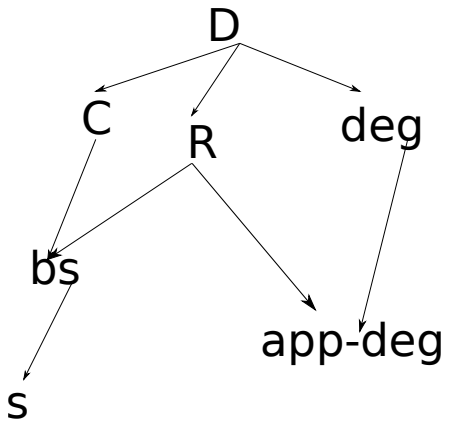

Fig. 1. Relations between complexity measures. Arrow from A to B implies $A = \Omega(B)$.

Exercise 11. Which of the relations are not clear to you in the above figure.

Most of the inequalities follow from definition. The lower bounds on $D(f)$ and $C(f)$ have been discussed in this lecture before. We are left with $R(f) \geq \frac{1}{3}\text{bs}(f)$, this follows from the observation that every block should have been queried with constant probability in any randomized algorithm.

Let us formally prove this. The claim is: every sensitive block, say B, needs to be queried at least with probability 1/3. Let x and $x^{\oplus B}$ be the two inputs which only differ on B but have different function value. Let p be the total probability of decision trees which query a variable from B (we want to show that it is bigger than 1/3).

The remaining decision trees (having probability $1-p$) will output according to x or x^B; suppose less (weight) of them agrees with x. So the correctness probability at x is bounded by $p + (1-p)/2$. Since it should be greater than 2/3, we get $p \geq 1/3$.

So the expected number of queries for any algorithm is at least $\frac{1}{3}\text{bs}(f)$, proving the result (remember that expected queries are smaller than worst case queries).

2 Polynomial relations between complexity measures

We would like to prove that all complexity measures are polynomially related; intuitively, they should be related because they indicate the complexity of a Boolean function.

A polynomial relation between two quantities A and B means,

$$A = O(B^{c_1}) \text{ and } B = O(A^{c_2}),$$

where c_1 and c_2 are two positive numbers. Ideally, we would like to come up with functions which show that these relations are tight.

For example, take the case of $D(f)$ and $C(f)$. We know $C(f) = O(D(f))$ and we will prove in the next subsection that $D(f) = O(C(f)^2)$. These relations are tight, for the first relation we can use OR, for the later one you can use tribes function (assignment).

Whether $C(f)$ is a lower bound on $R(f)$ is still an open question.

We want to polynomially relate all the quantities in Figure 1 (except sensitivity). The strategy would be to upper bound $D(f)$ (the biggest quantity) in terms of two smallest quantities ($\text{bs}(f)$ and $\widetilde{\deg}(f)$). We will start by upper bounding $D(f)$ with $\text{bs}(f)$ and $C(f)$.

The proofs in the next two subsections are inspired from [1].

2.1 Upper bound on deterministic query complexity

The first upper bound will be in terms of certificate complexity and block sensitivity. For a Boolean function $f : \{0,1\}^n \to \{0,1\}$,

$$D(f) \leq \text{bs}(f)C(f). \tag{1}$$

To prove this upper bound, we need to show a deterministic query algorithm which makes at most $\text{bs}(f)C(f)$ queries. The algorithm to compute $f(x)$ is simple to describe, query a consistent 1-certificate (consistent with all the queries till that point) $\text{bs}(f)$ times. If the queries are consistent with a certificate, output $f(x) = 1$. If at any point we don't have a 1-certificate left, output $f(x) = 0$. After $\text{bs}(f)$ iterations, pick any consistent y and output $f(x) = f(y)$.

Notice that at every stage, we can always pick a certificate with size $\leq C(f)$. This is because $C(f) = \max_x C(f,x)$, and a certificate of x will remain a certificate of x when the function is restricted. The certificates are picked $\text{bs}(f)$ times, implying that the number of queries are bounded by $\text{bs}(f)C(f)$. We only need to prove the correctness of the algorithm.

If we find a 1-certificate or if there is no such certificate, clearly algorithm's output is correct. Following claim takes care of the remaining case.

Claim. If the algorithm has still not given an output after $\mathrm{bs}(f)$ iterations, all remaining consistent inputs have the same value.

Proof. Let $b = \mathrm{bs}(f)$. Assume that the claim is false. This means there exist consistent y, z such that $f(y) = 0$ and $f(z) = 1$. We will create $b + 1$ disjoint sensitive blocks for y, showing contradiction.

Let the first b certificates be $C_1, C_2, \cdots C_b$ and C_{b+1} be the 1-certificate for z. The i-th sensitive block for y is defined to be indices at which C_i and y differ. Clearly there are $b + 1$ blocks and none of the block is empty $f(y) = 0$.

To show the disjointness, consider the k-th certificate C_k. Notice that some variables of C_k might have been queried earlier than the k-th iteration. Though, all elements in B_k will come from variables queried in the k-th iteration (C_k was consistent with whatever was queried before). This implies that all blocks are disjoint.

□

Note 1. This also shows that $D(f) = O(C(f)^2)$.

For polynomial relationship between $D(f)$ and $\mathrm{bs}(f)$, we would like to bound $D(f)$ in terms of $\mathrm{bs}(f)$ alone. This will be accomplished in the next section by bounding $C(f)$ in terms of $\mathrm{bs}(f)$.

2.2 Upper bound on certificate complexity

The second upper bound will be on certificate complexity. For a Boolean function $f : \{0,1\}^n \to \{0,1\}$,

$$C(f, x) \leq \mathrm{bs}(f, x)\mathrm{s}(f). \tag{2}$$

Let $b = \mathrm{bs}(f, x)$. For an input x, let $B_1, B_2, \cdots, B_b$ be the disjoint sensitive blocks achieving $\mathrm{bs}(f, x)$.

Claim. $\dot{\bigcup}_{i=1}^{b} B_i$ is a certificate for x with length $\mathrm{bs}(f, x)\mathrm{s}(f)$.

Proof. It is an easy exercise to show that $\dot{\bigcup}_{i=1}^{b} B_i$ is a certificate for x.

Exercise 12. Can you increase the number of disjoint sensitive blocks if it is not a certificate?

For the length, it is enough to show that the length of each block B_i is bounded by $\mathrm{s}(f)$. Notice that, each block B_i can be assumed *minimal*. This means, no subset of B_i is sensitive.

Exercise 13. Why can we make this assumption?

If B_i is minimal, then every variable of B_i is sensitive for x^{B_i} (the input with block B_i flipped). In other words, $|B_i| \leq \mathrm{s}(f)$.

Note 2. We can only bound the size of a minimal sensitive block by $\mathrm{s}(f)$ and not $\mathrm{s}(f, x)$.

Since there are at most $\mathrm{bs}(f, x)$ many blocks, we get a certificate of size $\mathrm{bs}(f, x)\mathrm{s}(f)$. Hence proved. □

The claim shows that $C(f) \leq \mathrm{bs}(f)\mathrm{s}(f)$. Combining with Equation 1, we get

$$D(f) \leq \mathrm{bs}(f)^2\mathrm{s}(f). \tag{3}$$

This shows that the biggest quantity $D(f)$ is polynomially bounded by the smallest quantity on the block sensitivity branch of Figure 1. Next, we will upper bound $\mathrm{bs}(f)$ with $\widetilde{\deg}(f)$ to show that all measures (except sensitivity) are polynomially related.

Exercise 14. Why is that sufficient?

2.3 Upper bound on Block sensitivity by approximate degree

The aim of this subsection is to show

$$\mathrm{bs}(f) = O(\widetilde{\deg}(f)^2). \tag{4}$$

The result was proved by Nisan and Szegedy [2]. Fortunately, we have seen this result already (almost)!

Exercise 15. Can you remember where?

It is a slight extension of the result that approximate degree of OR_n is $\sqrt{n}$. While showing a lower bound on approximate degree of OR, we observed that it even worked for Promise OR, i.e., the only properties used in the proof were,

- function value is different at Hamming weight 0 and Hamming weight 1,
- and function value is bounded ($\{0,1\}$) at every other point.

In other words, the lower bound proof only relies on the fact that there is a particular input z (all 0's in case of OR) such that n Hamming distance one inputs from z change their value. Specifically, if the function has maximum sensitivity at z, we can consider the *translated function*

$$f'(x) = f(x \oplus z).$$

Convince yourself that f' has a Promise OR structure on $\mathrm{s}(f)$ many bits.

Exercise 16. Show that the approximate degree of f is same as f'. Show that this implies,

$$\mathrm{s}(f) = O(\widetilde{\deg}(f)^2).$$

In other words, if you look at the function OR, it has kind of a flower structure (center at all 0 input and n petals coming out of it). For a function with sensitivity $\mathrm{s}(f)$, a similar *translated* flower structure is there with $\mathrm{s}(f)$ petals.

The bound on block sensitivity, Equation 4, follows by a similar argument where each block corresponds to a petal. Since we can always translate a function without affecting its block sensitivity and approximate degree, assume that f achieves its block sensitivity at 0 (all 0) input.

Formally, let $B_1, B_2, \cdots, B_b$ be maximum disjoint sensitive blocks possible for f where $b = \mathrm{bs}(f, 0) = \mathrm{bs}(f)$. Fix up all other variables which are not in any of the sensitive blocks according to x. We will abuse the notation and call the restricted function f.

Exercise 17. Show that the approximate degree of a restriction is always less than the approximate degree of the original function, so it is enough to show Equation 4 for the restricted function.

Construct a function f' on b variables such that

$$f'(z) = f(x_{1,z_1}, x_{2,z_2}, \cdots, x_{b,z_b}),$$

where $x_{i,0}$ sets variables of block B_i according to x and $x_{i,1}$ sets variables of block B_i according to x^{B_i}.

Exercise 18. Show that the approximate degree of f' is at least $\sqrt{b}$ (notice the flower structure).

The only thing left to show is that $\widetilde{\deg}(f') \leq \widetilde{\deg}(f)$ (to prove Equation 4).

Exercise 19. Show this by converting an approximating polynomial of f to f'. Hint: use the fact that $z_i = 0$ (respectively $z_i = 1$) means all variables in that block are 0 (respectively 1).

This shows that all complexity measures in Figure 1 are polynomially related (except sensitivity).

Symmetric functions: The bounds we have proved above can be improved (a lot) for the special class of symmetric functions. In particular, it can be shown that almost all the complexity measures are $\Theta(n)$ for any non-constant symmetric function. The only exception is approximate degree, which is $\Omega(\sqrt{n})$. The bound is tight for the OR function.

Monotone functions: Again, the specific case of monotone functions allow us to simplify things.

Theorem 1. *For a monotone Boolean function f, $C(f) = \mathrm{bs}(f) = \mathrm{s}(f)$.*

Proof. Since block sensitivity lies between sensitivity and certificate complexity, we just need to prove that $C(f) = \mathrm{s}(f)$.

Let C be a minimal 1-certificate. Given that f is monotone and C is minimal, C only specifies some indices to be 1. By minimality of C, all indices in C (having assigned 1) are sensitive. This shows that sensitivity is at least the size of C.

Similarly, let C be a minimal 0-certificate. Once again C only specifies indices to be 0 and every such index is sensitive. So, sensitivity is at least the size of C.

Since sensitivity is equal or bigger than any minimal certificate, $\mathrm{s}(f) = C(f)$. □

Though, unlike symmetric case, sensitivity need not be full.

Exercise 20. What is the sensitivity of Tribes function? Is Tribes monotone?

Remember that one of the reason to study monotone functions was that many graph properties were monotone. It can be shown that $D(f)$ for any monotone graph property is $\Theta(n)$. It is a big open problem if it is exactly n, and is called the *evasiveness conjecture* [1].

3 Assignment

Exercise 21. Show that $\mathrm{s}(f), \mathrm{bs}(f), C(f)$ are $\Theta(n)$ for any symmetric function on n variables.

Exercise 22. Find the $\mathrm{bs}f$ and $\mathrm{s}f$ for the addressing function.

Exercise 23. Can you think of an f and z to separate $\mathrm{s}_z(f)$ and $\mathrm{bs}_z(f)$.

Exercise 24. Tribes function is $\mathrm{AND}_n \circ \mathrm{OR}_n$. Show that D of tribes is $\Theta(n^2)$ and C of tribes is $\Theta(n)$.

Exercise 25. Difficult: Show a symmetric function where $\mathrm{bs}(f) > \mathrm{s}(f)$.

References

1. H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science, Volume 288, Issue 1, Pages 21-43*, 2002.
2. N. Nisan and M. Szegedy. On the degree of boolean functions as real polynomials. *Computational Complexity, volume 4, pages 301–313*, 1994.