Lecture 12: Sensitivity, related measures and Huang's proof

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. All of these measures and other measures introduced before, with the exception of sensitivity, were known to be polynomially related from a long time.

We will give one proof (out of many possible) of this polynomial relationship. In a breakthrough, Huang [2] recently showed that even sensitivity is polynomially related to all these measures (called sensitivity conjecture). For a perspective on this result, the conjecture was open for around 30 years, and finally Huang settled it by giving a beautiful proof which can arguably fit in one page.

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 $i \in [n]$.

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)?

The average sensitivity of the function f is,

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

Exercise 2. What is the relation between average sensitivity and Influence?

Exercise 3. 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 4. Show that $\mathrm{bs}(f,x) \geq \mathrm{s}(f,x)$. Can you show a function and an input where this inequality is strict?

Separation between $\mathrm{bs}(f)$ *and* $\mathrm{s}(f)$*:* The function constructed here is motivated from Rubinstein's result [3]. It seems that $\mathrm{bs}(f)$ can be much bigger than $\mathrm{s}(f)$. Can you think of a function where $\mathrm{bs}(f) > \mathrm{s}(f)$? It is a difficult question, let us take a look at a simpler question.

Define $\mathrm{s}_z(f)$ for $z \in \{0,1\}$ to be $\max_{x:\ f(x)=z} \mathrm{s}(f,x)$, we can similarly define $\mathrm{bs}_z(f)$. Can you think of an f and z to separate $\mathrm{s}_z(f)$ and $\mathrm{bs}_z(f)$? This is a question in the assignment, you might want to try it before going further.

We will give one such example, define $C_2 : \{0,1\}^n \to \{0,1\}$ to be the function iff there exist a block of two contiguous indices ($i-1$ and i) where the input is 1 and 0 otherwise. You can see that $\mathrm{s}_0(C_2) = 2$, but $\mathrm{bs}_0(C_2) = n/2$.

Exercise 5. Show that $\mathrm{s}_1(C_2) = n$.

So, we still don't get a separation between $\mathrm{bs}(f)$ and $\mathrm{s}(f)$. There is a standard technique to resolve this issue.

Define $g := \mathrm{OR}_n \circ f_m$, that means the arity of the inner function is m and outer function is n. Let us find the (block) sensitivities of this composed function. For a 0-input to g, all internal block of inputs should output 0 on f. The sensitivity of the input is the sum of sensitivities of all blocks.

$$\mathrm{s}_0(g) = n \cdot \mathrm{s}_0(f) \textit{ and } \mathrm{bs}_0(g) = n \cdot \mathrm{bs}_0(f).$$

On the other hand, for a 1-input, we should keep exactly one block outputting 1 on f (why?).

$$\mathrm{s}_1(g) = \mathrm{s}_1(f) \textit{ and } \mathrm{bs}_1(g) = \mathrm{bs}_1(f).$$

Exercise 6. The second part of the equation requires careful argument. Give that argument.

Suppose we compose C_2 with OR $(m = n)$. The sensitivity of the composed function will be $\Theta(n)$ and block sensitivity will be $\Theta(n^2)$.

Exercise 7. Check that changing m, n does not give a better separation.

This is the best possible separation known between $\mathrm{bs}(f)$ and $\mathrm{s}(f)$, it is a big open question to increase this gap. We only know that for any f, $\mathrm{bs}(f) \leq \mathrm{s}(f)^4$ [1].

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 $x_i = y_i \ \ \forall i \in C$, 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 8. 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 9. 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 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.

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. From polynomial method, we know $Q(f) = \Omega(\widetilde{\deg}(f))$. This only leaves $R(f) \geq \mathrm{bs}(f)$, this follows from the observation that every block should have been queried with constant probability in any randomized algorithm. A more direct proof is obtained by introducing fractional certificate complexity $\mathrm{RC}(f)$ and fraction block sensitivity $\mathrm{fbs}(f)$.

$$R(f) \geq \mathrm{RC}(f) = \Theta(\mathrm{fbs}(f)) \geq \mathrm{bs}(f).$$

The proof of these inequalities can be found in Kulkarni and Tal [6].

$$\begin{tikzcd}
 & & & \mathsf{D} \arrow[dll] \arrow[dl] \arrow[drr] & & \\
 & \mathsf{C} \arrow[d] & \mathsf{R} \arrow[dl] \arrow[dr] & & & \mathsf{deg} \arrow[ddl] \\
 & \mathsf{bs} \arrow[dl] & & \mathsf{Q} \arrow[dr] & & \\
\mathsf{s} & & & & \mathsf{app\text{-}deg} &
\end{tikzcd}$$

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

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,

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

where c_1 and c_2 are two natural 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).

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 ($\mathrm{bs}(f)$ and $\widetilde{\deg}(f)$). We will start by upper bounding $D(f)$ with $\mathrm{bs}(f)$ and $C(f)$.

The proofs in the next two subsections are inspired from [4].

2.1 Upper bound on deterministic tree complexity

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

$$D(f) \leq \mathrm{bs}(f)C(f). \qquad (1)$$

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

Notice that at every stage, we can always pick a certificate with size $\leq C(f)$ (since $C(f) = \max_x C(f,x)$). The certificates are picked $\mathrm{bs}(f)$ times, implying that the number of queries are bounded by $\mathrm{bs}(f)C(f)$. We only need to prove the correctness of the algorithm.

Since 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.

Hence proved. □

This also shows that $D(f) = O(C(f)^2)$. We wanted to bound $D(f)$ in terms of $\mathrm{bs}f$ alone, next task is to bound $C(f)$ 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 size of disjoint sensitive blocks if it is not a certificate?

For the length, we will 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, $|B_i| \leq \mathrm{s}(f)$.

Hence proved. □

Combining Equations 1 and 2, 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

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 [5]. Fortunately, we have seen this result (almost) already!

Exercise 15. Can you remember where?

It is a slight extension of showing that approximate degree of OR_n is $\sqrt{n}$. We will only give an idea, for the complete proof, see [5].

It is easier to argue the simpler result,

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

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, 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.

Exercise 16. Can you show

$$\widetilde{\deg}(f) = \Omega(\sqrt{\mathrm{s}(f)}).$$

The bound on block sensitivity, Equation 4, follows by a similar argument where each block corresponds to a petal. Formally, it is shown by constructing a function f' such that f' is like promise OR and $\widetilde{\deg}(f) \geq \widetilde{\deg}(f')$.

Exercise 17. Try doing this on your own, otherwise see [5].

This shows that all complexity measures in Figure 1 are polynomially related (except sensitivity). It took nearly 30 years, and finally in 2019 Huang [2] proved that even sensitivity is polynomially related.

3 Huang's proof of sensitivity conjecture

The result was shown by introducing a new quantity, spectral sensitivity, denoted $\lambda(f)$ (introduced in [2], formalized in [1]). It was a lower bound on sensitivity (follows easily from the definition), recently it has been shown to be a lower bound on approximate degree [1] (*not* needed for the proof of sensitivity conjecture). This modifies the relationship diagram to Figure 2.

Huang provided a polynomial upper bound on degree using $\lambda(f)$.

Exercise 18. Why is that sufficient?

First, we define spectral sensitivity for a Boolean function $f : \{0,1\}^n \to \{0,1\}$. To define spectral sensitivity, we need the concept of *sensitivity graph* of the function f, a subgraph of Boolean hypercube.

Exercise 19. What is Boolean hypercube (as a graph)?

The sensitivity graph of f, say G_f, is a subgraph of Boolean hypercube, i.e., there are 2^n vertices (for each input). An edge x, y is present in G_f iff $f(x) \neq f(y)$ and x, y is an edge in Boolean hypercube (they have Hamming distance 1).

$$\begin{tikzcd}
 & & \mathrm{D} \arrow[dll] \arrow[d] \arrow[drr] & & \\
\mathrm{C} \arrow[d] & & \mathrm{R} \arrow[dll] \arrow[dr] & & \mathrm{deg} \arrow[ddl] \\
\mathrm{bs} \arrow[d] & & & \mathrm{Q} \arrow[d] & \\
\mathrm{s} \arrow[drr] & & & \text{app-deg} \arrow[dl] & \\
 & & \text{sp-s} & &
\end{tikzcd}$$

Fig. 2. Relations diagram with spectral sensitivity. Arrow from A to B implies $A = \Omega(B)$.

Exercise 20. Find a function f whose sensitivity graph is the Boolean hypercube itself.

Exercise 21. How many edges are there in the sensitivity graph of OR_n.

Exercise 22. Show a subgraph of Boolean hypercube which is not a sensitivity graph for any function f.

We are interested in the eigenvalues of the adjacency matrix, say A_f, of the graph G_f. We first notice that the graph G_f is bipartite.

Exercise 23. Show that Boolean hypercube is bipartite.

That means, if u is an eigenvalue of G_f, then so is $-u$ (assignment). That means we can talk about the maximum eigenvalue (without clarifying if absolute value needs to be taken before taking maximum).

The spectral sensitivity of f, called $\lambda(f)$, is the maximum eigenvalue (also called spectral norm) of the adjacency matrix of G_f.

Exercise 24. What is the spectral sensitivity of PARITY?

Since the eigenvalue of a matrix is bounded by the maximum row sum (why), $\lambda(f) \leq \mathrm{s}(f)$. For $\lambda(f) \leq \widetilde{\deg}(f)$, refer to [1]. This completes the relationships given in Figure 2.

The main result of this section is the following upper bound on deg in terms of λ settling sensitivity conjecture.

Theorem 1 ([2]).
For any Boolean function $f : \{0,1\}^n \to \{0,1\}$,

$$\deg(f) \leq \lambda(f)^2.$$

The first simplification is that we can assume $\deg(f) = n$. If not, pick the monomial in the polynomial representation of f with highest degree, and set all other variables to some values. For the restricted function, $\deg(f)$ is same but $\lambda(f)$ can only be smaller (assignment).

That means we can assume $\deg(f) = n$ (any counterexample to Theorem 1 can be converted into a counterexample with full degree). In other words, we just need to prove that $\lambda(f) \geq \sqrt{n}$ when $\deg(f) = n$.

What can we say about sensitivity graph of f when $\deg(f) = n$? Define $V_0 = \{x : f(x) = \text{PARITY}(x)\}$ and $V_1 = \{x : f(x) \neq \text{PARITY}(x)\}$.

Exercise 25. Show that $\deg(f) = n$ is equivalent to saying that $|V_0| \neq |V_1|$.

The problems statement changes to, given that $|V_0| > 2^{n-1}$ (if $|V_0| < |V_1|$ then consider $1-f$), show that $\lambda(f) \geq \sqrt{n}$.

Exercise 26. Show that there is no edge between V_0 and V_1. Inside V_0 (and V_1), the edges are exactly the edges of Boolean hypercube.

This means that the eigenvalues of G_f are union of eigenvalues of the subgraph on V_0 and V_1. For any V with more than half the vertices, we will show that the induced subgraph from Boolean hypercube (say G_V) has eigenvalue more than $\sqrt{n}$. This will finish the proof.

An interesting lemma relates the eigenvalues of the induced subgraph with the eigenvalues of the original graph. It is called *Cauchy's interlacing theorem* [2], we will only use the following special case of it.

Lemma 1. *Let G be a graph on k vertices and its eigenvalues be $\lambda_1 \leq \lambda_2 \cdots \lambda_k$. If G_V is the induced subgraph on V with l vertices, then*

$$\|G_V\| \geq \lambda_l,$$

where $\|G_V\|$ denotes the maximum eigenvalue of G_V.

Proof. The adjacency matrix of G is an $k \times k$ matrix. The eigenvectors corresponding to bigger eigenvalues, $\{\lambda_k, \lambda_{k-1}, \cdots, \lambda_l\}$, span a vector space of dimension $k - l + 1$, say $\mathcal{S}_1$. The vector space corresponding to l standard basis vectors e_v where $v \in V$, say $\mathcal{S}_2$, spans a subspace of dimension l.

Exercise 27. Since the sum of dimensions of $\mathcal{S}_1$ and $\mathcal{S}_2$ is more than k, show that their intersection is non-empty.

For the common vector v, $Av = A_V v$ (where A, A_V are the adjacency matrices of G, G_V respectively), and the length of Av is more than λ_l times the length of v. So, we get

$$\|G_V\| := \|A_V\| \geq \lambda_l.$$

□

The adjacency matrix of Boolean hypercube (say H_n) has dimensions $2^n \times 2^n$. Arrange the eigenvalues of H in increasing order, $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{2^n}$. From Lemma 1, the maximum eigenvalue of G_V is more than $\lambda_{2^{n-1}+1}$.

What is $\lambda_{2^{n-1}+1}$? You will show in the assignment that the eigenvalues of Boolean hypercube has very simple structure. It has eigenvalue $-n + 2k$ with multiplicity $\binom{n}{k}$.

Exercise 28. What bound will this give on $\|G_V\|$ when $|V| > 2^{n-1}$?

Unfortunately the interlacing theorem applied on H_n doesn't seem to be of much help. It turns out, a small modification of the adjacency matrix of H_n will do the trick.

Proof of Theorem 1. The main idea of the proof is to construct a *signing* of the adjacency matrix of the Boolean hypercube. Applying interlacing theorem on that matrix will give the result. A signing of a $\{0,1\}$ matrix is assigning negative sign to some non-zero entries of the matrix. Let A_s be a signing of a $\{0,1\}$ matrix, then you will show in the assignment

$$\|A\| \geq \|A_s\|.$$

We will construct a signing s of Boolean hypercube such that half of its eigenvalues (2^{n-1} of them) will be $\sqrt{n}$ and the other half will be $-\sqrt{n}$. If A is the adjacency matrix of G_V,

$$\|A_V\| \geq \|(A_s)_V\|.$$

Here A is the adjacency matrix of H_n and A_V denote the induced matrix on the subset V.

By Lemma 1, $\|(A_s)_V\|$ should be greater than the $2^{n-1}+1$ highest eigenvalue of A_s, which is $\sqrt{n}$.

The only task is to construct the signing with required properties. It is defined inductively by,

$$(A_1)_s = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, (A_n)_s = \begin{pmatrix} A_{n-1} & I \\ I & -A_{n-1} \end{pmatrix}$$

You can easily show the following properties of this signing by induction.

- $(A_n)_s$ is a signing of H_n (it follows the structure of Boolean hypercube).
- Trace of $(A_n)_s$ is 0.
- $(A_n)_s^2 = nI$.

From the third property, each eigenvalue is either $\sqrt{n}$ or $-\sqrt{n}$. From the trace property, the multiplicity of each eigenvalue is 2^{n-1}. Thus, we have the signing with required property, showing that if V is a subset of vertices of H_n such that $|V| > 2^{n-1}$, then $\|G_V\| \geq \sqrt{n}$.

By the discussion before the proof, this implies that $\lambda(f) \geq \sqrt{n}$ for any f with degree n. □

Huang's result, using Equation 4, implies that $\mathrm{bs}(f) = O(\mathrm{s}(f)^4)$. We only know a function for which $\mathrm{bs}(f) = \Omega(\mathrm{s}(f)^2)$ [3]. It is an open problem to bridge this gap.

There have been interesting developments after this discovery, as mentioned before, it was proven that $\lambda(f) = O(\widetilde{\deg}(f))$ in [1]. They were able to use this to show that for any Boolean function f, $\deg(f) = O(\widetilde{\deg}(f)^2)$. This is known to be optimal by OR function.

4 Assignment

Exercise 29. Show that $\mathrm{s}(f), \mathrm{bs}f, C(f)$ are $\Theta(n)$ for any symmetric function on n variables.

Exercise 30. Show a symmetric function where $\mathrm{bs}(f) > \mathrm{s}(f)$.

Exercise 31. Can you think of an f and z to separate $\mathrm{s}_z(f)$ and $\mathrm{bs}_z(f)$.

Exercise 32. Suppose A is the adjacency matrix of a bipartite graph. Show that if u is an eigenvalue of A, then so is $-u$.

Exercise 33. Why is the λ of restricted function smaller than the λ of the original function?

Exercise 34. Let H_n be the Boolean hypercube on n elements. Show that H_n has eigenvalue $-n+2k$ with multiplicity $\binom{n}{k}$ for $0 \leq k \leq n$.

Hint: Use induction and structure of the adjacency matrix of Boolean hypercube.

Exercise 35. Just by looking at the eigenvalues of Boolean hypercube and Cauchy's interlacing theorem, you can come up with a statement like: if the degree n coefficient of f is *big enough* then $\lambda(f) \geq \sqrt{n}$. Make this statement precise and prove it.

Exercise 36. Show that if A_s is a signing of A, then

$$\|A\| \geq \|A_s\|.$$

Exercise 37. Tribes function is $\mathrm{AND}_n \circ \mathrm{OR}_n$. Show that D of tribes is $\Theta(n^2)$ and C of tribes is $\Theta(n)$.

References

1. Scott Aaronson, Shalev Ben-David, Robin Kothari, Shravas Rao and Avishay Tal, ‘’Degree vs. approximate degree and Quantum implications of Huang’s sensitivity theorem,” STOC, 2021.
2. Hao Huang, “Induced subgraphs of hypercubes and a proof of the Sensitivity Conjecture,” Annals of Mathematics, Volume 190, Pages 949-955, 2019.
3. David Rubinstein, “Sensitivity vs. block sensitivity of Boolean functions,” Combinatorica, Volume 15, Pages 297–299, 1995.
4. Harry Buhrman and Ronald de Wolf, “Complexity measures and decision tree complexity: a survey,” Theoretical Computer Science, Volume 288, Issue 1, Pages 21-43, 2002.
5. Noam Nisan and Mario Szegedy, “On the degree of boolean functions as real polynomials,” Computational Complexity, volume 4, pages 301–313, 1994.
6. Raghav Kulkarni and Avishay Tal, “On Fractional Block Sensitivity,” Chicago Journal of Theoretical Computer Science, Article 08, pages 1-16, 2016.