Sensitivity, Block Sensitivity and Certificate Complexity of Boolean Functions
(Master's Thesis)

Sourav Chakraborty
Thesis Advisor: László Babai

February, 2005

Abstract

We discuss several complexity measures for Boolean functions: sensitivity, block sensitivity, ℓ-block sensitivity, certificate complexity, average sensitivity and average block sensitivity. We survey various known bounds for these measures and relations between these measures for different classes of functions. For a general Boolean function our understanding of these measures is still very limited. Many of the relations that we have are far from being tight. For example, the best known upper bound on block sensitivity in terms of sensitivity is exponential [Simon 1983, Kenyon-Kutin 2004] while the largest known gap is quadratic [Rubinstein 1995].

For certain classes of functions (like functions with certain amount of symmetries) our understanding is much better. We discuss how the various measures perform on these classes of functions. The classes of functions we discuss are monotone functions, symmetric functions, graph properties, functions closed under some transitive group action and minterm-transitive functions. For monotone functions and symmetric functions we have quite a clear understanding of the measures. For graph properties we have almost tight bounds [Turan 1984].

Our main new contributions concern minterm-transitive functions defined by Chakraborty(2005). This class is a subclass of the class of functions invariant under some transitive group action. We establish a tight lower bound of $n^{1/3}$ for sensitivity of functions in this class. Using similar techniques we try to prove similar tight bounds for functions invariant under some transitive group action.

Most of these measures were initially defined to give nice bounds for the decision tree complexity for various models of computation. So we also review how these measures are related to decision tree complexity for deterministic, randomized and quantum model.

1 Introduction

In the last few decades computation of Boolean functions has been a major area of research in computer science. Various models of machines are used to compute Boolean functions. Some examples of machine models are deterministic model, non-deterministic model, randomized model, quantum model and parallel computing model. Complexity theory tries to understand the computational power of these models like finding tight bounds on the amount of time, space and query needed for computing a particular function. Unfortunately for many important problems no good bounds are known.

Decision tree complexity of a function is the minimum number of bits of the input that the optimal algorithm has to check (query) to be able to calculate the value of the function on that input. *Sensitivity, block sensitivity, certificate complexity, average sensitivity and block sensitivity* and *degree of representing polynomials* are some complexity measures on Boolean functions that are used to obtain good lower and upper bounds for decision tree complexity for various machine models. For example certificate complexity is exactly same as decision tree complexity for non-deterministic model, sensitivity gives tight bounds for CREW PRAM model while block sensitivity is used to compute lower bounds for the quantum model for cyclically invariant functions. So it is important to understand the properties of these measures.

We are interested in knowing the lower and upper bounds of these measure and how these measures relate to one another for certain classes of functions. Unfortunately in most classes of functions our knowledge is very limited. But for functions with certain amount of symmetry we have good bounds. The classes of functions that we survey are monotone functions, symmetric functions, graph properties, functions closed under some transitive group actions and minterm-transitive functions. For monotone functions, symmetric functions and graph properties we already have bounds which are tight (up to a factor of 4 for graph properties). But for functions closed under some transitive group actions the bounds we have seems to be far from being tight. Minterm-transitive is a subclass of functions closed under some transitive group actions. This class was introduce very recently in [8] and proved some tight bounds for this class. In this survey we will also try to extend the proofs of [8] to the bigger class of functions. We also try to extend the proof of [21] for graph properties to k-uniform hyper graph properties. We survey the known relations between the various measures and we also look at a number of examples that gives the best known gaps between the measures. For important theorems we have given sketches of proofs.

In this survey we focus mainly on the properties of sensitivity, block sensitivity, ℓ-block sensitivity, certificate complexity and average sensitivity and block sensitivity. Our main contribution in this survey is a detailed study on functions closed under transitive group actions. At the end we give an brief description of some of the other measures like degree of representing polynomials. We also study how some of these measures are related to the decision tree complexity for various machine models. But we will not go into details. A more detailed study on these topics are found in the survey of Harry Buhrman and Ronald de Wolf [6]. In their survey they gave more attention to how these measures are related to the decision tree complexity and quantum query complexity. Wegener [24] wrote a book called "Complexity of Boolean functions". Section 13.4 is on the complexity of PRAMs. There he has looked at sensitivity and certificate complexity.

The organization of this survey is as follows: In Section 2 we set up some notations that will be used later on. In Section 3 we look at some examples of Boolean functions. Later on in the survey we will study these examples for various measures. In Section 4 we define the measures and survey how they relate to each other for an arbitrary Boolean functions. In Section 5 we define several classes of functions and survey the known best bounds and relations of the measures for the classes of functions. Finally we have a quick survey on related topics. Through out the survey we point out various open problems.

2 Notations

We use the notation $[n] = \{1, 2, 3, ..., n\}$. In the rest of the survey (unless otherwise stated) f, g, h denotes boolean functions on n variables, *i.e.*, $f, g, h : \{0,1\}^n \to \{0,1\}$. We call the elements of $\{0,1\}^n$ "words" of length n. Also otherwise stated x, y, z denotes words of length n. For a word x we denote the ith bit as x_i. So x is actually $x_1 x_2 \cdots x_n$.

For any word x and $1 \leq i \leq n$ we denote by x^i the word obtained by switching the ith bit of x. For a word x and $A \subseteq [n]$ we use x^A to denote the word obtained from x by switching all the bits in A.

For a word $x = x_1 x_2 ... x_n$ we define $\mathrm{supp}(x)$ as $\{i \mid x_i = 1\}$. Weight of x, denoted $\mathrm{wt}(x)$, is $|\mathrm{supp}(x)|$, *i.e.*, number of 1s in x.

For a word $x = x_1 x_2 \cdots x_n$ and for $i \leq n$ we denote by $x|_{[i]}$ the word $x_1 x_2 \cdots x_i$, *i.e.*, the i length word formed by the first i bits of x.

The obvious ordering on the words is the lexicographic ordering. So unless otherwise stated when we speak of $x \leq y$ we mean by the lexicographic ordering.

Definition 2.1 *Given a function $f : \{0,1\}^n \to \{0,1\}$ we call the ith variable effective if there exist some word x for which $f(x) \neq f(x^i)$.*

Effectively the effective variables are the variables that matter to us. So a function can easily to modified to a function defined on smaller inputs such that all the variables in the input are effective.

We define something called *partial function.* It is used a lot in rest of the survey.

Definition 2.2 A *partial assignment* is a function $p : S \to \{0,1\}$ where $S \subseteq [n]$. We call S the support of this partial assignment. The weight of a partial assignment is the number of elements in S that is mapped to 1. We call x a (full) assignment if $x : [n] \to \{0,1\}$. (Note than any word $x \in \{0,1\}^n$ can be thought of as a full assignment.) We say $p \subseteq x$ if x is an extension of p, *i.e.*, the restriction of x to S denoted $x|_S = p$.

3 Examples

Before going any further with our survey we would look at some examples of interesting Boolean functions. These functions comes again and again in the finding lower bounds and upper bounds for various measures.

Example 3.1 *k-Threshold:- If x is a word of size n then $k - Threshold(x) = 1$ iff $\mathrm{wt}(x) \geq \frac{n}{2}$.*

Example 3.2 *Parity:- Given word* $x \in \{0,1\}^n$ *we define* $parity(x) = 1$ *iff* $|supp(x)|$ *is even.*

Example 3.3 *AND-of-OR:- Given a word* x *of length* k^2 *we can think* $x = x_{(1)}x_{(2)} \cdots x_{(k)}$ *where each* $x_{(i)}$ *is a word of length* k. *Now let* c_i *be the OR of all the bits in* $x_{(i)}$. *Then* f *is called AND-of-OR if it outputs* AND *of all the* c_i*'s.*

Example 3.4 *AND-OR tree Let* C *be a complete Boolean tree of depth* k *with* AND *gates at all the odd depth nodes and* OR *gates at the even depth nodes. Now* f *be the function corresponding to the circuit* C *with inputs as the leaves. The function* $f : \{0,1\}^{2^k} \to \{0,1\}$ *is called the AND-OR tree.*

Example 3.5 *Addressing Function:- Let* x *be a word of size* $n + 2^n$. *Thus* x *has two parts; first part is the first* n *bits of* x *and the second part is the remaining of* x. *The first part corresponds to a number* i *between 0 and* 2^n. *So* $f(x) = 1$ *iff the* i*th bit of the second part is 1. Since the first part acts as an address for the second part it is called the addressing function.*

Example 3.6 *k-Monotone Address function:- This is a modified version of the previous function. The input is a word of size* $n + 2^n$. *Just as in the last example we can divide the input* x *into two parts: first part* $x^{(1)}$ *of size* n *and the second part* $x^{(2)}$ *of size* 2^n. *Let* $0 \leq k \leq n$. *Now* $f(x) = 0$ *if* $\mathrm{wt}(x^{(1)}) < k$ *and* $f(x) = 1$ *if* $\mathrm{wt}(x^{(1)}) > k$. *If* $\mathrm{wt}(x^{(1)}) = k$ *then it acts as the usual address function.*

Note that the number of effective variables is $n + \binom{n}{k}$. *Because if* $\mathrm{wt}(x^{(1)}) > k$ *or* $< k$ *then the variables in* $x^{(2)}$ *is not used in computation of* f *and if* $\mathrm{wt}(x^{(1)}) = k$ *then only* $\binom{n}{k}$ *number of bits in* $x^{(2)}$ *can be addressed.*

Example 3.7 *Modified Rubinstein's function Let* $g : \{0,1\}^k \to \{0,1\}$ *be such that* $g(x) = 1$ *iff* x *contains two consecutive ones and the rest of the bits are 0. In function* $f' : \{0,1\}^{k^2} \to \{0,1\}$ *the variables are divided into groups* $B_1, \ldots, B_k$ *each containing* k *variables.* $f'(x) = g(B_1) \vee g(B_2) \vee \cdots \vee g(B_k)$. *Using* f' *we define the function* $f : \{0,1\}^{k^2} \to \{0,1\}$ *as* $f(x_1x_2\cdots x_n) = 1$ *iff* $f'(x_ix_{i+1}\cdots x_nx_1x_2\cdots x_i) = 1$ *for some* $1 \leq i \leq n$.

A slightly simplified version of this example was first given by Rubinstein [16]. This example has some nice properties that we will see later.

4 Various Measures on Boolean functions

4.1 Sensitivity

4.1.1 Definition

Cook, Dwork and Reischuk [9] originally introduced sensitivity as a simple combinatorial complexity measure for Boolean functions providing lower bounds on the time needed by a CREW PRAM.

Although the definition is very simple, only for a few simple classes of function the sensitivity is understood somewhat properly. It may be noted that initially this measure was called the *critical complexity*. In most of the earlier papers it is called by this name.

Intuitively, as the name suggests, the sensitivity of a function computes the number of bits of the input on which the function is sensitive. The formal definition is given below.

Definition 4.1 The *sensitivity* of f on the word x is defines as the number of bits on which the function is sensitive: $s(f,x) = |\{i : f(x) \neq f(x^i)\}|$.

We define the *sensitivity* of f as $s(f) = \max\{s(f,x) : x \in \{0,1\}^n\}$

We define *0-sensitivity* of f as $s^0(f) = \max\{s(f,x) : x \in \{0,1\}^n, f(x) = 0\}$

We define *1-sensitivity* of f as $s^1(f) = \max\{s(f,x) : x \in \{0,1\}^n, f(x) = 1\}$.

4.1.2 Properties of Sensitivity

By definition for any function $f : \{0,1\}^n \to \{0,1\}$ sensitivity is less than n. Also it is trivially observed that this upper bound is tight, *i. e.*, there are functions with sensitivity n.

For example the *Parity function*(Example 3.2) has sensitivity n. The *k-Threshold function* (Example 3.1) has sensitivity $\max\{k, n-k\}$. The *AND-of-OR function* (Example 3.3) has sensitivity k when the input size is k^2. The *addressing function* (Example 3.5) has sensitivity $(n+1)$ when the input size is $(n+2^n)$. The *Rubinstein's function* (Example 3.7) has sensitivity $2k$ when the input size is k^2.

Wegener [23] proved the following theorem.

Theorem 4.2 *The k-monotone addressing function (Example 3.6) with $k = \lfloor \frac{n}{2} \rfloor$ has sensitivity $\lceil \frac{n}{2} \rceil$ while the effective number of variables is $n + \binom{n}{\lfloor n/2 \rfloor}$.*

Proof: If x is an input of size $(n+2^n)$ with $k = \lfloor \frac{n}{2} \rfloor$ then in the case of *monotone addressing function* the number of effective variables is $n + \binom{n}{k} = n + \binom{n}{\lfloor n/2 \rfloor}$.

Let x be broken up into two parts. The first part x' be the address and $|x'|$ be n. Then if $\mathrm{wt}(x') < \lfloor \frac{n}{2} \rfloor$ (or $\mathrm{wt}(x') > \lfloor \frac{n}{2} \rfloor$) $s(f)$ is less than the number of 1 (or 0) in x', *i. e.*$\lceil \frac{n}{2} \rceil$.

If $\mathrm{wt}(x') = \lfloor \frac{n}{2} \rfloor$ and $f(x) = 0$ then to change the value of f we either have to increase the number of 1s in x' ($\lfloor \frac{n}{2} \rfloor$ possibilities) or we have to change the value at the position corresponding to the address (only one possibility). So $s(f) < \lceil \frac{n}{2} \rceil$. Similarly if $f(x) = 1$ we can see $s(f) < \lceil \frac{n}{2} \rceil$.

Hence in any case sensitivity is less than $s(f) < \lceil \frac{n}{2} \rceil$.

■

Lemma 4.3 *If* $m = n + \binom{n}{\lfloor n/2 \rfloor}$ *then* $\lceil \frac{n}{2} \rceil = \frac{1}{2}\log m + \frac{1}{4}\log\log m + O(1)$.

From Lemma 4.3 and Theorem 4.2 the following theorem follows.

Theorem 4.4 ([23]) *There is a Boolean function* $f : \{0,1\}^m \to \{0,1\}$ *with sensitivity less than* $\frac{1}{2}\log m + \frac{1}{4}\log\log m + O(1)$.

Wegener [23] proved that for the ($\lfloor n/2 \rfloor$)-monotone address the sensitivity is $\frac{1}{2}\log m + \frac{1}{4}\log\log m + O(1)$. Among any known functions this function in fact has the lowest sensitivity in terms of effective variable. Hans-Ulrich Simon [18] proved the following theorem:

Theorem 4.5 *For any boolean function* $f : \{0,1\}^n \to \{0,1\}$ *we have* $s(f) \geq (\frac{1}{2}\log n - \frac{1}{2}\log\log n + \frac{1}{2})$, *where* n *is the number of effective variables.*

Thus the *($\lfloor n/2 \rfloor$)-monotone addressing function* comes quite close to the possible lowest bound.

For various restricted classes of functions better bounds are known. We will have a look at the special classes in the next section.

4.2 Block sensitivity

4.2.1 Definitions

Nisan [12] introduced the concept of block sensitivity and demonstrated the remarkable fact that block sensitivity and CREW PRAM complexity are polynomially related.

Definition 4.6 The *block sensitivity* $bs(f, x)$ of a function f on an input x is the maximum number of disjoint subsets $B_1, B_2, ..., B_r$ of $[n]$ such that for all j, $f(x) \neq f(x^{B_j})$.

The *block sensitivity* of f, denoted $bs(f)$, is $\max_x bs(f, x)$.

Although the definition of block sensitivity is very similar to that of sensitivity it is surprising that no good relation is known between block sensitivity and sensitivity.

4.2.2 Properties of Block Sensitivity

Again by definition, for a function $f : \{0,1\}^n \to \{0,1\}$ the block sensitivity is less than n and clearly this upper bound is tight for any arbitrary function.

The definition of sensitivity is a special case of block sensitivity that is when all the blocks are of size 1. So block sensitivity is always more than sensitivity. But for an arbitrary Boolean function the best known upper bound on block sensitivity in terms of sensitivity is exponential. The lower bound on sensitivity of function f, given by H.-U. Simon [18], gives an upper bound of $O(s(f)4^{s(f)})$ on block sensitivity where $s(f)$ is the sensitivity of the function f. Kenyon and Kutin [11] gave the best known upper bound on block sensitivity in terms of sensitivity; their bound is $O\left(\frac{e}{\sqrt{2\pi}}e^{s(f)}\sqrt{s(f)}\right)$. Their proof uses some other measure called "*ell*-Block Sensitivity". In the next section we will define ℓ-block sensitivity.

But surprisingly the largest known gap between block sensitivity and sensitivity is quadratic, as shown by Rubinstein [16]. The *Modified Rubinstein's function* (example 3.5) demonstrates the quadratic gap. It has sensitivity of f is $2k$ while the block sensitivity is $\lfloor \frac{k^2}{2} \rfloor$.

There is a strong feeling that for an arbitrary function block sensitivity is polynomially related to sensitivity, in fact quadratically related.

Open Problem 4.7 *Is there a function $f : \{0,1\}^n \to \{0,1\}$ such that $bs(f) = \omega(s(f)^2)$.*

Open Problem 4.8 *Is there a constant c such that for any function $f : \{0,1\}^n \to \{0,1\}$ we have $bs(f) = O(s(f)^c)$.*

The following lemma follows from some simple observations.

Lemma 4.9 *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function and let $x \in \{0,1\}^n$ with $z = f(x)$. Let $B_1, B_2, \cdots, B_k$ be the minimal set of blocks corresponding to the block sensitivity for x. Then $\forall 1 \leq i \leq k$, $|B_i| \leq s^{1-z}(f)$.*

Proof: So $f(x) = z$ while $f(x^{B_i}) = (1-z)$. Now all the bits in B_i must be sensitive for x^{B_i}, otherwise there is $B' \subset B_i$ such that $f(x^{B'}) = (1-z)$ contradicting the minimality condition. ■

4.3 ℓ-Block Sensitivity

4.3.1 Definition

In the definition of block sensitivity (Definition 4.5) if we restrict the block size to be at most ℓ then we obtain the concept of ℓ-block sensitivity of the function f, denoted $bs_\ell(f)$. In [11] Kutin and Kenyon introduced this concept.

Definition 4.10 The *ℓ-block sensitivity* $bs_\ell(f,x)$ of a function f on an input x is the maximum number of disjoint subsets $B_1, B_2, ..., B_r$ of $[n]$ of size less than or equal to ℓ (*i.e.*, for all $1 \leq j \leq r$, $|B_j| \leq \ell$) such that for all j, $f(x) \neq f(x^{B_j})$.

The *block sensitivity* of f, denoted $bs_\ell(f)$, is $\max_x bs_\ell(f,x)$.

Similar to the definition of 0-sensitivity and 1-sensitivity we can define 0-ℓ-Block Sensitivity and 1-ℓ-Block Sensitivity denoted $bs^0_\ell(f)$ and $bs^1_\ell(f)$ respectively.

4.3.2 Properties of ℓ-Block Sensitivity

Kenyon and Kutin [11] proved the following theorems.

Theorem 4.11 *For $2 \leq \ell \leq s(f)$, and $z \in \{0,1\}$,*

$$bs^z_\ell(f) \leq \frac{4}{\ell} s^{1-z}(f) bs^z_{\ell-1}(f).$$

Also

$$bs^z_\ell(f) \leq c_\ell s^z(f) \left(s^{1-z}(f)\right)^{\ell-1},$$

where,

$$c_\ell = \frac{\left(1+\frac{1}{\ell+1}\right)^{\ell-1}}{(\ell-1)!} < \frac{e}{(\ell-1)!}.$$

Corollary 4.12 $s(f) \geq (\frac{bs_\ell(f)}{c_{ell}})^{1/\ell}$*, where c_ℓ is the constant of the previous Theorem.*

The above theorem helps us to prove the following relation between block sensitivity and sensitivity.

Theorem 4.13 *If $bs(f) \geq Kn$ then $s(f) = \Omega(n^{K/2})$.*

Theorem 4.14 $bs(f) \leq \frac{e}{\sqrt{2\Pi}} e^{s(f)} \sqrt{s(f)}$*.*

This is the currently the best known upper bound on block sensitivity in terms of sensitivity for general Boolean functions.

One important open problem here is asked by Kenyon and Kutin in their paper.

Open Problem 4.15 *Can the constant in the upper bound on ℓ-block sensitivity be improved? Actually a slight improvement in the constant can actually give us a sub-exponential bound on bock sensitivity in terms of sensitivity.*

4.4 Certificate Complexity

4.4.1 Definition

Certificate complexity was first introduced by Vishkin and Wigderson [22]. This measure was initially called sensitive complexity. In many of the earlier papers it is referred to by this name.

Definition 4.16 For $b = \{0, 1\}$, we define *b-certificate* as a partial assignment, $p : S \to \{0, 1\}$, which forces the value of the function to 1. Thus if $x|_S = p$ then $f(x) = 1$.

Definition 4.17 The *certificate complexity* of a function f on x, denoted $c_x(f)$, is the size of the smallest $f(x) - certificate$ that can be extended to x.

The *certificate complexity* of f is $\max_x c_x(f)$. For $b = \{0, 1\}$, we define the *b-certificate complexity* of f as $\max_{x:f(x)=b} c_x(f)$.

Just like 0-sensitivity and 1-sensitivity, we can define 0-certificate complexity $c^0(f)$ and 1-certificate complexity $c^1(f)$.

There is a way of looking at certificate complexity in terms of hyper cubes. We can think of a Boolean function as coloring the vertex on a n-dimension hyper cube using just two colors. Then certificate complexity is n minus the dimension of the largest monochromatic hypercube in the n-dimensional hypercube.

One interesting thing to note is that certificate complexity is exactly the decision tree complexity for non-deterministic machine model. That is because the non-deterministic machine can just guess the certificate that the input might have and it has to verify it and that will take at most $c(f)$ queries.

4.4.2 Properties of Certificate Complexity

Let x be a word for which the block sensitivity $bs_x(f) = bs(f)$. Note that any certificate for x must have at least one variable from each of the sensitive blocks of x. So $bs(f) \leq c(f)$ for all Boolean function f.

Hence the relation between sensitivity and block sensitivity and certificate complexity is

$$s(f) \leq bs(f) \leq c(f)$$

.

We have this slightly stronger result.

Lemma 4.18 *For $z \in \{0, 1\}$, $bs^z(f) \leq c^z(f)$.*

Proof: Let $x \in \{0,1\}^n$ with $B_1, B_2, \cdots, B_k$ be the minimal blocks corresponding to the block sensitivity of x. Let C_x be the certificate for x. Then C_x must intersect all the B_i. If not there is some block B_j which does not intersect with C_x. Then consider x^{B_j}. It contains C_x, so $f(x^{B_j}) = f(x)$. While by definition of the blocks $f(x^{B_j}) = 1 - f(x)$. Hence a contradiction. ∎

Nisan [12] gave the upper bound on certificate complexity in terms of sensitivity and block sensitivity.

Theorem 4.19 *If $f : \{0,1\}^n \to \{0,1\}$ is a boolean function on n variables then $c(f) \leq bs(f)s(f)$.*

This theorem follows from the following lemma.

Lemma 4.20 *For $x \in \{0,1\}$, $c(f,x) \leq bs(f,x)s^{1-f(x)}(f)$.*

Proof: Let $x \in \{0,1\}$ with $B_1, B_2, \cdots, B_k$ be the minimal blocks corresponding to the block sensitivity of x. Then $B = \cup_i B_i$ is a certificate of x. If not there is a y such that y is an extension of B but still $f(x) \neq f(y)$. Let $y = x^A$. Since y extends B, so A is disjoint from B. So by definition of block sensitivity $A \backslash B$ is a sensitive block and hence $A \subset B$, a contradiction.

So $c(f,x) \leq$B$\leq bs(f,x)s^{1-f(x)}(f)$ (from Lemma 4.8).

∎

This gives a quadratic bound on certificate complexity in terms of block sensitivity. But no example is known which has quadratic gap between certificate complexity and block sensitivity. It is thus an interesting open problem.

Open Problem 4.21 *Is there a boolean function for which there is a quadratic gap between certificate complexity and block sensitivity.*

Another interesting thing to investigate is how large can the ratio of certificate complexity to sensitivity be with respect to the input size. Simon's [18] lower bound on sensitivity tells us that $\frac{c(f)}{s(f)} = \Omega(\frac{n}{\log n})$. In [25] they constructed an example for which $\frac{c(f)}{s(f)} \geq n^{0.29248}$.

In Section 5.5 we construct a function for which sensitivity is $\Theta(n^{1/3})$ and certificate complexity is $\Theta(n^{2/3})$. So for this function $\frac{c(f)}{s(f)}$ is $\Theta(n^{1/3})$. But presently the largest gap known is for the *Rubinstein's function* or the *AND-of-OR function*. For that $\frac{c(f)}{s(f)}$ is $\sqrt{n}$. But still we are far from the upper bound we have.

Open Problem 4.22 *How large can $\frac{c(f)}{s(f)}$ be?*

4.5 Average Sensitivity and Block sensitivity

4.5.1 Definitions

One interesting measure to look at is the average sensitivity and average block sensitivity.

Definition 4.23 *We define average sensitivity* $\widehat{s(f)}$ *as*

$$\widehat{s(f)} = \frac{1}{2^n} \sum_x s(f, x) = \mathbb{E}[s(f, x)]$$

.

In case of block sensitivity there are two notions of average block sensitivity. They are called the average-block sensitivity and pseudo-average block sensitivity. But before we can define these we have to give a different way of looking at block sensitivity.

Let $\mathcal{D}$ be the set of all partitions of $[n]$. Let $d \in \mathcal{D}$. Let d be $\{S_1, S_2, \cdots, S_k\}$. We define the block sensitivity of function f on word x with respect to the partition d as

$$bs_d(f, x) = \sum_{i=1}^{k} [f(x) - f(x^{S_i})]^2$$

.

From this is can be observed that the block sensitivity $bs(f, x)$ of f on word x is $\max_{d \in \mathcal{D}} bs_d(f, x)$.

Using this notion of block sensitivity we have the following definitions.

Definition 4.24 *For* $d \in \mathcal{D}$ *we define d-average block sensitivity as*

$$\widehat{bs_d(f)} = \mathbb{E}[bs_d(f, x)] = \frac{1}{2^n} \sum_x bs_d(f, x)$$

.

Definition 4.25 *Average block sensitivity is defined as*

$$\widehat{bs(f)} = \mathbb{E}[bs(f, x)] = \frac{1}{2^n} \sum_x bs(f, x)$$

.

Definition 4.26 *Pseudo-average block sensitivity* $\widehat{\beta s(f)}$ *is defined as* $\max_{d \in \mathcal{D}} \widehat{bs_d(f, x)}$.

4.5.2 Properties of average sensitivity and average block sensitivity

Bernasconi [4] proved two very nice results involving these measure.

Theorem 4.27 *(1)* $\forall d \in \mathcal{D}$, $\widehat{bs_d(f)} \leq \widehat{s(f)}$.

(2) $\widehat{\beta s(f)} = \widehat{s(f)}$.

In case of sensitivity and block sensitivity the largest known gap is quadratic (for the Rubinstein's function). In fact it is widely conjectured to be the largest possible gap. Surprisingly in the average setting Bernasconi [4] showed that for the Rubinstein's function *average block sensitivity* is greater that $\sqrt{n}$, where as the *average sensitivity* has limit 0 as the input size goes to infinity. Thus in the average setting no upper bound on average block sensitivity in terms of average sensitivity can be expected.

Other interesting results relating to average sensitivity has also been proved. Ehud Friedgut [10] proved that functions with low average sensitivity depends on very few coordinates. More precisely functions with average sensitivity k can be approximated by functions depending on only c^k variables where c is a constant depending solely on the accuracy of approximation.

5 Simple classes of functions

We saw in the last section that for an arbitrary function our knowledge about how the various measures behave is quite limited. Also the relation between the measures is mostly far from understood. So we would like to restrict our attention to some much smaller class of functions which have a lot of nice properties. Unfortunately our knowledge in this classes is also not complete in many cases.

This section contains our main new contributions (Sections 5.3 - 5.5).

5.1 Monotone function

We call a function monotone if $f(x) \geq f(y)$ for $x \geq y$ where the ordering is lexicographic. Thus in other words if $f(x) = 1$ then by switching some of the 0s to 1s the values of the function does not change. Among the examples we saw in Section 3 *k-Threshold* and *k-Monotone addressing functions* are the only two monotone functions.

Nisan[12] proved the following proposition:

Proposition 5.1 *If* $f : \{0,1\}^n \rightarrow \{0,1\}$ *is a monotone function then* $s(f) = bs(f) = c(f)$.

Proof Let x be an input and C be the minimal certificate. Without loss of generality we can assume that $f(x) = 0$. Then by definition monotonicity C will not have any 1. Now let y be a word obtained by extending the partial function C with 1s. Now in y every bit of S is sensitive because if it is not then that bit can be dropped from the certificate. So $s(f, y) = |S| = c(f)$. Hence the proposition follow. ■

Wegener[23] proved the following theorem which we have already proved earlier in Section 4.1.

Theorem 5.2 *The k-monotone addressing function (Example 3.5) on inputs of size n has sensitivity $\frac{1}{2}\log n + \frac{1}{4}\log\log n + O(1)$.*

This is in fact the lowest known sensitivity and its sensitivity is very close to the lower bound on sensitivity of an arbitrary function given by Simon[18]. Also since monotone address function is a monotone function so from what Nisan pointed out it follows that the block sensitivity of monotone address function is $\frac{1}{2}\log n + \frac{1}{4}\log\log n + O(1)$. So we also have an example of a function that has very low block sensitivity.

5.2 Symmetric function

A function $f : \{0,1\}^n \to \{0,1\}$ is symmetric if the value of f only depends on the weight of the input, *i.e.*, the number of 1s in the input. So symmetric functions are very simple functions. Among the examples we saw in Section 3 only *Parity* and *k-Threshold* are symmetric functions.

Proposition 5.3 *[21] For non-constant symmetric functions on inputs of size n sensitivity is greater than $\lceil \frac{n+1}{2} \rceil$.*

Proof: Let $f : \{0,1\}^n \to \{0,1\}$ be a symmetric function. Since symmetric functions only depends on the weight of the inputs, so let $f(x) = 1$ if $\mathrm{wt}(x) = k$ and $f(x) = 0$ if $\mathrm{wt}(x) = (k-1)$. Then $s(f, x) \geq k$ if $\mathrm{wt}(x) = k$ and $s(f, x) \geq (n - (k-1))$ if $\mathrm{wt}(x) = (k-1)$.

So $s(f) \geq \max\{k, (n-k+1)\} = \lceil \frac{n+1}{2} \rceil$. ■

In fact the proposition is tight as sensitivity of the $\lfloor \frac{n}{2} \rfloor$-*Threshold function*(Example 3.1) is $\lceil \frac{n+1}{2} \rceil$.

Thus certificate complexity and block sensitivity are within a factor of two of sensitivity for symmetric functions.

5.3 Graph Property

Graph properties is a class of very natural functions. In short these are problems like whether a graph (encoded as a word) has a particular property that is invariant under isomorphism. So let $f : \{0,1\}^m \to \{0,1\}$ be a Boolean function that takes as input the encoding (*i.e.* the upper tri angle of the adjacency matrix) of a simple undirected graphs G. Now f evaluates to 1 iff the graph G has a given property. So the input size m is $\binom{|V|}{2} - |V|$ where $|V|$ is the number of vertices in the graph G. Also $f(G) = f(H)$ whenever G and H are isomorphic as graphs. Such a function f is called a *graph property*. When talking about graph property we will call the input $G = (V, E)$. So size of input is $\binom{|V|}{2}$ which is asymptotically equal to $|V|^2$.

Note that we take only the upper triangle of the adjacency matrix so that each undirected edge corresponds to only one 1 in the input. We can also define graph properties for simple directed graphs with self-loops. Then the input is the whole adjacency matrix of size $|V|^2$.

As observed by Turán [21] *graph property* is a class of functions that has a lot of symmetry but still not symmetric. The symmetry comes from the fact that the function is same for all isomorphic copies of graphs.

Let us look at some simple examples.

Example 5.4 *Let $f(G) = 1$ iff the graph G is connected.*

Example 5.5 *Let $f(G) = 1$ iff the graph G has an isolated vertex.*

Example 5.6 *$f(G) = 1$ iff the graph is 2-colorable.*

Example 5.7 *$f(G) = 1$ iff the graph is planar.*

Clearly the example are very common examples and hence of great interest. Many of the functions are well known NP-complete functions.

György Turán [21] gave almost a tight bound for the sensitivity of graph properties. He proved the following theorems.

Theorem 5.8 *There is a graph property which has sensitivity $|V| - 1$.*

Proof: The Example 5.5 has sensitivity $(|V| - 1)$. ■

Theorem 5.9 *If f is a graph property then $s(f) \geq \lfloor \frac{|V|}{2} \rfloor$.*

One interesting generalization is *hyper-graph property* for k-uniform hyper-graph. In this case the input is of size $\binom{n}{k}$.

Turán's theorem can be naturally generalized to k-uniform hyper-graph to prove the following theorem:

Theorem 5.10 *If f is a k-uniform graph property then sensitivity of f is $\Omega(|V|)$.*

But can we prove anything better? The example, "Is there an isolated vertex", has sensitivity $\binom{(|V|-1)}{(k-1)}$.

Open Problem 5.11 *For k-hyper-graph property is it true that sensitivity is $\Omega(\sqrt{n})$.*

Even finding examples of hyper-graph properties for which sensitivity is quite low is interesting.

5.4 Cyclically invariant function

One very interesting class of functions is the class of cyclically invariant functions. A slightly general class of functions is the class of functions invariant under transitive group actions. Most of the works in this section has been our contribution [8].

The definition of cyclically invariant function is simple. Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function and let $x = x_1 x_2 \cdots x_n$ be any input. Then f is cyclically invariant if for all x we have $f(x) = f(x')$, where $x' = x_2 x_3 \cdots x_n x_1$.

Now we give the definition of functions invariant closed under transitive group action.

Definition 5.12 Let G be a permutation group on $[n]$. G is called *transitive* if for all $1 \leq i, j \leq n$ there exists a $\pi \in G$ such that $\pi(i) = j$.

Definition 5.13 Let $S \subseteq [n]$ and let $\pi \in S_n$. Then we define S^π to be $\{\pi(i) \,|\, i \in S\}$.

Let G be a permutation group acting on $[n]$. Then the sets S^π, where $\pi \in G$, are called the *G-shifts* of S. If $p : S \to \{0,1\}$ is a partial assignment then we define $p^\pi : S^\pi \to \{0,1\}$ as $p^\pi(i) = p(\pi^{-1} i)$.

Definition 5.14 Let G be a subgroup of S_n, *i.e.*, a permutation group acting on $[n]$. A function $f : \{0,1\}^n \to \{0,1\}$ is said to be *invariant under the group* G if for all permutations $\pi \in G$ we have $f(x^\pi) = f(x)$ for all $x \in \{0,1\}^n$.

Definition 5.15 Let $x = x_1 x_2 ... x_n \in \{0,1\}^n$ be a word. Then for $0 < \ell < n$, we denote by $cs_\ell(x)$ the word $x_{\ell+1} x_{\ell+2} ... x_n x_1 x_2 ... x_\ell$, *i.e.*, the *cyclic shift* of the variables of x by ℓ positions.

Definition 5.16 A function $f : \{0,1\}^n \to \{0,1\}$ is called *cyclically invariant* if $f(x) = f(cs_1(x))$ for all $x \in \{0,1\}^n$.

Note that a cyclically invariant function is invariant under the group of cyclic shifts.

Cyclic invariant functions have many interesting properties and has been studied for many years. Various interesting observations have been posed as open problems earlier. Since cyclic invariant functions have a lot of symmetry in them it is believed that sensitivity of these functions are quite large.

Turán [21] after proving that for graph property sensitivity is $0(\sqrt{\textit{input size}})$ asked the following question.

Problem (Turán, 1984): *Does a lower bound of similar order hold still if we generalize graph properties to Boolean functions invariant under a transitive group of permutations?*

In the next section we give a cyclically invariant function with sensitivity $\Theta(n^{1/3})$. This example gives a negative answer to Turán's question.

Kenyon and Kutin [11] observed that for "nice" functions the product of 0-sensitivity and 1-sensitivity tends to be linear in the input length. Whether this observation extends to all "nice" functions was given as a (vaguely stated) open problem in that paper. In the next section we also construct a cyclically invariant Boolean function for which the product of 0-sensitivity and 1-sensitivity is $\Theta(\sqrt{n})$. Thus our function also gives a counterexample to Kenyon and Kutin's suggestion.

Certain other interesting things can be proved about functions invariant under transitive group actions.

Theorem 5.17 *For a non-constant boolean function $f : \{0,1\}^n \to \{0,1\}$ invariant under some transitive group action, certificate complexity $c(f) \geq \sqrt{n}$. And this inequality is tight. In the next section we will look at an example when the equality holds.*

Proof: We proove by contradiction. Let $c(f) \leq \sqrt{n}$. This means for all words $x \in \{0,1\}^n$ we have $c_x(f) \leq \sqrt{n}$. Since f is non-constant so there is a 0-certificate and a 1-certificate of size less than $\sqrt{n}$. Since f is cyclic and since product of the 0-certificate and 1-certificate is less than n so we can have a word with both the certificates. Then it will be a contradiction as the word will have both 0-certificate and 1-certificate. ■

Corollary 5.18 *For a function $f : \{0,1\}^n \to \{0,1\}$ invariant under some transitive group action the block sensitivity $bs(f) \geq n^{1/4}$.*

Proof: Since $c(f) \leq bs(f)^2$. ■

The inequality in the theorem is tight. In the next section we construct an function for which the equality holds. But the inequality in the corollary may not be tight. In the next section we come across an example for which the block sensitivity is $\Theta(\sqrt{n})$.

Open Problem 5.19 *Does a function invariant under transitive group action have block sensitivity $\Omega(\sqrt{n})$.*

It is widely believed that sensitivity of any cyclically invariant function is $\Omega(n^{1/3})$. The following lemmas may help us prove it.

Lemma 5.20 *If S is a subset of $[n]$, $|S| = k$ then there exist at least $\frac{n}{k^2}$ disjoint G-shifts of S.*

Proof: Let T be a maximal union of G-shifts of S. Since T is maximal T intersects with all G-shifts of S. So we must have $|T| \geq \frac{n}{k}$. So T must be a union of at least $\frac{n}{k^2}$ disjoint G-shifts of S. And this proves the lemma. ■

Lemma 5.21 *Let $f|\{0,1\}^n \to \{0,1\}$ be a function invariant under a transitive group action. Let c be a $f(x)$-certificate for some $x \in \{0,1\}^n$. Then $s^{\bar{f(x)}}(f) = \Omega(\frac{n}{k^2})$.*

Proof: Without loss of generality assume that $f(x) = 1$. By Lemma 5.20 we can have $\Omega(\frac{n}{k^2})$ disjoint G-shifts of c. The union of these disjoint G-shifts of c defines a partial assignment. Let $S = \{s_1, s_2, ..., s_r\}$ be the support of the partial assignment. And let Y_{s_i} be the value of the partial assignment in the s_i-th entry.

Since $k \neq 0$ the function f is not a constant function. Thus there exists a word z such that $f(z) = 0$. The i-th bit of z is denoted by z_i. We define,

$$T = \{j \mid z_j \neq Y_{s_m}, s_m = j\}$$

Now let $P \subseteq T$ be a maximal subset of T such that $f(z^P) = 0$. Since P is maximal, if we switch any other bit in $T \backslash P$ the value of the function f will change to 1.

So $s(f, z^P) \geq |(T \backslash P)|$. Now since $f(z^P) = 0$ we note that z^P does not contain any G-shift of c. But from Lemma 5.20 we know that z^T contains $\Omega(\frac{n}{k^2})$ disjoint G-shifts of y. So $|(T \backslash P)|$ is $\Omega(\frac{n}{k^2})$ and thus $s^0(f) \geq s(f, z^P) = \Omega(\frac{n}{k^2})$. ■

Now if the following conjecture is true then we can prove sensitivity of any cyclically invariant function is $\Omega(n^{1/3})$.

Conjecture 5.22 *For any function f invariant under some transitive group action there is a word x such that $s(f) \geq c(f, x)/d$ for some constant d.*

5.5 Minterm-cyclic function

In this section we first define two new class of functions called minterm-transitive and minterm-cyclic. The functions in the classes are functions invariant under some transitive group. This class was defined in [8].

Then we look at two examples that solves the open problems of Turán and Kenyon-Kutin.

Finally we prove that for the class minterm-transitive sensitivity is $\Omega(n^{1/3})$. This class of function is actually very simple but quite a big class of functions.

Definition 5.23 If $\mathcal{F}$ is a set of partial assignments then we define $m_{\mathcal{F}} : \{0,1\}^n \to \{0,1\}$ as $m_{\mathcal{F}}(x) = 1 \iff (\exists p \in \mathcal{F})$ such that $(p \subseteq x)$.

Note that each member of $\mathcal{F}$ is a 1-certificate for $m_{\mathcal{F}}$ and $m_{\mathcal{F}}$ is the unique smallest such function. (Here the ordering is pointwise, *i. e.*, $f \leq g$ if for all x we have $f(x) \leq g(x)$).

Definition 5.24 A *minterm* is a minimal 1-certificate, that is, no sub-assignment is a 1-certificate.

Proposition 5.25 *Let G be a permutation group. Let $p : S \to \{0,1\}$ be a partial assignment and let $\mathcal{F} = \{p^\pi \mid \pi \in G\}$. Then p is a minterm for the function $m_{\mathcal{F}}$.*

The function $m_{\mathcal{F}}$ will be denoted p^G. Note that the function p^G is invariant under the group G. When G is the group of cyclic shifts we denote the function p^{cyc}. The function p^{cyc} is cyclically invariant.

Proof of Proposition 5.25: If p has k zeros then for any word x with fewer than k zeros $m_{\mathcal{F}}(x) = 0$, since all the element of $\mathcal{F}$ has same number of 1s and 0s. But if q is a 1-certificate with fewer than k zeros we can have a word x by extending q to a full assignment by filling the rest with 1s, satisfying $f(x) = 1$ (since $q \subseteq x$). But x contains fewer than k zeros, a contradiction. So no minterm of $m_{\mathcal{F}}$ has fewer than k zeros.

Similarly no minterm of $\mathcal{F}$ has weight less than p. So no proper sub-assignment of p can be a 1-certificate. Hence p is a minterm of $m_{\mathcal{F}}$. ∎

Definition 5.26 Let $C(n,k)$ be the set of Boolean functions f on n variables such that there exists a partial assignment $p : S \to \{0,1\}$ with support $k (\neq 0)$ for which $f = p^{cyc}$. Let $C(n) = \cup_{k=1}^{n} C(n,k)$. We will call the functions in $C(n)$ **minterm-cyclic**. These are the simplest cyclically invariant functions.

Definition 5.27 Let G be a permutation group on $[n]$. We define $D_G(n,k)$ (for $k \neq 0$) to be the set of Boolean functions f on n variables such that there exists a partial assignment $p : S \to \{0,1\}$ with support k for which $f = p^G$. We define $D_G(n)$ to be $\cup_{k=1}^{n} D_G(n,k)$. This is a class of simple G-invariant Boolean functions. We define $D(n)$ to be $\cup_G D_G(n)$ where G ranges over all transitive groups. We call these functions **minterm-transitive**. Note that the class of minterm-cyclic functions is a subset of the class of minterm-transitive functions.

5.5.1 The new functions

In this section we will construct a cyclically invariant Boolean function which has sensitivity $\Theta(n^{1/3})$ and a cyclically invariant function for which the product of 0-sensitivity and 1-sensitivity is $\Theta(\sqrt{n})$.

Theorem 5.28 *There is a cyclically invariant function,* $f : \{0,1\}^n \to \{0,1\}$*, such that,* $s(f) = \Theta(n^{1/3})$*.*

Theorem 5.29 *There is a cyclically invariant function,* $f : \{0,1\}^n \to \{0,1\}$*, such that,* $s^0(f)s^1(f) = \Theta(\sqrt{n})$*.*

For proving the above theorems we will first define an auxiliary function g on k^2 variables ($k^2 \leq n$). Then we use g to define our new minterm-cyclic function f on n variables. If we set $k = \lfloor n^{2/3} \rfloor$, Theorem 5.28 will follow. Theorem 5.29 follows by setting $k = \lfloor \sqrt{n} \rfloor$.

The auxiliary function

We first define $g : \{0,1\}^{k^2} \to \{0,1\}$ where $k^2 \leq n$. We divide the input into k blocks of size k each. We define g by a regular expression.

$$g(z) = 1 \iff z \in \underbrace{110^{k-2}}_{k}\underbrace{11111\{0,1\}^{k-5}}_{k}\underbrace{11111\{0,1\}^{k-5}}_{k}\cdots\underbrace{11111\{0,1\}^{k-8}111}_{k} \qquad \cdots (1)$$

In other words, let $z \in \{0,1\}^{k^2}$ and let $z = z_1 z_2 \ldots z_k$, where each $z_i \in \{0,1\}^k$ for all $1 \leq i \leq k$, *i.e.*, z is broken up into k blocks of size k each. Then $g(z) = 1$ iff $z_1 = (11\underbrace{00\ldots0}_{k-2})$ and for all $2 \leq j \leq k$ the first five bits of z_j are 1 and also the last 3 bits of z_k are 1. Note that g does not depend on the rest of the bits.

The new function

Now we define the function f using the auxiliary function g. Let $x|_{[m]}$ denote the word formed by the first m bits of x. Let us set

$$f(x) = 1 \iff \exists \ell \text{ such that } g\left(cs_\ell(x)|_{[k^2]}\right) = 1.$$

In other words, viewing x as laid out on a cycle, $f(x) = 1$ iff x contains a contiguous substring y of length k^2 on which $g(y) = 1$.

In other words, let $z \in \{0,1\}^{k^2}$ and let $z = z_1 z_2 \ldots z_k$, where each $z_i \in \{0,1\}^k$ for all $1 \leq i \leq k$, *i.e.*, z is broken up into k blocks of size k each. Then $g(z) = 1$ iff $z_1 = (11\underbrace{00\ldots0}_{k-2})$ and for all $2 \leq j \leq k$

the first five bits of z_j are 1 and also the last 3 bits of z_k are 1. Note that g does not depend on the rest of the bits.

Properties of the new function

It follows directly from the definition that f is a cyclically invariant Boolean function.

It is important to note that the function g is so defined that the value of g on input z depends only on $(6k-2)$ bits of z.

Also note that the pattern defining g is so chosen that if $g(z) = 1$ then there is exactly one set of consecutive $(k-2)$ zeros in z and no other set of consecutive $(k-4)$ zeros.

Claim 5.30 *The function f has (a) 0-sensitivity $\Theta(\frac{n}{k^2})$ and (b) 1-sensitivity $\Theta(k)$.*

Proof of Claim: (a) Let x be a word such that the first k^2 bits are of the form (1) and the rest of the bits are 0. Now clearly $f(x) = 1$. Also it is easy to see that on this input x 1-sensitivity of f is $(6k-2)$ and therefore $s^1(f) = \Omega(k)$.

Now let $x \in \{0,1\}^n$ be such that $f(x) = 1$ and there exists $1 \leq i \leq n$ such that $f(x^i) = 0$. But $f(x) = 1$ implies that some cyclic shift of x contains a contiguous substring z of length k^2 of the form (1) (i. e., $g(z) = 1$). But since g depends only on the values of $(6k-2)$ positions so one of those bits has to be switched so that f evaluates to 0. Thus $s^1(f) = O(k)$.

Combined with the lower bound $s^1(f) = \Omega(k)$ we conclude $s^1(f) = \Theta(k)$.

(b) Let $\lfloor \frac{n}{k^2} \rfloor = m$ and $r = (n - k^2 m)$. Let $x = (1\underline{0}0^{k-2}(111110^{k-5})^{k-2}111110^{k-8}111)^m 0^r$. Then $f(x) = 0$ since no partial assignment of the form (1) exists in x. But if we switch any of the underlined zero the function evaluates to 1. Note that the function is not sensitive on any other bit. So on this input x the 0-sensitivity of f is $m = \lfloor \frac{n}{k^2} \rfloor$ and therefore $s^0(f) = \Omega(\frac{n}{k^2})$.

Now let $x \in \{0,1\}^n$ and assume $f(x) = 0$ while $f(x^i) = 1$ for some $1 \leq i \leq n$. By definition, the 0-sensitivity of f is the number of such values of i. For each such i there exists a partial assignment $z_i \subseteq x^i$ of the form (1). So z_i^i is a contiguous substring of x^i (or some cyclic shift of x^i) of length k^2. Now consider the $z_i^i \subseteq x$ (recall z_i^i denotes the partial assignment obtained by switching the ith bit of z_i). Due to the structure of the pattern (1) z_i has exactly one set of consecutive $(k-2)$ zeros. So z_i^i has exactly one set of consecutive $(k-2)$ bits with at most one of the bits being 1 while the remaining bits are zero. So the supports of any two z_i^i either have at least (k^2-1) positions in common or they have at most two positions in common (since the pattern (1) begins and ends with 11). Hence the number of distinct z_i^i is at most $\Theta(\frac{n}{k^2})$. Hence we have $s^0(f) = O(\frac{n}{k^2})$.

Combined with the lower bound $s^0(f) = \Omega(\frac{n}{k^2})$ we conclude that $s^0(f) = \Theta(\frac{n}{k^2})$. ■

Proof of Theorem 5.28: From Claim 5.30 it follows $s(f) = \max\left\{\Theta(k), \Theta(\frac{n}{k^2})\right\}$ (since $s(f) =$

$\max s^0(f), s^1(f))$. So if we set $k = \lfloor n^{2/3} \rfloor$ we obtain $s(f) = \Theta(n^{1/3})$. ∎

Proof of Theorem 5.29: From Claim 5.30 we obtain $s^0(f)s^1(f) = \Theta(\frac{n}{k})$. So if we set $k = \lfloor \sqrt{n} \rfloor$ we have $s^0(f)s^1(f) = \Theta(\sqrt{n})$. ∎

Theorem 5.28 answers Turán's problem [21] (see the Introduction) in the negative. In [11], Kenyon and Kutin asked whether $s^0(f)s^1(f) = \Omega(n)$ holds for all "nice" functions f. Although they do not define "nice," arguably our function in Theorem 5.29 is nice enough to answer the Kenyon-Kutin question is the negative.

Claim 5.31 *The function f has block sensitivity and certificate complexity* $\Theta(\max\{k, \frac{n}{k}\})$.

Hence for this function the gap between the sensitivity and block sensitivity is still quadratic.

In the next section we prove that for a minterm-transitive function, sensitivity is $\Omega(n^{1/3})$ and the product of 0-sensitivity and 1-sensitivity is $\Omega(\sqrt{n})$. Hence our examples are tight.

5.5.2 Minterm-transitive functions have sensitivity $\Omega(n^{1/3})$

Theorem 5.32 *If f is a minterm-transitive function on n variables then $s(f) = \Omega(n^{1/3})$ and $s^0(f)s^1(f) = \Omega(\sqrt{n})$.*

To prove this theorem we will use the following three lemmas. Since f is a minterm-transitive function, *i. e.*, $f \in D(n)$, we can say $f \in D_G(n, k)$ for some transitive group G and some $k \neq 0$.

Lemma 5.33 *If $f \in D_G(n, k)$ then $s^1(f) \geq \frac{k}{2}$.*

Proof: Let y be the minterm defining f. Without loss of generality $\mathrm{wt}(y) \geq \frac{k}{2}$. Let us extend y to a full assignment x by assigning zeros everywhere outside the support of y. Then switching any 1 to 0 changes the value of the function from 1 to 0. So we obtain $s(f, x) \geq \frac{k}{2}$. Hence $s^1(f) \geq \frac{k}{2}$. ∎

Lemma 5.34 *If $f \in D_G(n, k)$ then $s^0(f) = \Omega(\frac{n}{k^2})$.*

Proof: Follows from the Lemma 5.21. ∎

Proof of Theorem 5.32: From the Lemma 5.33 and Lemma 5.34 we obtain,

$$s(f) = \max\{s^0(f), s^1(f)\} = \max\left\{\Omega\left(\frac{n}{k^2}\right), \frac{k}{2}\right\}.$$

This implies $s(f) = \Omega(n^{1/3})$.

Now since $s^0(f)$ and $s^1(f)$ cannot be smaller than 1, it follows from the Lemma 5.33 and 5.34 that

$$s^0(f)s^1(f) = \max\left\{\Omega\left(\frac{n}{k}\right), \frac{k}{2}\right\}.$$

So $s^0(f)s^1(f) = \Omega(\sqrt{n})$. ■

The new function we looked at in Theorem 5.28 is minterm-transitive and has sensitivity $\Theta(n^{\frac{1}{3}})$. Thus this lower bound on sensitivity is tight for minterm-transitive functions. Similarly for the function in Theorem 5.29 the product of 0-sensitivity and 1-sensitivity is tight.

We can even give tight upper bound on block sensitivity for minterm-transitive functions in terms of sensitivity.

Theorem 5.35 *For a minterm-transitive function $f : \{0,1\}^n \to \{0,1\}$ we have $s(f) \geq \frac{bs(f)}{k}$, where k is the size of the minterm.*

Proof: Let x is the word for which the block sensitivity is $bs(f)$. Now let the minimal blocks be $B_1, B_2, \cdots, B_{bs(f)}$. By definition $f(x^{B_i}) \neq f(x)$. That means x^{B_i} has atleast a minterm. Choose any minterm. Now that minterm can intersect at most k of the blocks. Drop those blocks from the set of blocks. Do it for all the blocks. Finally atleast $\frac{bs(f)}{k}$ of the blocks are left. Let the union of these blocks be B. The blocks have the property that if we switch any block a minterm will be formed that does not intersects any other block.

Now let $A \subset B$ be the maximal set such that $f(x^A) = f(x)$. So x^A has sensitivity more than $B \backslash A$. And $B \backslash A$ must have have atleast one bit from all the blocks because if any block is switched fully then a minterm is formed because the minterm does not intersect any other block. So $|B \backslash A| \geq \frac{(bs(f)}{k}$. Hence $s(f, x^A) \geq \frac{bs(f)}{k}$. ■

Corollary 5.36 *For miniterm-transitive function, $bs(f) \leq s(f)^2$.*

Hence for minterm-transitive functions, sensitivity and block sensitivity does not have a gap of more than quadratic. And this is tight.

6 Other Related Topics

In this section we will do a quick survey on some of the related materials. We will look at two commonly used measures: degree of representing polynomials and degree of approximating polynomials. Here we define these measures and just state some of the main results related to them.

We also give an informal definition of three models of computation: deterministic, randomized and quantum. We have a quick look at how the various measures are related to the decision tree complexity of these three models of computation. We just state the main results. Much more elaborate survey for these things can be found in [6].

Definition 6.1 *Given a function $f : \{0,1\}^n \to \{0,1\}$, we call a polynomial $P : \mathbb{R}^n \to \mathbb{R}$ as a representing polynomial of f if for all $x \in \{0,1\}^n$, $f(x) = P(x)$. The degree of the representing polynomial for a function f, denoted $deg(f)$, is the lowest degree that a representing polynomial for f can have.*

Note that since for $x \in \{0,1\}$, $x^2 = x$ so we can actually only consider multilinear polynomials.

Definition 6.2 *A polynomial P is said to approximate f if for all $x \in \{0,1\}^n$, $|f(x) - P(x)| \leq \frac{1}{3}$. Similar to the definition of degree of the representing polynomial we define the concept of degree of approximating polynomials, denoted $\widetilde{deg}(f)$.*

Following are the some of the main results about $deg(f)$ and $\widetilde{deg}(f)$.

Theorem 6.3 *[13]*

1. *$bs(f) \leq 2deg(f)^2$.*
2. *$deg(f) \geq (\log n - O(\log\log n))$ for all Boolean function f.*
3. *$bs(f) \leq 6\widetilde{deg}(f)^2$ for all functions f.*

Theorem 6.4 *[1] for almost all functions f, we have $\widetilde{deg}(f) \geq \frac{n}{2} - O(\sqrt{n}\log n)$.*

Now we give the informal definition of three models of computation and survey the main relations between the decision tree complexity of the various models and the various measures.

6.1 Deterministic Machines

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. Let $\mathcal{M}$ be a deterministic machine that computes f. At every step it queries for a bit in the input. And according to the bit it branches and the computation tree we obtain is the *decision tree.*

Definition 6.5 *The decision tree complexity of a function for a deterministic machine is the depth of the decision tree of the optimal deterministic machine computing the functions f. It is denoted $D(f)$. Basically it is the number of queries the optimal machine will make.*

Similarly we can define non-deterministic decision tree and non-deterministic decision tree complexity. In this the machine can guess. Note that the decision tree complexity for a non-deterministic machine is exactly the certificate complexity of the function. Now the decision tree complexity of a deterministic machine is clearly more than the decision tree complexity of the non-deterministic machine. So from that is follows:

Theorem 6.6 *1. $c(f) \leq D(f)$.*

2. $deg(f) \leq D(f)$.

Beals *et al* [2] proved the following upper bound for the $D(f)$.

Theorem 6.7 *$D(f) \leq c^z(f)bs(f)$ for any $z \in \{0,1\}$.*

Hence we obtain a polynomial upper bound on decision tree complexity in terms of both certificate complexity and block sensitivity.

Corollary 6.8 *$D(f) \leq bs(f)^3$.*

Open Problem 6.9 *Is $D(f) \leq bs(f)^2$.*

Also since $D(f) \leq c^z(f)bs(f)$ for both $z = 0$ and $z = 1$ and since $bs(f) \leq \max\{c^0(f), c^1(f)\}$ so we obtain the following corollary.

Corollary 6.10 *1. $D(f) \leq c^0(f)c^1(f)$.*

2. In the case of monotone functions $c(f) = s(f)$. So we have: $D(f) \leq s(f)^2$.

Theorem 6.11 *1. [14] $D(f) \leq deg(f)^2 bs(f) \leq 2deg(f)^4$.*

2. [2, 13] $D(f) = O(\widetilde{deg}(f)^6)$.

6.2 Randomized Machines

Just like in the deterministic model we can define a decision tree and decision tree complexity of the randomized machine that computes the function with bounded error (*i.e.* with error less than $\frac{1}{3}$). The decision tree complexity is denoted $R(f)$.

Clearly since a randomized machine can also act as a deterministic machine so $R(f) \leq D(f)$.

Theorem 6.12 *[6] $\widetilde{deg}(f) \leq R(f)$.*

Theorem 6.13 *[12]* $bs(f) \leq 3R(f)$.

From the fact that $D(f) \leq bs(f)^3$ we obtain the following corollary.

Corollary 6.14 *[12]* $D(f) \leq 27R(f)^3$.

The obvious question to ask is what is the relation between $D(f)$ and $R(f)$. The largest gap known is 1.3. The *AND-OR-tree* (Example 3.4) has $D(f) = \Theta(R(f)^{1.3})$[17, 19].

Open Problem 6.15 *What is the biggest gap between $D(f)$ and $R(f)$?*

6.3 Quantum Machines

Similar to the definition of decision tree complexity we also define the quantum decision tree of quantum model. We will not give the formal definition.

Similar to the definition of the decision tree complexity *quantum query complexity* is the number of quantum queries a optimal quantum machine will make to compute f.

We denote $Q(f)$ to be the number of queries an optimal quantum machine will need to make to compute the function correctly with probability 1. We also denote $\widehat{Q}(f)$ the number of queries an optimal quantum machine will need to make to compute the function correctly with probability more than $\frac{1}{2}$.

Now since a quantum machine can also act like a deterministic machine we have the following theorem.

Theorem 6.16 $\widehat{Q}(f) \leq Q(f) \leq D(f)$

Now the obvious question is that how is the quantum query complexity related to the certificate complexity, block sensitivity and sensitivity.

Theorem 6.17 *[2]*

1. $deg(f) \leq 2Q(f)$.
2. $\widetilde{deg}(f) \leq 2\widehat{Q}(f)$.

Then combining the above theorem and Theorem 6.3 we obtain

Theorem 6.18 $bs(f) \leq 8Q(f)^2$ *and* $bs(f) \leq 72\widehat{Q}(f)^2$.

Theorem 6.19 *[2, 14]* $D(f) = O(Q(f)^4)$ *and* $D(f) = O(\widehat{Q}(f)^6$.

But the inequalities are not known to be tight.

Open Problem 6.20 *What is the biggest gap between* $D(f)$ *and* $Q(f)$*.*

For monotone functions we can state the stronger result.

Theorem 6.21 $D(f) = O(Q(f)^2)$ *and* $D(f) = O(\widehat{Q}(f)^4)$*.*

In [20] the following theorem is proved.

Theorem 6.22 *For a function* f *invariant under some transitive group action* $Q(f) \geq n^{1/4}$*.*

They used block sensitivity to proove the theorem.

It is impossible to state all possible results in this field. In this survey we did not look at all into related subjects like circuit complexity [24, 3, 5], communication complexity[15, 7], running time for parallel computing [9, 18, 12, 24] and many others. These subjects have very close relations to sensitivity, block sensitivity, certificate complexity and other measures.

Acknowledgements

I thank László Babai for introducing me to the subject and helping me with lots of useful ideas and suggestions. I also thank Lance Fortnow and Nanda Raghunathan for taking pains in reading the survey and giving useful suggestions. I also thank Samuel Kutin for giving me useful suggestions for my paper.

References

[1] A. Ambainis. *A note on quantum black-box complexity of almost all Boolean functions.* Information Processing Letters, 71(1):5-7, 1999.

[2] R. Beals, H. Buhrman, R. Cleve, M. Mosca and R. de Wolf. *Quantum lower bounds by polynomials* In Proceedings of 39th FOCS, 352-361, 1998.

[3] R. Beigel. *The polynomial methods in circuit complexity* In the Proceedings of the 8th IEEE Structure in Complexity Theory Conference, 82-95, 1993.

[4] A. Bernasconi. *Sensitivity vs. block sensitivity (an average-case study).* Information Processing Letters, vol 59(3):151-157 (1996)

[5] R.B.Boppana *The average sensitivity of bounded-depth circuits* Information Processing Letters, 63(5) 257-261 1997.

[6] H. Burhman and Ronald de Wolf. *Complexity Measures and Decision Tree Complexity: A survey* 2000.

[7] H. Buhrman and Ronald de Wolf. *Communication Complexity Lower bounds by polynomials*

[8] S. Chakraborty. *On the Sensitivity of Cyclically-Invariant Boolean Functions* ECCC Report No. TR05-020.

[9] Stephen Cook, Cynthia Dwork and Rüdiger Reischuk. *Upper and lower time bounds for parallel random access machines without simultaneous writes.* SIAM J.Comput. 15 (1986), no. 1, 87-97.

[10] E. Friedgut *Boolean functions with low average sensitivity depends on few coordinates*

[11] Claire Kenyon and Samuel Kutin. *Sensitivity, block sensitivity, and ℓ-block sensitivity of Boolean functions.* Information and Computation, vol 189 (2004), no. 1, 43-53.

[12] Noam Nisan. *CREW PRAMs and decision trees.* SIAM J. Comput. 20 (1991), no. 6, 999-1070.

[13] N. Nisan and M. Szegedy. *On the degree of Boolean functions as real polynomials* Computational Complexity, 4(4):301-313, 1994.

[14] N. Nisan and R. Smolensky. Unpublished. Cited in [6].

[15] N. Nisan and A. Wigderson. *On rank vs. communication complexity* Combinatorica, 15(4):557-565, 1995.

[16] David Rubinstein. *Sensitivity vs. block sensitivity of Boolean functions.* Combinatorica 15 (1995), no. 2, 297-299.

[17] M.Saks and A. Wigderson *Probabilistic Boolean decision trees and the complexity of evaluating game trees.* Proceedings of 27th FOCS, pages 29-38, 1986.

[18] Hans-Ulrich Simon. *A tight $\Omega(\log\log n)$-bound on the time for parallel RAM's to compute non-degenerated Boolean functions.* FCT vol 4 (1983), Lecture notes in Comp. Sci. 158.

[19] M. Snir *Lower bounds for probabilistic linear decision trees* Theoretical Computer Science 38:69-82, 1985.

[20] Xiaoming Sun, Andrew C. Yao and Shengyu Zhang. *Graph Properties and Circular Functions: How Low Can Quantum Query Complexity Go?* IEEE CCC 2004.

[21] György Turán. *The critical complexity of graph properties.* Inform. Process. Lett. 18 (1984), 151-153.

[22] U. Vishkin and A. Wigderson *Trade-offs between depth and width in parallel computation* SIAM J. Comput. 14 (1985) 303-314.

[23] I. Wegener *The critical complexity of all (monotone) boolean functions.* Information and Control, 67 (1985), Pages 212-222.

[24] I. Wegener *The Complexity of Boolean Functions*. Wiley-Teubner Series in Computer Science (Wiley, New York) 1987.

[25] I. Wegener and Laszlo Zádori. *A Note on the Relations Between Critical and Sensitive Complexity* Journal of Information Processing and Cybernatics 25(8/9):417-421, 1989.