Note on the average sensitivity of monotone Boolean functions*

Shengyu Zhang †

Abstract

We consider the average sensitivity $\bar{s}(f)$ of monotone functions f. First we give an exact representation of $\bar{s}(f)$ in terms of the number of 1-inputs with weight i ($i = 1, ..., n$). We then give a lower and an upper bound for $\bar{s}(f)$, both are tight for some monotone functions.

keyword computational complexity, sensitivity

1 Introduction

The sensitivity $s(f)$ is one of the most important and well studied complexity measures (see [6] by Buhrman and de Wolf for an excellent survey on many measures). The average sensitivity $\bar{s}(f)$, which is the sensitivity average on inputs, recently draws some attention. For example, Bernasconi shows large gaps between the average sensitivity and the average block sensitivity [4], Boppana considers the average sensitivity of bounded-depth circuits [5], and Shi shows that the average sensitivity is a lower bound of approximation polynomial degree, and thus can also serve as a lower bound of quantum query complexity [8].

In this note we consider the average sensitivity of monotone functions. In particular, we give an exact representation of $\bar{s}(f)$ in terms of the number of 1-inputs with weight i ($i = 1, ..., n$). We also derive a lower and a $\Theta(\sqrt{n})$ upper bound for $\bar{s}(f)$, both are tight for some monotone functions.

Quantum computing gets rapidly developed in the last decade. In the past several years, quantum query complexity is extensively studied, and many lower bounds are proven by the polynomial method proposed by Beal, Buhrman, Cleve, Mosca and de Wolf [3], and the quantum adversary method proposed by Ambainis [2, 1]. Some problems such as Triangle, k-Clique, Graph Matching, and And-Or Tree draw a lot of attention recently partly because their exact quantum query complexity is still unknown. Recently it has been independently showed by Zhang [10], Szegedy [9], Laplante and Magniez [7] that it is impossible to use the quantum adversary method to give lower bounds better than the current known ones for those problems. So we have to use other lower bound techniques to try to improve the current lower bounds. The average sensitivity is a lower bound of quantum query complexity [8]. However, the results in the present paper implies that the average sensitivity method is not strong enough either to improve the current lower bounds of those problems, because all the problems mentioned above are monotone functions, and the current lower bounds are already no less than $\Omega(\sqrt{n})$ [10].

2 The average sensitivity of monotone functions

The definition of the sensitivity and the average sensitivity are as follows.

Definition 1 *The sensitivity of a Boolean function f on input $x = x_1x_2...x_n \in \{0,1\}^n$ is*

$$s(f,x) = |\{i \in \{1,...,n\} : f(x) \neq f(x^{(i)})\}|,$$

*This research was supported in part by NSF grant CCR-0310466.

†Computer Science Department, Princeton University, NJ 08544, USA. Email: szhang@cs.princeton.edu

where $x^{(i)}$ is the n-bit string obtained from x by flipping x_i. The sensitivity of f is

$$s(f) = \max_{x\in\{0,1\}^n} s(f,x),$$

and the average sensitivity of f is

$$\bar{s}(f) = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} s(f,x).$$

Some notations are as follows. Let $[n]$ to denote the set $\{1,...,n\}$. For any input $x \in \{0,1\}^n$, denote $S_0(x) = \{i \in [n] : x_i = 0\}$ and $S_1(x) = \{i \in [n] : x_i = 1\}$. The weight of x is $|x| = |S_1(x)|$. We say x is a b-input if $f(x) = b$ ($b \in \{0,1\}$). Denote by $N_i(f)$ the number of 1-inputs that have weight i. Clearly, if f is not constant, then $f(00...0) = 0$ and $f(11...1) = 1$, thus we have $N_0(f) = 0$ and $N_n(f) = 1$.

It turns out that the average sensitivity of any non-constant monotone function only depends on $\{N_i(f)\}_{i=1,...,n-1}$, as showed by the following theorem.

Theorem 1 *For any non-constant monotone Boolean function f,*

$$\bar{s}(f) = [2\sum_{i=1}^{n} iN_i(f) - n\sum_{i=1}^{n} N_i(f)]/2^{n-1} \tag{1}$$

Proof We shall use notations omitting f when no confusion is caused. For example, N_i is short for $N_i(f)$. Let $s_{all}(f) = \sum_{x\in\{0,1\}^n} s(f,x)$, and we shall show $s_{all}(f) = 4\sum_{i=1}^n iN_i - 2n\sum_{i=1}^n N_i$, which immediately implies (1).

We first define

$$A_{i,\neq} = \{(x,y) : |x| = i,\ y = x^{(j)} \text{ for some } j \in S_0(x),\ f(x) \neq f(y)\}. \tag{2}$$

Then it is easy to check that $s_{all} = 2\sum_{i=0}^{n-1} |A_{i,\neq}|$. (The factor of 2 is because any $(x,y) \in A_{i,\neq}$ contributes 1 for $s(f,x)$, and also contribute 1 for $s(f,y)$, but the latter is not counted in $\sum_{i=0}^{n-1} |A_{i,\neq}|$.) We further define

$$A_{i,b} = \{(x,y) : |x| = i,\ y = x^{(j)} \text{ for some } j \in S_0(x),\ f(x) = f(y) = b\}, \tag{3}$$

for $b \in \{0,1\}$, and

$$A_i = \{(x,y) : |x| = i,\ y = x^{(j)} \text{ for some } j \in S_0(x)\} \tag{4}$$

Note that $A_{i,0}, A_{i,1}, A_{i,\neq}$ is a partition of A_i (*i.e.* $A_{i,0}, A_{i,1}, A_{i,\neq}$ are pairwise disjoint and the union of them is exactly A_i), so $|A_{i,\neq}| = |A_i| - |A_{i,0}| - |A_{i,1}|$. It is easy to see $|A_i| = \binom{n}{i}(n-i)$, and

$$A_{i,0} = \bigcup_{|y|=i+1, f(y)=0} \{(x,y) : y = x^{(j)} \text{ for some } j \in S_0(x)\} \tag{5}$$

by noting the monotonicity of f. Also note that the sets in the union above are disjoint, so we have

$$|A_{i,0}| = \left[\binom{n}{i+1} - N_{i+1}\right](i+1). \tag{6}$$

Similarly, we know

$$|A_{i,1}| = |\bigcup_{|x|=i, f(x)=1} \{(x,y) : y = x^{(j)} \text{ for some } j \in S_0(x)\}| = N_i(n-i). \tag{7}$$

Therefore,

$$
\begin{aligned}
& \textstyle\sum_{i=0}^{n-1} |A_{i,\neq}| \\
= \;& \textstyle\sum_{i=0}^{n-1} (|A_i| - |A_{i,0}| - |A_{i,1}|) \\
= \;& \textstyle\sum_{i=0}^{n-1} \binom{n}{i}(n-i) - \sum_{i=0}^{n-1} N_i(n-i) - \sum_{i=0}^{n-1} \left[\binom{n}{i+1} - N_{i+1}\right](i+1) \\
= \;& \textstyle n\sum_{i=0}^{n-1} \binom{n}{i} - \sum_{i=0}^{n-1} i\binom{n}{i} - n\sum_{i=0}^{n-1} N_i + \sum_{i=0}^{n-1} iN_i - \sum_{i=0}^{n-1}(i+1)\binom{n}{i+1} + \sum_{i=0}^{n-1}(i+1)N_{i+1} \\
= \;& \textstyle n(2^n-1) - 2\sum_{i=0}^{n-1} i\binom{n}{i} - n - n\sum_{i=0}^{n-1} N_i + 2\sum_{i=0}^{n-1} iN_i + n \\
= \;& \textstyle n(2^n-1) - 2\sum_{i=1}^{n-1} i\binom{n}{i} - n\sum_{i=1}^{n-1} N_i + 2\sum_{i=1}^{n-1} iN_i
\end{aligned}
\tag{8}
$$

Note that $2\sum_{i=1}^{n-1} i\binom{n}{i} = \sum_{i=1}^{n-1} i\binom{n}{i} + \sum_{i=1}^{n-1}(n-i)\binom{n}{i} = n\sum_{i=1}^{n-1}\binom{n}{i} = n(2^n-2)$. Thus we get

$$\sum_{i=0}^{n-1} |A_{i,\neq}| = n + 2\sum_{i=1}^{n-1} iN_i - n\sum_{i=1}^{n-1} N_i = 2\sum_{i=1}^{n} iN_i - n\sum_{i=1}^{n} N_i, \tag{9}$$

as desired. □

We now consider the range of the average sensitivity $\bar{s}(f)$. In general, $\bar{s}(f)$ can be tiny, for example, $\bar{s}(\textsc{And}) = \bar{s}(\textsc{Or}) = n/2^{n-1}$, and it is easy to argue that this is the smallest possible $\bar{s}(f)$ for non-constant f, no matter whether f is monotone or not. On the other hand, the average sensitivity can also be large, for example $\bar{s}(\textsc{Parity}) = n$. But it turns out that for monotone functions, we have the following tight bounds for the average sensitivity.

Theorem 2 *For any non-constant monotone Boolean function f,*

$$n/2^{n-1} \leq \bar{s}(f) \leq \binom{n}{\lfloor n/2 \rfloor}\lceil n/2\rceil/2^{n-1} = \Theta(\sqrt{n}). \tag{10}$$

The lower bound is tight for functions AND *and* OR*, and the upper bound is tight for the function* MAJORITY.

Proof Again let $s_{all}(f) = \sum_{x\in\{0,1\}^n} s(f,x)$, and we shall show

$$2n \leq s_{all}(f) \leq 2\binom{n}{\lfloor n/2 \rfloor}\lceil n/2\rceil. \tag{11}$$

The proof of the lower bound part of (11) is easy and omitted here; we now show the interesting upper bound part. Suppose $\bar{x}$ is one of those 1-inputs with minimal weight. That is, $f(\bar{x}) = 1$, and for any $i \in S_1(\bar{x})$, we have $f(\bar{x}^{(i)}) = 0$. Note that for any $i \in S_0(\bar{x})$, we have $f(\bar{x}^{(i)}) = 1$ because f is monotone. We define another function f' by

$$f'(x) = \begin{cases} 0 & \text{if } x = \bar{x} \\ f(x) & \text{if } x \neq \bar{x} \end{cases}.$$

Then f' is also monotone. Also note that from f to f', we only change function value for one input (*i.e.* $\bar{x}$), thus the the change of $s_{all}(f)$ to $s_{all}(f')$ is only due to this change. To be more precise, each $i \in S_1(\bar{x})$ contributes 2 in $s_{all}(f)$ (1 in $s(f,\bar{x})$ and 1 in $s(f,\bar{x}^{(i)})$), but does not contribute in $s(f')$; each $i \in S_0$ does exactly the opposite: it contributes 2 in $s(f')$ but does not contribute in $s(f)$. Therefore, we have

$$s_{all}(f') = s_{all}(f) - 2|\bar{x}| + 2(n - |\bar{x}|) \tag{12}$$

which implies $s_{all}(f') > s_{all}(f)$ if $|\bar{x}| < n/2$. So changing the function value for a minimum weight 1-input x increases s_{all} if $|x| < n/2$. We repeat this process until $f(x) = 0$ for all x with $|x| < n/2$, the the s_{all} increases during the course. Symmetrically, we use a similar way to let $f(x) = 1$ for all x with $|x| > n/2$, and s_{all} also increases during the course. We eventually end up with a function

f_{max} with $f_{max}(x) = 0$ if $|x| < n/2$ and $f_{max}(x) = 1$ if $|x| > n/2$. If n is odd, then this is exactly the function MAJORITY, whose s_{all} is $2\binom{n}{(n-1)/2}\frac{n+1}{2}$. If n is even, then the only inputs whose function values are not determined yet are those with weight equal to $n/2$. But actually whether the function value is 0 or 1 does not matter to s_{all} because of Equation (12). Thus we also have $s_{all}(f_{max}) = s_{all}(\text{MAJORITY}) = 2\binom{n}{n/2}n/2$. Combining the two cases completes the proof of the upper bound of (11). $\square$

References

[1] A. Ambainis. Polynomial degree vs. quantum query complexity, FOCS'03, 230-241, 2003. Earlier version at quant-ph/0305028

[2] A. Ambainis. Quantum lower bounds by quantum arguments, Journal of Computer and System Sciences, 64:750-767, 2002. Earlier version at STOC'00

[3] R. Beals, H. Buhrman, R. Cleve, M.Mosca, R. deWolf. Quantum lower bounds by polynomials. Journal of ACM, 48: 778-797, 2001. Earlier versions at FOCS'98 and quant-ph/9802049

[4] A. Bernasconi. Sensitivity vs. block sensitivity (an average-case study). Information Processing Letters, 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. Buhrman, R. de Wolf. Complexity measures and decision tree complexity: a survey. Theoretical Computer Science, Volume 288, Issue 1, 21-43, 2002

[7] S. Laplante, F. Magniez. Lower bounds for randomized and quantum query complexity using Kolmogorov arguments. 19th IEEE Conference on Computational Complexity, 294-304, 2004. Earlier version at quant-ph/0311189

[8] Y. Shi. Lower bounds of quantum black-box complexity and degree of approximating polynomials by influence of Boolean variables. Information Processing Letters, 75(1-2): 79-83, 2000. quant-ph/9904107

[9] M. Szegedy. A note on the limitations of quantum adversary method, manuscript

[10] Shengyu Zhang. On the power of Ambainis' lower bounds, 31st International Colloquium on Automata, Languages and Programming, 1238-1250. Earlier version at quant-ph/0311060