Pseudo-average block sensitivity equals average sensitivity

S. Venkatesh
School of Technology and Computer Science
Tata Institute of Fundamental Research
Homi Bhabha Road, Mumbai 400005, INDIA
email: venkat@tcs.tifr.res.in

Abstract

Bernasconi [1] proved that the *pseudo-average block sensitivity* and the *average sensitivity* of a Boolean function are equal. Her proof relies on identities from harmonic analysis. In this note, we give a direct proof.

keywords: combinatorial problems, Boolean function, sensitivity, block sensitivity.

1 Introduction

Sensitivity and *block sensitivity* are measures of the complexity of Boolean functions. Cook, Dwork and Reischuk [3] showed that any CREW PRAM computing a Boolean function requires time proportional to the logarithm of its sensitivity. Nisan [6] strengthened this relationship by introducing the notion of block sensitivity; he showed that the complexity of computing a function on a CREW PRAM is both upper and lower bounded (up to constant factors) by the logarithm of its block sensitivity. It is not known if the sensitivity and block sensitivity of a function are polynomially related; the widest gap known is due to Rubinstein [7], who showed an infinite family of Boolean functions with a quadratic gap between the two measures.

Bernasconi studied the relationship between the average versions of these two measures. She proposed two definitions for average block sensitivity. For the first definition, which she called the *average block sensitivity*, she showed that Rubinstein's function on n variables has average block sensitivity at least $\sqrt{n}$, whereas its average sensitivity goes to zero as n increases. For the second definition, which she called the *pseudo-average block sensitivity*, she showed that the two measures are equal in the average case. Her proof of this fact used results from harmonic analysis of Boolean functions. In this note, we give a simple direct proof of the same result.

2 Definitions

For a binary string $w \in \{0,1\}^n$, we denote by $w^{(i)}$, the string obtained from w by flipping its ith bit. Similarly, given a subset of indices $S \subseteq \{1, 2, \ldots, n\}$, we denote by $w^{(S)}$, the string obtained from w by flipping all bits in S. In what follows, f denotes a Boolean function on n variables.

Definition 1 (Sensitivity) *The sensitivity of f on the string $w \in \{0,1\}^n$, $S_w(f)$, is the number of indices i such that $f(w) \neq f(w^{(i)})$.*

Definition 2 (Average sensitivity) *The average sensitivity of f, $\overline{S}(f)$, is given by:*

$$\overline{S}(f) = \frac{1}{2^n} \sum_w S_w(f).$$

Definition 3 (Block sensitivity) *Let $d = \{S_1, S_2, ..., S_k\}$ be a partition of $\{1, 2, \ldots, n\}$. Then the block sensitivity of f on w with respect to d, $bs_{d,w}(f)$, is the number of blocks S_i such that $f(w) \neq f(w^{(S_i)})$.*

Definition 4 (Pseudo-average block sensitivity) *The Pseudo-average block sensitivity of a Boolean function f is given by:*

$$\widetilde{BS}(f) = \max_d \frac{1}{2^n} \sum_w bs_{d,w}(f),$$

where d ranges over all partitions of $\{1, 2, \ldots, n\}$.

Definition 5 (Average block sensitivity) *The Average block sensitivity of a Boolean function f is given by:*

$$\overline{BS}(f) = \frac{1}{2^n} \sum_w \max_d bs_{d,w}(f),$$

where d ranges over all partitions of $\{1, 2, \ldots, n\}$.

3 The computational significance of $\widetilde{BS}$

Why is *block sensitivity* an important notion? Nisan [6] showed that logarithm of the block sensitivity provides a lower bound for the time complexity of computing the function on a CREW PRAM. Before [6], it was known that logarithm of the sensitivity provides a lower bound. In order to strengthen this result, Nisan observed that for every function f, there is another function g, such that $S(g) \geq BS(f)$. Furthermore, any algorithm for f can be modified to compute g, without much increase in running time. The lower bound in terms of block sensitivity thus follows immediately from the earlier lower bound in terms of sensitivity.

We can ask if something similar is possible for average sensitivity. First, are there any computational lower bounds known that use average sensitivity? Yes. We have the following theorem of Boppana [2] (see also Linial, Mansour and Nisan [5]).

- Any depth d circuit computing a function f has size at least $2^{\Omega(\overline{S}^{1/(d-1)})}$, where $\overline{S}$ is the average sensitivity of f.

Can we replace average sensitivity by average block sensitivity (for some definition of average block sensitivity), just as Nisan did for CREW PRAM complexity? The answer is yes.

Proposition 6 *Any depth d circuit computing f has size at least $2^{\Omega(b^{1/(d-1)})}$, where b is the pseudo-average block sensitivity of f, i.e. $b = \widetilde{BS}(f)$.*

Proof: Let f be a Boolean function on n variables $\{x_1, x_2, \ldots, x_n\}$. Let the pseudo-average block sensitivity of f be achieved for the partition $\{B_1, B_2, \ldots, B_k\}$. Then, let g be a function on $n + k$ variables $\{z_1, z_2, \ldots, z_n, y_1, y_2, \ldots, y_k\}$, obtained from f by replacing the variable x_i by $z_i \oplus y_j$, where B_j is the unique block containing i. It is easy to check that $\overline{S}(g) \geq \widetilde{BS}(f)$. Hence, by appropriately fixing the values for $\{z_1, \ldots, z_n\}$(using an averaging argument), we can derive from g a Boolean function h on k variables $\{y_1, y_2, \ldots, y_k\}$ such that the following claims hold.

1. $\overline{S}(h) \geq \widetilde{BS}(f)$.
2. If f can be computed by a circuit of size ℓ and depth d, then there is a circuit of size ℓ and depth d computing h.

The proposition follows from the above result of Boppana relating circuit complexity to average sensitivity. □

Remark 1: Note that the above proof uses the standard method of substitution employed in Nisan's paper, and is independent of Bernasconi's result that $\overline{S} = \widetilde{BS}$. No such claim appears to be true for $\overline{BS}(f)$. In fact, there cannot be, because Rubinfeld's function has $\overline{BS}$ equal to $\Omega(\sqrt{n})$, but it can be computed in depth two and size $O(n^{3/2})$!

Remark 2: We note that Kahn, Kalai and Linial [4] study average sensitivity as sum of influences of different variables. In this setting, each variable is controlled by a different player, and we wish to know how many players on an average have control over the value of the function. If this is generalised to allow each player to control more than one variable (each player can simultaneously flip all the variables he controls), in such a way that no two players control the same variable, then the sum of the influences of the players is exactly $\widetilde{BS}(f)$. There doesn't appear to be any corresponding interpretation for $\overline{BS}(f)$.

4 Pseudo-average block sensitivity versus average sensitivity

Proposition 6 proved in the previous section gives us the following: lower bounds on $\widetilde{BS}$ imply lower bounds on circuit complexity. Unfortunately, this technique is unlikely to yield improved lower bounds; Bernasconi has shown that $\widetilde{BS}$ is equal to $\overline{S}$. Her proof makes essential use of the following relationship between average sensitivity and the Fourier transform of the function shown by Kahn, Kalai and Linial [4].

$$\overline{S}(f) = 4 \sum_{S} |S| \hat{f}(S)^2.$$

Here $\hat{f}$ is the Fourier transform of f.
In this note, we give a direct combinatorial proof of the same result.

Theorem 7 (Bernasconi [1]) $\widetilde{BS}(f) = \overline{S}(f)$.

Our proof uses the following lemma.

Lemma 8 *For every partition d, $\sum_{w} bs_{d,w}(f) \leq \sum_{w} S_w(f)$.*

Proof: With every $(w, S) \in \{0,1\}^n \times d$ such that $f(w) \neq f(w^{(S)})$, we will associate a pair $(x, i) \in \{0,1\}^n \times \{1, 2, \ldots, n\}$ such that $f(x^{(i)}) \neq f(x)$. Furthermore, this association will be one-to-one. The lemma follows from this.

Consider a string w and a block S such that $f(w) \neq f(w^{(S)})$. We now describe the procedure to obtain (x, i) from (w, S). Let $S = \{i_1, i_2, \ldots, i_t\}, i_1 < i_2 < \ldots < i_t$. We start with w, and first flip bit i_1, then bit i_2, and so on until the value of f changes. Suppose this happens when bit i_l is flipped. Such a bit exists since the value of f changes when all bits in S are flipped. We associate (w, S) with (x, i_l), where $x = w^{(\{i_1,\ldots,i_{l-1}\})}$; note that $f(x) \neq f(x^{(i_l)})$.

Suppose the above procedure maps (w, S) to (x, i). Let $S_i \in d$ be the block containing i and $T_i = \{j \in S_i \mid j < i\}$. Then it can be checked that $S = S_i$ and $w = x^{(T_i)}$. Hence, the mapping is one-to-one. □

Proof of Theorem: Now, $\overline{S}(f) = \frac{1}{2^n} \sum_w bs_{d,w}(f)$, where d is the partition of $\{1, 2, \ldots, n\}$ into singleton sets. Thus, $\overline{S}(f) \leq \widetilde{BS}(f)$. On the other hand, from the above lemmai :e get that for all partitions d, $\frac{1}{2^n} \sum_{w} bs_{d,w}(f) \leq \frac{1}{2^n} \sum_{w} S_w(f)$. Thus, $\widetilde{BS}(f) \leq \overline{S}(f)$. □

References

[1] A. Bernasconi: Sensitivity vs. Block Sensitivity (an average-case study), *Information processing letters*, 59(3), 1996, 151-158.

[2] R. Boppana: The average sensitivity of bounded-depth circuits, *Information processing letters*, 63(5), 1997, 257-261.

[3] S. Cook, C. Dwork, R. Reischuk: Upper and lower time bounds for parallel random access machines without simultaneous writes, *SIAM Journal of Computing*, 15(1), 1986, 87-97.

[4] J. Kahn, G. Kalai, N. Linial: The Influence of Variables on Boolean Functions, in *proceedings of the 29th FOCS*, 1988, 68-80.

[5] N. Linial, Y. Mansour, N. Nisan: Constant depth Circuits, Fourier Transform, and Learnability, *Journal of the ACM*, 40(3), 1993, 607-620.

[6] Noam Nisan: CREW PRAMS and Decision Trees, *SIAM Journal of Computing*, 20(6), 1991, 999-1007.

[7] D. Rubinstein: Sensitivity vs. Block Sensitivity of Boolean Functions, *Combinatorica* 15(2), 1995, 297-299.