Sensitivity vs block sensitivity of Boolean functions

Madars Virza
madars@gmail.com
Advisor: Andris Ambainis

Sensitivity of a Boolean function

Sensitivity to a bit

Function f is *sensitive* to bit i on x if

$$f(x_1, \ldots, x_i, \ldots, x_n) \neq f(x_1, \ldots, 1 - x_i, \ldots, x_n)$$

Sensitivity on a word

The sensitivity of f on x is the number of sensitive bits on x:

$$s(f, x) = |\{i | f(x_1, \ldots, x_i, \ldots, x_n) \neq f(x_1, \ldots, 1 - x_i, \ldots, x_n)\}|$$

Sensitivity of a Boolean function

The sensitivity of f is the maximum of sensitivities on a word over all inputs: $s(f) = \max_w s(f, w)$.

Sensitivity: an example

Three-argument majority function

$$f(x_1, x_2, x_3) = \begin{cases} 1 & \text{if at least two of } x_1, x_2, x_3 \text{ are 1,} \\ 0 & \text{otherwise.} \end{cases}$$

Sensitivity of majority

- $s(f, 011) = 2$, because there are exactly two bits, namely x_2 ($f(011) \neq f(010)$) and x_3 ($f(011) \neq f(001)$), whose change would change the value of f;
- $s(f, 000) = 0$ $\quad s(f, 001) = 2$ $\quad s(f, 010) = 2$ $\quad s(f, 011) = 2$
  $s(f, 100) = 2$ $\quad s(f, 110) = 2$ $\quad s(f, 101) = 2$ $\quad s(f, 111) = 0$;
- we conclude that the sensitivity of f is 2.

Block sensitivity of a Boolean function

Blocks

Let x be a Boolean string of length n and let S be any subset of indices, we will call S a "block". By x^S we will mean x with all the bits in S flipped.

Sensitivity to block

Function f is *sensitive* to block S on x if $f(x) \neq f(x^S)$

Example

The three-argument majority function is sensitive to block $1, 2$ on 000, because $f(0, 0, 0) \neq f(1, 1, 0)$.

Block sensitivity of a Boolean function (cont.)

Block sensitivity on word

The *block sensitivity* $bs(f, x)$ of f on input x is defined as the maximum number k of disjoint subsets $B_1, \ldots, B_k$ of $\{1, 2, \ldots, n\}$ such that for each B_i, $f(x) \neq f(x^{B_i})$.

Block sensitivity of a function

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

A generalization of sensitivity

The relation to sensitivity is immediate: block sensivity generalizes different bits to disjoint blocks.

The block sensitivity problem

The main open problem

Is there a constant c such that $bs(f) = O(s^c(f))$?

Progress on the block sensitivity problem

- the best known upper bound of block sensitivity in terms of sensitivity is exponential [Kenyon04];
- the best separation is quadratic: $bs(f) = \frac{1}{2}s(f)^2$ for Rubinstein's function [Rubinstein95];
- the gap has been exponential for more than 20 years.

The importance of the block sensitivity problem

A possible proof technique

- note that for most functions it is *very* easy to determine their sensitivity. We can't say the same about other complexity measures;
- also note that block sensitivity is polynomially related to almost every other complexity measure: deterministic query complexity, quantum query complexity, certificate complexity, etc.;
- if sensitivity and block sensitivity are polynomially related, then we would have an immense number of new results about other complexity measures.

Approaching the block sensitivity problem

The results of Kenyon and Kutin

- this paper demonstrates the best upper bound (though exponential). Their proof is via l-block sensitivity, which limits the block size to at most l.
- at the end of the paper there is an interesting open question Q about bs_2 and s;
- even an improvement to the constants in the relationship between bs_2 and s could lead to a subexponential upper bound of block sensitivity in terms of sensitivity;

A possible attack – trying small examples

- investigate small examples looking for improvements to Q;
- if there is a small example that substantially improves the solution to Q and we are able to generalize it, then we have proved a new upper bound of block sensitivity!

Investigating Boolean functions of low degree

Exhaustive search

- the number of n variable Boolean functions is 2^{2^n};
- an exhaustive search is unfeasible for even $n = 5$;
- a result obtained by exhaustive search: a short proof of sub-quadratic separation between sensitivity and block sensitivity.

An idea from cryptography – reducing the problem to SAT

- build a SAT instance by considering 2^n variables corresponding to the values of $f(x_1, x_2, \dots x_n)$;
- add additional variables and clauses for constraints $s(f) \leq s$ and $bs(f) \geq bs$, for arbitrary constants s, bs;
- use a SAT solver on the resulting problem instances.

Results

Results of computer search

- we found a 9-argument function with a somewhat simple structure and $bs(f) > \frac{1}{2}s(f)^2$;
- our experiments give a complete characterization of possible s and bs pairs for every n not exceeding 12.

The main result – improved separation

- we were able to generalize our function ($bs(f) = \frac{1}{2}s(f)^2 + \frac{1}{2}s(f)$), thereby improving the best known separation;
- our function also improves the best results about question from paper by Kenyon and Kutin, however this improvement is not strong enough to prove the subexponential bound we seeked.

The main result

The main theorem

For every non-negative integer k there exists a Boolean function f of $n = (2k+1)^2$ variables, for which $s(f) = 2k+1$ and $bs(f) = (2k+1)(k+1)$.

Our function

An example of such a function is given by dividing variables into $2k+1$ disjoint sections with $2k+1$ variables in each section. We define f to be 1 iff there is a section $x_1, x_2, \ldots, x_{2k+1}$ such that either:

(i) $x_{2i-1} = x_{2i} = 1$ for some $1 \leq i \leq k$ and all other x_j's are 0, or

(ii) $x_{2k+1} = 1$ and all other x_j's are 0.

Our function

A concrete example: $n = 9^2$

	1	2	3	4	5	6	7	8	9	
1	0	0	**1**	**1**	0	0	0	0	0	
2	0	0	0	0	0	0	**1**	**1**	0	
3	**1**	**1**	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	**1**	
5	0	0	**1**	**0**	0	0	0	0	0	incomplete pair
6	0	0	**1**	**1**	0	**1**	0	0	0	extra 1 bit
7	0	**1**	**1**	0	0	**1**	0	0	0	misaligned pair
8	0	0	0	0	0	0	0	0	0	all zeroes
9	0	0	0	0	0	0	0	0	0	all zeroes

A row is "good" if either:

- there is exactly one 1 bit and it is in the last column;
- there is exactly two 1 bits, they are "paired" and the pair is correctly aligned.

Proof of the main result

$s(f) \geq 2k+1$, $bs(f) \geq (2k+1)(k+1)$

- we observe that for input $w = 0 \dots 0$ we have $s(f, w) \geq 2k+1$ and $bs(f, w) \geq (2k+1)(k+1)$

$bs(f) = (2k+1)(k+1)$

- assume we have already proved that $s(f) = 2k+1$;
- assume that the maximal block sensitivity is achieved using u blocks of size 1 and v blocks of size a least 2;
- from the sensitivity we have $u \leq 2k+1$;
- from the total number of variables we have $u + 2v \leq (2k+1)^2$;
- taken together: $bs(f) = u + v \leq \frac{1}{2}((2k+1) + (2k+1)^2) = (2k+1)(k+1)$.

Proof of the main result (cont.)

$s(f) \leq 2k + 1$

We consider two cases for arbitrary input w:

- $f(w) = 1$
  - if there is only one "good" section, we have at most $2k + 1$ choices for the bit to alter;
  - if there are at least two "good" sections, we can't change the value of f by flipping just one input bit.
- $f(w) = 0$
  - we prove that for each of the $2k + 1$ sections there is at most one bit whose change could flip the value of f;
  - proof by case analysis (consider how 1 bits could be distributed among pairs and unpaired bit).

Questions

Questions?

More details in our paper:

`http://tinyurl.com/blocksensitivity`