Computational aspects for the nonlinearity of Boolean functions

Massimiliano Sala
(with Alessio Meneghetti)

University of Trento

maxsalacodes@gmail.com

BFA 2018 Loen (Norway) - June 19, 2018

Definitions

$$(\mathbb{F}_2)^n = \{v_1, \ldots, v_{2^n}\}, \qquad f : (\mathbb{F}_2)^n \to \mathbb{F}_2$$

Algebraic Normal Form: $f = \sum_{v \in (\mathbb{F}_2)^n} f_v X^v$, e.g. $X^{(110)} = x_1 x_2$

$$(f_{v_1}, f_{v_2}, \ldots, f_{v_{2^n}}) \in (\mathbb{F}_2)^{2^n}$$

Lookup Table: $\{v \to f(v)\}$

$$(f(v_1), f(v_2), \ldots, f(v_{2^n})) \in (\mathbb{F}_2)^{2^n}$$

Evaluation and the binary Moebius transform

$$f_v = \bar{f}(v)$$

$$\begin{array}{ccc} f = \sum_{v \in (\mathbb{F}_2)^n} f_v X^v & \longrightarrow & (f(v_1), f(v_2), \ldots, f(v_{2^n})) \\ \uparrow & & \downarrow \\ (\bar{f}(v_1), \bar{f}(v_2), \ldots, \bar{f}(v_{2^n})) & \longleftarrow & \bar{f} = \sum_{v \in (\mathbb{F}_2)^n} f(v) X^v \end{array}$$

Complexity considerations (?)

The computational effort required to go from a representation to the other is $O(n2^n)$ binary operations.

The actual complexity is still unknown.

Affine Functions

$$\alpha : (\mathbb{F}_2)^n \to \mathbb{F}_2$$

Algebraic Normal Form: $\alpha = a_0 + a_1 x_1 + \ldots + a_n x_n$

$$(a_0, a_1, \ldots, a_n) \in (\mathbb{F}_2)^{n+1}$$

Nonlinearity of a Boolean function

Distance between functions

The **distance** $\mathrm{d}(f, g)$ between two Boolean functions f and g is the number of $v \in (\mathbb{F}_2)^n$ for which $f(v) \neq g(v)$.

Again

The **distance** $\mathrm{d}(f, g)$ between f and g is the Hamming distance between the corresponding evaluation vectors.

Nonlinearity of a Boolean function

Nonlinearity

The nonlinearity of f is the minimum of the distances between f and any affine function α

$$\mathrm{nl}(f) = \min_{\alpha} \mathrm{d}(f, \alpha)$$

Maximum nonlinearity

$$\mathrm{nl}(f) \ \leq \ 2^{n-1} - 2^{\frac{n}{2}-1}$$

Bent function

f is bent iff $\mathrm{nl}(f) \ = \ 2^{n-1} - 2^{\frac{n}{2}-1}$.

Decision problems

For any $n \geq 1$, let us consider a sequence of sets $\mathcal{I}_n$.

A **decision problem** $\mathcal{P}$ is a function

$$\forall n, \quad \mathcal{I}_n \mapsto \{true, false\}.$$

- An element of $\mathcal{I}_n$ is called an **instance of the problem** $\mathcal{P}$
- n is called the **complexity parameter**,
- so,
  $\mathcal{I}_n$ is also called the **set of inputs**
  (implicitely assuming parameter complexity n).

Example of decision problems

If $\mathcal{I}_n$ is the set of all Boolean functions, we have many interesting decision problems:

- is f bent?
- is f affine?
- is $\mathrm{nl}(f) = 3$?

From decision problems to other problems

The last example suggests that, in our context, decision problems may be used as **building blocks** of any interesting problem.

How to measure complexity

There are many notions of complexity, which I found very confusing when I started approaching this area.
To measure complexity you have to make some inevitable choices:

- what you are measuring?
  I am considering only field operations in $\mathbb{F}_2$;
  I am **not** considering the cost of storing memory;
- how much? I am counting as **one operation** any bit addition, multiplication or memory reading.
- how to compare
  I am using only the big-O notation and for any n I am considering only worst-case complexity.

Decision problems as Boolean functions

Recall:
A **decision problem** $\mathcal{P}$ is a function

$$\forall n, \quad \mathcal{I}_n \mapsto \{true, false\} \, .$$

In our Boolean context, $\mathcal{I}_n \subset (\mathbb{F}_2)^N$, so
a **decision problem** $\mathcal{P}$ is the evaluation of a Boolean function

$$(\mathbb{F}_2)^N \mapsto \{true, false\} = \mathbb{F}_2 \, .$$

However, the problem is not given in ANF or other convenient form!

Difficult decision problems

NP-complete

We do not give a formal definition, but believe me that (decision) NP-complete problems are, in some sense, the most difficult problems to solve.
If you find an algorithm that solves an NP-complete problem in strictly less than exponential time, then you have done a major step in both Mathematics and Computer Science!

An NP-complete problem I love

Given a Boolean function f whose evaluation in each point requires $O(n^3)$ operations, decide whether

$$f = 1$$

or equivalently, if f has any root.

A result by Pan

The problem with understanding the actual complexity of problems is that it is very difficult to find lower bounds:
you must show that any algorithm solving $\mathcal{P}$ needs **at least** xxxx operations.

Leaving the Boolean world

Let $\mathcal{I}_n$ be the set of all univariate polynomial with complex coefficient with degree n.
Let us consider the problem $\mathcal{P}$ of (exactly) evaluating a polynomial in any (complex) point, counting (complex) multiplication and (complex) additions.

A result by Pan II

Theorem (Viktor Y. Pan, 1966)

To solve $\mathcal{P}$ you need at least n operations.

Nonlinearity as a Coding Problem

A Reed-Muller code of first order is the linear binary code obtained by evaluating all affine functions. It is a $[2^n, n+1, 2^{n-1}]_2$ code.

$$\mathrm{nl}(f) \longleftrightarrow \mathrm{decode}\left(f(v_1), \ldots, f(v_{2^n})\right)$$

Complexity considerations

If $\mathrm{nl}(f) < 2^{n-2}$ then we can compute it in $O(n^3)$ operations.
Recent works suggest that this bound can be significantly lowered.

The complexity of correcting beyond the distance is not known.
For general linear codes it is NP-hard.

The Walsh transform

$$\begin{array}{rccc} f: & (\mathbb{F}_2)^n & \longrightarrow & \mathbb{F}_2 \\ & & \downarrow & \\ \hat{f}: & (\mathbb{F}_2)^n & \longrightarrow & \mathbb{Z} \end{array}$$

$$\hat{f}(x) = \sum_{y \in (\mathbb{F}_2)^n} (-1)^{x \cdot y + f(y)}$$

Complexity considerations

The computation of the Walsh spectrum of f from its evaluation vector requires $O(n2^n)$ integer operations.

Open problem

Faster computation of the Walsh transform.

The Walsh transform

$$\mathrm{nl}(f) = \min_{y \in (\mathbb{F}_2)^n} \left\{ 2^{n-1} - \frac{1}{2}\hat{f}(v) \right\} = 2^{n-1} - \max_{y \in (\mathbb{F}_2)^n} \hat{f}(y)$$

Complexity considerations

From the evaluation vector, the computation of $\mathrm{nl}(f)$ using the Walsh transform requires $O(n2^n)$ integer operations.
Indeed, we obtain the same asymptotic cost starting from the ANF of f.

Numerical Normal Form of a function

Let f be a function on $\{0,1\}^n$ taking values in a field $\mathbb{K}$.
Its representation as a polynomial

$$f = \sum_{v \in \{0,1\}^n} \lambda_v X^v,$$

where $\lambda_v \in \mathbb{K}$, is called the Numerical Normal Form (NNF) of f.
Any Boolean function admits a unique NNF.

Complexity Considerations

The NNF of f can be computed from its truth table, and it requires $O(n2^n)$ additions over $\mathbb{K}$.

Multivariate Approach

In 2006 I have started considering the problem of nonlinearity for Boolean functions, using an approach based on multivariate polynomials.

Along the way, several researchers have contributed:

Emanuele Bellini, Eleonora Guerrini, Alessio Meneghetti, Theo Mora, Emmanuela Orsini, Ilaria Simonetti.

Notation

- $E[X] = E[x_1, \ldots, x_N] = \{x_1^2 - x_1, \ldots, x_N^2 - x_N\}$
- $\mathcal{M}_{N,t}$ is the set of all square-free monomials of degree t in $\mathbb{F}_2[x_1, \ldots, x_N]$.
- σ_i is the i-th elementary symmetric function $\sum_{\mathcal{M}_{N,t}} \mathrm{m}$.
- $I_{N,t} = \langle \{\sigma_t, \ldots, \sigma_N\} \cup E[X] \rangle$.
- $S_{N,t}$ is the Hamming Ball, $S_{N,t} = \{v \in (\mathbb{F}_2)^N \mid \mathrm{w}_H(v) \leq t\}$.
- $\varphi_{N,t}$ is the Boolean function vanishing exactly at $S_{N,t-1}$.

Vanishing Ideal of a Hamming Ball centred at zero

Theorem (Guerrini, Orsini, -)

Let $1 \leq t \leq N$. The vanishing ideal of $S_{N,t}$ is $I_{N,t+1}$.
Its reduced Groebner basis G (w.r.t any ordering) is

$$G = E[X] \cup \mathcal{M}_{N,t}, \quad \text{for } t \geq 2$$
$$G = \{x_1, \ldots, x_N\}, \quad \text{for } t = 1.$$

Theorem (Meneghetti)

In terms of the elementary symmetric functions, the ANF of $\varphi_t^{(N)}$ can be computed in $O(N \log N)$ operations. Moreover

$$I_{N,t} = \langle \{\varphi_{N,t}\} \cup E[X] \rangle$$

Generic affine Boolean functions

Let $A = \{a_i\}_{0 \le i \le n}$ be a variable set of $n + 1$ unknowns.

The polynomial $\alpha = a_0 + \sum_{i=1}^{n} a_i x_i$ in $\mathbb{F}_2[A, x_1, \ldots, x_n]$ represents a generic affine Boolean function in n variables.

Let $\overline{\alpha}$ be the evaluation vector of α:

$$\overline{\alpha} = (\alpha(A, v_1), \ldots, \alpha(A, v_{2^n})) \in (\mathbb{F}_2[A])^{2^n}$$

Note that $\overline{\alpha}$ is a vector of polynomials.

Simonetti's Ideal

Let $J_t^n(f)$ be the ideal in $\mathbb{F}_2[A]$ defined by

$$\left\langle \left\{ \mathrm{m}\left(\overline{\alpha} + \overline{\mathrm{f}}\right) \;\middle|\; \mathrm{m} \in \mathcal{M}_{\mathrm{N,t}} \right\} \cup E[A] \right\rangle$$

where $N = 2^n$.

Remark

$E[A] \subset J_t^n(f) \quad \Rightarrow \quad J_t^n(f)$ is zero-dimensional and radical.

Simonetti's Ideal

Lemma (Simonetti, -)

For any $1 \leq t \leq 2^n$ the following statements are equivalent:

1. $\mathcal{V}(J_t^n(f)) \neq \emptyset$
2. $\exists\, u \in \{\bar{\alpha} + \bar{f}\}$ such that $\mathrm{w}_H(u) \leq t - 1$
3. $\exists\, \alpha$ such that $\mathrm{d}(f, \alpha) \leq t - 1$

Theorem (Simonetti, -)

$\mathrm{nl}(f)$ is the minimum t for which $\mathcal{V}(J_t^n(f)) \neq \emptyset$

Simonetti's Ideal

Complexity Considerations

- A direct application of this method becomes impractical even for small values of n, since $\binom{2^n}{t}$ monomials should be evaluated.
- Computational experiments by E. Bellini suggest that only a few monomials need to be evaluated. Unfortunately there is no obvious way to select those monomials.

Open problem

- Given f, select the monomials in Simonetti's ideal that need to be evaluated.
- Find classes of Boolean functions such that the complexity of the method is low.

Meneghetti's method

For each $i = 1, \ldots, N = 2^n$, let

$$\beta_i(A) = \alpha(A, v_i) + f(v_i) \in \mathbb{F}_2[A].$$

Theorem (Meneghetti)

$$\mathrm{nl}(f) \geq t \qquad \Longleftrightarrow \qquad \varphi_{N,t}\,(\beta_1(A), \ldots, \beta_N(A)) = \varphi_{n+1,1}(A).$$

Meneghetti's method

Complexity Considerations

As the previous method, the computation of $\mathrm{nl}(f)$ is impractical, since $\binom{2^n}{t}$ multiplications involving affine functions are required.

Open problems

- Exploit symmetries of $\varphi_{N,t}$ to lower the complexity.
- Exploit symmetries of the set $\{f_i(A)\}_i$ to lower the complexity.
- Find classes of Boolean functions such that the complexity of the method is low.

Bellini's approach

Recall: $\beta_i(A) = \alpha(A, v_i) + f(v_i) \in \mathbb{F}_2[A]$.

Define:

- $\beta_i^{\mathbb{Z}}(A)$ is the NNF of $\beta_i(A)$.
- $\mathfrak{n}_f(A) = \beta_1^{\mathbb{Z}} + \ldots + \beta_{2^n}^{\mathbb{Z}}$ $\leftarrow$ nonlinearity polynomial
- $E_{\mathbb{Q}}[A] = \{a_0^2 - a_0, a_1^2 - a_1, \ldots, a_n^2 - a_n\} \subset \mathbb{Q}[A]$

Bellini's approach

Let us consider the projection

$$(\mathbb{F}_2)^{n+1} \to (\mathbb{F}_2)^n, \qquad v = (v_0, v_1, \ldots, v_n) \mapsto \tilde{v} = (v_1, \ldots, v_n)$$

Theorem (Bellini, -)

Let $\{c_v\}_{v\in(\mathbb{F}_2)^{n+1}}$ be such that $\mathfrak{n}_f(A) = \sum_{v\in\{0,1\}^{n+1}} c_v A^v$. Then

$$c_0 = \sum_{u\in(\mathbb{F}_2)^n} f(u)$$

$$c_v = (-2)^{\mathrm{w}_H(v)} \sum_{\tilde{v}\preceq u} \left[f(u) - \tfrac{1}{2}\right]$$

Complexity Considerations

Using a fast butterfly scheme, the computation of the nonlinearity polynomial requires $O(n2^n)$ integer sums ad doublings.

Bellini's approach

Let $\mathcal{N}_f^t = \langle E_{\mathbb{Q}}[A] \cup \{\mathfrak{n}_f - t\}\rangle$.

Theorem (Bellini, -)

$$\mathcal{V}(\mathcal{N}_f^t) \neq \emptyset \text{ if and only if } \mathrm{nl}(f) = t.$$

Complexity Considerations

The computation of $\mathrm{nl}(f)$ relies on a multivariate polynomial system. If we treat it as a generic ideal, we need to compute a Groebner basis.

Bellini's approach

Remark

The evaluation vector of the nonlinearity polynomial $\mathfrak{n}_f(A)$ represents the distances of f from all possible affine Boolean functions.

Theorem (Bellini, -)

$$\mathrm{nl}(f) = \min_{v \in \{0,1\}^{n+1}} \{\overline{\mathfrak{n}}_f(v)\}$$

Groebner description, natural and linear representations

Let:

- $\mathbb{K}$ be any field
- $J \subset \mathbb{K}[X]$ be a zero-dimensional ideal with $\deg(J) = s$
- $A = \mathbb{K}[X]/J$ the corresponding quotient algebra, with $\dim_{\mathbb{K}}(A) = s$

With a slight abuse of notation, we denote with $f \in A$ the residue class modulo J of $f \in \mathbb{K}[X]$. Let:

- $\phi_f(g)$ be the endomorphism $A \to A$ mapping g to $fg \in A$
- $\mathfrak{b} = \{b_1, \ldots, b_s\}$ a $\mathbb{K}-$basis of A

Groebner description, natural and linear representations

Any element $g \in A$ admits a unique representation of the form

$$g = \sum_j \gamma_j^{(\mathfrak{b})}(g) b_j.$$

The vector

$$\mathrm{Rep}(g, \mathfrak{b}) = \left(\gamma_1^{(\mathfrak{b})}(g), \ldots, \gamma_s^{(\mathfrak{b})}(g)\right)$$

is known as the Groebner description of g.

Remark

The endomorphism ϕ_f is represented by the square matrix

$$M_{f,\mathfrak{b}} = \left[\gamma_j^{(\mathfrak{b})}(fb_i)\right].$$

Groebner description, natural and linear representations

A natural representation of the ideal J consists of

- a $\mathbb{K}-$ basis $\mathfrak{b} \subset A$
- the square matrices $M_{x_1,\mathfrak{b}}, \ldots, M_{x_n,\mathfrak{b}}$.

Remark

An (optional) third object of a natural representation is the assignment of

- s^3 values $\gamma_{ijl} \in \mathbb{K}$ such that

$$b_i \cdot b_j = \sum_l \gamma_{ijl} b_l.$$

Groebner description, natural and linear representations

A set of monomials $N \subset \mathcal{M}$ is an escalier if it is an order ideal, i.e. if for each pair $\lambda, \tau \in \mathcal{M}$ for which $\lambda\tau \in N$, then $\tau \in N$.

A natural representation is called a linear representation if the basis $\mathfrak{b}$ of the reprensentation is an escalier.

A Groebner representation based algorithm

Input:

- The natural representation $\mathfrak{b}, M$ of a zero-dimensional ideal $I \subset \mathbb{F}_q[X]$
- The Groebner descriptions of a finite set of elements $F = \{f_1, \ldots, f_m\} \subset \mathbb{F}_q[X]$;

Output:

- The linear representation of the ideal $J = I \cup \langle F \rangle$.

A Groebner representation based algorithm

Idea:

- Start with $J = I$;
- Add to J an element of F at a time, by updating its natural representation.

At each step, some elements of the basis $\mathfrak{b}$ may be removed.

At the end, what remains is a natural representation $\mathfrak{b}'$ of J.

Complexity Considerations

This algorithm, known as Traverso's algorithm, needs to perform at most s loops each costing $O(ns^2)$.

A Groebner representation based algorithm

Theorem

Computing the Nonlinearity through Simonetti's system using Traverso's algorithm requires $O(n2^{2n})$ elementary operations.