Spectral Norm, Economical Sieve, and Linear Invariance Testing of Boolean functions

Chandrima Kayal

Université Paris Cité, CNRS, IRIF, Paris, France

Joint work with
Swarnalipa Datta (Indian Statistical Institute, Kolkata), Arijit Ghosh (Indian Statistical Institute, Kolkata), Manaswi Paraashar (IIT Hyderabad) and Manmatha Roy (Indian Statistical Institute, Kolkata)

STACS 2026

Contents

Definition: Property tester of functions

An algorithm (property tester) $\mathcal{A}$ $(\epsilon, \epsilon + \tau)$-tests $\mathcal{C}$, for a class of functions $f : \mathbb{F}_2^n \to \{-1, +1\}$, if given access to the truth table of a function f, whether

- f is "ϵ-close to $\mathcal{C}$"
- f is "$\epsilon + \tau$-far from $\mathcal{C}$"

can be tested using $\mathcal{A}$ with success probability (called the *confidence*) $\geq 2/3$.

Definition: Property tester of functions

An algorithm (property tester) $\mathcal{A}$ $(\epsilon, \epsilon + \tau)$-tests $\mathcal{C}$, for a class of functions $f : \mathbb{F}_2^n \to \{-1, +1\}$, if given access to the truth table of a function f, whether

- f is "ϵ-close to $\mathcal{C}$"
- f is "$\epsilon + \tau$-far from $\mathcal{C}$"

can be tested using $\mathcal{A}$ with success probability (called the *confidence*) $\geq 2/3$.

- The number of queries to the truth-table of f made by $\mathcal{A}$ is called the **query complexity** of $\mathcal{A}$.
- When $\epsilon = 0$, we call the property tester **non-tolerant**.
- When $\epsilon > 0$, we call the property tester **tolerant**.

Function Isomorphism

- $f : \mathbb{F}_2^n \to \{-1, +1\}$ and,

$$(f \circ \sigma)(x_1, \ldots, x_n) := f(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$$

where $\sigma \in S_n$ (the group of all permutations of $[n]$).

One Fundamental Question

(f:known, g:unknown)

- Given query access to g,
- if there exists a permutation $\sigma \in S_n$ such that $f \circ \sigma = g$,
- g is far from f?

Function Isomorphism

- $f : \mathbb{F}_2^n \to \{-1, +1\}$ and,

$$(f \circ \sigma)(x_1, \ldots, x_n) := f(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$$

where $\sigma \in S_n$ (the group of all permutations of $[n]$).

One Fundamental Question

(f:known, g:unknown)

- Given query access to g,
- if there exists a permutation $\sigma \in S_n$ such that $f \circ \sigma = g$,
- g is far from f?

- The **Hamming distance** between two functions $f_1 : \mathbb{F}_2^n \to \{-1, +1\}$ and $f_2 : \mathbb{F}_2^n \to \{-1, +1\}$ is defined as

$$\delta(f_1, f_2) = \Pr_x[f_1(x) \neq f_2(x)].$$

Linear Isomorphism

- For $A \in \mathbb{F}_2^{n \times n}$, let $f \circ A : \mathbb{F}_2^n \to \{-1, +1\}$ be the function $f \circ A(x) = f(Ax)$ for all $x \in \mathbb{F}_2^n$.

Linear Isomorphism

- For $A \in \mathbb{F}_2^{n \times n}$, let $f \circ A : \mathbb{F}_2^n \to \{-1, +1\}$ be the function $f \circ A(x) = f(Ax)$ for all $x \in \mathbb{F}_2^n$.
- Two Boolean functions $f, g : \mathbb{F}_2^n \to \{-1, +1\}$ are **Linearly isomorphic** to each other if

$$f = g \circ A,$$

where $A : V \to W$ is a linear transformation.

Linear Isomorphism

- For $A \in \mathbb{F}_2^{n \times n}$, let $f \circ A : \mathbb{F}_2^n \to \{-1, +1\}$ be the function $f \circ A(x) = f(Ax)$ for all $x \in \mathbb{F}_2^n$.
- Two Boolean functions $f, g : \mathbb{F}_2^n \to \{-1, +1\}$ are **Linearly isomorphic** to each other if

$$f = g \circ A,$$

  where $A : V \to W$ is a linear transformation.
- A linear transformation:
  1. $A(x + y) = A(x) + A(y)$,
  2. $A(cx) = cA(x)$, where $c \in \mathbb{F}$.

Linear Isomorphism

- For $A \in \mathbb{F}_2^{n \times n}$, let $f \circ A : \mathbb{F}_2^n \to \{-1, +1\}$ be the function $f \circ A(x) = f(Ax)$ for all $x \in \mathbb{F}_2^n$.
- Two Boolean functions $f, g : \mathbb{F}_2^n \to \{-1, +1\}$ are **Linearly isomorphic** to each other if

$$f = g \circ A,$$

  where $A : V \to W$ is a linear transformation.
- A linear transformation:
  1. $A(x + y) = A(x) + A(y)$,
  2. $A(cx) = cA(x)$, where $c \in \mathbb{F}$.
- **Question:** If f and g be linear transformation of each other?

- The **Hamming distance** between two functions $f_1 : \mathbb{F}_2^n \to \{-1, +1\}$ and $f_2 : \mathbb{F}_2^n \to \{-1, +1\}$ is defined as

$$\delta(f_1, f_2) = \Pr_x[f_1(x) \neq f_2(x)].$$

Linear Isomorphism Distance

The **Linear Isomorphism Distance** between two functions $f : \mathbb{F}_2^n \to \{-1, +1\}$ and $g : \mathbb{F}_2^n \to \{-1, +1\}$ is defined as

$$\mathrm{dist}_{\mathbb{F}_2^n}(f, g) = \min_{[A \in \mathbb{F}_2^{n \times n} : A \text{ is non-singular}]} \delta(f \circ A, g).$$

Testing Isomorphism of Boolean Functions

Let $f, g : \mathbb{F}_2^n \to \{-1, +1\}$ be Boolean functions.

- g is known. Given query access to f. For any $\epsilon \geq 0$ and $\tau > 0$, determine if
- $\exists$ linear transformation $A \in \mathbb{F}_2^{n \times n}$ such that $\delta(f \circ A, g) \leq \epsilon$, or
- $\forall$ linear transformation $A \in \mathbb{F}_2^{n \times n}$, $\delta(f \circ A, g) \geq \epsilon + \tau$.

In other words,

- $\mathrm{dist}_{\mathbb{F}_2^n}(f, g) \leq \epsilon$.
- $\mathrm{dist}_{\mathbb{F}_2^n}(f, g) \geq \epsilon + \tau$.

Why do we care?

Many important Boolean function classes are closed under function isomorphism:

- k-**juntas:** functions depending on at most k variables
- **Sparse functions:** functions having s many non-zero Fourier coefficients
- **Symmetric functions:** value depends only on Hamming weight

Why do we care?

Many important Boolean function classes are closed under function isomorphism:

- k-**juntas:** functions depending on at most k variables
- **Sparse functions:** functions having s many non-zero Fourier coefficients
- **Symmetric functions:** value depends only on Hamming weight
- **Graph properties:** viewed as Boolean functions on adjacency matrices

Question: Can we efficiently detect structural equivalence under function isomorphism?

Related works

- Fischer et al. were the first to initiate this where A is a permutation matrix.
- Followed by a long line of work by Alon and Blais [RANDOM'2010], Blais and O'Donnell [CCC'10], Chakraborty et al. [CCC'12], and many more.
- Testing linear isomorphism of Boolean functions was studied by Wimmer and Yoshida [ICALP'13], followed by Grigorescu, Wimmer and Xie [RANDOM'13].

Wimmer and Yoshida [ICALP 2013]

- Wimmer and Yoshida [ICALP 2013] solved this using $O\left(\frac{m^{24}}{\tau^{24}}\right)$ many queries.
- Here m is the upper bound on the spectral norm of g. That is,

$$\|\widehat{g}\|_1 = \sum_{r \in \mathbb{Z}_2^n} |\widehat{g}(\chi_r)| \leq m.$$

- Solves Affine-ismorphism as well.

Wimmer and Yoshida [ICALP 2013]

- Wimmer and Yoshida [ICALP 2013] solved this using $O\left(\frac{m^{24}}{\tau^{24}}\right)$ many queries.
- Here m is the upper bound on the spectral norm of g. That is,

$$\|\widehat{g}\|_1 = \sum_{r \in \mathbb{Z}_2^n} |\widehat{g}(\chi_r)| \leq m.$$

- Solves Affine-ismorphism as well.
- Kushilevitz-Mansour learning algorithm in implicit set-up.

Contents

Our result: Upper Bound

Theorem [Datta, Ghosh, Kayal, Paraashar, Roy]

- The problem **"Testing Isomorphism of Boolean Functions"** can be solved using $O\left(\frac{m^4}{\tau^4}\right)$ many queries.
- Here m is the upper bound on the spectral norm of the known function g. That is,

$$\|\widehat{g}\|_1 = \sum_{r \in \mathbb{Z}_2^n} |\widehat{g}(\chi_r)| \leq m.$$

Our Result: Lower Bound

- Maiorana–McFarland functions!

Theorem [Datta, Ghosh, Kayal, Paraashar, Roy]

For any $m > 0$,

- There exists $h : \mathbb{F}_2^n \to \{-1, +1\}$ with spectral norm at most m,
- Every adaptive algorithm for the $(0, 1/4)$-Testing Isomorphism of Boolean Functions problem with respect to h requires $\Omega(m^2)$ queries to the unknown function.

- Wimmer and Yoshida [ICALP 2013] proved an $\Omega(m)$ lower bound for the $(0, \tau)$-'Testing Isomorphism of Boolean Functions' problem, for $\tau = \frac{1}{m}$.

Contents

Testing Isomorphism of Boolean Functions

Let $f, g : \mathbb{F}_2^n \to \{-1, +1\}$ be Boolean functions.

- g is known. Given query access to f. For any $\epsilon \geq 0$ and $\tau > 0$, determine if
- $\exists$ a non-singular matrix $A \in \mathbb{F}_2^{n \times n}$ such that $\delta(f \circ A, g) \leq \epsilon$, or
- $\forall$ non-singular matrices $A \in \mathbb{F}_2^{n \times n}$, $\delta(f \circ A, g) \geq \epsilon + \tau$.

The algorithm executes in two major steps:

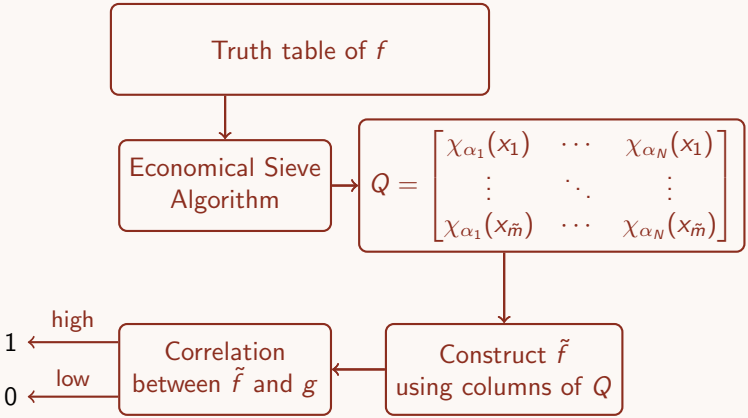
1. It learns the heavy Fourier co-efficients of the unknown function **implicitly**. **'Economical Sieve Algorithm'**.

The algorithm executes in two major steps:

1. It learns the heavy Fourier co-efficients of the unknown function **implicitly**. **'Economical Sieve Algorithm'**.
2. Use the Economical Sieve Algorithm algorithm to create a function $\tilde{f}$ that approximates f. Check the Isomorphism **off-line**.

The query complexity of the isomorphism testing algorithm solely depends on the Economical Sieve algorithm.

Algorithm Diagram



Economical Sieve

- Economical Seive is a query-efficient version of Implicit Seive by Wimmer and Yoshida.

Economical Sieve

- Economical Seive is a query-efficient version of Implicit Seive by Wimmer and Yoshida.

The **Implicit/Economical Sieve Algorithm** does the following:

- It will try to count and capture the **heavy Fourier** characters.

Economical Sieve

- Economical Seive is a query-efficient version of Implicit Seive by Wimmer and Yoshida.

The **Implicit/Economical Sieve Algorithm** does the following:

- It will try to count and capture the **heavy Fourier** characters.
- Input: a threshold θ, a set $M = \{x_1, \ldots, x_{\tilde{m}}\}$ of independently and uniformly chosen random points, and the unknown function f.

Economical Sieve

- Economical Seive is a query-efficient version of Implicit Seive by Wimmer and Yoshida.

The **Implicit/Economical Sieve Algorithm** does the following:

- It will try to count and capture the **heavy Fourier** characters.
- Input: a threshold θ, a set $M = \{x_1, \ldots, x_{\tilde{m}}\}$ of independently and uniformly chosen random points, and the unknown function f.
- It returns some values $\{\chi_{\alpha_j}(x_i), j \in [N]\}$, for all $x_i \in M$, where $S = \{\chi_{\alpha_1}, \ldots, \chi_{\alpha_N}\}$ is the set of *'heavy weight'* ($|\widehat{f}(\chi_{\alpha_i})| \geq \theta$) characters.
- So the set S is unknown.

- Fourier representation:

$$f(x) = \sum_{\alpha \in \mathbb{F}_2^n} \widehat{f}(\alpha)\chi_\alpha(x).$$

Random bucketing/Coset-Hashing

- $H =< \beta_1, \ldots, \beta_k >^{\perp}$, where $\beta_i \in \mathbb{F}_2^n$'s are chosen independently and uniformly.
- **Affine subspaces/cosets/buckets** $C(b) :=$ cosets of $\mathbb{F}_2^n/H$.
- Choose H in such a way that every Heavy Fourier coefficient end up in different buckets.

Let's first derive when f is sparse!

If f is sparse and we are re-using the samples following Wimmer and Yoshida [ICALP 2013], one can derive $O(\frac{m^4}{\tau^4})$ algorithm.

- A Boolean function f is *s-sparse* if it has s-many nonzero Fourier coefficients.
- The Fourier coefficients of f belong to different buckets with high probability on taking $O(s^2)$ many buckets.

If f is not sparse?

- In each bucket there is a heavy Fourier co-efficient + extra!

If f is not sparse?

- In each bucket there is a heavy Fourier co-efficient + extra!
- **Goal:** Sign of the bucket is determined by sign of the heavy Fourier co-efficient.
- We do this via **Local-list correction** (ℓ_1 concentration): Precisely we bound the following:

$$\mathcal{P}_C(z) - \mathcal{P}_C^*(z)$$

where $\mathcal{P}_C f(z) := \sum_{\beta \in C} \widehat{f}(\beta)\, \chi_\beta(z)$.

If f is not sparse?

- In each bucket there is a heavy Fourier co-efficient + extra!
- **Goal:** Sign of the bucket is determined by sign of the heavy Fourier co-efficient.
- We do this via **Local-list correction** (ℓ_1 concentration): Precisely we bound the following:

$$\mathcal{P}_C(z) - \mathcal{P}_C^*(z)$$

where $\mathcal{P}_C f(z) := \sum_{\beta \in C} \widehat{f}(\beta)\, \chi_\beta(z)$.

- How P_C relates sign?

$$P_C f(x) = \mathbb{E}_y[f(y)\chi_\alpha(y)\chi_\alpha(x)].$$

- Note that

$$P_C f(y_j) \cdot P_C f(y_j - x_i) = |\widehat{f}(\chi_\alpha)|^2\, \chi_\alpha(x_i).$$

- So $\operatorname{sign}(P_C f(y_j) \times P_C f(y_j - x_i)) = \chi_\alpha(x_i)$.

Key Differences from the Implicit Sieve

Implicit Sieve [Wimmer and Yoshida]

- Discards lightweight buckets in *two* rounds.
- First using wt_2, then using wt_4.
- Needed to estimate error for each bucket.

Economical Sieve

- Discards lightweight buckets in *one* round.
- Uses only wt_2 estimation.
- Uses ℓ_1-concentration.

Query complexity improvement:

- Reusing queries: $O\left(\frac{m^{24}}{\tau^{24}}\right) \to O\left(\frac{m^8}{\tau^8}\right)$.
- No wt_4 estimation: $O\left(\frac{m^8}{\tau^8}\right) \to O\left(\frac{m^4}{\tau^4}\right)$.

Contents

Conclusion!

- Our upper bound is $O(m^4)$ and lower bound is $\Omega(m^2)$. There's still a quadratic gap.
- Note that the lower bounds are non-tolerant. A lower-bound that is purely tolerant?

Thank You!