Testing Juntas and Function Isomorphism

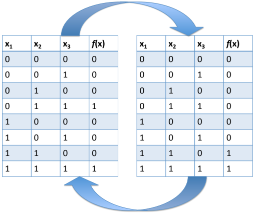
Eric Blais

Carnegie Mellon University
Pittsburgh, PA
eblais@cs.cmu.edu

January 8, 2009

Overview

Testing juntas



x_1	x_2	x_3	$f(x)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

x_1	x_2	x_3	$f(x)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Testing function isomorphism

Testing juntas

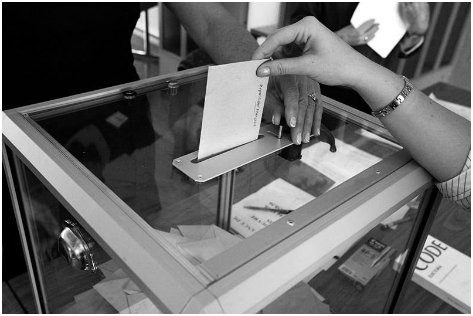
Definitions

The function $f : \{-1,1\}^n \to \{-1,1\}$ ...

... is a k-junta if it depends on at most k variables.

... is ϵ-far from being a k-junta if for every k-junta g, $\Pr_x[f(x) \neq g(x)] \geq \epsilon$.

Influence



The *influence* of a set S of variables in $f : \{-1,1\}^n \to \{-1,1\}$ is

$$\mathrm{Inf}_f(S) = \Pr_{x,y}[f(x) \neq f(x_{\bar{S}} y_S)].$$

Definitions

When the function $f : \{-1,1\}^n \to \{-1,1\}$ ...

...is a k-junta, then there is a set $J \in \binom{[n]}{k}$ such that $\mathrm{Inf}_f(\bar{J}) = 0$.

...is ϵ-far from being a k-junta, then for every set $J \in \binom{[n]}{k}$, $\mathrm{Inf}_f(\bar{J}) \geq \epsilon$.

Testing Juntas

An algorithm is an ϵ-tester for k-juntas if it accepts k-juntas and rejects functions that are ϵ-far from being k-juntas w.p. $\geq 2/3$.

vs.

The Question

What is the minimum number of queries required to ϵ-test k-juntas?

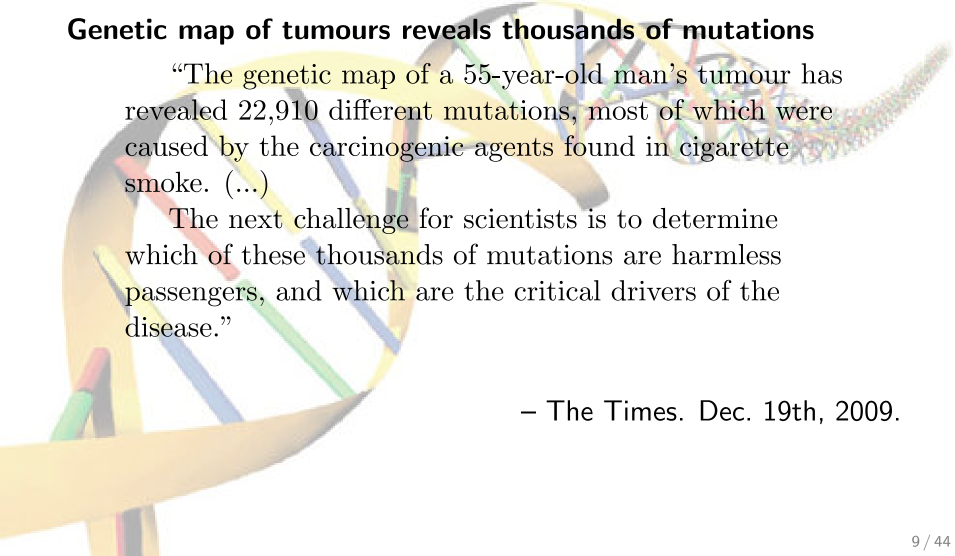
What is known

The minimum number $Q_{n,k,\epsilon}$ of queries required to ϵ-test k-juntas lies in the range

$$\Omega(k) \leq Q_{n,k,\epsilon} \leq \tilde{O}(k^2/\epsilon).$$

[Fischer, Kindler, Ron, Safra, Samorodnitsky '04]
[Chockler, Gutfreund '04]

Motivation - Machine Learning

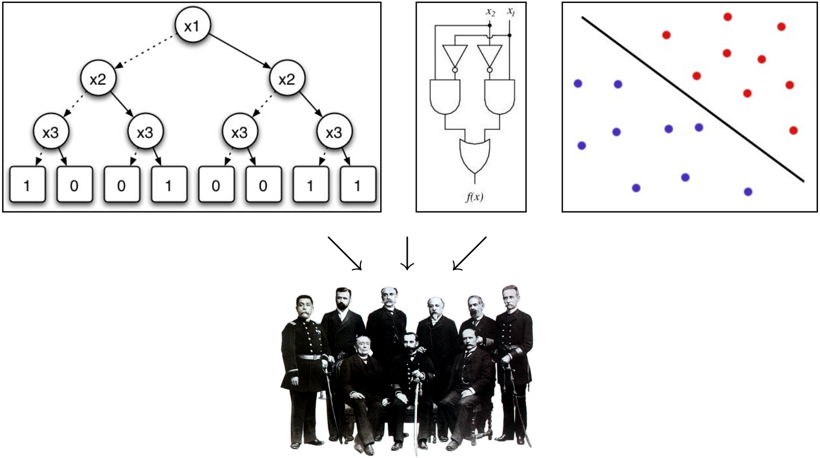


Genetic map of tumours reveals thousands of mutations

> "The genetic map of a 55-year-old man's tumour has revealed 22,910 different mutations, most of which were caused by the carcinogenic agents found in cigarette smoke. (...)
>
> The next challenge for scientists is to determine which of these thousands of mutations are harmless passengers, and which are the critical drivers of the disease."

– The Times. Dec. 19th, 2009.

Motivation - Property Testing



[Diakonikolas, Lee, Matulef, Onak, Rubinfeld, Servedio, Wan '07]
[Matulef, O'Donnell, Rubinfeld, Servedio '09]

Our results

Theorem 1 (RANDOM '08)

The minimum number $Q^{(\text{n.a.})}_{n,k,\epsilon}$ of queries required to ϵ-test k-juntas *non-adaptively* is at most

$$Q^{(\text{n.a.})}_{n,k,\epsilon} \leq \tilde{O}(k^{3/2}/\epsilon).$$

Theorem 2 (STOC '09)

The minimum number $Q_{n,k,\epsilon}$ of queries required to ϵ-test k-juntas is at most

$$Q_{n,k,\epsilon} \leq O(k \log k + k/\epsilon).$$

Basic building block

Let $\mathcal{I}$ be a partition of the variables in $[n]$.

IDENTIFY RELEVANT PART $(f, S, \mathcal{I})$

- Generate $x, y \in \{-1, 1\}^n$ uniformly at random conditioned on $x_{\overline{S}} = y_{\overline{S}}$ until we find a pair where $f(x) \neq f(y)$.
- Do a binary search on the hybrid inputs between x and y to find a part containing a relevant variable and output it.

Identifying a relevant part

$$
\begin{aligned}
f(101\ 110\ 011\ 010) &= 1\\
111\ 110\ 011\ 010&\\
111\ 010\ 011\ 010&\\
111\ 010\ 110\ 010&\\
f(111\ 010\ 110\ 001) &= 0
\end{aligned}
$$

Identifying a relevant part

$$f(101\ 110\ 011\ 010) = 1$$

$$111\ 110\ 011\ 010$$

$$f(111\ 010\ 011\ 010) = 1$$

$$111\ 010\ 110\ 010$$

$$f(111\ 010\ 110\ 001) = 0$$

Identifying a relevant part

$$\begin{aligned}
&f(101\ 110\ 011\ 010) = 1\\
&111\ 110\ 011\ 010\\
\longrightarrow\ &f(111\ 010\ 011\ 010) = 1\\
\longrightarrow\ &f(111\ 010\ 110\ 010) = 0\\
&f(111\ 010\ 110\ 001) = 0
\end{aligned}$$

Basic building block

Let $\mathcal{I}$ be a partition of the variables in $[n]$.

IDENTIFY RELEVANT PART $(f, S, \mathcal{I})$

- Generate $x, y \in \{-1, 1\}^n$ uniformly at random conditioned on $x_{\overline{S}} = y_{\overline{S}}$ until we find a pair where $f(x) \neq f(y)$.
- Do a binary search on the hybrid inputs between x and y to find a part containing a relevant variable and output it.

When $\mathrm{Inf}_f(S) \geq \epsilon$, the expected number of queries q made by a call to the algorithm is

$$\mathbf{E}[q] \leq 2/\epsilon + \log |\mathcal{I}|.$$

The Algorithm

```
Junta Tester $(f, q, s)$
- Randomly partition the variables into $s$ parts.
- Initialize $J \leftarrow \emptyset$.
- Repeat until at most $q$ queries have been made:
    - Let $j \leftarrow$ Identify Relevant Part$(f, \bar{J}, \mathcal{I})$.
    - Update $J \leftarrow J \cup \{j\}$.
    - If $|J| > k$, Reject.
- If $|J| \leq k$, Accept.
```

Claim: The algorithm is a valid ϵ-tester for k-juntas when $s = \mathrm{poly}(k)$ and $q = \Theta(k/\epsilon + k \log \theta) = \Theta(k/\epsilon + k \log k)$.

Main Technical Lemma

Lemma

When $f : \{-1,1\}^n \to \{-1,1\}$ is ϵ-far from being a k-junta and $\mathcal{I}$ is a random partition with $s = \mathrm{poly}(k)$ parts, then w.h.p. for every set J formed by taking the union of k parts, $\mathrm{Inf}_f(\bar{J}) \geq \epsilon/2$.

Tools for proving the Lemma:

$$\begin{aligned} \mathrm{Inf}_f(S) &= \tfrac{1}{2} \sum_{T : S \cap T \neq \emptyset} \hat{f}(T)^2. \\ \mathrm{Inf}_f^{\leq \tau}(S) &\overset{def}{=} \tfrac{1}{2} \sum_{T : S \cap T \neq \emptyset, |T| \leq \tau} \hat{f}(T)^2. \end{aligned}$$

Main Technical Lemma

Lemma

When $f : \{-1,1\}^n \to \{-1,1\}$ is ϵ-far from being a k-junta and $\mathcal{I}$ is a random partition with $s = \mathrm{poly}(k)$ parts, then w.h.p. for every set J formed by taking the union of k parts, $\mathrm{Inf}_f(\bar{J}) \geq \epsilon/2$.

Proof Sketch:

$$\begin{aligned} \mathrm{Inf}_f(\bar{J}) &= \tfrac{1}{2} - \tfrac{1}{2} \sum_{S \subseteq J} \hat{f}(S)^2 \\ &\geq \tfrac{1}{2} - \tfrac{1}{2} \sum_{S \subseteq J, |S| \leq \tau} \hat{f}(S)^2 - \epsilon/4 \quad \text{(w.h.p.)} \\ &= \tfrac{1}{2} - \mathrm{Inf}_f^{\leq \tau}(J) - \epsilon/4 \end{aligned}$$

To complete the proof, we want to show that $\mathrm{Inf}_f^{\leq}(J) \leq \frac{1}{2} - \frac{3}{4}\epsilon$.

Main Technical Lemma

Proof Sketch: (continued.)
We want to show that $\mathrm{Inf}_f^{\leq\tau}(J) \leq \frac{1}{2} - \frac{3}{4}\epsilon$. Let

$$J^* = \max_{S \subseteq J, |S| = k} \mathrm{Inf}_f^{\leq\tau}(S).$$

Then

$$\begin{aligned} \mathrm{Inf}_f^{\leq\tau}(J) &\leq \mathrm{Inf}_f^{\leq\tau}(J^*) + \sum_{i \in J \setminus J^*} \mathrm{Inf}_f^{\leq\tau}(i) \\ &\leq \frac{1}{2} - \epsilon + \sum_{i \in J \setminus J^*} \mathrm{Inf}_f^{\leq\tau}(i). \end{aligned}$$

To complete the proof, we note

- Only a small number of variables have non-negligible $\mathrm{Inf}_f^{\leq\tau}(i)$.
- W.h.p., those variables are split by the random partition.
- So w.h.p. $\sum_{i \in J \setminus J^*} \mathrm{Inf}_f^{\leq\tau}(i) \leq \epsilon/4$. □

Summary and open problems

The minimum number $Q_{n,k,\epsilon}$ of queries required to ϵ-test k-juntas lies in the range $\Omega(k) \le Q_{n,k,\epsilon} \le O(k \log k + k/\epsilon)$.

Open Problem 1

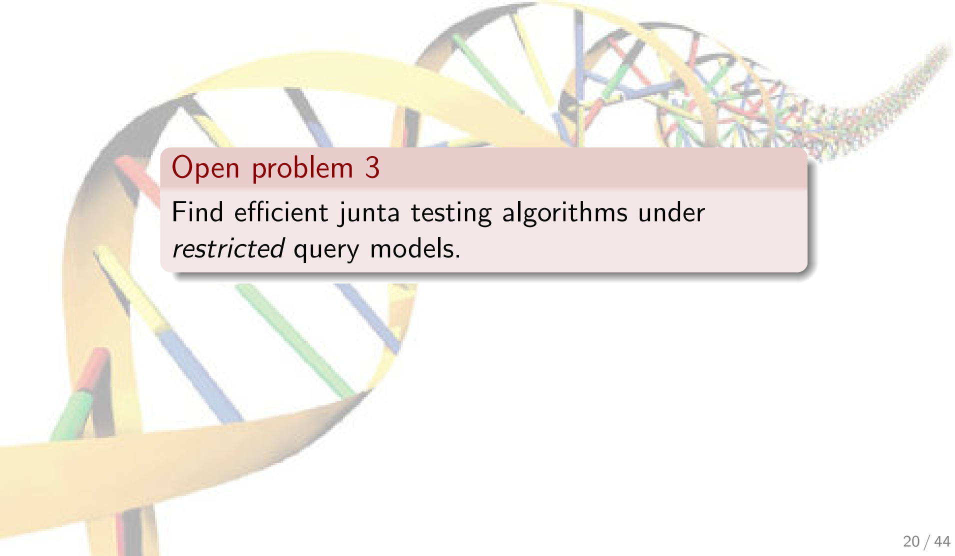
Close the gap on the value of $Q_{n,k,\epsilon}$.

The minimum number of queries required to ϵ-test k-juntas non-adaptively is $Q_{n,k,\epsilon}^{(n.a.)} \le \tilde{O}(k^{3/2}/\epsilon)$.

Open Problem 2

Improve the non-adaptive upper bound to $Q_{n,k,\epsilon}^{(n.a.)} \le \tilde{O}(k/\epsilon)$, or show that this is impossible.

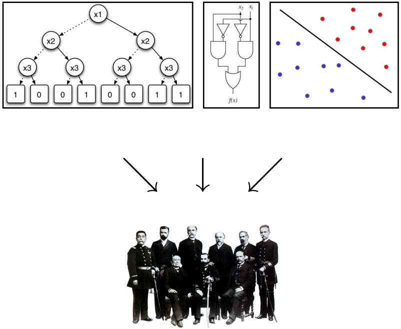
Motivation - Machine Learning



Open problem 3

Find efficient junta testing algorithms under *restricted* query models.

Open problems – Testing by implicit learning

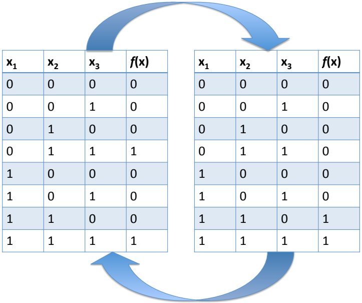


Open Problem 4

Use improved junta testing algorithm to improve the bounds for testing succinct DNFs, decision trees, circuits, etc.

Open Problem 4.a

Devise an efficient *tolerant* junta testing algorithm.



x_1	x_2	x_3	$f(x)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

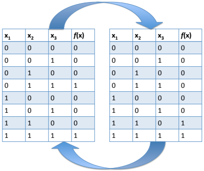
x_1	x_2	x_3	$f(x)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Testing function isomorphism

Joint work with Ryan O'Donnell.

Function isomorphism

Fix $g : \{-1,1\}^n \to \{-1,1\}$. The function $f : \{-1,1\}^n \to \{-1,1\}$...



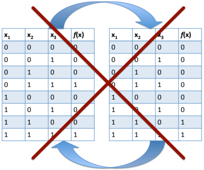
x_1	x_2	x_3	f(x)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

x_1	x_2	x_3	f(x)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

... is isomorphic to g if both functions are identical up to permutation of the input variables.

... is ϵ-far from being isomorphic to g if for every function h isomorphic to g,

$$\Pr_x[f(x) \neq h(x)] \geq \epsilon.$$

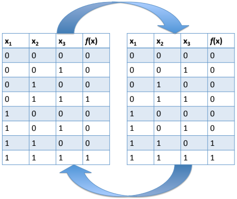


x_1	x_2	x_3	f(x)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

x_1	x_2	x_3	f(x)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

The problem

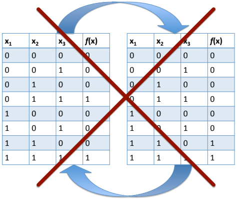
An algorithm is an ϵ-tester for g-isomorphism if it accepts functions isomorphic to g and rejects functions ϵ-far from being isomorphic to g with probability at least $2/3$.



x_1	x_2	x_3	$f(x)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

x_1	x_2	x_3	$f(x)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

vs.



x_1	x_2	x_3	$f(x)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

x_1	x_2	x_3	$f(x)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Question

For every function g, what is the minimum number of queries required to ϵ-test g-isomorphism?

Motivation – Characterization

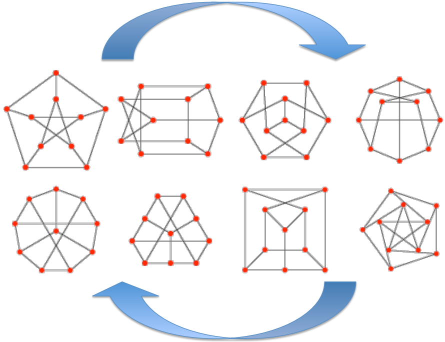
Overall Goal

- Characterize the query complexity required to test all function properties.

Sub-goal

- Characterize the query complexity required to test isomorphism against all boolean functions.

Motivation – Graph Isomorphism



[Fischer '04]

What is known

Function class	Lower bound	Upper bound
Fully symmetric functions		$O(1/\epsilon)$
Dictator functions		$O(1/\epsilon)$ *
And$_k$		$O(1/\epsilon)$ †
k-juntas		$\mathrm{poly}(k/\epsilon)$ §
Parity$_k$ ($k \leq o(\sqrt{n})$)	$\tilde{\Omega}(\sqrt{k})$ (n.a.) § $\Omega(\log k)$ §	

* [Bellare, Goldreich, Sudan '98]

† [Parnas, Ron, Samorodnitsky '03]

§ [Fischer, Kindler, Ron, Safra, Samorodnitsky '04]

What is known

Function class	Lower bound	Upper bound
Fully symmetric functions		$O(1/\epsilon)$
Dictator functions		$O(1/\epsilon)$ *
And$_k$		$O(1/\epsilon)$ †
k-juntas	**???**	$\text{poly}(k/\epsilon)$ §
Parity$_k$ ($k \leq o(\sqrt{n})$)	$\tilde{\Omega}(\sqrt{k})$ (n.a.) § $\Omega(\log k)$ §	

* [Bellare, Goldreich, Sudan '98]

† [Parnas, Ron, Samorodnitsky '03]

§ [Fischer, Kindler, Ron, Safra, Samorodnitsky '04]

Our first result

Theorem 1

Fix $0 < \ell < k < n$. Let $g : \{0,1\}^n \to \{0,1\}$ be a k-junta and be ϵ-far from being a $(k-\ell)$-junta. Then ϵ-testing g-isomorphism non-adaptively requires

$$\Omega\left(\log \frac{\min(k, n-k)}{\ell^2}\right)$$

queries.

Proof Idea

Theorem 1 (special case)

Let $g : \{0,1\}^n \to \{0,1\}$ be a k-junta that is ϵ-far from being a $(k-1)$-junta, for some $k \leq n/2$. Then any non-adaptive algorithm for ϵ-testing g-isomorphism must make $\Omega(\log k)$ queries.

Proof Sketch:

Use Yao's Minimax Method. Show that no deterministic algorithm making $q = o(\log k)$ queries can distinguish between the functions drawn from $\mathcal{F}_{yes}$ or $\mathcal{F}_{no}$, where

- $\mathcal{F}_{yes}$ is the uniform distribution over functions isomorphic to g,
- $\mathcal{F}_{no}$ is a distribution over $(k-1)$-juntas (to be defined later).

Notation

Definition (g_{core})

Let $g : \{0,1\}^n \to \{0,1\}$ be a k-junta with relevant variables $i_1, \ldots, i_k \in [n]$. Then we define $g_{core} : \{0,1\}^k \to \{0,1\}$ to be the function defined such that for every $x \in \{0,1\}^n$,

$$g(x_1, \ldots, x_n) = g_{core}(x_{i_1}, \ldots, x_{i_k}).$$

Query matrix

A deterministic non-adaptive testing algorithm must fix its queries in advance:

$$\begin{array}{rcl} f(x_{1,1},\ x_{1,2},\ \ldots,\ x_{1,n}) & = & ? \\ f(x_{2,1},\ x_{2,2},\ \ldots,\ x_{2,n}) & = & ? \\ & \vdots & \\ f(x_{q,1},\ x_{q,2},\ \ldots,\ x_{q,n}) & = & ? \end{array}$$

Those queries naturally define a query matrix

$$Q = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{q,1} & x_{q,2} & \cdots & x_{q,n} \end{pmatrix}.$$

Process for $\mathcal{R}_{yes}$

$$Q = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Process for $\mathcal{R}_{yes}$

$$Q = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \xrightarrow{\text{Select}} \overset{S_{yes}}{\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}}$$

Process for $\mathcal{R}_{yes}$

$$Q = \begin{pmatrix} {\color{gray}1} & {\color{gray}1} & 1 & 0 & {\color{gray}0} & 1 \\ {\color{gray}0} & {\color{gray}1} & 0 & 1 & {\color{gray}1} & 1 \\ {\color{gray}1} & {\color{gray}0} & 1 & 1 & {\color{gray}0} & 1 \\ {\color{gray}0} & {\color{gray}0} & 1 & 0 & {\color{gray}1} & 1 \end{pmatrix} \xrightarrow{\text{Select}} \overset{S_{yes}}{\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}} \xrightarrow{\text{Permute}} \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

Process for $\mathcal{R}_{yes}$

$$\begin{tikzcd}
Q = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \arrow[r, "\text{Select}"] \arrow[dr] & \overset{S_{yes}}{\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}} \arrow[r, "\text{Permute}"] & \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \arrow[dl] \\
& \text{RETURN:} \quad \begin{aligned} f(1,1,1,0,0,1) &= g_{core}(0,1,1) \\ f(0,1,0,1,1,1) &= g_{core}(1,0,1) \\ f(1,0,1,1,0,1) &= g_{core}(1,1,1) \\ f(0,0,1,0,1,1) &= g_{core}(0,1,1) \end{aligned} &
\end{tikzcd}$$

Process for $\mathcal{R}_{no}$

$$Q = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Process for $\mathcal{R}_{no}$

$$\begin{tikzcd}
Q = \begin{array}{cccccc} & & & & & \times 2 \\ \multicolumn{6}{c}{\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}} \end{array} \arrow[r, "\text{Select}"] \arrow[dr] & \overset{S_{no}}{\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}} \arrow[r, "\text{Permute}"] & \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \arrow[dl] \\
 & \textsc{Return:} \quad \begin{aligned} f(1,1,1,0,0,1) &= g_{core}(1,1,1) \\ f(0,1,0,1,1,1) &= g_{core}(1,0,1) \\ f(1,0,1,1,0,1) &= g_{core}(1,1,1) \\ f(0,0,1,0,1,1) &= g_{core}(1,1,1) \end{aligned} &
\end{tikzcd}$$

Analysis

Key idea:

- When $2^q \ll n$, there are many copies of each *type* of column.
- This makes it hard to distinguish between $\mathcal{R}_{yes}$ and $\mathcal{R}_{no}$.

Rest of the analysis:

- Bounding $d_{TV}(\mathcal{R}_{yes}, \mathcal{R}_{no})$ reduces to bounding the distance between distributions of S_{yes} and S_{no}.
- In term, bounding distributions of S_{yes} and S_{no} reduces to bounding the distance between similar Hypergeometric distributions.

Extensions

Our lower bound applies to all k-juntas that are ϵ-far from being $(k-\ell)$-juntas for some $\ell \ll \sqrt{k}$.

Question: Are there interesting classes of "honest" k-juntas that do not fit this definition?

Answer: Yes! Most notably, Maj_k functions.

Second result

Theorem 2 (Simplified)

There exists an $\epsilon > 0$ s.t. for any $1/\epsilon \leq k \leq \frac{1}{2}n$, any non-adaptive algorithm for ϵ-testing Maj_k-isomorphism must make at least $\Omega(k^{1/12})$ queries.

Proof sketch:

Again apply Yao's Minimax Method, with

- $\mathcal{F}_{yes}$ as the uniform distribution over Maj_k functions, and
- $\mathcal{F}_{no}$ as a uniform distribution over a class of threshold functions;

$\mathcal{F}_{yes}$ and $\mathcal{F}_{no}$

To generate $f_{yes} \sim \mathcal{F}_{yes}$, let

$$f_{yes} = \operatorname{sgn}\left(x_{i_1} + \cdots + x_{i_k}\right),$$

and to generate $f_{no} \sim \mathcal{F}_{no}$, let

$$f_{no} = \operatorname{sgn}\left(\sum_{j \in [k/3]} \tfrac{1}{2}x_{i_{4j-3}} + \tfrac{1}{2}x_{i_{4j-2}} + \tfrac{1}{2}x_{i_{4j-1}} + \tfrac{3}{2}x_{i_{4j}}\right),$$

where in both cases the indices $i_1, i_2, \ldots$ are chosen uniformly at random without replacement from $[n]$.

$\mathcal{R}_{yes}$ and $\mathcal{R}_{no}$

Given a query matrix Q, to generate a response in the yes case, we return the orthant that contains the vector

$$V_{yes} = v_1 + \cdots + v_k.$$

To generate a response in the no case we return the orthant that contains

$$V_{no} = \sum_j \tfrac{1}{2} v_{4j-3} + \tfrac{1}{2} v_{4j-2} + \tfrac{1}{2} v_{4j-1} + \tfrac{3}{2} v_{4j},$$

where in both cases $v_1, \dots, v_k \in \{-1, 1\}^q$ are drawn uniformly at random without replacement from the columns of Q.

Analysis

Uses a multidimensional invariance principle to show:

Lemma

For any union of orthants $\mathcal{O}$ and any query matrix Q with q queries,

$$|\Pr[V_{yes} \in \mathcal{O} - \Pr[V_{no} \in \mathcal{O}]| \leq \frac{q^{3/2}}{k^{1/8}}.$$

Open problems

Main problem: provide tight lower and upper bounds for the query complexity of testing g-isomorphism for *all* boolean functions g.

Open Problem 1

Let g be a k-junta that is ϵ-far from being a k'-junta for *any* $k' < k$. Show that the query complexity of testing g-isomorphism must depend on k'.

Open problems

Main problem: provide tight lower and upper bounds for the query complexity of testing g-isomorphism for *all* boolean functions g.

Open Problem 1

Let g be a k-junta that is ϵ-far from being a k'-junta for *any* $k' < k$. Show that the query complexity of testing g-isomorphism must depend on k'.

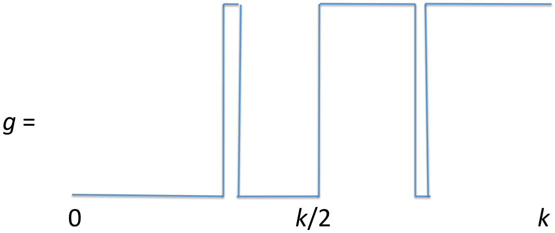
Open Problem 2

Show that the statement in Open Problem 1 holds for *symmetric* functions.

Open problems

Open Problem 3

Let g be a symmetric function that is $o(1)$-close to a Maj_k function. Show that the query complexity for testing g-isomorphism depends on k.



Open problems

Open Problem 4

Establish lower bounds for the query complexity of testing isomorphism to linear threshold functions.

Thank you!