Isomorphism Testing for Boolean Functions Computable by Small-Depth Circuits

V. Arvind and Yadu Vasudev

The Institute of Mathematical Sciences
Chennai, India

9 March
LATA 2012
A Coruña, Spain

The Problem

Given two boolean functions f and g over n variables $x_1, \ldots, x_n$, check if there exists a permutation $\pi : [n] \to [n]$ such that for all $x_1, \ldots, x_n \in \{0, 1\}$, $f(x_1, \ldots, x_n) = g(x_{\pi(1)}, \ldots, x_{\pi(n)})$.

What is known

- When f and g are represented as truth tables, the problem is equivalent to hypergraph isomorphism.
- Fastest known algorithm takes $2^{O(n)}$ time [Luks '99].

What is known

- Functions given as boolean circuits computing them.
- co-NP hard when the functions are given as DNF formulas. It is in Σ_2^p but is not complete for that class unless PH collapses to Σ_3^p [Agrawal and Thierauf '00].

What can be done?

What can be done?

- A notion of approximate isomorphism?

What can be done?

- A notion of approximate isomorphism?
- When are two boolean functions f and g "close" to each other?

What can be done?

- A notion of approximate isomorphism?
- When are two boolean functions f and g "close" to each other?

For a significant fraction of $x \in \{0,1\}^n$, $f(x) = g(x)$

The approximation version of the problem

$$\exists \pi : [n] \to [n],\ \forall x_1, \ldots, x_n,\ f(x_1, \ldots, x_n) = g(x_{\pi(1)}, \ldots, x_{\pi(n)})$$

Given two boolean functions f and g which are isomorphic , compute a permutation $\sigma : [n] \to [n]$ such that f and g^σ are ε-close .

$$\Pr[f(x) \neq g^\sigma(x)] \leq \varepsilon$$

$$g^\sigma(x_1, \ldots, x_n) = g(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$$

The approximation version of the problem

$$\exists \pi : [n] \to [n],\ \forall x_1, \ldots, x_n,\ f(x_1, \ldots, x_n) = g(x_{\pi(1)}, \ldots, x_{\pi(n)})$$

How?

Given two boolean functions f and g which are isomorphic , compute a permutation $\sigma : [n] \to [n]$ such that f and g^σ are ε-close .

$$\Pr[f(x) \neq g^\sigma(x)] \leq \varepsilon$$

$$g^\sigma(x_1, \ldots, x_n) = g(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$$

Our Result

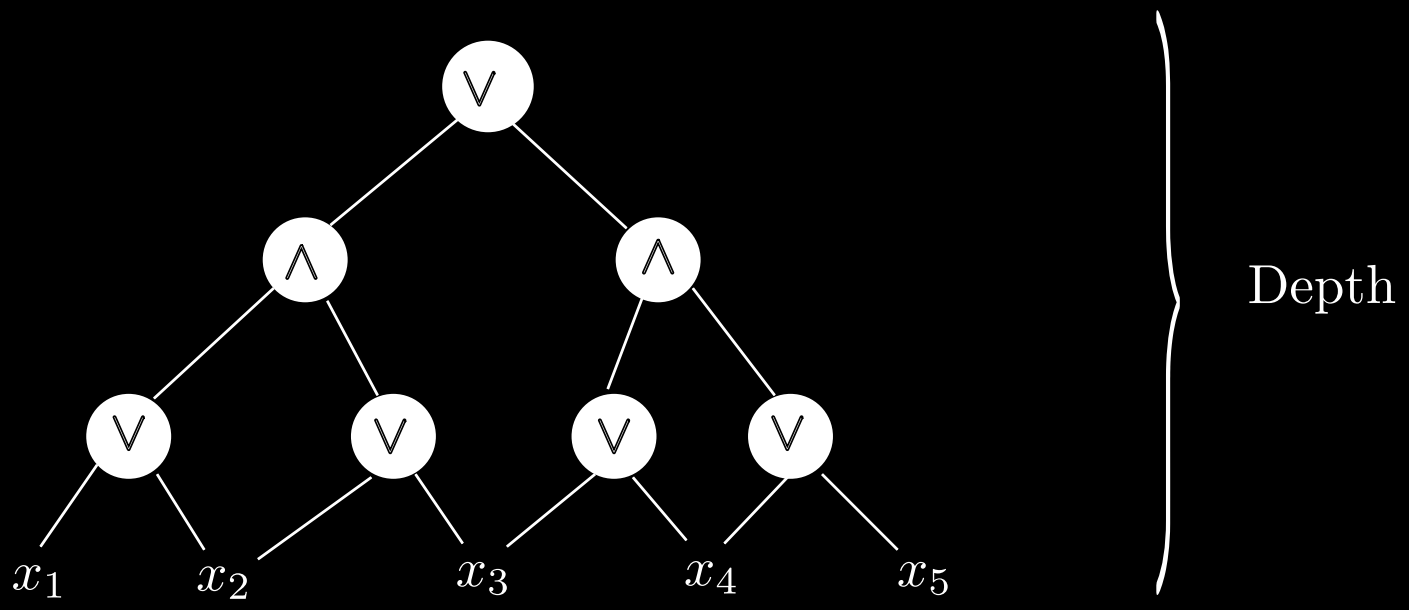
Theorem

Given two boolean functions $f : \{0,1\}^n \to \{0,1\}$ *and* $g : \{0,1\}^n \to \{0,1\}$ *which are isomorphic, represented as circuits of small size and depth, there is a randomized algorithm running in time* $2^{\widetilde{O}(\sqrt{n})}$ *which computes a permutation* $\sigma : [n] \to [n]$ *such that* f *and* g^σ *are* $\frac{1}{100}$*-close.*

The idea

- Approximate f and g using low degree polynomials which can be computed efficiently.
- Compute an exact isomorphism between the low degree polynomials and extend it to a permutation for f and g.

Boolean Circuits



Size = number of nodes in the circuit.

Fourier Analysis of Boolean Functions

$$\mathcal{F} = \{f : \{-1,1\}^n \to \mathbb{R}\}.$$

Fourier Analysis of Boolean Functions

$$\mathcal{F} = \{f : \{-1, 1\}^n \to \mathbb{R}\}.$$

Basis

$$f_i(x) = \begin{cases} 1 & \text{if } x = i \\ 0 & \text{otherwise} \end{cases}$$

for $i \in \{0, \ldots, 2^n\}$.

An inner product

$$\text{For } f, g \in \mathcal{F}, \langle f, g \rangle = \mathbb{E}_x \left[f(x) g(x) \right]$$
$$= \frac{1}{2^n} \sum_{x \in \{-1,1\}^n} f(x) g(x)$$

Fourier Basis for Boolean Functions

$\chi_S(x_1, \ldots, x_n) = \prod_{i \in S} x_i$ for all subsets $S \subseteq [n]$

Fourier Basis for Boolean Functions

$\chi_S(x_1, \ldots, x_n) = \prod_{i \in S} x_i$ for all subsets $S \subseteq [n]$

- $\langle \chi_S, \chi_T \rangle = \mathbb{E}_x [\chi_{S \triangle T}(x)]$.
- $\mathbb{E}_x[\chi_S(x)] = 0$, $S \neq \phi$.
- $\mathbb{E}_x[\chi_\phi(x)] = 1$.

Fourier Basis for Boolean Functions

$\chi_S(x_1, \ldots, x_n) = \prod_{i \in S} x_i$ for all subsets $S \subseteq [n]$

- $\langle \chi_S, \chi_T \rangle = \mathbb{E}_x [\chi_{S \triangle T}(x)]$.
- $\mathbb{E}_x[\chi_S(x)] = 0$, $S \neq \phi$.
- $\mathbb{E}_x[\chi_\phi(x)] = 1$.

$f = \sum_{S \subseteq [n]} \widehat{f}(S) \chi_S$, $\widehat{f}(S) = \langle f, \chi_S \rangle$

Representing boolean functions with polynomials

Representing boolean functions with polynomials

- Any boolean function can be expressed as a real polynomial of degree n.

Representing boolean functions with polynomials

- Any boolean function can be expressed as a real polynomial of degree n.
- $\mathrm{AND}(x_1, \cdots, x_n) = 1 - 2 \prod_{i \in [n]} \left(\frac{1 - x_i}{2} \right)$.
- $\mathrm{OR}(x_1, \cdots, x_n) = 2 \prod_{i \in [n]} \left(\frac{1 + x_i}{2} \right) - 1$.

Low degree approximations of boolean functions

How well can a degree $d \ll n$ real polynomial $p(x_1, \cdots, x_n)$ approximate a boolean function $f(x_1, \cdots, x_n)$?

Low degree approximations of boolean functions

How well can a degree $d \ll n$ real polynomial $p(x_1, \cdots, x_n)$ approximate a boolean function $f(x_1, \cdots, x_n)$?

$\mathbb{E}_{x \in \{-1,1\}^n} \left[(f(x) - p(x))^2 \right] \leq \varepsilon.$

Approximating small size, depth circuits with real polynomials

Theorem (Linial, Mansour and Nisan '93)

Let $f : \{-1,1\}^n \to \{-1,1\}$ be computed by circuits of size s and depth d. Then for all $t > 0$,

$$\sum_{S \subseteq [n],\ |S| \geq t} \widehat{f}(S)^2 \leq 2s2^{-t^{1/d}/20}.$$

Approximating small size, depth circuits with real polynomials

Theorem (Linial, Mansour and Nisan '93)

Let $f : \{-1,1\}^n \to \{-1,1\}$ be computed by circuits of size s and depth d. Then for all $t > 0$,

$$\sum_{S \subseteq [n],\ |S| \geq t} \widehat{f}(S)^2 \leq 2s2^{-t^{1/d}/20}.$$

$$\widetilde{f} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{f}(S)\chi_S = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{f}(S) \prod_{i \in S} x_i$$

Approximating small size, depth circuits with real polynomials

Theorem (Linial, Mansour and Nisan '93)

Let $f : \{-1,1\}^n \to \{-1,1\}$ be computed by circuits of size s and depth d. Then for all $t > 0$,

$$\sum_{S \subseteq [n],\ |S| \geq t} \widehat{f}(S)^2 \leq 2s2^{-t^{1/d}/20}.$$

$$\widetilde{f} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{f}(S)\chi_S = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{f}(S) \prod_{i \in S} x_i$$

$$\mathbb{E}\left[(f(x) - \widetilde{f}(x))^2\right] \leq 2s2^{-t^{1/d}/20}$$

Approximating small size, depth circuits with real polynomials

Theorem (Linial, Mansour and Nisan '93)

Let $f : \{-1,1\}^n \to \{-1,1\}$ be computed by circuits of size s and depth d. Then for all $t > 0$,

$$\sum_{S \subseteq [n],\ |S| \geq t} \widehat{f}(S)^2 \leq 2s2^{-t^{1/d}/20}.$$

$$\widetilde{f} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{f}(S)\chi_S = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{f}(S) \prod_{i \in S} x_i$$

$$\|f - \widetilde{f}\|_2^2 \leq 2s2^{-t^{1/d}/20}$$

Two notions of “closeness”

Two notions of "closeness"

Proposition

Let f and g be boolean functions, then

$$\|f - g\|_2^2 = 4 \Pr\left[f(x) \neq g(x)\right].$$

A simple observation

$$\widetilde{f} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{f}(S)\chi_S, \quad \widetilde{g} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{g}(S)\chi_S$$

A simple observation

$$\widetilde{f} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{f}(S)\chi_S, \quad \widetilde{g} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{g}(S)\chi_S$$

$$\widetilde{f} \xrightarrow{\pi} \widetilde{g}$$

A simple observation

$$\widetilde{f} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{f}(S)\chi_S, \quad \widetilde{g} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{g}(S)\chi_S$$

$$\widetilde{f} \xrightarrow{\pi} \widetilde{g}$$

$$\begin{aligned} \|f - g^{\pi}\|_2 &= \|f - \widetilde{f} + \widetilde{f} - \widetilde{g^{\pi}} + \widetilde{g^{\pi}} - g^{\pi}\|_2 \\ &\leq \|f - \widetilde{f}\|_2 + \|\widetilde{f} - \widetilde{g^{\pi}}\|_2 + \|\widetilde{g^{\pi}} - g^{\pi}\|_2 \end{aligned}$$

A simple observation

$$\widetilde{f} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{f}(S)\chi_S, \quad \widetilde{g} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{g}(S)\chi_S$$

$$\widetilde{f} \xrightarrow{\pi} \widetilde{g}$$

$$\begin{aligned} \|f - g^{\pi}\|_2 &= \|f - \widetilde{f} + \widetilde{f} - \widetilde{g^{\pi}} + \widetilde{g^{\pi}} - g^{\pi}\|_2 \\ &\leq \|f - \widetilde{f}\|_2 + 0 + \|\widetilde{g^{\pi}} - g^{\pi}\|_2 \\ &\leq 4s2^{-t^{1/d}/20} \end{aligned}$$

Computing the Fourier coefficients

- Exact computation of $\widehat{f}(S)$?

$$\widehat{f}(S) = \frac{1}{2^n} \sum_{x \in \{-1,1\}^n} f(x)\chi_S(x).$$

Estimating the Fourier coefficients [Linial, Mansour and Nisan '93]

Estimating the Fourier coefficients [Linial, Mansour and Nisan '93]

- For $i \leftarrow 1$ to m, pick x_i uniformly at random from $\{-1, 1\}^n$.
- Compute $\alpha_f(S) = \frac{1}{m} \sum f(x_i)\chi_S(x_i)$.

Estimating the Fourier coefficients [Linial, Mansour and Nisan '93]

- For $i \leftarrow 1$ to m, pick x_i uniformly at random from $\{-1,1\}^n$.
- Compute $\alpha_f(S) = \frac{1}{m}\sum f(x_i)\chi_S(x_i)$.

Theorem (Chernoff-Hoeffding bounds)

$$\Pr\left[|\widehat{f}(S) - \alpha_f(S)| \geq \lambda\right] \leq 2e^{-\lambda^2 m/2}$$

Estimating the Fourier coefficients [Linial, Mansour and Nisan '93]

Theorem (Chernoff-Hoeffding bounds)

$$\Pr\left[|\widehat{f}(S) - \alpha_f(S)| \geq \lambda\right] \leq 2e^{-\lambda^2 m/2}$$

Fix $\lambda = \frac{1}{2^{\lceil \ell/2 \rceil - 1}}$, $m = tn \log n 2^{\ell}$.

Estimating the Fourier coefficients [Linial, Mansour and Nisan '93]

Theorem (Chernoff-Hoeffding bounds)

$$\Pr\left[|\widehat{f}(S) - \alpha_f(S)| \geq \lambda\right] \leq 2e^{-\lambda^2 m/2}$$

Fix $\lambda = \frac{1}{2^{\lfloor \ell/2 \rfloor - 1}}$, $m = tn \log n 2^{\ell}$.

$$|\widehat{f}(S) - \alpha_f(S)| \leq \frac{1}{2^{\lfloor \ell/2 \rfloor - 1}} \text{ with high probability.}$$

Estimating the Fourier coefficients [Linial, Mansour and Nisan '93]

Theorem (Chernoff-Hoeffding bounds)

$$\Pr\left[|\widehat{f}(S) - \alpha_f(S)| \geq \lambda\right] \leq 2e^{-\lambda^2 m/2}$$

Fix $\lambda = \frac{1}{2^{\lfloor \ell/2 \rfloor - 1}}$, $m = tn \log n 2^{\ell}$.

$$|\widehat{f}(S) - \alpha_f(S)| \leq \frac{1}{2^{\lfloor \ell/2 \rfloor - 1}} \text{ with high probability.}$$

$\widehat{f_\ell}(S)$ is the truncation of $\alpha_f(S)$ to $\lfloor \frac{\ell}{2} \rfloor - 1$ bits of precision.

Low degree polynomial approximations

$$\widetilde{f_\ell} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{f_\ell}(S)\chi_S, \quad \widetilde{g_\ell} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{g_\ell}(S)\chi_S$$

Low degree polynomial approximations

$$\widetilde{f_\ell} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{f_\ell}(S)\chi_S, \quad \widetilde{g_\ell} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{g_\ell}(S)\chi_S$$

Lemma

There is a randomized algorithm which computes $\widetilde{f_\ell}$ and $\widetilde{g_\ell}$ in time $O(n^t \cdot 2^\ell)$ given access to the circuits computing f and g.

Low degree polynomial approximations

$$\widetilde{f_\ell} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{f_\ell}(S)\chi_S, \quad \widetilde{g_\ell} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{g_\ell}(S)\chi_S$$

Lemma

There is a randomized algorithm which computes $\widetilde{f_\ell}$ and $\widetilde{g_\ell}$ in time $O(n^t \cdot 2^\ell)$ given access to the circuits computing f and g.

$$\|\widetilde{f} - \widetilde{f_\ell}\|_2^2 = \sum_{S \subseteq [n],\ |S| \leq t} \left(\widehat{f}(S) - \widehat{f_\ell}(S)\right)^2 \leq \frac{4n^t}{2^{\ell-1}}$$

A simple observation (2)

$$\widetilde{f_\ell} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{f_\ell}(S)\chi_S, \quad \widetilde{g_\ell} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{g_\ell}(S)\chi_S$$

A simple observation (2)

$$\widetilde{f_\ell} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{f_\ell}(S)\chi_S, \qquad \widetilde{g_\ell} = \sum_{S \subseteq [n],\ |S| \leq t} \widehat{g_\ell}(S)\chi_S$$

$$\widetilde{f_\ell} \xrightarrow{\pi} \widetilde{g_\ell}$$

A simple observation (2)

$$\widetilde{f}_\ell = \sum_{S\subseteq[n],\ |S|\leq t} \widehat{f}_\ell(S)\chi_S, \quad \widetilde{g}_\ell = \sum_{S\subseteq[n],\ |S|\leq t} \widehat{g}_\ell(S)\chi_S$$

$$\widetilde{f}_\ell \xrightarrow{\pi} \widetilde{g}_\ell$$

$$\begin{aligned}\|f - g^\pi\|_2 &= \|f - \widetilde{f} + \widetilde{f} - \widetilde{f}_\ell + \widetilde{f}_\ell - \widetilde{g_\ell^\pi} + \widetilde{g_\ell^\pi} - \widetilde{g^\pi} + \widetilde{g^\pi} - g^\pi\|_2 \\ &\leq \|f - \widetilde{f}\|_2 + \|\widetilde{f} - \widetilde{f}_\ell\|_2 + \|\widetilde{f}_\ell - \widetilde{g_\ell^\pi}\|_2 + \|\widetilde{g_\ell^\pi} - \widetilde{g^\pi}\|_2 + \\ &\quad \|\widetilde{g^\pi} - g^\pi\|_2\end{aligned}$$

A simple observation (2)

$$\widetilde{f_\ell} = \sum_{S\subseteq[n],\ |S|\le t} \widehat{f_\ell}(S)\chi_S, \quad \widetilde{g_\ell} = \sum_{S\subseteq[n],\ |S|\le t} \widehat{g_\ell}(S)\chi_S$$

$$\widetilde{f_\ell} \xrightarrow{\pi} \widetilde{g_\ell}$$

$$\begin{aligned}
\|f - g^\pi\|_2 &= \|f - \widetilde{f} + \widetilde{f} - \widetilde{f_\ell} + \widetilde{f_\ell} - \widetilde{g_\ell^\pi} + \widetilde{g_\ell^\pi} - \widetilde{g^\pi} + \widetilde{g^\pi} - g^\pi\|_2 \\
&\le \|f - \widetilde{f}\|_2 + \|\widetilde{f} - \widetilde{f_\ell}\|_2 + \boxed{0} + \|\widetilde{g_\ell^\pi} - \widetilde{g^\pi}\|_2 + \|\widetilde{g^\pi} - g^\pi\|_2 \\
&\le 4s2^{-t^{1/d}/20} + \frac{4n^{t/2}}{2^{(\ell-1)/2}}
\end{aligned}$$

An approximate isomorphism

$$\widetilde{f_\ell} = \sum_{S \subseteq [n],\ |S| \le t} \widehat{f_\ell}(S)\chi_S, \qquad \widetilde{g_\ell} = \sum_{S \subseteq [n],\ |S| \le t} \widehat{g_\ell}(S)\chi_S$$

$$\widetilde{f_\ell} \xrightarrow{\pi} \widetilde{g_\ell}$$

$$\|f - g^\pi\|_2^2 \le \left(4s2^{-t^{1/d}/20} + \frac{4n^{t/2}}{2^{(\ell-1)/2}}\right)^2$$

An approximate isomorphism

$$\widetilde{f_\ell} = \sum_{S \subseteq [n],\ |S| \le t} \widehat{f_\ell}(S)\chi_S, \qquad \widetilde{g_\ell} = \sum_{S \subseteq [n],\ |S| \le t} \widehat{g_\ell}(S)\chi_S$$

$$\widetilde{f_\ell} \xrightarrow{\pi} \widetilde{g_\ell}$$

$$4 \Pr\left[f(x) \neq g^\pi(x)\right] \le \left(4s2^{-t^{1/d}/20} + \frac{4n^{t/2}}{2^{(\ell-1)/2}}\right)^2$$

An approximate isomorphism

$$\widetilde{f_\ell} = \sum_{S \subseteq [n],\ |S| \le t} \widehat{f_\ell}(S)\chi_S, \qquad \widetilde{g_\ell} = \sum_{S \subseteq [n],\ |S| \le t} \widehat{g_\ell}(S)\chi_S$$

$$\widetilde{f_\ell} \xrightarrow{\pi} \widetilde{g_\ell}$$

$$\Pr\left[f(x) \neq g^\pi(x)\right] \le 2^{-(\log n)^{O(1)}},$$

when $\ell = (\log n + \log s)^{O(d)}$, $t = (\log n + \log s)^{O(d)}$.

Exact isomorphism of low-degree polynomials

Given

$$\widetilde{f_\ell} = \sum_{S \subseteq [n],\ |S| \le t} \widehat{f_\ell}(S)\chi_S, \quad \widetilde{g_\ell} = \sum_{S \subseteq [n],\ |S| \le t} \widehat{g_\ell}(S)\chi_S,$$

compute a permutation $\pi : [n] \to [n]$ such that $\forall x,\ \widetilde{f_\ell}(x) = \widetilde{g_\ell^\pi}(x)$.

Exact isomorphism of low-degree polynomials

Given

$$\widetilde{f_\ell} = \sum_{S \subseteq [n],\ |S| \le t} \widehat{f_\ell}(S)\chi_S, \quad \widetilde{g_\ell} = \sum_{S \subseteq [n],\ |S| \le t} \widehat{g_\ell}(S)\chi_S,$$

compute a permutation $\pi : [n] \to [n]$ such that $\forall x,\ \widetilde{f_\ell}(x) = \widetilde{g_\ell^\pi}(x)$.

The coefficients $\widehat{f_\ell}(S)$ and $\widehat{g_\ell}(S)$ are represented by $\lfloor \frac{\ell}{2} \rfloor - 1$ bit strings.

Reducing to hypergraph isomorphism

Reducing to hypergraph isomorphism

Define weighted hypergraphs G_f and G_g :

Reducing to hypergraph isomorphism

Define weighted hypergraphs G_f and G_g :

$$V = [n]$$
$$E_f, E_g = S \subseteq [n],\ |S| \leq t$$

Reducing to hypergraph isomorphism

Define weighted hypergraphs G_f and G_g :

$$V = [n]$$
$$E_f, E_g = S \subseteq [n],\ |S| \leq t$$

$$w_f(S) = \begin{cases} \widehat{f_\ell}(S) & \forall S \subseteq [n], |S| \leq t \\ 0 & \text{otherwise} \end{cases}$$

Reducing to hypergraph isomorphism

Define weighted hypergraphs G_f and G_g :

$$V = [n]$$
$$E_f, E_g = S \subseteq [n],\ |S| \leq t$$

$$w_g(S) = \begin{cases} \widehat{g_\ell}(S) & \forall S \subseteq [n], |S| \leq t \\ 0 & \text{otherwise} \end{cases}$$

Reducing to hypergraph isomorphism

Define weighted hypergraphs G_f and G_g :

$$V = [n]$$
$$E_f, E_g = S \subseteq [n],\ |S| \leq t$$

$$w_g(S) = \begin{cases} \widehat{g}_\ell(S) & \forall S \subseteq [n], |S| \leq t \\ 0 & \text{otherwise} \end{cases}$$

Lemma

Suppose there exists an isomorphism π from G_f to G_g such that for all edges $S \in E_f$, $w_f(S) = w_g(\pi(S))$. Then π is an isomorphism from $\widetilde{f}_\ell$ to $\widetilde{g}_\ell$.

Reducing to hypergraph isomorphism

Define hypergraphs $\widetilde{G_f}$ and $\widetilde{G_g}$:

$$V = [n] \cup \{v_1, \ldots, v_r\}, r = \lfloor \frac{\ell}{2} \rfloor - 1$$

Reducing to hypergraph isomorphism

Define hypergraphs $\widetilde{G_f}$ and $\widetilde{G_g}$:

$$V = [n] \cup \{v_1, \ldots, v_r\}, r = \lfloor \frac{\ell}{2} \rfloor - 1$$

$$\widehat{f_\ell}(S) = 0.11001, \quad S \cup \{v_1, v_2, v_5\} \in E_f$$

Reducing to hypergraph isomorphism

Define hypergraphs $\widetilde{G_f}$ and $\widetilde{G_g}$:

$$V = [n] \cup \{v_1, \ldots, v_r\}, r = \lfloor \frac{\ell}{2} \rfloor - 1$$

$$\widehat{f_\ell}(S) = 0.11001, \quad S \cup \{v_1, v_2, v_5\} \in E_f$$

$$\widehat{g_\ell}(S) = 0.10111, \quad T \cup \{v_1, v_3, v_4, v_5\} \in E_g$$

Reducing to hypergraph isomorphism

Define hypergraphs $\widetilde{G_f}$ and $\widetilde{G_g}$:

$$V = [n] \cup \{v_1, \ldots, v_r\}, r = \lfloor \frac{\ell}{2} \rfloor - 1$$

$$\widehat{f_\ell}(S) = 0.11001, \quad S \cup \{v_1, v_2, v_5\} \in E_f$$

$$\widehat{g_\ell}(S) = 0.10111, \quad T \cup \{v_1, v_3, v_4, v_5\} \in E_g$$

Lemma

Suppose there exists an isomorphism π from $\widetilde{G_f}$ to $\widetilde{G_g}$ such that for all $i \in [r]$, $\pi(v_i) = v_i$. Then π is an isomorphism from $\widetilde{f_\ell}$ to $\widetilde{g_\ell}$.

Hypergraph Isomorphism

$$\widetilde{G_f}(V_f, E_f) \xrightarrow[?]{\pi} \widetilde{G_g}(V_g, E_g)$$

Hypergraph Isomorphism

$$\widetilde{G_f}(V_f, E_f) \xrightarrow[?]{\pi} \widetilde{G_g}(V_g, E_g)$$

For any $S \in E_f$,

$$|S| \leq t + \ell \leq (\log n)^{O(1)}$$

Hypergraph Isomorphism

$$\widetilde{G_f}(V_f, E_f) \xrightarrow[?]{\pi} \widetilde{G_g}(V_g, E_g)$$

For any $S \in E_f$,

$$|S| \leq t + \ell \leq (\log n)^{O(1)}$$

Theorem (Babai, Codenotti)

Given two hypergraphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ such that for all $S \in E_1$ and $T \in E_2$, $|S|, |T| \leq k$, there is a deterministic algorithm running in time $2^{\widetilde{O}(k^2\sqrt{n})}$ which computes an isomorphism between G_1 and G_2 if it exists or answers FALSE if no such isomorphism exists.

The Algorithm

The Algorithm

- Given circuits for f and g, compute polynomials $\widetilde{f}_\ell$ and $\widetilde{g}_\ell$.

The Algorithm

- Given circuits for f and g, compute polynomials $\widetilde{f_\ell}$ and $\widetilde{g_\ell}$.
- Construct hypergraphs $\widetilde{G_f}$ and $\widetilde{G_g}$ and compute the isomorphism using the algorithm of Babai and Codenotti.

The Algorithm

- Given circuits for f and g, compute polynomials $\widetilde{f_\ell}$ and $\widetilde{g_\ell}$.
- Construct hypergraphs $\widetilde{G_f}$ and $\widetilde{G_g}$ and compute the isomorphism using the algorithm of Babai and Codenotti.

Running time of the algorithm

$$O(n^t \cdot 2^\ell) + O(2^{(t+\ell)^2\sqrt{n}})$$

The Algorithm

- Given circuits for f and g, compute polynomials $\widetilde{f_\ell}$ and $\widetilde{g_\ell}$.
- Construct hypergraphs $\widetilde{G_f}$ and $\widetilde{G_g}$ and compute the isomorphism using the algorithm of Babai and Codenotti.

Running time of the algorithm

$$O(2^{(\log n)^{O(1)}}) + O(2^{(\log n)^{O(1)}\sqrt{n}})$$

Restating our result. . .

Theorem

Given two isomorphic boolean functions f and g computed by boolean circuits of size $n^{O(1)}$ and constant depth, there is a randomized algorithm with running time $2^{\widetilde{O}(\sqrt{n})}$ that computes a permutation π such that $\Pr[f(x) \neq g^{\pi}(x)] \leq \frac{1}{2^{(\log n)^{O(1)}}}$.

Conclusion

- Can this approximation guarantee be improved?
- Characterize the guarantees for different classes of boolean functions.

Thank You!