Information Processing Letters 116 (2016) 237–240

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Efficient computation of approximate isomorphisms between Boolean functions

Hans Ulrich Simon

Horst Görtz Institute for IT-Security and Department of Mathematics, Ruhr-University Bochum, D-44780 Bochum, Germany

ARTICLE INFO

Article history:
Received 24 March 2015
Received in revised form 9 November 2015
Accepted 9 November 2015
Available online 27 November 2015
Communicated by M. Chrobak

Keywords:
Randomized algorithms
Approximation algorithms
Analysis of algorithms
Boolean functions
Isomorphism problem

ABSTRACT

Arvind and Vasudev [2] have introduced the notion of an approximate isomorphism between two Boolean functions f and g. They furthermore designed two algorithms that construct an approximate isomorphism when given oracle access to f and g. The first of these algorithms is specialized to Boolean functions which are computable by constant-depth circuits. The second one applies to any pair of Boolean functions. It runs in exponential time and achieves optimality up to a factor of order $\sqrt{n}$. In this paper, we present an improved analysis and come up with a variant of the second algorithm that runs in polynomial time and achieves optimality up to a factor of (approximately) 2.

1. Introduction

Two Boolean functions are said to be isomorphic if they are equal up to a permutation of the variables. The problem of deciding whether two functions $f, g : \{0,1\}^n \to \{0,1\}$ are isomorphic is known to be coNP-hard even when f and g are given as DNF formulas. The problem is in Σ_2^p but it is not known to be in coNP. Agrawal and Thierauf [1] have shown that the problem is not Σ_2^p-hard unless the polynomial hierarchy collapses to Σ_3^p.

Arvind and Vasudev [2] have introduced the notion of approximate isomorphisms where the Boolean functions f, g are said to be $(1-\rho)$-approximately isomorphic if g, after applying an appropriate permutation to its variables, assigns the same binary labels as f on a fraction $\rho \in [0,1]$ of the Boolean domain. Clearly $\rho = 1$ means that f and g are (fully) isomorphic. The notion of an approximate isomorphism naturally leads to the following maximization problem: given oracle access to f and g, find a permutation of the variables of g which makes the agreement rate ρ with the function f as large as possible.

Arvind and Vasudev have designed two approximation algorithms. The first one is specialized to Boolean functions f, g which are computable by constant-depth circuits, say by circuits of depth d. It achieves the following: if f, g are (fully) isomorphic, then it runs in time $2^{\log^{O(d)}(n/\varepsilon)\sqrt{n}}$ and, with a probability of at least $1 - 2^{-\Omega(n)}$, it returns a permutation of the variables of g that leads to an agreement rate of $1-\varepsilon$ with the function f. The second algorithm in [2] applies to any pair of Boolean functions. It runs in exponential time and returns a permutation σ of the variables of g so that the resulting agreement rate with f differs from the optimal one at most by a factor of order $\sqrt{n}$. In this paper, we present an improved analysis and come up with an algorithm that runs in polynomial time and achieves optimality up to a factor of (approximately) 2. See Theorem 3.4 for the precise result.

The remainder of the paper is organized as follows. In Section 2, we specify the main problem more formally and fix some notations. In Section 3, we present our main results. In Section 4, we briefly discuss how small the agreement rate between f and g^σ can possibly become when

E-mail address: hans.simon@rub.de.

http://dx.doi.org/10.1016/j.ipl.2015.11.009
0020-0190/© 2015 Elsevier B.V. All rights reserved.

the *worst* permutation σ is chosen for two (fully) isomorphic functions f and g.

2. The problem MAX-BFI

The constants 0 and 1 are called *binary labels* or simply *labels* in this paper. For any Boolean function $g : \{0,1\}^n \to \{0,1\}$, any vector $x = (x[1], \ldots, x[n])$ and any permutation σ of $1, \ldots, n$, we introduce the notations

$$x^\sigma = (x[\sigma(1)], \ldots, x[\sigma(n)]) \text{ and } g^\sigma(x) = g(x^\sigma) \ .$$

Two Boolean functions $f, g : \{0,1\}^n \to \{0,1\}$ are said to be *isomorphic* if there exists a permutation σ such that $f = g^\sigma$. Boolean Function Isomorphism (BFI) is the problem of deciding whether two Boolean functions, given by oracle access,[1] are isomorphic.

As in [2], we replace the strict notion of an isomorphism by a measure ranging over the interval $[0,1]$ that quantifies "how isomorphic" two Boolean functions are. To this end, we define

$$\rho(f,g) = 2^{-n} \cdot |\{x \in \{0,1\}^n : f(x) = g(x)\}| \text{ and}$$
$$\rho^*(f,g) = \max_\sigma \rho(f, g^\sigma) \ .$$

We will refer to $\rho(f,g)$ as the *agreement rate* of f and g. Note that f and g are (fully) isomorphic iff $\rho^*(f,g) = 1$. In the sequel, we will discuss the following optimization problem:

MAX-BFI Given oracle access to $f, g : \{0,1\}^n \to \{0,1\}$, compute a maximizer σ^* of $\rho(f, g^\sigma)$, i.e. compute a permutation σ^* such that $\rho(f, g^{\sigma^*}) = \rho^*(f,g)$.

In the following section, we will present a randomized approximation algorithm for this problem.

Notational conventions We denote by $\mathcal{S}_n$ the set of permutations of $1, \ldots, n$. The notation $s \in_R S$ for some finite set S means that s is chosen uniformly at random from S. Probabilities (resp. expected values) involving parameters $s \in_R S$ are then written in the form $\Pr_{s \in_R S}[\cdot]$ (resp. $\mathbb{E}_{s \in_R S}[\cdot]$).

3. The main results

The key result in this section, Theorem 3.2 below, states that random permutations achieve, on the average, an agreement rate that differs from the optimal one by factor 1/2 only. But before we state and prove this formally, we discuss a very simple optimization problem in two real variables that will play a prominent role in the proof of the key result:

Lemma 3.1. *Let α, β be constants such that $0 \le \alpha \le 1 - \beta \le 1$. Let the function h be given by $h(a,b) = ab + (1-a)(1-b)$. Then,*

$$\left\{ \min_{a,b} h(a,b) \textbf{ s.t. } \alpha \le a, b \le 1 - \beta \right\} \ge \frac{\alpha + \beta}{2} \ . \qquad (1)$$

Proof. For each fixed a, the function $h_a(b) = h(a,b)$ is linear in b. Thus $h_a(b)$ is monotonically increasing or monotonically decreasing. Hence $h_a(b)$ is minimized for some $b \in \{\alpha, 1-\beta\}$. For reasons of symmetry, the analogous remark is valid with the roles of a and b exchanged. Therefore at least one of the sets $\{(\alpha, 1-\beta), (1-\beta, \alpha)\}$ and $\{(\alpha,\alpha), (1-\beta, 1-\beta)\}$ must contain an optimal solution (a^*, b^*) of the minimization problem on the left-hand side in (1). In the former case,

$$h(a^*, b^*) = \alpha(1-\beta) + (1-\alpha)\beta = \alpha + \beta - 2\alpha\beta$$

whereas, in the latter case,

$$h(a^*, b^*) \ge \min\{\alpha^2 + (1-\alpha)^2, \beta^2 + (1-\beta)^2\} \ .$$

Our assumptions on α and β imply that $\alpha, \beta \ge 0$ and $\alpha + \beta \le 1$. Since the geometric mean is upper-bounded by the arithmetic mean, we have $\alpha\beta \le (\alpha+\beta)^2/4 \le (\alpha+\beta)/4$ and $\alpha(1-\alpha) \le 1/4$. Thus $\alpha + \beta - 2\alpha\beta \ge (\alpha+\beta)/2$. Moreover,

$$\alpha^2 + (1-\alpha)^2 = 1 - 2\alpha(1-\alpha) \ge 1 - \frac{1}{2} = \frac{1}{2} \ge \frac{\alpha+\beta}{2}$$

and, for reasons of symmetry, the analogous inequality holds with the roles of α and β exchanged. In any case, one gets $h(a^*, b^*) \ge (\alpha+\beta)/2$, as desired. □

One can easily extend the proof of Lemma 3.1 and show that the optimal solution (a^*, b^*) is an element of $\{(\alpha, 1-\beta), (1-\beta, \alpha)\}$ if $\alpha, \beta \le 1/2$, whereas it equals (α, α) (resp. (β, β)) if $\alpha > 1/2$ (resp. if $\beta > 1/2$). Since we do not need this extension in the sequel, we omit the details.

We are now well prepared for the key result in this section:

Theorem 3.2. *For each pair $f, g : \{0,1\}^n \to \{0,1\}$ of Boolean functions, we have*

$$\mathbb{E}_{\sigma \in_R \mathcal{S}_n}[\rho(f, g^\sigma)] \ge \frac{\rho^*(f,g)}{2} \ .$$

Proof. For the sake of brevity, let $B = \{0,1\}^n$ and, for $i = 0, \ldots, n$, let $B_i \subseteq B$ be the subset consisting of all points with Hamming weight i. Let $s_i(f) = |B_i \cap f^{-1}(1)|$ denote the number of points in B_i to which f assigns the label 1. Let $X(u, \sigma)$ be the function that evaluates to 1 if $f(u) = g^\sigma(u) = g(u^\sigma)$ and that evaluates to 0 otherwise. If u is chosen uniformly at random from B_i and σ is chosen uniformly at random from $\mathcal{S}_n$, then (u, u^σ) is uniformly distributed over $B_i \times B_i$. From this and from the fact that $|B_i| = \binom{n}{i}$, it follows that

$$\mathbb{E}_{u \in_R B_i, \sigma \in_R \mathcal{S}_n}[X(u,\sigma)] = \frac{s_i(f) s_i(g)}{\binom{n}{i}^2} + \frac{\left(\binom{n}{i} - s_i(f)\right)\left(\binom{n}{i} - s_i(g)\right)}{\binom{n}{i}^2} \ .$$

Let $Y_i(\sigma)$ count the total number of agreements between f and g^σ on B_i, i.e., $Y_i(\sigma) = \sum_{u \in B_i} X(u, \sigma)$. Note that, for each $\sigma_0 \in \mathcal{S}_n$, we have

[1] An oracle for $f : \{0,1\}^n \to \{0,1\}$ returns $f(x)$ when called on x.