数理解析研究所講究録
第 876 巻 1994 年 60-68

Hardness of Learning Binary Decision Diagrams

Yasuhiko TAKENAGA and Shuzo YAJIMA
武永康彦　　矢島脩三
Faculty of Engineering, Kyoto University

Abstract

An ordered binary decision diagram (BDD) is a representation of a Boolean function, which is regarded as a kind of branching program. In this paper, we consider minimum BDD identification problems: given positive and negative examples of a Boolean function, output the BDD with minimum number of nodes (or with minimum width) that is consistent with all the examples. We prove in this paper that the problems are NP-complete. The result implies that k-width BDD and k-node BDD are not learnable under PAC-learning model unless NP = RP.

1 Introduction

An Ordered Binary Decision Diagram (or simply BDD) [1, 2] is a graph representation of a Boolean function. The BDD representation of Boolean functions has the following good properties. (1) Every Boolean function has a unique canonical BDD representation. (2) Many of practical Boolean functions are represented in feasible size. (3) Various basic operations such as reduction (minimization) and Boolean operations are executed efficiently. Owing to the excellent properties, BDD's have come to be indispensable in application programs of logic design verification, fault diagnosis of logic circuits, logic synthesis and so on. In the applications, the use of BDD's enables us to deal with large scale circuits efficiently.

Theoretically, a BDD is regarded as a kind of restricted branching program. For all the paths from the source node to a sink node, the variables are evaluated in a fixed order.

In this paper, we consider the problems to identify the minimum BDD that satisfies given positive examples and negative examples. We assume in this paper that the variable ordering is fixed, however, the size of BDD's may vary exponentially according to the variable ordering. The width and the number of nodes are used as the measure of minimality. We prove that, in both cases, the minimum BDD identification is NP-complete. If we regard that the value of the function is 'don't care' for the assignments which do not appear in the examples, the given examples represent an incompletely specified Boolean function. That is, minimum BDD identification is, in another word, a problem to find the simplest completely specified Boolean function that is consistent with a given incompletely specified function.

These problems are closely related to computational learning theory. In the PAC(Probably Approximately Correct)-learning model [3, 4], the learner generates a hypothesis based on the examples given by the teacher. We say that a concept is learnable if an approximately correct hypothesis can be presented with very high probability within polynomial time. It is known that k-term DNF, k-clause CNF, μ-formulas are not learnable under the PAC-learning model unless NP = RP [5]. We can conclude from the NP-hardness of minimum BDD identification that k-width BDD and k-size BDD are not learnable in polynomial time under PAC-learning model unless NP = RP.

2 Binary Decision Diagrams

An Ordered Binary Decision Diagram (BDD) [1, 2] is a directed acyclic graph that represents a Boolean function. The nodes of a BDD consist of variable nodes and two value nodes. Each variable node is labeled with a variable and has two outgoing edges, which are called 0-edge and 1-edge. Two value nodes are called 0-node and 1-node. There is a total ordering of variables for a BDD. On every path from the root node to a value node, each variable appears at most once according to the total ordering. The value of the function is given by traversing from the root node to a value node. At a variable node, one of the outgoing edges is selected according to the assignment to the variable. The value of the function is 0 if the terminal node is 0-node, and 1 if the terminal node is 1-node.

A BDD is represented by a permutation π on $\{1, 2, \cdots N\}$ and a set of 4-tuples $(i, index(i), low(i), high(i))$ that represent variable nodes, where

π is the total ordering of variables,

i is a node number,

$index(i) \in \{1, 2, \cdots N\}$ (N is the number of variables) is an index of the variable that is assigned to the node, and

$low(i), high(i)$ are the numbers of the nodes pointed by the 0-edge and the 1-edge respectively.

For a permutation π on $\{1, 2, \cdots N\}$, every node of a BDD must satisfy

$\pi(index(i)) < \pi(index(low(i)))$ and

$\pi(index(i)) < \pi(index(high(i)))$,

except when $low(i)$ or $high(i)$ is a value node. π is called a variable ordering. $\pi(index(i))$ is denoted by $level(i)$ and is called the level of node i or the level of $x_{index(i)}$.

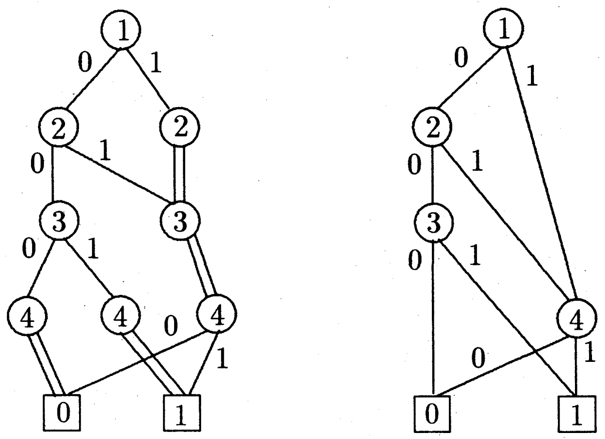
The Boolean function that is represented by node i, denoted by f_i, is defined as follows by Shannon's expansion :

$f_0 = 0, \ f_1 = 1,$

$f_i = x_{index(i)} \cdot f_{high(i)} + \overline{x_{index(i)}} \cdot f_{low(i)}.$

When the source node of a BDD A is a, the function represented by A is $f_A = f_a$.

When two nodes i and j represent the same function, they are called to be equivalent and denoted by $i \equiv j$. More precisely speaking, this equivalent relation is inductively defined as follows : $i \equiv j$ iff $level(i) = level(j)$, $high(i) \equiv high(j)$ and $low(i) \equiv low(j)$. When *high*(i) = *low*(i), node i is called to be redundant. A BDD is called a dense BDD when all the variable nodes satisfy *level*(i) + 1 = *level*(*low*(i)) = *level*(*high*(i)). Any BDD can be transformed to a dense BDD by adding redundant nodes. A dense BDD which has no equivalent nodes is called a quasi-reduced BDD [6]. A BDD which has no equivalent nodes and no redundant nodes is called a reduced BDD. Fig.1 shows the quasi-reduced BDD and the reduced BDD that represent $f = \bar{x}_1\bar{x}_2x_3 + (x_1 + x_2)x_4$.



(a) A quasi-reduced BDD. (b) A reduced BDD.

○ : variable node □ : value node

Figure 1: A quasi-reduced BDD and a reduced BDD.

Let *width*(k) be the sum of the number of nodes in level k and the number of edges that passes through level k. The width of each level is minimized in a quasi-reduced BDD. The minimum width of level k means the number of different subfunctions in level k. The width of a BDD is defined as $max_{1 \le k \le n} width(k)$.

3 Minimum Binary Decision Diagram Identification

3.1 NP-completeness of Minimum Binary Decision Diagram Identification Problems

In this section, we consider the complexity of identifying the minimum BDD from positive examples and negative examples. We assume in this paper that the variable ordering of a BDD is fixed.

Definition : MINIMUM WIDTH BDD IDENTIFICATION

Instance : A set EX of examples and a positive integer k.

Question : Is there a BDD of width less than or equal to k that satisfies all the examples?

Definition : MINIMUM BDD IDENTIFICATION

Instance : A set EX of examples and a positive integer k.

Question : Is there a BDD which has less than or equal to k nodes that satisfies all the examples?

Note that an example is a pair $\langle x, f(x)\rangle$, where $x \in \{0,1\}^n$ is an assignment for variables $x_1, x_2, \cdots, x_n$, and $f(x) \in \{0,1\}$ is the value of f for assignment x. The variable ordering of the BDD is fixed as $\pi(x_i) = i, 1 \leq i \leq n$.

When we assign values to $x_1, x_2, \cdots, x_k$, a function that satisfies EX is considered as $(n-k)$-variable incompletely specified Boolean function. Let f, g, h be incompletely specified Boolean functions. We denote $f \sqsubseteq g$ when $g(x) = 1$ if $f(x) = 1$ and $g(x) = 0$ if $f(x) = 0$ for all x. f and g can be unified iff there exists h s.t. $f \sqsubseteq h,\ g \sqsubseteq h$. Let $H = \{h | f \sqsubseteq h,\ g \sqsubseteq h\}$, then $h' = \sqcup\{f, g\}$ is defined as $h' \in H,\ \forall h \in H\ h' \sqsubseteq h$.

Theorem 1 MINIMUM WIDTH BDD IDENTIFICATION is NP-complete.

Proof First, we show a nondeterministic polynomial time algorithm for MINIMUM WIDTH BDD IDENTIFICATION.

Let $prefix_i(x)$ denote the $i (0 \leq i \leq n)$ highest bits of x, and let '$\cdot$' mean concatenation of sequences.

[Algorithm MinIdent]

1 : $P = \{prefix_i(x) \mid \langle x, f(x)\rangle \in EX,\ 1 \leq i \leq n\}$. For all $y \in P$, $1 \leq |y| < n$, guess $g(y) \in \{1, 2, \cdots, min(k, 2^{|y|})\}$. For $y \in P$ s.t. $|y| = n$, let $g(y) = f(y)$.

2 : For $1 \leq i \leq n, 1 \leq j \leq k$, let $P_{i,j} = \{prefix_i(x) | g(prefix_i(x)) = j\}$.

3 : For $1 \leq i \leq n$, $1 \leq j \leq k$, check whether the following conditions are satisfied. 1) $g(r \cdot 0) = g(s \cdot 0)$ for all r, s s.t. $r, s \in P_{i,j}$, $r \cdot 0 \in P$ and $s \cdot 0 \in P$, 2) $g(r \cdot 1) = g(s \cdot 1)$ for all r, s s.t. $r, s \in P_{i,j}$, $r \cdot 1 \in P$ and $s \cdot 1 \in P$.

If the conditions are satisfied for all i, j, r, s, then there exists a BDD of width less than or equal to k.

We can see that $|P| \leq |EX| \times n$ and $|P_{i,j}| \leq |EX|$. Therefore the time requirement of Algorithm MinIdent is bounded by a polynomial of n and $|EX|$.

We shall claim the correctness of Algorithm MinIdent. We can construct a BDD as follows. The path corresponding to an assignment x is on the $g(prefix_{i-1}(x))$-th node in level i. For each $1 \leq i \leq n$, $r \in P(|r| = i-1)$, the 0-edge from the $g(r)$-th node of level i points $g(r \cdot 0)$-th node of level $i+1$ if $r \cdot 0 \in P$, and the 1-edge points $g(r \cdot 1)$-th node if $r \cdot 1 \in P$.

If the conditions of 3: are not satisfied, there exists a node that has more than one 0-edges or 1-edges. Otherwise, each node has at most one 0-edge and 1-edge, and the generated graph is a subgraph of a k-width BDD. Moreover, we can easily see that there are paths from the source node to value nodes for all the assignments given as examples. The edges which are not generated by the above method may point any node.

Next, we show the NP-hardness of MINIMUM WIDTH BDD IDENTIFICATION by reduction from GRAPH K-COLORABILITY.

Definition : GRAPH K-COLORABILITY
Instance : An undirected graph $G(V, E)$ and a positive integer k.
Question : Is there a function $f : V \rightarrow \{1, 2, \cdots, k\}$ s.t. $f(i) \neq f(j)$ for all the edges $(i, j) \in E$?

Let N denote the number of nodes in G. We can assume without loss of generality that N is a power of 2.

The Boolean function of the reduced problem has $6logN + 2$ variables. The set of examples are as follows :

$\langle B_i \cdot B_j \cdot B_p \cdot B_q \cdot 00 \cdot B_r \cdot B_s\ ,\ f_i(r, s)\rangle\ \ (r < s)$,
$\langle B_i \cdot B_j \cdot B_p \cdot B_q \cdot 01 \cdot B_r \cdot B_s\ ,\ f_j(r, s)\rangle\ \ (r < s)$,
$\langle B_i \cdot B_j \cdot B_p \cdot B_q \cdot 10 \cdot B_r \cdot B_s\ ,\ f_p(r, s)\rangle\ \ (r < s)$,
$\langle B_{N-1} \cdot B_{N-1} \cdot B_{N-1} \cdot B_q \cdot 11 \cdot B_r \cdot B_s\ ,\ g_q(r, s)\rangle\ (r < s, (r, s) \in E)$ and
$\langle B_i \cdot B_j \cdot B_p \cdot B_q \cdot 11 \cdot B_r \cdot B_s\ ,\ f_q(r, s)\rangle\ (r < s$, excepting $i = j = p = N - 1)$,

where $0 \leq i, j, p, q, r, s \leq N-1$ and B_i is a binary representation of an integer i. $f_0, f_1, \cdots, f_{N-1}$ and $g_0, g_1, \cdots, g_{N-1}$ are defined as follows :

$f_t(B_t, B_s) = 0$ iff $t < s$,
$f_t(B_r, B_t) = 1$ iff $r < t$,
$g_t(B_t, B_s) = 0$ iff $t < s$ and $(t, s) \in E$ and
$g_t(B_r, B_t) = 1$ iff $r < t$ and $(r, t) \in E$.

The positive integer to bound the width of the BDD is $N^4 - N + k$.

The number of examples is
$(N - 1)(4N^4 - N) + 2|E| = O(N^5)$.

The examples can be generated using $O(logn)$ space.

We shall prove that there exists a $(N^4 - N + k)$-width BDD that satisfy all the examples iff graph G is k-colorable. In order to count the width of each level, we use the following propositions.

Propositions 1. For any i, j $(i \neq j,\ 0 \leq i \leq N - 1)$, f_i and f_j cannot be unified.

2. g_i and g_j $(i \neq j)$ can be unified iff $(i, j) \notin E$.

3. $g_i \sqsubseteq f_i$ $(0 \leq i \leq N - 1)$.

4. If $g_{i_1}, g_{i_2}, \cdots g_{i_m}$ can be unified, $g' = \sqcup\{g_{i_1}, g_{i_2}, \cdots g_{i_m}\}$ can be unified with any of f_{i_j} $(0 \leq j \leq m)$.

Proof 1. When $i < j$, $f_i(i, j) = 0$, $f_j(i, j) = 1$.

4. We have only to prove the case where $m = 2$. f_i and f_j $(i < j)$ differ only when the parameters are i and j. However, g_i and g_j can be unified, because $(i, j) \notin E$, that is, $g_i(i, j)$ and $g_j(i, j)$ are undefined. Therefore, f_i and g_j (f_j and g_i) can be unified. □

The next lemma follows from Proposition 2.

Lemma 1 $g_i (0 \leq i \leq N-1)$ can be devided into k subsets all of whose elements can be unified iff G is k-colorable. □

The minimum width of each level is as follows. In this case, the width is minimized in each level ($\leq 4logN + 3$) at the same time.

1. $1 \leq level \leq 4logN$

   $width(level) \leq 2^{level-1}$. Especially, $width(level) \leq N^4/2$ when $level = 4logN$.

2. $level = 4logN + 1$

   There are N^4 nodes in this level, some of which can be unified. In case $i = j = p = N-1$, there are N functions of the form $x_{4logN+1} \cdot x_{4logN+2} \cdot g_a + \overline{x_{4logN+1} \cdot x_{4logN+2}} \cdot f_{N-1}$, $0 \leq a \leq N-1$. Therefore, the functions differ only when $x_{4logN+1} = x_{4logN+2} = 1$. From Lemma 1, these N nodes can be reduced to k nodes.

   Otherwise, for at least one assignment to $x_{4logN+1}$ and $x_{4logN+2}$, different functions are selected among f_a, $0 < a < N-1$. From proposition 1, the functions cannot be unified. Hence, $width(4logN + 1) = N^4 - N + k$.

3. $level = 4logN + 2$

   In this level, there are N^2 functions of the form $\overline{x_{4logN+2}} \cdot f_a + x_{4logN+2} \cdot f_b, 0 \leq a, b \leq N-1$ and k functions of the form $\overline{x_{4logN+2}} \cdot f_{N-1} + x_{4logN+2} \cdot h_c$, $0 \leq c \leq k$, where $h_c = \sqcup\{g_{i_1}, g_{i_2}, \cdots g_{i_m}\}$. The former ones cannot be unified each other. The latter ones can be unified with one of the former functions from Proposition 4. Therefore, $width(4logN + 2) = N^2$.

   Even though the width is not minimized in level $4logN + 1$, the width of this level is N^2 by the same argument.

4. $level = 4logN + 3$

   As is the case of 3, $width(4logN + 3) = N$.

5. $4logN + 4 < level \leq 6logN + 2$

   In general, the width of a level is at most twice the width of the preceding level. Therefore $width(level) \leq N \times 2^{level-4logN-3} \leq N^3$.

As $N^4 - N + k > N^4/2 \geq 3/2N^3$ for $N \geq 2$, the width of this BDD is $N^4 - N + k$. □

The proof shows that it is still NP-complete to minimize only the width of a specified level.

Theorem 2 MINIMUM BDD IDENTIFICATION is NP-complete.

Proof We can easily see that MINIMUM BDD IDENTIFICATION is in NP by extending Algorithm MinIdent. For the proof of NP-hardness, we use a reduction from GRAPH K-COLORABILITY. The basic idea of this proof is similar to that of Theorem 1.

The Boolean function of the reduced problem has $7logN+4$ variables. The set of examples are as follows :

$\langle B_i \cdot B_j \cdot B_p \cdot B_q \cdot B_m \cdot *000 \cdot B_r \cdot B_s\ ,\ f_i(r,s)\rangle\ \ (r<s)$,
$\langle B_i \cdot B_j \cdot B_p \cdot B_q \cdot B_m \cdot *001 \cdot B_r \cdot B_s\ ,\ f_j(r,s)\rangle\ \ (r<s)$,
$\langle B_i \cdot B_j \cdot B_p \cdot B_q \cdot B_m \cdot *010 \cdot B_r \cdot B_s\ ,\ f_p(r,s)\rangle\ \ (r<s)$,
$\langle B_i \cdot B_j \cdot B_p \cdot B_q \cdot B_m \cdot *011 \cdot B_r \cdot B_s\ ,\ f_q(r,s)\rangle\ \ (r<s)$,
$\langle B_i \cdot B_j \cdot B_p \cdot B_q \cdot B_m \cdot 0100 \cdot B_r \cdot B_s\ ,\ f_m(r,s)\rangle\ \ (r<s)$,
$\langle B_i \cdot B_j \cdot B_p \cdot B_q \cdot B_m \cdot 1100 \cdot B_r \cdot B_s\ ,\ g_m(r,s)\rangle\ (r<s,(r,s)\in E)$,
$\langle B_i \cdot B_j \cdot B_p \cdot B_q \cdot B_m \cdot 0101 \cdot B_r \cdot B_s\ ,\ f_0(r,s)\rangle\ \ (r<s)$,
$\langle B_i \cdot B_j \cdot B_p \cdot B_q \cdot B_m \cdot 1101 \cdot B_r \cdot B_s\ ,\ f_1(r,s)\rangle\ \ (r<s)$,

where $0 \le i,j,p,q,m,r,s \le N-1$ and $*$ means both 0 and 1. $f_0, f_1, \cdots, f_{N-1}$ and $g_0, g_1, \cdots, g_{N-1}$ are the same as those defined in the proof of Theorem 1. The positive integer to bound the number of nodes is $3N^5+(k+2)N^4-2$.

The number of examples is
$11N^5(N-1)+2|E|N^4 = O(N^6)$.

To count the number of nodes, we must remove redundant nodes from the width of each level. The minimum number of nodes in each level, denoted by $node(level)$, is as follows. In this case, the number of nodes can be minimized in each level ($\le 5logN+5$) at the same time.

1. $1 \le level \le 5logN+1$

   There are N^5 nodes and any two nodes cannot be unified. Therefore, $node(level) = 2^{level-1}$. The total number of nodes is $\sum_{1\le level\le 5logN+1} node(level) = 2N^5-1$.

2. $level = 5logN+2$

   When i,j,p,q are fixed and $x_{5logN+1}=1$, there are N functions which differ only when $x_{5logN+2}=1, x_{5logN+3}=x_{5logN+4}=0$. They can be reduced to k functions iff G is k-colorable. In any other cases, the nodes in this level cannot be unified. Hence $node(5logN+2) = N^5+kN^4$ iff G is k-colorable.

3. $level = 5logN+3$

   In this level, there are N^4 different functions of the form $\overline{x_{5logN+3}}\cdot\overline{x_{5logN+4}}\cdot f_a + \overline{x_{5logN+3}}\cdot x_{5logN+4}\cdot f_b + x_{5logN+3}\cdot\overline{x_{5logN+4}}\cdot f_c + x_{5logN+3}\cdot x_{5logN+4}\cdot f_d$, $0 \le a,b,c,d \le N-1$ and k functions of the form $\overline{x_{5logN+3}}\cdot\overline{x_{5logN+4}}\cdot h_a + \overline{x_{5logN+3}}\cdot x_{5logN+4}\cdot f_1$, $0\le a\le k$. The former ones cannot be unified each other. Among them, N^2 nodes can be removed as redundant nodes. The latter ones can be unified with the former ones from Proposition 4. Therefore, for any G, $node(5logN+3) = N^4-N^2$.

4. $level = 5logN+4$

   As is the case of 3, $node(5logN+4) = N^2-N$.

5. $5logn + 5 \leq level \leq 7logN + 4$

   $node(5logN + 5) = N$. Then the total number of nodes is less than $\sum_{1 \leq i \leq 2logN} N \cdot 2^{i-1} = N^3 - N$.

From the above discussion, when G is exactly k-colorable, the total number of nodes is at least

$$\begin{aligned} & num_{min}(N,k) \\ &= (2N^5 - 1) + (N^5 + kN^4) + (N^4 - N^2) + (N^2 - N) + N \\ &= 3N^5 + (k+1)N^4 - 1 \end{aligned}$$

and is not more than

$$\begin{aligned} & num_{max}(N,k) \\ &= (2N^5 - 1) + (N^5 + kN^4) + (N^4 - N^2) + (N^2 - N) + (N^3 - N) \\ &= 3N^5 + (k+1)N^4 + N^3 - 2N - 1. \end{aligned}$$

As $num_{min}(N, k+1) > 3N^5 + (k+2)N^4 - 2 > num_{max}(N,k)$, the number of nodes is less than $3N^5 + (k+2)N^4 - 2$ iff G is k-colorable. □

3.2 Hardness of Learning Binary Decision Diagrams

The identification of the minimum BDD from examples is closely related to computational learning theory. On the PAC-learning model [3, 4], the goal is to find a good approximation of an unknown Boolean function from random examples. When the learner requests an example, it is drawn according to an arbitrary distribution P on $\{0,1\}^n$. The error of a hypothesis g for unknown f is defined to be the probability that $f(x) \neq g(x)$ for an assignment $x \in \{0,1\}^n$ drawn randomly according to P.

We call that a Boolean function is learnable by a class X of concepts iff there is a learning algorithm that runs in polynomial time and outputs, with probability at least $1-\delta$, a hypothesis that approximates the unknown Boolean function with error at most ϵ.

From Theorem 1 and 2, we can make the same discussion as [5], on the learnability of k-width BDD and k-node BDD. If there is a polynomial time learning algorithm, we can solve GRAPH K-COLORABILITY using the learning algorithm and the examples shown in the reduction, which implies NP =RP.

Corollary 1 k-width BDD and k-node BDD are not learnable under PAC-learning model unless NP = RP.

4 Conclusion

In this paper, we proved the NP-completeness of identifying the minimum BDD. The results also imply the hardness of learning k-width BDD and k-node BDD. It is our future work to consider the case when we allow to change the variable ordering because the size of a BDD greatly varies according to the variable ordering.

References

[1] S. B. Akers : "Binary Decision Diagrams", IEEE Trans. Comput. Vol.C-27, No.6, pp.509-516 (1978).

[2] R. E. Bryant : "Graph-based Algorithms for Boolean Function Manipulation", IEEE Trans. Comput. Vol.C35, No.8, pp.677-691 (1986).

[3] L. G. Valiant : "A Theory of the Learnable", Comm. ACM, Vol.27, No.11, pp.1134-1142 (1984).

[4] M. Kearns, M. Li, L. Pitt and L.G. Valiant : "On the Learnability of Boolean Formulae", Proc. 19th STOC, pp.285-295 (1987).

[5] L. Pitt and L. G. Valiant : "Computational Limitations on Learning from Examples", J. ACM, Vol.35, No.4, pp.965-984 (1988).

[6] N. Ishiura : "Synthesis of Multi-level Logic Circuits from Binary Decision Diagrams", Proc. Synthesis and Simulation Meeting and International Interchange '92, pp.74-83 (1992).