On the Hardness of Approximating the Minimum Consistent OBDD Problem

Kouichi Hirata * Shinichi Shimozono † Ayumi Shinohara ‡

Kyushu Inst. of Tech. Kyushu Inst. of Tech. Kyushu University

Abstract

Ordered binary decision diagrams (OBDD, for short) represent Boolean functions as directed acyclic graphs. The minimum consistent OBDD problem is, given an incomplete truth table of a function, to find the smallest OBDD that is consistent with the truth table with respect to a fixed variable ordering. First, we show that this problem is NP-hard. Then, we prove that there is a constant $\epsilon > 0$ such that no polynomial time algorithm can approximate the minimum consistent OBDD within the ratio n^ϵ unless P=NP, where n is the number of variables. These results suggest that OBDDs are unlikely to be polynomial time learnable in PAC-learning model. Furthermore, we give a polynomial time learnable subclass of OBDDs representing symmetric functions.

1 Introduction

An ordered binary decision diagram (OBDD for short) [9] represents a Boolean function as a directed acyclic graph whose internal nodes correspond to the input variables and terminal nodes associated with the output values. Many useful Boolean functions, such as symmetric functions and threshold functions, can be expressed succinctly by OBDDs [9]. The size of an OBDD is, however, strongly dependent on the variable ordering [9], and it is known that finding the optimal ordering that realizes the minimum size OBDD representing the function is intractable [8]. For a fixed variable ordering, the minimum size OBDD representing the function can be computed in linear time from the *complete* truth table [19].

*Address: Department of Artificial Intelligence, Kyushu Institute of Technology, Kawazu 680-4, Iizuka, 820 JAPAN. E-mail: `hirata@ai.kyutech.ac.jp`

†Address: Department of Control Engineering and Science, Kyushu Institute of Technology, Kawazu 680-4, Iizuka, 820 JAPAN. E-mail: `sin@ces.kyutech.ac.jp`

‡Address: Research Institute of Fundamental Information Science, Kyushu University, Fukuoka 812 JAPAN. Fax: +81-92-611-2668, E-mail: `ayumi@rifis.kyushu-u.ac.jp`

However, to our knowledge, the problem of finding the minimum size OBDD from an *incomplete* truth table has not been investigated. It corresponds to the *minimum consistent problem* to find the smallest hypothesis that explains given positive and negative examples correctly, from the viewpoint of computational learning theory. Computational complexity of the minimum consistent problem for a class of representations of hypotheses is closely related to the efficiency of learning algorithms.

For the class of deterministic finite automata (DFA, for short), the problem of determining the minimum state DFA consistent with given examples was shown to be NP-hard [1, 13]. This negative result was enhanced as the lower bound on approximation ratios due to Li and Vazirani [15]. They showed that, unless P=NP, the minimum consistent DFA cannot be approximated within the ratio $\frac{9}{8}$ in polynomial time. Pitt and Warmuth [18] improved to the ratio opt^k, where opt is the minimum number of states and k is any positive integer.

Hancock *et al.* [14] investigated the problem for decision lists and decision trees that represent n-ary Boolean functions. They showed that decision lists cannot be approximated in polynomial time within a factor of n^c for some constant $c > 0$, unless P=NP. They also showed that decision trees cannot be approximated in polynomial time within a factor of n^c for any $c > 0$ unless NP is included in DTIME[$2^{poly \log n}$].

In this paper, we analyze the complexity of the minimum consistent problem for OBDD with a fixed variable ordering. First, we show that the problem is NP-hard, even the number of positive examples is only one. Then, we show that there is a constant $\epsilon > 0$ such that no polynomial time algorithm can approximate the minimum consistent OBDD within the ratio n^ϵ unless P=NP. The proof employed in our result does not use cyclic transitions, which is the essence of the hardness for DFA in [18]. Our results suggest that any efficient algorithm for learning OBDDs would have to produce very large hypotheses.

On the learnability of OBDDs from examples, Gavaldà and Guijarro [12] studied in the model of *exact learning via queries* [2, 3, 4]. They claimed that learning algorithm for DFA using both membership and equivalence queries can be applied to learning OBDDs if the variable ordering is fixed. Ergün *et al.* [10] introduced *bounded-width* OBDDs[1] and investigated learnability of OBDDs in *probably approximately correct* (*PAC* for short) learning model [17, 20]. They have shown that, for a fixed variable ordering, width-2 OBDDs are polynomial time PAC-learnable[2], although learning

[1]The paper [10] deals with the notion of branching programs instead of OBDDs.

[2]Note that the result in [10] is not a *proper* PAC-learnability.

width-k OBDDs is as hard as learning DNF formulas for $k \geq 3$. In this paper, we also consider a learnable subclass of OBDDs, and show that OBDDs representing symmetric functions are polynomial time PAC-learnable.

2 Minimum Consistent OBDD Problem

We first give definitions and notations for OBDDs. Then we introduce the minimum consistent problem of OBDD as a combinatorial optimization problem, and show that the problem is NP-hard.

Definition 1. Let X be a set $\{x_1, \ldots, x_n\}$ of variables, and let V be a set $\{1, \ldots, m\}$ of positive integers. A *binary decision diagram* $D = (V, E_1, E_0, X, var)$ over X is a rooted, directed acyclic graph $(V, E_0 \cup E_1)$ whose nodes are labeled with variables in X, except terminal nodes labeled with output values 0 and 1. Labels are associated with the nodes by the mapping $var : V \to X \cup \{0, 1\}$. Two sets E_0 and E_1 of arcs are disjoint. Every internal node u has exactly two arcs $(u, v_1) \in E_1$ and $(u, v_0) \in E_0$, which indicate successors v_1 and v_0 with respect to values 1 and 0 assigned to $var(u)$. The root node is the only node in D that has no incoming arcs. The size of an OBDD is the number of internal nodes.

Let $<$ be a total order $x_1 < \cdots < x_n$ on X. We say $D = (V, E_0, E_1, X, var)$ is an *ordered binary decision diagram* (OBDD for short) if the order satisfy $var(u) < var(v)$ whenever v is reachable from u.

For a given input w in $\{0, 1\}^n$, the output value of D is determined by tracing the arcs from the root node to terminal nodes according to the values $w[1], \ldots, w[n]$ assigned to the associated variables $x_1, \ldots, x_n$. We call the terminal nodes labeled with 1 and 0 by 1-terminal and 0-terminal, respectively.

We define an OBDD $D_{(v)}$ by the partial diagram of D whose root node is v. Then we say that u and v are *duplicated nodes* if $D_{(u)}$ and $D_{(v)}$ are the same partial diagrams. Let P and N be disjoint subsets of $\{0, 1\}^n$. We say that D is *consistent with P and N* if D outputs 1 for any input in P and outputs 0 for any input in N. We say that two OBDDs D and D' are *equivalent with respect to P and N* if both of them are consistent with P and N. Let D be an OBDD that is consistent with $P, N \subseteq \{0, 1\}^n$. Let u be a node of D with arcs $(u, v_1) \in E_1$ and $(u, v_0) \in E_0$, and let D' and D'' be the OBDD obtained from D by removing u and redirecting all incoming arcs of u to either v_1 or v_0, respectively. If D is equivalent to D' or D'', then we say that u is *redundant*. We say that an OBDD D is *reduced* if D has no duplicated nodes and no redundant nodes. Note that redundant nodes defined here are generalization of those introduced in [9].

Given disjoint sets $P, N \subseteq \{0,1\}^n$ of strings, a reduced OBDD that is consistent with P and N can be obtained by a trivial polynomial-time algorithm: It first produces a tree whose path to a terminal corresponds to each string in $\{0,1\}^n$, and leaves are duplicated terminal nodes, then it reduces duplicated and redundant nodes in the tree.

Now we consider the following problem.

Definition 2. MINIMUM CONSISTENT OBDD

Instance: Two disjoint sets $P, N \subseteq \{0,1\}^n$.

Solution: An n-variable OBDD D that is consistent with P and N.

Cost: The size of D.

The goal of this combinatorial optimization problem is to find an OBDD of the smallest size. In the case $P \cup N = \{0,1\}^n$, a reduced OBDD is unique and minimum [9]. However, in general, the following theorem holds.

Theorem 1. MINIMUM CONSISTENT OBDD is NP-hard.

Proof. We present a log-space reduction from an NP-hard combinatorial optimization problem MINIMUM COVER [11] that has presented in [14] to show that the shortest consistent monomial cannot be approximated within a ratio $\log n$ in polynomial time. Given a collection C of subsets over U, MINIMUM COVER is the problem to find a subcollection $C' \subseteq C$ that covers U as small as possible. We provide, for C over U with $n = |C|$, the sets P and N of strings as follows: P consists of only one string 1^n, and N consists of $|U|$ strings each of which, for every $i \in U$, is obtained by placing 0s on 1^n at all positions k for c_k such that $i \in c_k$. Then any OBDD consistent with P and N has only one path to 1-terminal on which the kth node detect all negative strings that correspond to elements contained in c_k. Thus there is an OBDD that has m nodes and is consistent with P and N if there is a cover $C' \subseteq C$ for U such that $|C'| = m$. □

3 Approximating the Minimum Consistent OBDD

Since the problem to find a minimum consistent OBDD is NP-hard, it is natural to ask whether there is an efficient algorithm that approximates the minimum OBDD with a ratio nearly 1. Notice that the reduction referred in Lemma 1 preserves the cost between MINIMUM COVER and MINIMUM CONSISTENT OBDD. Hence, even for an instance with only one positive example, we do not have any polynomial-time algorithm that can approximate the minimum consistent OBDD within the ratio $\log n$, unless NP $\subseteq$ DTIME$[n^{poly \log n}]$ [16]. We raise this lower bound by a reduction that involves many positive examples.

To obtain our result, we invoke the non-approximability result on the well-known combinatorial optimization problem GRAPH COLORING (CHROMATIC NUMBER) [11, 16]. Let $G = (V, E)$ be an undirected graph. We say that G is *k-colorable* if there is a k-coloring of G, that is, a mapping $f : V \to \{1, \ldots, k\}$ such that $f(u) \neq f(v)$ whenever $(u, v) \in E$. The *chromatic number* of G is the minimum number k such that G is k-colorable. We denote the chromatic number of G by $K(G)$. Then the problem GRAPH COLORING is, given a graph G, to find a k-coloring for G such that k is as small as possible. It is known in [16] that there is a constant $c > 0$ such that GRAPH COLORING cannot be approximated within the ratio n^c unless P=NP. Thus, there is $c > 0$ such that no polynomial-time algorithm can always find a k-coloring satisfying $k/K(G) < n^c$ for all G.

Let us start with introducing a translation from a graph to sets of strings for dealing with OBDDs. Let $G = (V, E)$ be a graph with the set $V = \{1, \ldots, n\}$ of nodes. For each node $i \in V$, we define the *adjacency sets* P_i and N_i as follows: (i) P_i consists of 1^n and $p_{ij} = 1^{j-1}\,0\,1^{n-j}$ for every $(i, j) \in E$, and (ii) N_i consists of only one string $q_i = 1^{i-1}\,0\,1^{n-i}$ corresponding to the node i. Note that, for any $i \in V$, P_i and N_i are disjoint.

Let U be a subset of V. We denote the union $\bigcup_{i \in U} P_i$ by P_U and $\bigcup_{i \in U} N_i$ by N_U. We identify a mapping $f : V \to \{1, \ldots, k\}$ with the partition of nodes $U_1, \ldots, U_k \subseteq V$ defined by $U_i = \{v \in V \mid f(v) = i\}$.

Lemma 1. A partition $U_1, \ldots, U_k$ of V is a k-coloring of G if and only if P_{U_i} and N_{U_i} are disjoint for all $1 \leq i \leq k$.

Proof. This holds since there is $(i, j) \in E$ such that $i, j \in U_h$ for some $1 \leq h \leq k$ if and only if there are the same strings p_{ij} in $P_i \subseteq P_{U_h}$ and q_j in $N_j \subseteq N_{U_h}$.

Lemma 2. Let $G = (V, E)$ be a graph and let P_U and N_U be disjoint unions of the adjacency sets for $U \subseteq V$ of G. Then a reduced OBDD that is consistent with P_U and N_U is the minimum OBDD and has exactly $|U|$ nodes with only one path to 1-terminal.

Proof. Every string in $P_U \cup N_U$ includes exactly one 0, except 1^n in P_U. Therefore, any non-terminal node detects either one negative string or one positive string from the others. Since there are only $|U|$ negative strings in N_U, a reduced OBDD has exactly $|U|$ nodes on the only path to 1-terminal each of which detects one string in N_U.

The translation of graphs to strings presented here is rather different from those provided for the reductions to other problem, such as k-term DNF and k-Decision List [14, 17]. This is because OBDDs can count easily the number of 0s (or 1s) in the input string. Now we are ready to prove our main theorem.

Theorem 2. There is a constant $0 < \epsilon < 1$ such that Minimum Consistent OBDD cannot be approximated within a ratio n^ϵ in polynomial time unless P = NP.

Proof. We show a reduction preserving the approximation ratio from Graph Coloring to Minimum Consistent OBDD. Let $G = (V, E)$ be a graph with the set $V = \{1, \ldots, n\}$ of nodes. We denote the binary representation of $i - 1$ of the length $l = \lceil \log n \rceil$ by $\langle i \rangle$ for $1 \leq i \leq n$. For G, we define the sets P and N of strings, which consist of three parts, as follows:

$$\begin{aligned} P &= \{\langle i \rangle\, 1^n\, p \mid i \in V \wedge p \in P_i\} \\ N &= \{\langle i \rangle\, 1^n\, q \mid i \in V \wedge q \in N_i\} \cup \{\langle i \rangle\, 1^{j-1}\, 0\, 1^{n-j}\, 1^n \mid j \in V\}\,, \end{aligned}$$

where P_i and N_i for $i \in V$ are the adjacency sets of G. Notice that P and N are disjoint.

First, we show the following: Any reduced OBDD that is consistent with P and N partitions P_V and N_V by $l + n$ prefix of strings, i.e., the header part and the middle part, into $P_{U_1}, \ldots, P_{U_k}$ and $N_{U_1}, \ldots, N_{U_k}$ corresponding to k-coloring of G.

By a complete binary tree that examines all the first l symbols of the header part, P_i and N_i for all $i \in V$ can be separated from other P_j and N_j with $i \neq j$. Of course, a complete binary tree is the "worst case" that uses $n - 1$ nodes. However, we ignore how many nodes are involved to make a partition: The number of nodes needed to make a partition that corresponds to a coloring will be considered later.

Let D be a reduced OBDD that is consistent with P and N. Any path of D to 1-terminal has every node labeled with variables $x_{l+1}, \ldots, x_{l+n}$ to detect all negative strings $\langle i\rangle 1^{i-1}01^{n-i}1^n$ for $1 \leq i \leq n$. This is the same discussion in the proof of Lemma 2. We will call this part on a path consisting of n nodes a *conduit*. We say a conduit C *encloses* a string s if all nodes of C are traced when computing the output of D for input s , i.e., s "flows" through C.

Let u be a of node of a conduit enclosing $\langle i\rangle 1^n 1^n$. The node u is a duplicate node if and only if there is a node v of a conduit enclosing $\langle j\rangle 1^n 1^n$ such that $P_i \cup P_j$ and $N_i \cup N_j$ are disjoint. Therefore, for $U \subseteq V$, all conduits enclosing any string with header $\langle i\rangle$ for $i \in U$ can be overlayed into one conduit if and only if P_U and N_U are disjoint.

Suppose that D classifies the strings in P and N into a partition consisting of k groups by overlaying conduits. Let $U_1, \ldots, U_k \subseteq V$ be a partition representing the groups classified by the header part. Then, for any $1 \leq i \leq k$, P_{U_i} and N_{U_i} must be disjoint. If the partition makes k conduits, there needs nk nodes in the diagram D to construct the conduits.

The rest of each path only needs to detect 0 of q_i as we have seen in Lemma 2. Therefore, D must have exactly n nodes labeled with $x_{l+n+1}, \ldots, x_{l+n+n}$.

Now we consider the number of nodes to partition the strings into a pairs of disjoint sets. For classifying n strings by their header part into k groups, there needs at least $k-1$ nodes and at most $n-1$ nodes, depending on how the numbers representing nodes are classified. Therefore, if a reduced and consistent OBDD has k conduits, then the number of nodes of D is at least $k-1+nk+n$ and at most $n-1+nk+n$.

Let us consider the approximation ratio of Minimum Consistent OBDD. Let D be a OBDD consistent with P and N. We can compute a reduced OBDD D' with respect to P and N from D in polynomial time with $|P \cup N|$. Let k be the number of conduits in D'. Then, for $k \geq K(G) > 2$, the approximation ratio r of D satisfies

$$\frac{k}{2K(G)} < \frac{k-1+nk+n}{n-1+nK(G)+n} \leq r\,.$$

Since conduits in D' corresponds to a k-coloring for G, the lower bound $n^c < \frac{k}{K(G)}$ for Graph Coloring also gives the lower bound of the ratio for Minimum Consistent OBDD. Let m be the length of strings in P and N (so the number of variables). Then $m = l + 2n < 3n$ and thus the inequality $\frac{1}{2}(\frac{m}{3})^c < \frac{1}{2}n^c < r$ holds. Therefore, there is $0 < \epsilon < c < 1$ such that $n^\epsilon < r$ for sufficiently large m satisfying $(2 \cdot 3^c)^{\frac{1}{c-\epsilon}} < m$ unless P = NP. □

4 A Learnable Subclass of OBDDs

In this section, we discuss the relationship between the learnability of OBDDs and our results on MINIMUM CONSISTENT OBDD.

Valiant [20] has proposed a criterion of correct identification of a concept from examples in a stochastic setting. The idea of this model is that after randomly sampled examples and non-examples of a concept are given, an identification procedure should conjecture a concept with "high probability" that is "not too different" from the correct concept. This model is called a *probably approximately correct* (*PAC*, for short) learning model.

Angluin [2, 3] has introduced the "minimally adequate teacher" in order to learning DFAs, and developed the learning model which is allowed to make several types of queries. The goal of this model is *exact* identification of a concept from a concept class, that is, to succeed the algorithm must halt and output a concept which is equivalent to a target concept. In this section, the learning algorithm gathers information about a target concept using two types of queries; membership and equivalence queries.

Let D_* be a target OBDD. For any $w \in \{0,1\}^n$, the answer to the *membership* query is *yes* if w satisfies D_*; otherwise, *no*. For a hypothesis OBDD D, the answer to the *equivalence* query is *yes* if D is equivalent to D_*; otherwise, the answer is *no*, and the query returns a *counterexample* w in symmetric difference of D and D_*, that is $w \in (D - D_*) \cup (D_* - D)$.

By regarding OBDDs as DFAs and by applying Angluin's results [2, 4] to OBDDs, Gavaldà and Guijarro [12] have obtained the following results: (1) The OBDDs are polynomial time learnable with both membership and equivalence queries; (2) The OBDDs are not polynomial time learnable with equivalence queries alone.

It is known that if a concept class is not polynomial time PAC-learnable, then it is not polynomial time learnable with equivalence queries, but the converse does not hold [3]. Hence, the above theorem does not guarantee that OBDDs are not PAC-learnable. Also our results on MINIMUM CONSISTENT OBDD do not show that OBDDs are not PAC-learnable. However, our results do show that any efficient

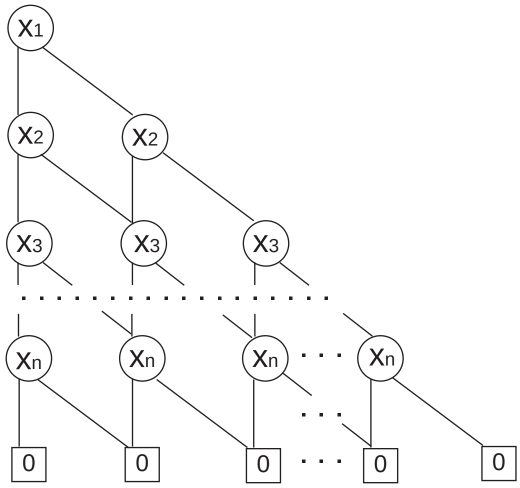

Figure 1: An initial internal OBDD

algorithm for learning OBDDs would have to produce very large hypothesis.

Our results suggest that OBDDs are unlikely to be polynomial time learnable in PAC-learning model. Then, in the reminder of this section, we investigate polynomial time learnable subclasses of OBDDs. Ergün *et al.* [10] have already shown that width-2 OBDDs are polynomial time PAC-learnable, although learning width-k OBDDs is as hard as learning DNF formulas for $k \geq 3$. The learnability of width-2 OBDDs follows that OBDDs representing parity functions and monomials are PAC-learnable.

As another learnable subclass, we can consider OBDDs representing symmetric functions. A function $f(x_1, \cdots, x_n)$ is called *symmetric* if f remains unchanged for every permutation of its variables $x_1, \cdots, x_n$, that is, the value of f depends only on the number of variables equal to 1. Note that n-ary symmetric functions are representable as OBDDs with $O(n^2)$ nodes. Hence, we can show that OBDDs representing symmetric functions are polynomial time learnable with equivalence queries as follows. This result is related to the paper [6].

Fix the order $x_1 < x_2 < \cdots < x_n$ of variables. Let D_* be a target OBDD and D be an internal OBDD in our learning algorithm. Assume that our learning algorithm returns a reduced OBDD D' of D as a hypothesis. The initial internal OBDD is described as Figure 1. While an hypothesis D' is not equivalent to D_*, our learning algorithm receives a counterexample $w \in \{0,1\}^n$ from equivalence queries, and exchanges the label 0 of the terminal node connected by w in D to 1.

It is $O(|D|)$ time to generate the initial internal OBDD. By the form of OBDD, the number of equivalence queries is at most $n+1$. It is $O(n)$ time to determine the terminal node connected by w. The time complexity of the reduction of D is $O(|D|)$ [19], where $|D|$ is the number of nodes in the OBDD D. Since $|D| \leq \frac{1}{2}(n+1)(n+2) = O(n^2)$, the total learning time is: $O(n^2)+(n+1)(O(n)+O(n^2)) = O(n^3)$. Hence, we obtain the following result:

Theorem 3. The OBDDs representing n-ary symmetric functions are learnable in time $O(n^3)$ with at most $n+1$ equivalence queries. Hence, they are also polynomial time PAC-learnable.

5 Conclusion

We have shown some intractabilities of the problem to find the minimum consistent OBDD from given examples, where the variable ordering is fixed. We first have shown that the problem is NP-hard, even the number of positive examples is only one. Secondly, we have proved that the problem is hard to approximate within a factor n^ϵ in polynomial time, where n is the number of variables and $\epsilon > 0$ is some constant.

Our results provide some interesting contrasts to the similar results on the problems to find the minimum consistent decision lists and decision trees due to Hancock *et al.* [14]. The first result is essentially the same to their result on hardness to find the shortest monomial. According to the second result, however, the difficulties of finding minimum decision lists and decision trees mainly relies on choosing the optimal ordering of the variables, while the variable ordering is fixed in our problem. Thus, the proof strategy to reduce from GRAPH COLORING completely differs from theirs, although they have also reduced from the same problem.

On the other hand, since the variable ordering is fixed in our setting, OBDDs can be regarded as a kind of DFA. Pitt and Warmuth [18] have shown that finding the minimum consistent DFA cannot be approximated within opt^k states, where opt is the minimum number of states and k is any constant greater than 1. They have used a special form of DFAs, so called *counter-like DFAs*, where the cyclic transitions plays an essential role to show their hardness. In contrast to their reduction, since OBDDs cannot have any cycles, our results would suggest that finding the minimum consistent DFA is still very difficult even when the DFA is restricted to be acyclic. In future works, we will deal with the problem of finding the minimum consistent acyclic

DFA directly.

Finally, we note that our hardness results *do not* imply that OBDDs are not polynomial time learnable in PAC-model, since an *Occam algorithm* [7] is allowed to produce an OBDD whose size is also dependent on the number of given examples. Our result gives a partial negative result on the polynomial time learnability, in the same sense of the results on DFAs [18], decision lists and decision trees [14]: Under the assumption P≠NP, there exists an $\epsilon > 0$ such that there is no polynomial time learning algorithm which always outputs an OBDD of size at most $s^{1+\epsilon}$, where s is the size of the target OBDD.

References

[1] Angluin, D.: *On the complexity of minimum inference of regular sets*, Information and Control **39**, 337–350, 1978.

[2] Angluin, D.: *Learning regular sets from queries and counterexamples*, Information and Computation **75**, 87–106, 1987.

[3] Angluin, D.: *Queries and concept learning*, Machine Learning **2**, 319–342, 1988.

[4] Angluin, D.: *Negative results for equivalence queries*, Machine Learning **5**, 121–150, 1990.

[5] Blum, A.: *New approximation algorithms for graph coloring*, Journal of the Association for Computing Machinery **41**, 470–516, 1994.

[6] Blum, A., Chalasani, P. and Jackson, J.: *On learning embedded symmetric concepts*, Proc. 6th International Workshop on Computational Learning Theory, 337–346, 1993.

[7] Board, R. and Pitt, L.: *On the necessity of Occam algorithms*, Theoretical Computer Science **100**, 157–184, 1992.

[8] Bolling, B. and Wegener, I.: *Improving the variable ordering of OBDDs is NP-complete*, Technical Report, Universität Dortmund, 1994.

[9] Bryant, R. E.: *Symbolic Boolean manipulation with ordered binary-decision diagrams*, ACM Computing Surveys **24**, 293–318, 1992.

[10] Ergün, F., Kumar, S. R. and Rubinfeld, R.: *On learning bounded-width branching programs*, Proc. 8th International Workshop on Computational Learning Theory, 361–368, 1995.

[11] Garey, M. and Johnson, D. S.: *Computers and intractability: A guide to the theory of NP-completeness*, W. H. Freeman and Company, 1978.

[12] Gavaldà, R. and Guijarro, D.: *Learning ordered binary decision diagrams*, Proc. 6th International Workshop on Algorithmic Learning Theory, 228–238, LNAI **997**, 1995.

[13] Gold, E. M.: *Complexity of automaton identification from given data*, Information and Control **37**, 302–320, 1978.

[14] Hancock, T., Jiang, T., Li, M. and Tromp, J.: *Lower bounds on learning decision lists and trees*, draft, 1996.

[15] Li, M. and Vazirani, U.: *On the learnability of finite automata*, Proc. 1988 Workshop on Computational Learning Theory, 359–370, 1988.

[16] Lund, C. and Yannakakis, M.: *On the hardness of approximating minimization problems*, Proc. 25th Annual ACM Symposium on Theory of Computing, 286–293, 1993.

[17] Pitt. L and Valiant, L. G.: *Computational limitation on learning from examples*, Journal of the Association for Computing Machinery **35**, 965–984, 1988.

[18] Pitt, L. and Warmuth, M. K.: *The minimum consistent DFA problem cannot be approximated within any polynomial*, Journal of the Association for Computing Machinery **40**, 95–142, 1993.

[19] Sieling, D. and Wegener, I.: *Reduction of OBDDs in linear time*, Information Processing Letter **48**, 139–144, 1993.

[20] Valiant, L. G.: *A theory of the learnable*, Communications of ACM **27**, 1134–1142, 1984.