ELSEVIER

Information Processing Letters 66 (1998) 165–170

Information Processing Letters

On the hardness of approximating the minimum consistent acyclic DFA and decision diagram

Shinichi Shimozono [a,*], Kouichi Hirata [a,1], Ayumi Shinohara [b,2]

[a] Department of Artificial Intelligence, Kyushu Institute of Technology, Iizuka 820-8502, Japan

[b] Department of Informatics, Kyushu University, Fukuoka 812-8581, Japan

Received 15 April 1997; received in revised form 26 March 1998
Communicated by T. Asano

Abstract

We show that the problems to find the smallest acyclic DFA and OBDD with a fixed order of variables that are consistent with given sets of positive and negative examples are not approximable in polynomial time within worst case factors $n^{1/28}$ and $n^{1/21}$, respectively, with the input size n unless P = NP. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Finite state machine minimization; Approximation; Combinatorial problems; Computational complexity; Data structures

1. Introduction

The minimum consistency problem for deterministic finite automata (DFAs) is, given two sets of strings as positive examples and negative examples, to find a DFA with the minimum number of states that accepts all the positive examples and no negative examples. This problem captures the computational complexity of learning a DFA from a sample of a language and the state minimization of a finite state machine specified by an incomplete input–output table. The corresponding problems for important and various classes of representations of languages have been intensively investigated [9,12]. If a minimum consistency problem for a class is intractable, then an efficient algorithm that produces a consistent representation whose size is not too large compared to the smallest one is important in not only practical fields but also learning theory, since it immediately implies that the class is polynomial-time learnable in Valiant's PAC-learning model [4].

For a class of DFAs, the minimum consistency problem was first shown to be NP-hard by Gold [8] and Angluin [1]. Li and Vazirani [11], and Simon [15] strengthened this hardness by proving that there exists a constant $c > 1$ such that no polynomial-time algorithm can guarantee a consistent DFA whose size is at most c times the minimum, unless P = NP. Pitt and Warmuth presented in [13] a remarkable result: they showed that the problem cannot be approximated in polynomial time within ratios both opt^k for any constant $k > 0$ and $n^{1/14}$, unless P = NP, where opt is the possible minimum size and n is the size of an input. Interestingly, a DFA which must be constructed in their proof is formed from simply a loop of transitions whose periodical length plays an essential role to accept no negative examples.

* Corresponding author. Email: sin@ai.kyutech.ac.jp.
[1] Email: hirata@ai.kyutech.ac.jp.
[2] Email: ayumi@i.kyushu-u.ac.jp.

0020-0190/98/$19.00 © 1998 Elsevier Science B.V. All rights reserved.
PII: S0020-0190(98)00065-9

An acyclic DFA is restricted to having no cyclic transitions, and thus is applied to specify a finite set of strings. Branching programs and ordered binary decision diagrams (OBDDs), which are representations of Boolean functions in various practical fields [5,7], are tied with acyclic DFAs in computational complexity. For both DFAs and OBDDs, the identification from a completely specified sample is well studied (e.g., [3, 5,17]). On the other hand, there are only a few results dealing with the identification of OBDDs from incompletely specified samples [6,14,16]. Although the proof in [1] by Angluin indirectly indicates the NP-hardness of the minimum consistency problem for acyclic DFAs, further approximability and nonapproximability were not known.

In this paper, we show that even the identification of an approximately small acyclic DFA is intractable. We prove that the minimum consistency problem for acyclic DFAs cannot be approximated within a ratio $n^{1/28}$ in polynomial time unless P = NP, where n is the number of symbols in a given sample. By a modified version of the proof, we also show a tighter bound $n^{1/21}$ for the minimum consistency problem for OBDDs with respect to a fixed order of variables. Our result suggests that both the classes of acyclic DFAs and OBDDs are unlikely to be PAC-learnable.

2. Acyclic DFA and minimum consistency problem

As far as we are concerned in this paper, we only deal with OBDDs whose orders of variables are specified in samples as the orders of symbols.

A set $\Sigma = \{0, 1\}$ is the alphabet throughout this paper. The set of all strings formed from the symbols of Σ is denoted by Σ^*, and the set of all stings of fixed length $n > 0$ is denoted by Σ^n. Let s and t be strings in Σ^*. Then $|s|$ denotes the length of s, and both $t \cdot s$ and ts denote the concatenation of t and s. The unique empty string in Σ^* of length 0 is denoted by λ.

Definition 1. A *deterministic finite automaton (DFA)* M is a quadruple (Q, δ, q_0, F), where Q is a finite set of *states*, δ is a partial function from $Q \times \Sigma$ to Q, $q_0 \in Q$ is the *initial state* and $F \subseteq Q$ is the set of *accepting states*. The function δ is called the *transition function* and is extended in a straightforward way to the partial mapping from $Q \times \Sigma^*$ to Q. A DFA M *accepts* a string $s \in \Sigma^*$ if and only if $\delta(q_0, s)$ is defined and in F, and M *rejects* s if and only if M does not accept s. The *size* of DFA M, denoted by $|M|$, refers to the number of states of M.

An *acyclic DFA (ADFA)* $M = (Q, \delta, q_0, F)$ is a DFA which has no cyclic transitions, i.e., for any $q \in Q$ and $s \in \Sigma^* - \{\lambda\}$, if the value $\delta(q, s)$ is defined then $\delta(q, s) \neq q$. A state q of M is said to be *redundant* if the two transitions from q are defined and direct the same state $\delta(q, 0) = \delta(q, 1)$. An *OBDD* is an acyclic DFA whose accepting state is unique and size refers to the number of states that are not redundant.

An ADFA $M = (Q, \delta, q_0, F)$ is implicitly identified with a directed acyclic graph, where each node is a state and each edge is a transition labeled with a symbol in Σ as either a *0-transition* or a *1-transition*. A *computation path* with a string s is the path on the graph representation specified with the transitions $\delta(q_0, s)$, if $\delta(q_0, s)$ is defined. The *distance* of a state q is the length of a shortest string with which the computation path reaches to q.

For a DFA M, $L(M) \subseteq \Sigma^*$ denotes the set of strings accepted by M. Note that for every language L of a finite number of strings there is an ADFA M such that $L = L(M)$. A *sample* $S = \langle S_1, S_0\rangle$ is a pair of disjoint finite sets S_1 and S_0 of strings, whose elements are called *positive examples* and *negative examples*, respectively. An ADFA M is *consistent* with $S = \langle S_1, S_0\rangle$ if M accepts all positive examples in S_1 and rejects all negative examples in S_0. The *size* of a sample S is the total length of examples in S.

Then, the minimum consistency problem for ADFAs is defined as follows:

Definition 2 (MINIMUM CONSISTENT ACYCLIC DFA (MIN CON ADFA)).

(i) an instance is a sample $S = \langle S_1, S_0\rangle$,

(ii) a solution is an ADFA M consistent with the sample S, and

(iii) the measure of a solution is the size $|M|$ of the ADFA M.

The goal of this problem is to find a solution whose measure is minimum.

MIN CON OBDD is defined in the same way.

Let Π be a minimization problem, and x an instance of Π. A *polynomial-time approximation algorithm* A for Π is a polynomial-time algorithm that produces a solution $A(x)$ for any instance x of Π. The *performance ratio* $R(x,s)$ of a solution s of x is the ratio of the measure of s to the measure of an optimum solution of x. We say that an algorithm A *approximates* the problem Π within a ratio r if, for any instance x of Π, A always produces a solution $A(x)$ such that $R(x, A(x)) \leqslant r$.

3. Nonapproximability of the minimum consistency problems

The next lemma shows a simple nonapproximability of MIN CON OBDD, which is due to the hardness in choosing appropriate nodes as redundant nodes. Note that the corresponding subproblem of MIN CON ADFA is trivially solvable.

Lemma 3. *Even if the number of positive examples in a sample is one,* MIN CON OBDD *cannot be approximated in polynomial time within any constant unless* P = NP. *If* NP $\not\subseteq$ DTIME$[n^{poly \log n}]$ *is assumed, then the problem cannot be approximated within a ratio* $\frac{1}{4}\log\gamma$ *in polynomial time, where* γ *is the length of an example.*

This follows from both the nonapproximability of the minimum consistency problem for monomials [9] and the fact that an OBDD consistent with such a sample can be formed from the unique path reaching to the accepting state, i.e., equivalent to a monomial.

The following theorem shows the hardness to deal with the both classes of ADFAs and OBDDs.

Theorem 4. *For any* $\varepsilon > 0$, MIN CON ADFA *cannot be approximated in polynomial time within a ratio* $n^{1/28-\varepsilon}$, *and* MIN CON OBDD *within* $n^{1/21-\varepsilon}$, *unless* P = NP, *where* n *is the size of a given sample.*

Proof. We show an approximation-preserving reduction from MIN GRAPH COLORING (CHROMATIC NUMBER) and its variant that prove this theorem. The structure of an ADFA which would be constructed from a sample in the reduction is essentially equivalent to that of OBDDs devised independently in [14, 10].

Let $G = (V, E)$ be an undirected graph with m nodes in $V = \{1, \ldots, m\}$. A *coloring* of G is a mapping f from V to a set of positive integers such that, for any edge $(i, j) \in E$, $f(i) \neq f(j)$ holds. A k-coloring is a coloring by k integers. The problem MIN GRAPH COLORING is, given a graph $G = (V, E)$, to find a k-coloring of G with the minimum number k.

For each node $i \in V$ of a graph $G = (V, E)$, we associate a *node-string* $v_i = 1^{i-1}01^{m-i}$ and an *adjacency-set* $adj(i) = \{v_j \mid (i, j) \in E\}$. Also, for a subset $V' \subseteq V$ of nodes, we define $P_G[V'] = \bigcup_{i\in V'} adj(i)$ and $N_G[V'] = \{v_i \mid i \in V'\}$. Note that the node-string v_i is not in $adj(i)$ for any $i \in V$, and a subset $V' \subseteq V$ is an independent set of G if and only if $P_G[V']$ and $N_G[V']$ are disjoint. A k-*partition* $V_1, \ldots, V_k$ of V is a sequence of k mutually disjoint subsets of V whose union equals V. We identify a k-partition of V with a mapping $f: V \to \{1, \ldots, k\}$ in the obvious way. It is immediate that a k-partition of V is a k-coloring of G if and only if, for any $1 \leqslant i \leqslant k$, $P_G[V_i]$ and $N_G[V_i]$ are disjoint.

The translation procedure from a graph G to the corresponding sample S_G is given as follows. For a string $s \in \Sigma^*$, we denote by $pre(s, h)$ the substring formed from the first $h < |s|$ symbols of s. Given a graph $G = (V, E)$ with m nodes, we define $l = m^2$ and construct the sets T, P, N and T' of examples as follows:

(i) $P = \{\langle i\rangle 1^l \cdot v_j \mid i \in V \text{ and } v_j \in adj(i)\}$,
(ii) $N = \{\langle i\rangle 1^l \cdot v_i \mid i \in V\}$,
(iii) $T = \{\langle i\rangle 1^l \cdot 1^m \mid i \in V\}$,
(iv) $T' = \{pre(s, h) \mid s \in T \text{ and } \lceil\log m\rceil + l + 1 \leqslant h \leqslant \lceil\log m\rceil + l + m - 1\}$,

where $\langle i\rangle$ is the binary representation of $i - 1$ in $\lceil\log m\rceil$ bits. Then, the sample $S_G = \langle S_1, S_0\rangle$ is given by $S_1 = P \cup T$ and $S_0 = N \cup T'$. Note that S_1 and S_0 are disjoint since $\langle i\rangle 1^l$ is unique to every node $i \in V$, and $P_G[\{i\}]$ and $N_G[\{i\}]$ are disjoint.

Now we define for a sample given as above a class of ADFAs in which a structure representing a coloring of a graph exists. A *partitioning-ADFA* M *for* S_G is an ADFA that is consistent with S_G and has only states included in the computation paths with examples in T. A state of M whose distance is $\lceil\log m\rceil + l$ is said to be a *partition-identity*. A chain of l states with 1-transitions proceeding a partition-identity is called a *conduit*. We require that in a partitioning-ADFA no two distinct conduits succeed to the same partition-

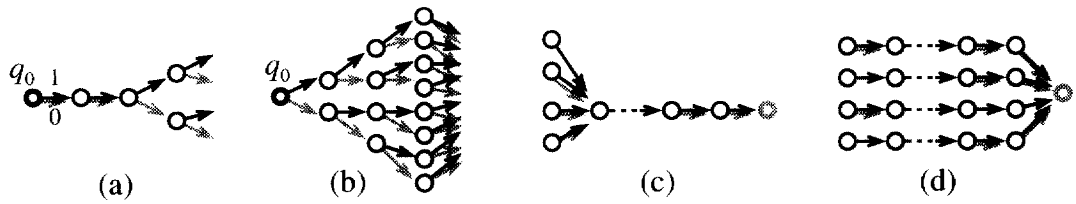

Fig. 1. Examples giving the lower and upper bounds of the number of classifier-states (a) and (b), and of rejector-states (c) and (d) for $\lceil \log m \rceil = 4$ and $k = 4$.

identity. A state whose distance is at most $\lceil \log m \rceil - 1$ is called a *classifier-state*, and is at least $\lceil \log m \rceil + l$ except the accepting state is called a *rejector-state*. Note that T' forces the distance of states reachable by examples in S_1 to be the same. Thus we can suppose that an accepting state of M is unique, and the distance of every state equals the length of any computation path to it.

Lemma 5. *Any ADFA consistent with S_G can be reduced to a partitioning-ADFA M_k for some integer $k > 0$ which has exactly k distinct partition-identities and can be translated into a k-coloring of G.*

Proof. To obtain a partitioning-ADFA, we apply the following log-space procedure:

(a) From every classifier-state and state forming a conduit, remove all the transitions not traversed by a computation path of a positive example in S_G.

(b) From every rejector-state, remove the 0-transition if no computation paths of positive examples traverse it. Otherwise, direct the 0-transition to the state indicated by the 1-transition. This works correctly since every node-string contains just one '0', which is the unique chance in rejector-states to distinguish an example in P from other examples.

(c) Remove all the states (and transitions from them) that has no incoming transitions, and repeat it until there is no such unnecessary state. Finally, for each partition-identity, merge all conduits proceeding to it into one.

Let q^j for $1 \leqslant j \leqslant k$ be a partition-identity of M_k, and associate it with the set

$$V^j = \{ i \in V \mid \delta(q_0, \langle i \rangle 1^l) = q^j \}$$

where q_0 is the initial state of M_k. Then the sequence $V^1, \ldots, V^k$ represents a k-coloring of G, since (a) a computation path with any identity $\langle i \rangle 1^l$ for $i \in V$ reaches to a partition-identity and (b) M_k is consistent with S_G if and only if

$$P_G[V^i] \cap N_G[V^i] = \emptyset \quad \text{for all } 1 \leqslant i \leqslant k. \qquad \square$$

Lemma 6. *Let M_k and $M_{k'}$ be partitioning-ADFAs for S_G with k and k' partition-identities, respectively. Then, if $k > k'$,*

$$|M_k| > |M_{k'}| \quad \textit{and} \quad \frac{1}{2} \cdot \frac{k}{k'} \leqslant |M_k|/|M_{k'}|$$

hold.

Proof. We denote the number of the classifier-states and the rejector-states in M_k by *classifier*(M_k) and *rejector*(M_k), respectively. Then the following inequalities hold (see Fig. 1):

$$\lceil \log m \rceil - \lceil \log k \rceil + k - 1 \leqslant \mathit{classifier}(M_k) \leqslant 2m - 1,$$
$$m - 1 + k \leqslant \mathit{rejector}(M_k) \leqslant km.$$

With $lk = m^2 k$ states of the k conduits and one accepting-state, the lower bound $LB(m,k)$ and the upper bound $UB(m,k)$ of the size of M_k are defined as follows:

$$LB(m,k) = \lceil \log m \rceil - \lceil \log k \rceil + m^2 k + 2k + m - 1,$$
$$UB(m,k) = m^2 k + m(k+2).$$

Therefore,

$$LB(m,k) > UB(m,k') \quad \text{and}$$
$$\frac{k}{2k'} \leqslant \frac{LB(m,k)}{UB(m,k')} \leqslant |M_k|/|M_{k'}| \quad \text{for any } k > k'. \qquad \square$$

Now we are ready to complete our proof of the theorem for ADFA. Let k^* be the possible minimum number of colors for a coloring of G, and let M^* be the smallest ADFA consistent with S_G. Then, for a k-coloring f, $R(G, f) = k/k^*$, and for an ADFA M,

$$R(S_G, M) = |M|/|M^*| \geqslant |M|/UB(m, k^*).$$

By Lemmas 5, 6 and the construction of S_G, we can see that there is an approximation-preserving reduction from MIN GRAPH COLORING to MIN CON ADFA by two log-space computable functions ρ and τ with the following properties:

(i) Any graph G with m nodes is translated by ρ into a sample S_G with at most $2m^2 + 2m$ examples of each length at most $m^2 + 2m$.

(ii) Any ADFA M that is consistent with S_G is translated by τ into a coloring f_M for G.

(iii) Between the translations (i) and (ii) the inequality $2R(S_G, M) > R(G, f_M)$ holds.

It is known from [2] that no polynomial-time algorithm approximates MIN GRAPH COLORING within a ratio $m^{1/7-c}$ for any $c > 0$ unless P = NP. The reduction pair (ρ, τ) guarantees that the size n of a sample S_G for G with m nodes is $O(m^4)$. It implies that for any $\varepsilon > 0$ and for any polynomial-time algorithm there exists an instance such that

$$n^{1/28-\varepsilon} < \frac{1}{2} R(G, f_M) < R(S_G, M).$$

The reduction to MIN CON OBDD can be given as a variant of that to MIN CON ADFA, where the major difference is that the estimation of the number of states must regard only non-redundant states. By the definition of OBDDs, we can omit examples in T' restricting the distance of accepting states, and have to add a new set of negative examples

$$T'' = \{\langle i\rangle 1^{j-1} 0 1^{l-j} \cdot 1^m \mid 1 \leqslant j \leqslant l\}$$

to prevent the states in conduits from being redundant. Note that m non-redundant states are always required in rejector-states. Lemma 6 holds for examples with the shorter middle part $l = 3m$ with the bounds

$$LB(m, k) = k - 1 + m(3k + 1),$$
$$UB(m, k) = 3m(k + 1) - 1.$$

Thus n is $O(m^3)$ for OBDDs, and it leads to the ratio $n^{1/21-\varepsilon}$ for any $\varepsilon > 0$. □

If we accept a stronger assumption on the complexity classes, then we can raise the ratios by tighter bounds for MIN GRAPH COLORING in [2]:

Corollary 7. MIN CON ADFA *cannot be approximated within* $n^{1/20-\varepsilon}$ *and* MIN CON OBDD *within* $n^{1/15-\varepsilon}$ *for any* $\varepsilon > 0$ *in polynomial time, if* coRP $\neq$ NP *is assumed.*

4. Conclusion

We have shown the hardness of approximating the problems to find an acyclic DFA and an OBDD that are smallest and consistent with a given sample of a language. The nonapproximability obtained in the theorem would agree with an intuition that the minimum consistency problem for acyclic DFAs should not be harder than that for DFAs. Our result shows some interesting contrasts to the corresponding results on the problems to find the minimum consistent decision lists and decision trees due to Hancock et al. [9]. The difficulties of finding minimum decision lists and decision trees mainly rely on choosing the optimal order of variables in their results, while the order of variables is fixed in our problem.

It is known in [3] that, given a completely specified sample of a Boolean function, finding an optimal order of variables with which an OBDD is minimized is intractable. Our result itself does not give the hardness if a variable order can be chosen arbitrary. In the sense of the OBDD research, we still have basic open issues in dealing with incompletely specified samples.

Acknowledgments

Authors are very grateful to the anonymous referees for the painstaking attention to detail and many helpful comments. We thank Hiroki Arimura for the valuable discussions.

References

[1] D. Angluin, On the complexity of minimum inference of regular sets, Inform. and Control 39 (1978) 337–350.

[2] M. Bellare, O. Goldreich, M. Sudan, Free bits, PCPs and non-approximability—towards tight results, in: Proc. 36th Annual Symposium on Foundations of Computer Science, 1995, pp. 422–431.

[3] B. Bollig, I. Wegener, Improving the variable ordering of OBDDs is NP-complete, IEEE Trans. Comput. 45 (1996) 993–1002.

[4] A. Blumer, A. Ehrenfeucht, D. Haussler, M.K. Warmuth, Occam's razor, Inform. Process. Lett. 24 (1987) 377–380.
[5] R.E. Bryant, Symbolic Boolean manipulation with ordered binary-decision diagrams, ACM Comput. Surveys 24 (1992) 293–318.
[6] R. Gavaldá, D. Guijarro, Learning ordered binary decision diagrams, in: Proc. 6th International Workshop on Algorithmic Learning Theory, Springer, Berlin, 1995, pp. 228–238.
[7] J. Gergov, C. Meinel, On the complexity of analysis and manipulation of Boolean functions in terms of decision graphs, Inform. Process. Lett. 50 (1994) 317–322.
[8] E.M. Gold, Complexity of automaton identification from given data, Inform. and Control 37 (1978) 302–320.
[9] T. Hancock, T. Jiang, M. Li, J. Tromp, Lower bounds on learning decision lists and trees, in: Proc. 12th Annual Symposium on Theoretical Aspects of Computer Science, Springer, Berlin, 1995, pp. 527–538.
[10] K. Hirata, S. Shimozono, A. Shinohara, On the hardness of approximating the minimum consistent OBDD problem, in: Proc. Fifth Scandinavian Workshop on Algorithm Theory, Springer, Berlin, 1996, pp. 112–123.
[11] M. Li, U. Vazirani, On the learnability of finite automata, in: Proc. 1988 Workshop on Computational Learning Theory, Morgan Kaufmann, San Mateo, CA, 1988, pp. 359–370.
[12] L. Pitt, L.G. Valiant, Computational limitations on learning from examples, J. ACM 35 (1988) 965–984.
[13] L. Pitt, M.K. Warmuth, The minimum consistent DFA problem cannot be approximated within any polynomial, J. ACM 40 (1993) 95–142.
[14] M. Sauerhoff, I. Wegener, On the complexity of minimizing the OBDD size for incompletely specified functions, IEEE Trans. C.A.D. Integ. Circuits and Syst. 15 (1996) 1435–1437.
[15] H.U. Simon, On approximate solutions for combinatorial optimization problems, SIAM J. Discrete Math. 3 (1990) 294–310.
[16] Y. Takenaga, S. Yajima, Hardness of identifying the minimum ordered binary decision diagram, Technical Report, Department of Information Science, Kyoto University, 1993.
[17] B.A. Trakhtenbrot, Y.M. Barzdin, Finite Automata, North-Holland, Amsterdam, 1973.