九州大学学術情報リポジトリ
Kyushu University Institutional Repository

Unions of a Bounded Number of Tree Pattern Languages Are Hard To Learn

Arimura, Hiroki
Department of Artificial Intelligence Kyushu Institute of Technology

https://hdl.handle.net/2324/3180

出版情報：RIFIS Technical Report. 81, 1994-02-13. Research Institute of Fundamental Information Science, Kyushu University
バージョン：
権利関係：

RIFIS-TR-CS-81

RIFIS Technical Report

Unions of a Bounded Number of Tree Pattern Languages Are Hard To Learn

Hiroki Arimura

February 13, 1994

Research Institute of Fundamental Information Science
Kyushu University 33
Fukuoka 812, Japan
E-mail: arim@ai.kyutech.ac.jp Phone: 0948-29-7638

Unions of a Bounded Number of Tree Pattern Languages Are Hard to Learn

Hiroki Arimura
Department of Artificial Intelligence
Kyushu Institute of Technology
Kawazu 680-4, Iizuka 820, JAPAN

Abstract

In this paper, we show that for every positive integer k, the class $\mathcal{TPL}^k$ of unions of at most k tree pattern languages is not learnable unless $P = NP$ in the framework of PAC-learnability.

1 Preliminary

In this section, we give a brief summary on PAC-leanability. First, we give the following basic definitions concerning to PAC-learnability according to Natarajan [4] and Blumer *et al.* [2]. Let Σ be an alphabet. A *concept* is a set $c \subseteq \Sigma^*$ of strings and a *concept class* is a nonempty set $\mathcal{C}$ of concepts. An *example* of c is a pair $\langle x, d \rangle$, where $d = 1$ if $x \in c$ and $d = 0$ otherwise. A concept class $\mathcal{C}$ is *PAC-learnable* if there is an algorithm $\mathcal{A}$ which satisfies the following conditions:

1. $\mathcal{A}$ runs in polynomial time with respect to the length of the input.
2. There is a polynomial $p(\cdot, \cdot, \cdot)$ such that for any integer $n \geq 0$, any concept $c \in \mathcal{C}$, any real number ε, δ $(0 \leq \varepsilon, \delta \leq 1)$, and any probability distribution P over $\Sigma^{\leq n}$, if $\mathcal{A}$ receives $p(n, 1/\varepsilon, 1/\delta)$ examples which are generated randomly according to P, then $\mathcal{A}$ outputs a representation of a hypothesis h such that $P(c \oplus h) \leq \varepsilon$ with probability at least $1 - \delta$.

A concept class $\mathcal{C}$ has *polynomial dimension* if there is a polynomial $p(\cdot)$ such that $|\mathcal{C}_n| \leq 2^{p(n)}$, where $\mathcal{C}_n = \{ c \cap \Sigma^{\leq n} \mid c \in \mathcal{C} \}$ for all $n \geq 0$. A *randomized polynomial time hypothesis finder* is a randomized polynomial time algorithm that, given disjoint finite sets P and N of strings such that $P \cap N = \emptyset$, computes the representation of a concept $c \in \mathcal{C}$ *consistent* with P and N, that is, $P \subseteq c$ and $N \cap c = \emptyset$.

Theorem 1 (Natarajan [4] and Blumer et al. [2]) *If $\mathcal{C}$ is PAC-learnable then there is a randomized polynomial time hypothesis finder for $\mathcal{C}$.*

A *tree pattern* over Σ is a first-order term p over Σ, and the *language defined by* p is the set $L(p)$ of all ground instances of p. We say p is *regular* if any variable occurs at most once in p. A set L of ground trees is called a *(regular) tree pattern language* if $L = L(p)$ for some (regular) p. The class to be investigated in this report is the class $\mathcal{TPL}^k$ of unions of at most k tree pattern languages [1]. We write $p \leq q$ if $p = q\theta$ for some θ.

2 The class $\mathcal{TPL}^k$ is hard to learn

The problem here is the consistency problem for the class $\mathcal{C}$ stated as

> **CONSISTENCY($\mathcal{C}$)**
> **Instance:** Disjoint finite sets $Pos, Neg \subseteq \Sigma^*$
> **Question:** Is there a concept $L \in \mathcal{C}$ consistent with positive examples in Pos and negative examples in Neg, that is, $Pos \subseteq L$ and $Neg \cap L = \emptyset$.

We use a reduction from the following theorem to the consistency problem for $\mathcal{TPL}^k$ to show the hardness of PAC-learning of the class $\mathcal{TPL}^k$.

Theorem 2 (Theorem 3.2 of [5]) *For all $k \geq 2$, the consistency problem for k-term DNF is* NP*-complete.*

Now, we give the main theorem in this paper.

Theorem 3 *For every $k \geq 2$, the consistency problem for $\mathcal{TPL}^k$ is* NP*-complete.*

Proof. First we see the problem is in NP. If $w \leq p$ then the size of p is less than or equal to the size of w. Thus, there is a nondeterministic algorithm that find an answer in polynomial time by guessing at most k tree patterns p with $L(p) \cap P \neq \emptyset$ and checking their consistency.

We next give a polynomial time reduction from the consistency problem for k-term DNF to the problem. Let $U = \{u_1, \ldots, u_n\}$ be variables, P and N be a set of positive and negative examples, respectively, for the consistency problem for k-term DNF. We assume $P \neq \emptyset$. We use trees over $\Sigma = \{\mathbf{a}, \mathbf{f}, \mathbf{g}\}$ to represent assignments in P and N, where $\mathbf{a}$ is of arity zero, $\mathbf{f}$ and $\mathbf{g}$ are of arity n function symbols. We denote by $\mathbf{1}_0$ the function symbol $\mathbf{a}$, and for any $1 \leq i \leq n$, we recursively define the term $\mathbf{1}_i$ as $\mathbf{1}_i = \mathbf{f}(\mathbf{a}, \ldots, \mathbf{a}, \mathbf{1}_{i-1})$ that the argument only at n-th position is set to be $\mathbf{1}_{i-1}$ and other arguments are set to be $\mathbf{a}$. For any $1 \leq i \leq n$, we denote by $\mathbf{0}_i$ the term $\mathbf{g}(\mathbf{a}, \ldots, \mathbf{a}, \mathbf{1}_{i-1})$ that the argument only at n-th position is set to be $\mathbf{1}_{i-1}$ and the others are set to $\mathbf{a}$. Note that any function symbols at the same position of $\mathbf{1}_i$ and $\mathbf{0}_i$ are the same symbol, except those at the root position. Given an assignment $A = \hat{u}_1, \ldots, \hat{u}_n \in \{0,1\}^n$, we define the tree $\Phi(A)$ over Σ as

$$\Phi(A) = \mathbf{f}(a_1, \ldots, a_n), \text{ where } a_i = \begin{cases} \mathbf{0}_i & \text{, if } \hat{u}_i = 0, \\ \mathbf{1}_i & \text{, if } \hat{u}_i = 1. \end{cases}$$

Then, we define positive examples and negative examples for the consistency problem for $\mathcal{TPL}^k$ as $\Phi(P)$ and $\Phi(N)$, respectively.

First, we assume that there is a set $Q = \{q_1, \ldots, q_l\}$ $(l \leq k)$ of tree patterns over Σ consistent with $\Phi(P)$ and $\Phi(N)$. Without loss of generality, we can assume that $\Phi(P) \cap L(q_j) = \emptyset$ for any $1 \leq j \leq l$. By the following claim, we can assume that any tree pattern in Q is regular.

Claim. Let D be any subset of $\Phi(\{0,1\}^n)$. Then, for any tree pattern $q \in Q$, there is a regular tree pattern q' such that $D \cap L(q) = D \cap L(q')$.

Proof for Claim. Assume that a variable x occurs more than once in q. Let α and β be the postitions that x occurs. For a position α and a tree pattern p, we denote by q/α the subterm of p occurring at α. Then, if q matches some $w \in D$, that is, $w \leq q$ then the subterms w/α and w/β must be identical because both of q/α and q/β are x. By construction of D, any two arguments a_i and a_j of w are distinct for $i \neq j$. Thus, if $w \leq q$ then one of α or β, say α, must be an properly internal position of some argument of q. Let d be w/α. By construction of D, the subterm u/α is d for every u in D. Let q' be the term $q' = q\{x := d\}$. Then, we know

$u \in L(q)$ iff $u \in L(q')$ for every u in D. In this way, we can eliminate all variables in q that occur more than once. (*End of the proof for Claim*)

Let $F = T_1 + \cdots + T_1$ be a k-term DNF to be consistent with P and N, where each term $T_j = L_1 \cdots L_n$ corresponding to $q_j = \mathbf{f}(t_1, \ldots, t_n)$ is defined as for any $1 \leq i \leq n$,

$$L_i = \begin{cases} u_i & \text{, if } \mathbf{0}_i \not\leq t_i \text{ and } \mathbf{1}_i \leq t_i, \\ \bar{u}_i & \text{, if } \mathbf{0}_i \leq t_i \text{ and } \mathbf{1}_i \not\leq t_i, \\ 1 & \text{, if } \mathbf{0}_i \leq t_i \text{ and } \mathbf{1}_i \leq t_i. \end{cases}$$

By the assumption, either $\mathbf{0}_i \leq t_i$ or $\mathbf{1}_i \leq t_i$ for any $1 \leq i \leq n$. Thus, the above L_i is well defined. Then, we consider $T_j = L_1 \cdots L_n$. Assume T_j is true under a truth assignment $A = \hat{u}_1, \ldots, \hat{u}_n \in \{0,1\}^n$. Since each L_i is true, if $\hat{u}_i = 1$, then L_i is either 1 or u_i. Thus, $\mathbf{1}_i \leq t_i$. In this case, $a_i \leq t_i$ because $a_i = \mathbf{1}_i$. If $\hat{u}_i = 0$, similar argument also shows that $a_i \leq t_i$. Since q_j is regular, $\Phi(A) \leq q_j$

Conversely, we assume that $\Phi(A) \leq q_j$ for a truth assignment $A = \hat{u}_1, \ldots, \hat{u}_n \in \{0,1\}^n$. Let $\Phi(A) = \mathbf{f}(a_1, \ldots, a_n)$. Since $\Phi(A) \leq q_j$, $a_i \leq t_i$ for any $1 \leq i \leq n$. Since a_i is either $\mathbf{0}_i$ or $\mathbf{1}_i$, we can easily see that each L_i is true under A in both cases and that T_j is true under A. Thus, $\Phi(A) \leq q_j$ iff T_j is true under A for any $A \in \{0,1\}^n$. Hence, F is a consistent k-term DNF with P and N.

Next, we assume that there is a k-term DNF $F = T_1 + \cdots + T_l$ $(l \leq k)$ consistent with P and N. Without loss of generality, we can assume that T_j does not contain both of u_i and $\bar{u}_i$ for any $1 \leq i \leq n$. Then, we construct a set $Q = \{q_1, \ldots, q_l\}$ of tree patterns over Σ. For T_j, we put $q_j = \mathbf{f}(t_1, \ldots, t_n)$ as for any $1 \leq i \leq n$,

$$t_i = \begin{cases} \mathbf{1}_i & \text{, if } T_j \text{ contains } u_i, \text{ but not } \bar{u}_i, \\ \mathbf{0}_i & \text{, if } T_j \text{ contains } \bar{u}_i, \text{ but not } u_i, \\ x_i & \text{, if } T_j \text{ does not contain both of } u_i \text{ and } \bar{u}_i, \end{cases}$$

where $x_1, \ldots, x_n$ are mutually distinct variables. Clearly, T_j is true under A iff $\Phi(A) \leq q_j$ for any $A \in \{0,1\}^n$. Hence, Q is a set of at most k tree patterns consistent with $\Phi(P)$ and $\Phi(N)$. Finally, it is not difficult to see that the reduction can be computed in deterministic log space. □

Example 1 For positive examples $\{0010, 0110, 1001\}$ and negative examples $\{0000, 0111, 1010\}$, we construct positive examples $\{\mathbf{f}(\mathbf{0}_1, \mathbf{0}_2, \mathbf{1}_3, \mathbf{0}_4), \mathbf{f}(\mathbf{0}_1, \mathbf{1}_2, \mathbf{1}_3, \mathbf{0}_4), \mathbf{f}(\mathbf{1}_1, \mathbf{0}_2, \mathbf{0}_3, \mathbf{1}_4)\}$ and negative examples $\{\mathbf{f}(\mathbf{0}_1, \mathbf{0}_2, \mathbf{0}_3, \mathbf{0}_4), \mathbf{f}(\mathbf{0}_1, \mathbf{1}_2, \mathbf{1}_3, \mathbf{1}_4), \mathbf{f}(\mathbf{1}_1, \mathbf{0}_2, \mathbf{1}_3, \mathbf{0}_4)\}$ over $\Sigma = \{\mathbf{a}, \mathbf{f}, \mathbf{g}\}$. Then, there is a set $\{\mathbf{f}(\mathbf{1}_1, \mathbf{0}_2, x_3, \mathbf{1}_4), \mathbf{f}(\mathbf{0}_1, x_2, \mathbf{1}_3, \mathbf{0}_4)\}$ of two tree patterns consistent with these examples, which corresponds to the 2-term DNF $u_1 \cdot \bar{u}_2 \cdot u_4 + \bar{u}_1 \cdot u_3 \cdot \bar{u}_4$.

For the class $\mathcal{TPL}^k_{reg}$ of unions of at most k regular tree pattern languages. we know the result is also true from the proof of Theorem 3. In this case, we can prove the result even for $\sharp\Sigma = 2$ by using a simpler reduction, while Theorem 3 needs $\sharp\Sigma \geq 3$. The reduction is constructed in a similar way as in the proof of Theorem 3 except that $\Sigma = \{\mathbf{a}, \mathbf{f}\}$, $\mathbf{0}_1 = \cdots = \mathbf{0}_n = \mathbf{a}$, and $\mathbf{1}_1 = \cdots = \mathbf{1}_n = \mathbf{f}(\mathbf{a}, \ldots, \mathbf{a})$. Hence, we have the following corollary from Theorem 1.

Theorem 4 *For every $k \geq 2$, the consistency problem for $\mathcal{TPL}^k_{reg}$ is* NP*-complete.*

Corollary 5 *For every $k \geq 2$, both of classes $\mathcal{TPL}^k$ and $\mathcal{TPL}^k_{reg}$ are not PAC-learnable unless $P = NP$.*

Arimura *et al.* [1] reported that if the class $\mathcal{TPL}^k$ has a property called *the compactness with respect to containment* then the class is polynomial update time inferable from positive data. Note that the results we have shown in this report do not change even if the class has the compactness with respect to containment.

3 Discussion

In this section, we present some additional results on the class $\mathcal{TPL}^k$ of unions of a bounded number of tree patterns. In this sections we will show these results without proofs. First, we consider the problem concerning to polynomial update time inference from positive data. A *k-mimimal multiple generalization* (k-mmg, for short) of a finite set S of ground tree patterns is a set of at most k tree patterns that defines a minimal language containing S within the class $\mathcal{TPL}^k$. The following theorem states that the problem of computing (or printing) *all* of the k-mmgs of S will not polynomial time computable, while the problem of computing *one* of them is polynomial time computable if $\sharp\Sigma > k$ as reported in Arimura *et al.* [1]. The theorem is an immediate consequence of Theorem 3 stated in Section 2.

Theorem 6 *For every $k \geq 2$, the problem of printing all the k-mmgs of S is* NP*-hard for every alphabet Σ with $\sharp\Sigma \geq 3$.*

The second problem is to find a single *maximal* tree pattern consistent with given positive and negative examples. We can consider such a hypothesis as a *simplest* hypothesis that descriminates positive and negative examples. If we remove the restriction that the hypothesis to be found must be a maximal one, then the problem is obviously solvable in polynomial time by taking the lgg of all positive examples. However, under the restriction of maximality, the problem becomes hard to compute even for single tree patterns. The reduction used in the proof is similar to one for Theorem 3.

Theorem 7 *The problem of finding a single maximal tree pattern consistent with given positive and negative examples is* NP*-complete.*

References

[1] Arimura, H., Shinohara, T. and Otsuki, S. A polynomial time algorithm for finding finite unions of tree pattern languages. In *Proc. of the 2nd International Workshop on Nonmonotonic and Inductive Logic.* LNAI 659, pp. 118–131. Springer, 1993.

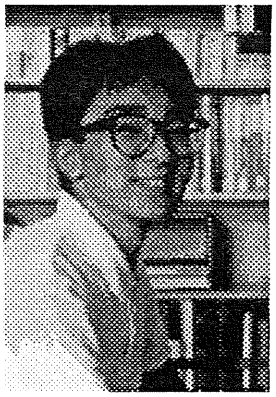
[2] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the vapnik-chervonenkis dimension. *JACM*, Vol. 36, No. 4, pp. 929–965, 1989.

[3] S. Miyano, A. Shinohara, and T. Shinohara. Which classes of elementary formal systems are polynomial-time learnable? In S. Arikawa, A. Maruoka, and T. Sato, editors, *Proceedings of the Second Workshop on Algorithmic Learning Theory*, pp. 139–150, 1991.

[4] B. K. Natarajan. On learning sets and functions. *Machine Learning*, Vol. 4, No. 1, pp. 67–97, 1989.

[5] L. Pitt and L. G. Valiant. Computational limitations on learning from examples. *JACM*, Vol. 35, No. 4, pp. 965–984, 1988.

About the Author



Hiroki Arimura (有村博紀) was born in Fukuoka on June 7, 1965. He received the B.S. degree in 1988 in Phisics, and the M.S. degree in 1990 in Information Systems from Kyusyu University. Presently, he is an Assistant of Kyushu Institute of Technology. His research interests are in logic programming and computational learning theory.

Research Institute of Fundamental Information Scinece, Kyushu University 33, Fukuoka 812, Japan.