2009－MPS－73 (4)
2009／3／5

繰り返し内部構造変数を持つ木パターンの有限和の質問学習

松本 哲志†, 鈴木 祐介††, 正代 隆義‡, 宮原 哲浩††, 内田 智之††

† 東海大学理学部情報数理学科
†† 広島市立大学情報科学研究科知能工学専攻
‡ 九州大学大学院システム情報科学研究院情報理学部門

Dana Angluin により質問を用いた学習の数理モデル (質問学習モデル) が提案されている．これまでの研究の多くは文字列を要素とする言語を対象としており，パターン言語や正規言語などの言語族が多項式時間で学習可能であることが示されてきた．現在，Web 上の HTML/XML ファイルなどのような木構造データが大量に存在する．我々は，木構造データに共通する構造を表現するパターンとして，項木という木構造パターンを提案している．本論文では，特徴的木構造を柔軟に表現するために，内部構造変数の繰り返しを許す項木で定義される言語の有限な和集合のクラスを考え，このクラスが質問学習モデルにおいて多項式時間で学習可能であることを示す．

Learning of Finite Unions of Tree Patterns with Repeated Internal Structured Variables from Queries

Satoshi MATSUMOTO†, Yusuke SUZUKI††, Takayoshi SHOUDAI ‡, Tetsuhiro MIYAHARA†† and Tomoyuki UCHIDA††

† Department of Mathematical Sciences, Tokai University
†† Department of Intelligent Systems, Hiroshima City University
‡ Department of Informatics, Kyushu University

A term tree is an ordered tree pattern, which have ordered tree structure and variables, and is suited for a representation of a tree structured pattern. A term tree t is allowed to have a repeated variable which occurs in t more than once. In this paper, we consider the learnability of finite unions of term trees with repeated variables in the exact learning model of Angluin (1988), which is a mathematical model of learning via queries in computational learning theory. We present polynomial time learning algorithms for finite unions of term trees with repeated variables by using superset and restricted equivalence queries. Moreover we show that there exists no polynomial time learning algorithm for finite unions of term trees by using restricted equivalence, membership and subset queries. This result indicates the hardness of learning finite unions of term trees in the exact learning model.

1 Introduction

In the field of Web mining, Web documents such as HTML/XML files have tree structures and are called tree structured data. In order to extract meaningful knowledge from given data, many data mining tools need to collaborate with experts or users in mining processes. Many of such tools are designed in query learning scheme. We are interested in clustering of heterogeneous tree structured data having no rigid structure. From these motivations, in this paper, we consider polynomial time learnabilities of finite unions of tree structured patterns in the exact learning model by Angluin [3].

A term tree is a rooted tree pattern which consists of an ordered tree structure, ordered children and internal structured variables [5, 6]. A variable in a term tree is a list of two vertices and it can be substituted by an arbitrary tree. For example, the term tree $t = (V_t, E_t, H_t)$ in Figure 1 is defined as follows. $V_t = \{v_1, \ldots, v_{11}\}$, $E_t = \{(v_1, v_2), (v_2, v_3), (v_1, v_4), (v_7, v_8), (v_1, v_{10}), (v_{10}, v_{11})\}$ with root v_1 and sibling relation displayed in Figure 1. $H_t = \{[v_4, v_5], [v_1, v_6], [v_6, v_7], [v_6, v_9]\}$.

A variable with a variable label x in a term tree t is said to be *repeated* if x occurs in t more than once. In this paper, we treat a term

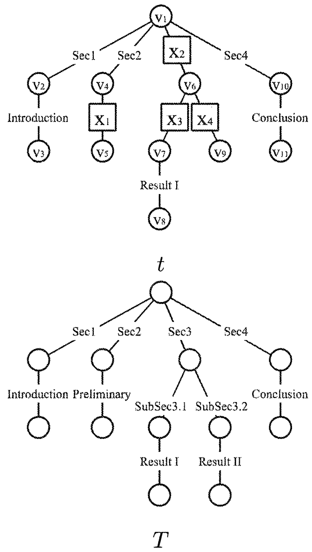

Figure 1: A term tree t explains a tree T. A variable is represented by a box with lines to its elements. The label inside a box is the variable label of the variable.

tree with repeated variables. In [4], Arimura et al. discussed the polynomial time learnabilities of ordered gapped forests without repeated gap-variables in the exact learning model. In this paper, we discuss polynomial time learnabilities of finite unions of term trees with repeated variables in the exact learning model. For a tree T which represents tree structured data such as Web documents, string data such as tags or texts are assigned to edges of T. Hence, we assume naturally that the cardinality of a set of edge labels is infinite. Let Λ be a set of strings used in tree structured data. Then, our target class of learning is the class, denoted by $\mathcal{OTF}_\Lambda$, of all finite sets of term trees all of whose edges are labeled with elements in Λ . The *term tree language* of a term tree t, denoted by $L_\Lambda(t)$, is the set of all labeled ordered trees which are obtained from t by substituting arbitrary labeled trees for all variables in t. The language represented by a finite set of term trees $R = \{t_1, t_2, \ldots, t_m\}$ in $\mathcal{OTF}_\Lambda$ is the finite union of m term tree languages $L_\Lambda(R) = L_\Lambda(t_1) \cup L_\Lambda(t_2) \cup \ldots \cup L_\Lambda(t_m)$. In particular, we define $L_\Lambda(\emptyset) = \emptyset$.

In the exact learning model by Angluin [3], a learning algorithm is said to *exactly learn* a target finite set R_* of term trees if it outputs a finite set R of term trees such that $L_\Lambda(R) = L_\Lambda(R_*)$ and halts, after it uses some queries. In this paper, firstly, we present a polynomial time algorithm which exactly learns any finite set in $\mathcal{OTF}_\Lambda$ having m_* term trees by using superset queries for a known number m_*. Secondly, we present a polynomial time algorithm for the same setting as above except that the number of term trees in R_* is unknown. Finally, we show that there exists no polynomial time learning algorithm for finite unions of term trees by using restricted equivalence, membership and subset queries. This result indicates the hardness of learning finite unions of term trees in the exact learning model.

In the exact learning model, many researchers [1, 2, 4, 5] showed the exact learnabilities of several kinds of tree structured patterns. A term tree t is said to be *linear* (or *repetition-free*) if all variable labels in t are mutually distinct. In [5], we showed the polynomial time exact learnability of finite unions of linear term trees, denoted by $\mu\mathcal{OTF}_\Lambda$, using restricted subset queries and equivalence queries. As other learning models, in [6], we showed the class of single regular term trees is polynomial time inductively inferable from positive data.

2 Preliminaries

Let X be an infinite alphabet whose element is called a *variable label*, and Λ an alphabet where $\Lambda \cap X = \emptyset$. We call an element in Λ an *edge label*, and in this paper, we assume that $|\Lambda|$ is infinite.

Let $T = (V_T, E_T)$ be an edge-labeled rooted tree with ordered children which has a set V_T of vertices and a set E_T of edges labeled with elements of $\Lambda \cup X$. Let H_t be the set of all edges in E_T whose labels are in X. Let $V_t = V_T$ and $E_t = E_T - H_t$ (i.e., $E_t \cup H_t = E_T$ and $E_t \cap H_t = \emptyset$). A triplet $t = (V_t, E_t, H_t)$ is called a *term tree*, and elements in V_t, E_t and H_t are called a *vertex*, an *edge* and a *variable*, respectively. We denote by (v, v') the edge in E_t and $[v, v']$ the variable in H_t.

Let f and g be term trees with at least two vertices. Let $h = [v, v']$ be a variable in f with the variable label x and $\sigma = [u, u']$ a list of two distinct vertices in g, where u is the root of g and u' is a leaf of g. The form $x := [g, \sigma]$ is called a *binding* for x. A new term tree $f' = f\{x := [g, \sigma]\}$ is obtained by applying the binding $x := [g, \sigma]$ to f in the following way. Let $e_1 = [v_1, v'_1], \ldots, e_m = [v_m, v'_m]$ be the variables in f with the variable label x. Let $g_1, \ldots, g_m$ be m copies of g and u_i, u'_i the vertices of g_i corresponding to u, u' of g, respectively. For each variable $e_i = [v_i, v'_i]$, we attach g_i to f by removing the variable e_i from H_f and by identifying the vertices v_i, v'_i with the vertices u_i, u'_i of g_i.

A *substitution* θ is a finite collection of bindings $\{x_1 := [g_1, \sigma_1], \cdots, x_n := [g_n, \sigma_n]\}$, where x_i's are mutually distinct variable labels in X. The term tree $f\theta$, called the *instance* of f by θ, is obtained by applying all the bindings $x_i := [g_i, \sigma_i]$ on f simultaneously. Then the instance $t\theta$ of the term tree t by θ is isomorphic to the tree T in Figure 1. Let t and t' be term trees. We write $t \preceq t'$ if there exists a substitution θ such that t is isomorphic to $t'\theta$. If $t \preceq t'$ and t is not isomorphic to t', then we write $t \prec t'$.

3 Learning model

In this paper, let R_* be a set of term trees in $\mathcal{OTF}_\Lambda$ to be identified, and we say that the set R_* is a *target*. Without loss of generality, we assume that $L_\Lambda(R_*) \neq L_\Lambda(R_* - \{r\})$ for any $r \in R_*$.

We introduce the exact learning model via queries due to Angluin [3]. In this model, learning algorithms can access to *oracles* that answer specific kinds of queries about the unknown term tree language $L_\Lambda(R_*)$. We consider the following oracles. (1) *Superset query* Sup_{R_*}: The input is a set R in $\mathcal{OTF}_\Lambda$. If $L_\Lambda(R) \supseteq L_\Lambda(R_*)$, then the output is *"yes"*. Otherwise, it returns a *counterexample* $t \in L_\Lambda(R_*) - L_\Lambda(R)$. (2) *Restricted equivalence query* $rEquiv_{R_*}$: The input is a set R in $\mathcal{OTF}_\Lambda$. The output is *"yes"* if $L_\Lambda(R) = L_\Lambda(R_*)$ and *"no"* otherwise. (3) *Membership query* Mem_{R_*}: The input is a labeled tree t. The output is *"yes"* if $t \in L_\Lambda(R_*)$, and *"no"* otherwise. (4) *Subset query* Sub_{R_*}: The input is a set R in $\mathcal{OTF}_\Lambda$. The output is *"yes"* if $L_\Lambda(R) \subseteq L_\Lambda(R_*)$. Otherwise, it returns a counterexample $t' \in L_\Lambda(R) - L_\Lambda(R_*)$.

A learning algorithm $\mathcal{A}$ collects information about $L_\Lambda(R_*)$ by using queries and output a set

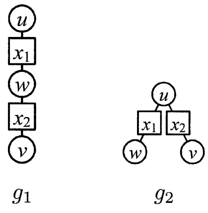

Figure 2: Linear term trees g_1,g_2.

R in $\mathcal{OTF}_\Lambda$. We say that a learning algorithm $\mathcal{A}$ *exactly identifies* a target R_* in polynomial time using a certain type of queries if $\mathcal{A}$ halts in polynomial time and outputs a set $R \in \mathcal{OTF}_\Lambda$ such that $L_\Lambda(R) = L_\Lambda(R_*)$ using queries of the specified type.

4 Learnability and Hardness

We denote by $\mathcal{ES}(r)$ the set of all linear term trees which are obtained from r by replacing a variable of r with g_1 or g_2 given Figure 2. Note that $|r'| > |r|$ and $r' \prec r$ for any $r' \in \mathcal{ES}(r)$, and $|\mathcal{ES}(r)| \leq 3|r|$.

Let m be a positive integer, R a set of term trees and r a term tree such that $m = |R_*|$, $L_\Lambda(R_*) \subseteq L_\Lambda(R \cup \{r\})$ and $L_\Lambda(R_*) \not\subseteq L_\Lambda(R)$. In the algorithm *LEARN_KNOWN*, the algorithm *L_OTT*(m,R,r) outputs a set S of term trees such that $S \subseteq R_*$ and $r_* \preceq r$ for any $r_* \in S$.

When the size of R_* is known in advance, we have the following theorem.

Theorem 1 If the algorithm *LEARN_KNOWN* of Figure 3 takes an integer m with $m \geq |R_*|$ as input, then it exactly identifies a set $R_* \in \mathcal{OTF}_\Lambda$ in polynomial time with respect to n and m using superset queries, where n is the maximum size of term trees in R_*.

When the size of R_* is unknown, we have the following theorem.

Theorem 2 The algorithm *LEARN_OTF* of Figure 4 exactly identifies any set $R_* \in \mathcal{OTF}_\Lambda$ in polynomial time with respect to n and m_* using superset queries and restricted equivalence queries, where n is the maximum size of term trees in R_*.

Finally, we show the insufficiency of learning of $\mathcal{OTF}_\Lambda$ in the exact learning model.

```
Algorithm LEARN_KNOWN
Input:   an integer m with m ≥ |R_*|;
Output:  A set R ∈ OTF_Λ with L_Λ(R) = L_Λ(R_*);
begin
  Let R_hypo := ∅;
  if Sup_{R_*}(R_hypo) = "yes" then
    output R_hypo;
  else begin
    Let r = ({u, v}, ∅, {[u, v]}) ∈ μOTT_Λ;
    R = {r}; R_hypo := R_nocheck := R;
    while R_nocheck ≠ ∅ do begin
      foreach r ∈ R_nocheck do
        if Sup_{R_*}((R_hypo − {r}) ∪ ES(r)) = "yes"
        then begin
          R_hypo := (R_hypo − {r}) ∪ ES(r);
          R_nocheck := (R_nocheck − {r}) ∪ ES(r);
          foreach r' ∈ ES(r) do begin
            if Sup_{R_*}(R_hypo − {r'}) = "yes" then
            begin
              R_hypo := R_hypo − {r'};
              R_nocheck := R_nocheck − {r'};
            end;
          end;
        end
        else begin
          R' := L_OTT(m,(R_hypo − {r}) ∪ ES(r),r);
          R_hypo := (R_hypo − {r}) ∪ R' ∪ ES(r);
          R_nocheck := (R_nocheck − {r}) ∪ ES(r);
          foreach r' ∈ ES(r) do begin
            if Sup_{R_*}(R_hypo − {r'}) = "yes" then
            begin
              R_hypo := R_hypo − {r'};
              R_nocheck := R_nocheck − {r'};
            end;
          end;
        end;
      end;
    end;
  output R_hypo;
end.
```

Figure 3: Algorithm *LEARN_KNOWN*

Theorem 3 Any learning algorithm that exactly identifies all finite sets of the term trees of size n using restricted equivalence, membership and subset queries must make $\Omega(2^n)$ queries in the worst case, where $n \geq 6$ and $|\Lambda| \geq 1$.

5 Conclusions

We have studied the learnability of $\mathcal{OTF}_\Lambda$ in the exact learning model. We have presented polynomial time learning algorithms for $\mathcal{OTF}_\Lambda$ by using superset and restricted equivalence queries. Moreover we show the hardness of learning $\mathcal{OTF}_\Lambda$ in the exact learning model.

```
Algorithm LEARN_OTF
Output: A set R ∈ OTF_Λ with L_Λ(R) = L_Λ(R_*).
begin
  m := 0; R := ∅;
  repeat
    m := m + 1;
    R := LEARN_KNOWN(m);
  until rEquiv_{R_*}(R) = "yes";
  output R;
end.
```

Figure 4: Algorithm *LEARN_OTF*

We will study the learnabilities of $\mu\mathcal{OTF}_\Lambda$ and $\mathcal{OTF}_\Lambda$ in the framework of polynomial time inductive inference from positive data.

References

[1] T. R. Amoth, P. Cull, and P. Tadepalli. Exact learning of tree patterns from queries and counterexamples. *Proc. COLT-98, ACM Press*, pp. 175–186, 1998.

[2] T. R. Amoth, P. Cull, and P. Tadepalli. Exact learning of unordered tree patterns from queries. *Proc. COLT-99, ACM Press*, pp. 323–332, 1999.

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

[4] H. Arimura, H. Sakamoto, and S. Arikawa. Efficient learning of semi-structured data from queries. *Proc. ALT-2001, Springer-Verlag, LNAI 2225*, pp. 315–331, 2001.

[5] S. Matsumoto, T. Shoudai, T. Uchida, T. Miyahara, and Y. Suzuki. Learning of finite unions of tree patterns with internal structured variables from queries. *IEICE Trans*, Vol. E91-D, No. 2, pp. 222–230, 2008.

[6] Y. Suzuki, R. Akanuma, T. Shoudai, T. Miyahara, and T. Uchida. Polynomial time inductive inference of ordered tree patterns with internal structured variables from positive data. *Proc. COLT-2002, Springer-Verlag, LNAI 2375*, pp. 169–184, 2002.