International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 16 (2017) pp. 6050-6054

Learning k-edge Deterministic Finite Automata in the Framework of Active Learning

Anuchit Jitpattanakul*

Department of Mathematics, Faculty of Applied Science,
King Mong's University of Technology North Bangkok, Thailand.

**Author for correspondence;*

Abstract

One of most attractive topics in grammatical inference is theoretically study on learnability of some classes of automata corresponding with defined formal languages. For last two decades a number of theoretical results have been reported and played as essential knowledge for applications in other fileds such as speech recognition and music-style recognition. In this paper, we consider the problem of learning *k*-edge determinisitc finite automata in the framework of active learning. The results show that the class of *k*-edge deterministic finite automata identification in the limit with membership queries and equivalence queries.

Keywords: *k*-edge deterministic finite automata; identification in the limit with quries; learnability;

INTRODUCTION

In research field of grammatical inference (GI), learning refers to a process of identifying a formal language in terms of its grammatical representation by a learner who is given information of the formal language. One of most attractive topics in GI is of theoretically studying on an important property of classes of defined automata. This property is called learnability. Typically, the study is mostly based on three learning models e.g. Gold's passive learning model [1], Angluin's active learning model [2], Valiant's approximately learning model [3].

The Gold's learning model is viewed as a framework of passive learning. In the process of learning an unknown language, a number of examples will be provided at each time to a learner who is to hypothesize a grammatical representation of the language on the basis of the examples received so far. The process continues repeatly. The success of learning process is considered by using a criterion called identification in the limit [1]. It was developed by adding some constraints of complexity [4]. One of those criteria that are widely used in a learning model called identification in the limit from polynomial time and data introduced by Higuera [5]. However, there are several situations where the learning can actively interact with its environment. The mathematical setting to do this is called active learning, where queries are made to an oracle. In this learning framework, the learner has access to a truthfully oracle which is allowed to answer specific type of queries.

Active learning is a paradigm firstly introduced with theoretical motivations but that for a number of reasons can today be considered also as a pragmatic approach [6]. Some of the theoretical reasons is to make use of additional information that can be measured. For practical view, the active learning is an important field of GI because it is becoming more widely used in case of problems where labeling the examples in the training data set is expensive.

The problem of active learning in GI is mainly studied on the class of regular languages. In Angluin's work, a polynomial time query learning algorithm for the class of minimal complete deterministic finite automata (DFA) is given, in which the learner can ask membership queries (MQ) and equivalence queries (EQ). There are though other types of possible queries: subset, superset, disjointness and exhaustive queries [7], structured membership queries [8], etc. This is because the class is almost only one class of formal languages that is both efficiently learnable and general enough to represent many nontrivial real-life phenomena. The learnability of various grammatical representation of formal languages has been also studied in active learning framework through identification of specifically defined automata [9] such as regular expression [10], and multiplicity tree automata [11]. Learning algorithms from these works have been also experimentally tested to real-world applications such as DNA sequences analysis [12], music style recognition [13], and speech recognition [14]. The obtained results show that the algorithms are an effective and efficient alternative to solve the problems. Unfortunately, one disadvantage of using those algorithms is about a size of returned automata depending on size of alphabet of inputs. This leads to inconvenience in practical ways.

In order to solve these problems, *k*-edge deterministic finite automata (*k*-DFA) were firstly introduced by Higuera [15] to recognize languages defined over an ordered alphabet. The languages are recognized by *k*-DFA called *k*-acceptable languages. An interesting property of *k*-DFA is that its size

depends on the k value instead of the size of alphabet. Recently the class of k-DFA has been proved that it is learnable in the limit from only positive examples and negative examples. But for using only positive examples, this class is not learnable from them [16].

In this paper, we focus our attention on the learnability on Angluin's active learning model. In this paper, two types of queries, *i.e*. membership queries and equivalence queries will be theoretically investigated. We study the identification in the limit with queries of k-edge deterministic finite automata by using membership and equivalence queries.

The remains are organized as follows. Section 2 presents basic definitions and notations. In section 3, we introduced k-edge deterministic finite automata and proved some properties of this class. In section 4 we investigate learnability of the class of k-edge deterministic finite automata in framework of active learning. The last section provides the conclusion of this work.

PRELIMINARIES

The basic definitions and notations used throughout this paper are provided in this section.

Formal languages and automata

Let Σ be an *alphabet* that is a finite and nonempty set of letters. The size of Σ is a number of letters, denoted by $|\Sigma|$. A finite sequence of letters from Σ is called a *string*. Given a string w, the length of strings is the total number of letters appearing in w and it is denoted by $|w|$. The string with length zero is called the *null string* denoted by λ. The infinite set of all possible strings over Σ, denoted by Σ^*, is the set of all finite-length strings generated by concatenating zero or more letters of Σ. An alphabet Σ is called an *ordered alphabet* denoted by $\Sigma_{\leq}$. Given an order relation $<$ on Σ, we can define a *lexicographic-length order* over Σ^* for two strings $u, v \in \Sigma^*$, by setting $u <_{lex} v$ if and only if $|u| < |v|$ or there exist strings $w, u', v' \in \Sigma^*$ and two letters $x < y \in \Sigma$ such that $|u'| < |v'|$ and $u = wxu'$, $v = wyv'$. A *language* over Σ denoted by L is any subset of Σ^*. The family of languages over Σ, denoted by $\mathcal{L}$, is called *a class of languages*.

A *finite automaton* is a grammatical representation that is typically defined as a 5-tuple $M = (\Sigma, Q, q_0, F, \delta)$,where Σ is a finite alphabet, Q is a finite non-empty set of states, $q_0 \in Q$ is an initial state, $F \subseteq Q$ is a set of final states, and $\delta: Q \times \Sigma \to Q$ is a state transition function. The state transition function δ can be extended to a mapping $\delta^*: Q \times \Sigma^* \to Q$ in the following inductive way: (i) $\delta^*(q, \lambda) = q$, for each state $q \in Q$, where λ is the null string, and (ii) $\delta^*(q, wa) = \delta(\delta^*(q, w), a)$, for each state $q \in Q$, each letter $a \in \Sigma$, and each string $w \in \Sigma^*$. The finite automaton M is *deterministic* if $|\delta(q, a)| \leq 1$ for each $q \in Q$ and for each $a \in \Sigma$. Then M is called *deterministic finite automata* (shortly denoted by *DFA*s).

Theoretically, an automaton plays an important role as a language recognizer. A string w is *recognized* by an automaton $M = (\Sigma, Q, q_0, F, \delta)$ if $\delta^*(q, w) \in F$. The language recognized by M, denoted by $\mathbb{L}(M)$, is the set of all strings which are recognized by the automaton M and this set is called a *regular language*. A language L is *recognizable* if there exists an automaton M such that $L = \mathbb{L}(M)$.

Active learning and convergence criterions

In grammatical inference, a *learning algorithm* $\mathcal{A}_L$ is a mapping function defined as $\mathcal{A}_L : \mathcal{S} \to \mathcal{G}$, where $\mathcal{S}$ is a set of all presentation and this set is used for learning any language L in a language class $\mathcal{L}$ by identifying a grammatical representation G in a class $\mathcal{G}$ of corresponding grammatical representations. For learning an unknown language L, we say that the algorithm $\mathcal{A}_L$ *converges* to $G \in \mathcal{G}$ from $S \in \mathcal{S}$ if and only if $\mathbb{L}(\mathcal{A}_L(S)) = L$.

The active learning is based on the existence of an oracle which can be seen in principle as a device that knows the language and has to answer correctly and can only answer queries from a given set of queries. In this section, we study on learnability of the class of k-edge deterministic finite automata by using two different types of queries. Firstly, only membership queries are available in learning context. Secondly, equivalence queries are additionally given. The two different type of queries defined as below.

Definition 2.1

A *membership query* is made by proposing a string to the oracle, who answers Yes if the string belong to the language and NO if not. We will denote this formally by

MQ : $\Sigma^* \to$ {Yes, No}.

Definition 2.2

An *equivalence query* is made by proposing a grammatical representation G to the oracle. The oracle answers Yes if the grammatical representation G is equivalent to the target and NO if not. We will denote this formally by

MQ : $\mathcal{G} \to$ {Yes, No}.

We describe the active learning process. We define a class of grammatical representation G and the sort of queries we are allowed to make and the oracle will have to answer. We call this class of queries QUER. Typically if the learner is only

allowed to make membership queries, we will have QUER = {MQ}.

The learning criterion that will be used in this paper can be found in [1] and if is well known as *identification in the limit with queries*. A formal definition of this criterion is restated as follows.

Definition 2.3

A class $\mathcal{G}$ is *identifiable in the limit with queries* from QUER if there exists an algorithm $\mathcal{A}$ such that given any grammatical representation G in $\mathcal{G}$, $\mathcal{A}$ identifies $\mathcal{G}$ in the limit, i.e. returns a grammatical representation G' equivalent to G and halts.

k-edge Deterministic Finite Automata And Their Properties

We begin this section with giving a formal definition of an automaton called a *k*-edge deterministic finite automaton. Moreover, some properties of the *k*-acceptable languages will be investigated their learnability.

Definition 3.1

A *k-edge deterministic finite automaton* (*k*-DFA) is a 6-tuple $M_k = (\Sigma_{\leq}, Q, q_0, F_A, F_R, \delta_k)$ where $\Sigma_{\leq}$ is a finite ordered alphabet, Q is a finite set of states, q_0 is the initial state, $F_A \subseteq Q$ is a set of accepting states and $F_R \subseteq Q$ is a set of rejecting states, $\delta : Q \times \Sigma_{\leq} \times \Sigma_{\leq} \rightarrow Q$ is the transition function defined as for any $q \in Q$, $\#(\delta) = |\{[x, y]: \delta_k(q, x, y) \neq \emptyset\}| \leq k$, and if $\delta_k(q, a_1, b_1) \neq \delta_k(q, a_2, b_2)$ then $\{z : a_1 \leq z \leq b_1\} \cap \{z : a_2 \leq z \leq b_2\} = \emptyset$. The extended transition function $\delta_k^*: Q \times \Sigma^* \rightarrow Q$ is defined as $\delta_k^*(q, \lambda) = q$ and $\delta_k^*(q, aw) = \delta_k^*(q', w)$ where $x \leq a \leq y$ and $\delta_k(q, x, y) = q'$ such that $q, q' \in Q$, $a, x, y \in \Sigma_{\leq}$, $w \in \Sigma_{\leq}^*$.

Example 3.1

Let L be a language defined over $\Sigma_{\leq} = \{a_1, a_2, a_3, a_4, a_5\}$. The language L is $((a_1+a_2+a_3)+(a_4+a_5)\, a_1^*(a_2+a_3+a_4+a_5))^*$. This automaton is S_2-DFA because for any $q \in Q$, $|\{(x, y) : \delta_2(q, x, y) \neq \emptyset\}| = 2$ and for state $q_0 : \delta_2(q_0, 1, 3) \neq \delta_2(q_0, 4, 5) \neq \emptyset$ then $\{z : a_1 \leq z \leq a_3\} \cap \{z : a_4 \leq z \leq a_5\} = \emptyset$ for state $q_3 : \delta_2(q_3, a_1, a_1) \neq \delta_2(q_3, a_2, a_5) \neq \emptyset$ then $\{z : a_1 \leq z \leq a_1\} \cap \{z : a_2 \leq z \leq a_5\} = \emptyset$. The finite automaton in this example is depicted in Fig. 1

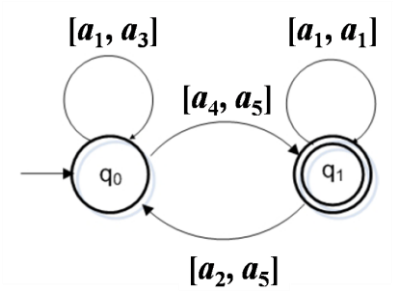

Figure 1: S_2-DFA

Definition 3.2

A language recognized by *k*-DFA $M_k = (\Sigma_{\leq}, Q, q_0, F_A, F_R, \delta_k)$ is called a *k-acceptable languages* defined as $L = \{w : \delta^*_k(q_0, w) \in F_A\}$. A set of all *strictly k*-acceptable languages is called that a *class of k-acceptable language* denoted by *k*-$\mathcal{ACC}$ for any integer $k \geq 0$.

To study learnability of *k*-$\mathcal{ACC}$, some properties are needed. In this section we have proved some propositions that will be referred in next section.

Proposition 3.1 *For any integer $k \geq 0$, a k-DFA is a (k+1)-DFA.*

Proof: Let $M_k = (\Sigma_{\leq}, Q, q_0, F_A, F_R, \delta_k)$ be a *k*-DFA. Suppose that q is a state in Q such that $q \in Q$, $|\{[x, y] : \delta_k(q, x, y) \neq \emptyset\}| = k$ and if $\delta_k(q, a_1, b_1) \neq \delta_k(q, a_2, b_2)$ then $\{z : a_1 \leq z \leq b_1\} \cap \{z : a_2 \leq z \leq b_2\} = \emptyset$. For all integer k, it is algebraically obvious that $k < k + 1$. It follows that $|\{[x, y] : \delta_k(q, x, y) \neq \emptyset\}| \leq k+1$. Therefore, M_k is a $(k+1)$-DFA by definition. ■

Proposition 3.2 *For any integer $k \geq 0$, k-$\mathcal{ACC} \subseteq$ (k+1)-$\mathcal{ACC}$.*

Proof: Let $M_k = (\Sigma_{\leq}, Q, q_0, F_A, F_R, \delta_k)$ be a *k*-DFA and $L_k \in k$-$\mathcal{ACC}$ be a *k*-acceptable language recognized by M_k. In order to prove this proposition, we will show that L_k is a language in the class $(k+1)$-$\mathcal{ACC}$. In other word, we have to show there exists a $(k+1)$-DFA to recognize the language L_k. Suppose $M_m = (\Sigma_{\leq}, Q, q_0, F_A, F_R, \delta_m)$ is a *m*-DFA. We define the transition function δ_{k+1} of M_m as $\delta_m = \delta_k - \{(q_0, a_0, a_i, p) : a_i \in \Sigma_{\leq}, p \in Q\} \cup \{(q_0, a_0, a_0, p), (q_0, a_1, a_i, p)\}$. It follows that $q \in Q$, $|\{[x, y] : \delta_m(q, x, y) \neq \emptyset\}| = k+1$. Thus, M_m is a $(k+1)$-DFA recognizing the language L_k. That is $L_k \in (k+1)$-$\mathcal{ACC}$. Thus we can conclude that k-$\mathcal{ACC} \subseteq (k+1)$-$\mathcal{ACC}$ for $k \geq 0$. ■

Proposition 3.3 *For any integer $k \geq 0$, k-$\mathcal{ACC} \subseteq \mathcal{REG}$.*

Proof: The idea of this proof is to show for all *k*-acceptable languages are in the class of regular languages. That is if L is a *k*-acceptable language recognized by *k*-edge deterministic finite automata then the language L is recognized by deterministic finite automata.

Let $M_k = (\Sigma_{\leq}, Q, q_0, F_A, F_R, \delta_k)$ be a *k*-DFA that each tuple defined as $\Sigma_{\leq} = \{a_1, a_2, \dots, a_n\}$, $\delta_k = \{(p, a, b, q) : p, q \in Q$ and $a, b \in \Sigma_{\leq}\}$ which $\#(\delta_k) = k$. This *k*-DFA M_k recognized the language *L*.

We can construct the deterministic finite automaton $M = (\Sigma, Q, q_0, F_A, F_R, \delta)$ recognized L. The construction of M shows as follows: $\Sigma = \{a : a \in \Sigma_{\leq}\}$, $\delta = \{(p, z, q) : a \leq z \leq b$ for all $(p, a, b, q) \in \delta_k\}$. That is the deterministic finite automata M can recognize the language L. Thus, we conclude that $k\text{-}\mathcal{ACC} \subseteq \mathcal{REG}$ for any integer $k \geq 0$. ■

Proposition 3.4 $\cup^{\infty}_{k=1}\ k\text{-}\mathcal{ACC} = \mathcal{REG}$

Proof: The idea of this proof is to show that $\cup^{\infty}_{k=1}\ k\text{-}\mathcal{ACC} \subseteq \mathcal{REG}$ and $\mathcal{REG} \subseteq \cup^{\infty}_{k=1}\ k\text{-}\mathcal{ACC}$.

Let $M = (\Sigma, Q, q_0, F_A, F_R, \delta)$ be a deterministic finite automata recognizing a regular languages *L*. Each tuple of M is defined as $\Sigma = \{a_1, a_2, \dots, a_n\}$, $\delta = \{(p, a, q) : p, q \in Q$ and $a, \in \Sigma\}$. We can construct a *k*-edge deterministic finite automaton $M_k = (\Sigma_{\leq}, Q, q_0, F_A, F_R, \delta_k)$ defined by $\Sigma_{\leq} = \{a_i : a_i \in \Sigma\}$ and $\delta_k = \{(p, a, a, q)$: for all $(p, a, q) \in \delta\}$. It is easy to see that M_k can recognize L .Thus, it follows that $\mathcal{REG} \subseteq \cup^{\infty}_{k=1}\ k\text{-}\mathcal{ACC}$.

From proposition 3.3, we have proved that for any integer $k \geq 0$, $k\text{-}\mathcal{ACC} \subseteq \mathcal{REG}$. Therefore, we conclude that $\cup^{\infty}_{k=1}\ k\text{-}\mathcal{ACC} = \mathcal{REG}$ ■

LEARNING k-DFA IN FRAMEWORK OF ACTIVE LEARNING

The active learning is based on the existence of an oracle which can be seen in principle as a device that knows the language and has to answer correctly and can only answer queries from a given set of queries. In this section, we study on learnability of the class of *k*-edge deterministic finite automata by using two different types of queries.

Learning *k*-acceptable languages by using membership and equivalence queries :

We theoretically show that the class of *k*-edge deterministic finite automata is identifiable in the limit by using both membership queries and equivalence queries. A reduction technique will be used in this proof.

The reduction technique for grammatical inference have been firstly formalized in [17] by Higuera. This algebraic technique allows us to refine previous theoretical results of learnability.

Theorem 4.1 (from Theorem 2 in [18])

If the $\mathcal{B}$ of languages is learnable in terms of R($\mathcal{B}$) from Pres($\mathcal{B}$), and there exists a computable function χ : R($\mathcal{B}$) $\rightarrow$ R($\mathcal{A}$) such that $\psi \circ \chi$ = Id, and ξ is a computable reduction, then the class $\mathcal{A}$ of languages is learnable in term of R($\mathcal{A}$) from Pres($\mathcal{A}$).

Proof: see in [18].

To get better understanding, a diagram representing the situation is shown in Fig. 2.

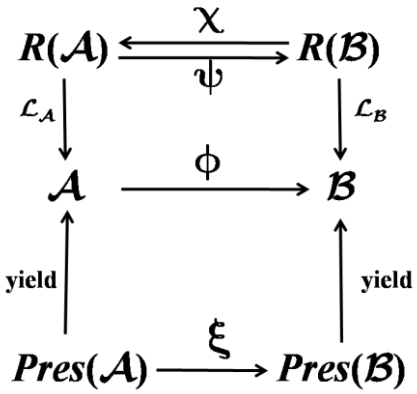

Figure 2: The commutation diagram

In [16]. the class of deterministic finite automata ($\mathcal{DFA}$) of regular languages ($\mathcal{REG}$) have been proved that it is identifiable in the limit with queries by using both membership queries and equivalence queries. For this work, we refine the theoretical results.

Theorem 4.2 *The class of k-edge deterministic finite automata is identifiable in the limit with queries by using both membership queries and equivalence queries*.

Proof: Let $\mathcal{A}_{REG}$ be a learning algorithm that identifies languages in the class of regular languages ($\mathcal{REG}$). Consider

algorithm $\mathcal{A}_k$ below, that learns a set $QUER_k$ of membership queries (MQ) and equivalence queries (EQ) and then output a k-DFA corresponding the learning set. The algorithm $\mathcal{A}_k$ executes as follows.

Algorithm : $\mathcal{A}_k$

Input : $QUER_k = \{MQ, EQ\}$

Output : k-DFA

1: $QUER_{REG} \leftarrow \xi(QUER_k)$;

2: DFA$\leftarrow \mathcal{A}_{REG}(QUER_{REG})$

3: k-DFA$\leftarrow \chi$(DFA)

Return k-DFA

Since ξ is identity, and χ is the natural transformation. Hence the k-edge deterministic finite automata is identifiable in the limit with queries by using both positive and negative examples. ■

CONCLUSION

In this work we study learnability of the k-edge deterministic finite automata on active learning model with two different types of queries. We have proved that the class of k-edge deterministic finite automata is identifiable in the limit with queries by using both membership queries and equivalence queries.

ACKNOWLEDGMENT

This research was funded by King Mongkut's University of Technology North Bangkok. Contract no. KMUTNB-GEN-57-20.

REFERENCES

[1] Gold E.M., 1967, "Language identification in the limit," Information and Control, vol. 10, no. 5,pp. 447–474.

[2] Angluin D., 1987, "Queries and concept learning," Machine Learning Journal, 2, pp. 319–342.

[3] Valiant L. G., 1984, "A theory of the learnable," Comm. ACM , vol. 27, no. 11, pp. 1134–1142.

[4] Pitt L., 1989, "Inductive inference, DFA's, and computational complexity," Proc. of the Int. Workshop AII'89, pp. 18–44.

[5] de la Higuera C., 1997, "Characteristic sets for polynomial grammatical inference," Machine Learning Journal, 27, pp. 125–138.

[6] Angluin, D., 1987, "Learning regular sets from queries and counterexamples," Information and Computation 75 (2),pp. 87–106.

[7] Angluin, D., 1988, "Queries and concept learning," Machine Learning 2 (4), pp. 319–342.

[8] Sakakibara, Y., 1990, "Learning context-free grammars from structural data in polynomial time," Theoretical Computer Science 76, pp. 223–242.

[9] de la Higuera C., 2005, "A bibliographical study of grammatical inference," Pattern Recognition, 38, pp. 1332–1348.

[10] Kinber, E.B., "On learning regular expressions and patterns via membership and correction queries," [33] 125–138

[11] Amaury, H., 2006, "Learning multiplicity tree automata," Proc. of ICGI'06, pp.268-280.

[12] Sakakibara Y., 2005, "Grammatical Inference in Bioinformatics," IEEE Trans. Pattern Anal. Mach. Intell., 27, no. 7, pp. 1051–106.

[13] Cruz P. and Vidal E., 1998, "Learning regular grammars to model musical style: Comparing different coding scheme," Proc. of ICGI'98, pp. 211-222.

[14] García P. et al., 1994, "On the use of the morphic generator grammatical inference (mggi) methodology in automatic speech recognition," Int. J. Pattern Recogn. Artif. Intell., vol. 4, pp. 667–685.

[15] de la Higuera C., 2006, "Ten open problems in grammatical inference," Proc. of ICGI'06, pp. 32-44.

[16] Jitpattanakul A. and Surarerks A., 2012, "The study of learnability of the class of k-acceptable languages on Gold's learning model," Chiang Mai Journal of Science.

[17] de la Higuera C., 2010 "Grammatical inference: learning automata and grammars" Cambridge University Press.

[18] de la Higuera C.,2005, "Complexity and reductions issues in Grammatical