Simple DFA are Polynomially Probably Exactly Learnable from Simple Examples

Rajesh Parekh
Allstate Research and Planning Center
321 Middlefield Road
Menlo Park, CA 94025.
rpare@allstate.com

Vasant Honavar
Artificial Intelligence Research Laboratory
Department of Computer Science
Iowa State University
Ames, IA 50011-1040.
honavar@cs.iastate.edu

Abstract

Efficient learning of DFA is a challenging research problem in *grammatical inference*. Both exact and approximate (in the PAC sense) identifiability of DFA from examples is known to be hard. Pitt, in his seminal paper posed the following open research problem: "*Are DFA PAC-identifiable if examples are drawn from the uniform distribution, or some other known simple distribution?*" (Pitt, 1989). We demonstrate that the class of *simple* DFA (i.e., DFA whose canonical representations have logarithmic Kolmogorov complexity) is efficiently *probably exactly learnable* under the Solomonoff Levin *universal distribution* $\mathbf{m}$ (wherein an instance x with Kolmogorov complexity $K(x)$ is sampled with probability that is proportional to $2^{-K(x)}$). The *simple distribution independent learning theorem* states that a concept class is learnable under universal distribution $\mathbf{m}$ *iff* it is learnable under the entire class of *simple distributions* provided the examples are drawn according to the universal distribution (Li & Vitányi, 1991). The class of simple distributions includes all *computable distributions*. Thus, it follows that the class of simple DFA is learnable under a sufficiently general class of distributions.

1 INTRODUCTION

DFAs are recognizers for *regular* languages which constitute the simplest class in the Chomsky hierarchy of formal languages (Chomsky, 1956). DFA induction from finite sets of labeled examples finds applications in several domains including *syntactic pattern recognition* (Fu, 1982), *intelligent autonomous agents* (Carmel & Markovitch, 1996), and *language acquisition* (Feldman *et al.*, 1990).

Exact learning of the target DFA is known to be a hard problem in that no efficient (polynomial time) algorithms are known that exactly learn a target DFA from an arbitrary set of labeled examples (Gold, 1978; Pitt & Warmuth, 1988). DFA learning can be made tractable if we assume that some additional information is available to the learner. Trakhtenbrot and Barzdin described a polynomial time algorithm for constructing the smallest DFA consistent with a *complete labeled sample* i.e., a sample that includes all labeled examples up to a particular length (Trakhtenbrot & Barzdin, 1973). Angluin showed that DFA can be exactly learned in polynomial time with the help of a *minimally adequate teacher* (MAT) capable of answering membership and equivalence queries. Oncina and Garcia's *regular positive and negative inference* (RPNI) algorithm is guaranteed to identify a DFA that is consistent with a given sample S in time polynomial in the sum of lengths of the examples in S (Oncina & García, 1992). Further, if S is a superset of a *characteristic set* (see section 2.1) for the target DFA then the DFA output by RPNI is guaranteed to be equivalent to the target (Oncina & García, 1992; Dupont, 1996b).

Even approximate learnability of DFA (in the *probably approximately correct* (PAC) sense (Valiant, 1984)) was proven to be a hard problem (see (Pitt & Warmuth, 1989; Kearns & Valiant, 1989)). The PAC model's requirement of learnability under all conceivable distributions is often considered too stringent for practical learning scenarios. Pitt's paper identified the following open research problem: "*Are DFA's PAC-identifiable if examples are drawn from the uni-*

form distribution, or some other known simple distribution?" (Pitt, 1989). Using a variant of Trakhtenbrot and Barzdin's algorithm (Trakhtenbrot & Barzdin, 1973), Lang empirically demonstrated that random DFAs are approximately learnable from a sparse uniform sample (Lang, 1992). However, exact identification of the target DFA was not possible even in the average case with a randomly drawn training sample.

Li and Vitányi proposed a model for PAC learning with *simple* examples called the *simple PAC* model wherein the class of underlying probability distributions is restricted to *simple* distributions. In this paper, we present a method for efficient PAC learning of DFA from simple examples and prove that the class of *simple* DFA is learnable under the simple PAC model. We demonstrate that for each DFA in the class of *simple* DFA there exists a polynomial sized characteristic set S_c of *simple examples*. Further, we show that a polynomial sized set of examples S drawn at random according to the *universal distribution* ($\mathbf{m}$) contains the characteristic set S_c with arbitrarily high probability. The RPNI algorithm on input S is thus guaranteed to return a canonical representation of the target DFA (with very high probability).

2 Preliminaries

Let Σ be a finite set of symbols called the *alphabet*; Σ^* be the set of strings over Σ; α, β, γ be strings in Σ^*; and $|\alpha|$ be the length of the string α. λ is a special string called the *null* string and has length 0. Given a string $\alpha = \beta\gamma$, β is the *prefix* of α and γ is the *suffix* of α. Let $Pref(\alpha)$ denote the set of all prefixes of α. A *language* L is a subset of Σ^*. The set $Pref(L) = \{\alpha \mid \alpha\beta \in L\}$ is the set of *prefixes* of the language and the set $L_\alpha = \{\beta \mid \alpha\beta \in L\}$ is the set of *tails* of α in L. The *standard order* of strings of the alphabet Σ is denoted by $<$. The standard enumeration of strings over $\Sigma = \{a, b\}$ is $\lambda, a, b, aa, ab, ba, bb, aaa, \ldots$ The set of *short prefixes* $S_p(L)$ of a language L is defined as $S_p(L) = \{\alpha \in Pref(L) \mid \not\exists \beta \in \Sigma^*$ such that $L_\alpha = L_\beta$ and $\beta < \alpha\}$. The *kernel* $N(L)$ of a language L is defined as $N(L) = \{\lambda\} \cup \{\alpha a \mid \alpha \in S_p(L), a \in \Sigma, \alpha a \in Pref(L)\}$. Given two sets S_1 and S_2, let $S_1 \backslash S_2$ and $S_1 \oplus S_2$ denote the *set difference* and the *symmetric difference* respectively. Let ln and lg denote the log to the bases e and 2 respectively.

2.1 FINITE AUTOMATA

A *deterministic* finite state automaton (DFA) is a quintuple $A = (Q, \delta, \Sigma, q_0, F)$ where, Q is a finite set of states, Σ is the finite alphabet, $q_0 \in Q$ is the start state, $F \subseteq Q$ is the set of accepting states, and δ is the transition function: $Q \times \Sigma \longrightarrow Q$. A state $d_0 \in Q$ such that $\forall a \in \Sigma$, $\delta(d_0, a) = d_0$ is called a *dead* state. A state $q \in Q$ that is not dead is called a *live* state. The extension of δ to handle input strings is standard and is denoted by δ^*. The set of all strings accepted by A is its language, $L(A)$. The language $L(A)$ of a DFA is called a *regular language*.

Given any FSA $A^{'}$, there exists a minimum state DFA (also called the *canonical DFA*, A) such that $L(A) = L(A^{'})$. Without loss of generality, we will assume that the target DFA being learned is a canonical DFA. Let N denote the number of states of A. It can be shown that any canonical DFA has at most one dead state (Hopcroft & Ullman, 1979). One can define a standard encoding of DFA as binary strings such that any DFA with N states is encoded as a string of length $O(N \lg N)$. A labeled example $(\alpha,\ c(\alpha))$ for A is such that $\alpha \in \Sigma^*$ and $c(\alpha) = +$ if $\alpha \in L(A)$ (i.e., α is a positive example) or $c(\alpha) = -$ if $\alpha \notin L(A)$ (i.e., α is a negative example). Let S^+ and S^- denote the set of *positive* and *negative* examples of A respectively. A is consistent with a *sample* $S = S^+ \cup S^-$ if it accepts all positive examples and rejects all negative examples. A set S^+ is said to be *structurally complete* with respect to a DFA A if S^+ covers each transition of A (except the transitions associated with the dead state d_0) and uses every element of the set of final states of A as an accepting state (Pao & Carr, 1978; Parekh & Honavar, 1993; Dupont *et al.*, 1994). It can be verified that the set $S^+ = \{b, aa, aaaa\}$ is structurally complete with respect to the DFA in Fig. 1.

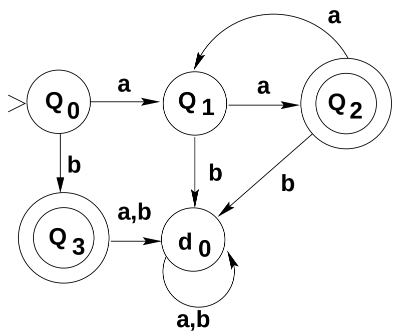

Figure 1: Deterministic Finite State Automaton.

A sample $S = S^+ \cup S^-$ is said to be *characteristic* with respect to a regular language L (with a canonical acceptor A) if it satisfies the following two conditions (Oncina & García, 1992):

- $\forall \alpha \in N(L)$, if $\alpha \in L$ then $\alpha \in S^+$ else $\exists \beta \in \Sigma^*$ such that $\alpha\beta \in S^+$.

- $\forall \alpha \in S_p(L), \forall \beta \in N(L)$, if $L_\alpha \neq L_\beta$ then $\exists \gamma \in \Sigma^*$ such that ($\alpha\gamma \in S^+$ and $\beta\gamma \in S^-$) or ($\beta\gamma \in S^+$ and $\alpha\gamma \in S^-$).

Intuitively, $S_p(L)$, the set of short prefixes of L is a *live complete* set with respect to A in that for each live state $q \in Q$, there is a string $\alpha \in S_p(L)$ such that $\delta^*(q_0, \alpha) = q$. The kernel $N(L)$ includes the set of short prefixes as a subset. Thus, $N(L)$ is also a live complete set with respect to A. Further, $N(L)$ covers every transition between each pair of live states of A. i.e., for all live states $q_i, q_j \in Q$, for all $a \in \Sigma$, if $\delta(q_i, a) = q_j$ then there exists a string $\beta \in N(L)$ such that $\beta = \alpha a$ and $\delta^*(q_0, \alpha) = q_i$. Thus, condition 1 above which identifies a suitably defined suffix $\beta \in \Sigma^*$ for each string $\alpha \in N(L)$ such that the augmented string $\alpha\beta \in L$ implies structural completeness with respect to A. Condition 2 implies that for any two distinct states of A there is a suffix γ that would correctly distinguish them. In other words, for any $q_i, q_j \in Q$ where $q_i \not\equiv q_j$, $\exists \gamma \in \Sigma^*$ such that $\delta^*(q_i, \gamma) \in F$ and $\delta^*(q_j, \gamma) \notin F$ or vice-versa. Given the language L corresponding to the DFA A in Fig. 1, the set of short prefixes is $S_p(L) = \{\lambda, a, b, aa\}$ and the kernel is $N(L) = \{\lambda, a, b, aa, aaa\}$. It can be easily verified that the set $S = S^+ \cup S^-$ where $S^+ = \{b, aa, aaaa\}$ and $S^- = \{\lambda, a, aaa, baa\}$ is a characteristic sample for L.

2.2 PROBABLY APPROXIMATELY CORRECT AND PROBABLY EXACTLY CORRECT LEARNING OF DFA

Let $\mathcal{X}$ denote the *sample space* defined as the set of all strings Σ^*. Let $x \subseteq \mathcal{X}$ denote a *concept*. For our purpose, x is a *regular language*. We identify the concept with the corresponding DFA and denote the class of all DFA as the *concept class* $\mathcal{C}$. The *representation* $\mathcal{R}$ that assigns a name to each DFA in $\mathcal{C}$ is defined as a function $\mathcal{R} : \mathcal{C} \longrightarrow \{0,1\}^*$. $\mathcal{R}$ is the set of standard encodings of the DFA in $\mathcal{C}$. Assume that there is an unknown and arbitrary but fixed distribution $\mathcal{D}$ according to which the examples of the target concept are drawn. In the context of learning DFA, $\mathcal{D}$ is restricted to a probability distribution on strings of Σ^* of length at most m.

Definition 1 *(due to (Pitt, 1989))*
DFAs are PAC-identifiable iff *there exists a (possibly randomized) algorithm $\mathcal{A}$ such that on input of any parameters ϵ and δ, for any DFA M of size N, for any number m, and for any probability distribution $\mathcal{D}$ on strings of Σ^* of length at most m, if $\mathcal{A}$ obtains labeled examples of M generated according to the distribution $\mathcal{D}$, then $\mathcal{A}$ produces a DFA M' such that with probability at least $1 - \delta$, the probability (with respect to distribution $\mathcal{D}$) of the set $\{\alpha \mid \alpha \in L(M) \oplus L(M')\}$ is at most ϵ ($0 < \epsilon < 1$ and $0 < \delta < 1$). For polynomial PAC-learnability, the run time of $\mathcal{A}$ (and hence the number of randomly generated examples obtained by $\mathcal{A}$) is required to be polynomial in N, m, $1/\epsilon$, $1/\delta$, and $|\Sigma|$.*

Definition 2 *DFAs are probably exactly learnable* iff *there exists an algorithm A such that on input of a parameter δ, for any DFA M of size N, for any number m, and for any probability distribution $\mathcal{D}$ on strings of Σ^* of length at most m, if $\mathcal{A}$ obtains labeled examples of M generated according to the distribution $\mathcal{D}$, then $\mathcal{A}$ produces a DFA M' such that with probability at least $1 - \delta$, M' is equivalent to M (i.e., $\Pr_{\mathcal{D}}(\{\alpha \mid \alpha \in L(M) \oplus L(M')\}) = 0$). For polynomial probably exact learnability, the run time of $\mathcal{A}$ is polynomial in N, m, $1/\delta$ and $|\Sigma|$.*

2.3 KOLMOGOROV COMPLEXITY

Note that the definition of PAC learning requires that the concept class (in this case the class of DFA) must be learnable under any arbitrary (but fixed) probability distribution. This requirement is often considered too stringent in practical learning scenarios where it is not unreasonable to assume that a learner is first provided with *simple* and *representative* examples of the target concept. Intuitively, when we teach a child the rules of *multiplication* we are more likely to first give simple examples like 3 × 4 than examples like 1377 × 428. A *representative set* of examples is one that would enable the learner to identify the target concept exactly. For example, the characteristic set of a DFA would constitute a suitable representative set. The question now is whether we can formalize what simple examples mean. *Kolmogorov complexity* provides a machine independent notion of *simplicity* of objects. The Kolmogorov complexity of an object (represented by a binary string α) is the length of the shortest binary program that computes α. Objects that have regularity in their structure (i.e., objects that can be easily compressed) have low Kolmogorov complexity. For example, consider the string $s_1 = 010101...01 = (01)^{500}$. On a particular machine M, a program to compute this string would be "*Print*

01 500 times". On the other hand consider a totally random string s_2 that cannot be compressed. A program to compute s_2 on M would would have to explicitly specify the entire string s_2. The length of the program that computes s_2 is thus shorter than that of the program that computes s_1. Thus, we could argue that s_1 has lower Kolmogorov complexity than s_2 with respect to the machine M.

We will consider the *prefix* version of the Kolmogorov complexity that is measured with respect to prefix Turing machines and denoted by K. Consider a prefix Turing machine that implements the partial recursive function $\phi : \{0,1\}^* \overset{partial}{\longrightarrow} \{0,1\}^*$. For any string $\alpha \in \{0,1\}^*$, the Kolmogorov complexity of α relative to ϕ is defined as $K_\phi(\alpha) = min\{|\pi| \mid \phi(\pi) = \alpha\}$ where $\pi \in \{0,1\}^*$ is a program input to the Turing machine. Prefix Turing machines can be effectively enumerated and there exists a *Universal Turing Machine* (U) capable of simulating every prefix Turing machine. Assume that the universal Turing machine implements the partial function ψ. The *Optimality Theorem* for Kolmogorov Complexity guarantees that for any prefix Turing machine ϕ there exists a constant c_ϕ such that for any string α, $K_\psi(\alpha) \leq K_\phi(\alpha) + c_\phi$. Note that we use the name of the Turing Machine (say M) and the partial function it implements (say ϕ) interchangeably i.e., $K_\phi(\alpha) = K_M(\alpha)$. Further, by the *Invariance Theorem* it can be shown that for any two universal machines ψ_1 and ψ_2 there is a constant $\eta \in \mathcal{N}$ (where $\mathcal{N}$ is the set of natural numbers) such that for all strings α, $|K_{\psi_1}(\alpha) - K_{\psi_2}(\alpha)| \leq \eta$. Thus, we can fix a single universal Turing machine U and denote $K(\alpha) = K_U(\alpha)$. Note that there exists a Turing machine that computes the identity function $\chi : \{0,1\}^* \longrightarrow \{0,1\}^*$ where $\forall\alpha,\ \chi(\alpha) = \alpha$. Thus, it can be shown that the Kolmogorov complexity of an object is bounded by its length i.e., $K(\alpha) \leq |\alpha| + K(|\alpha|) + \eta$ where η is a constant independent of α.

2.4 UNIVERSAL DISTRIBUTION

The set of programs for a string α relative to a Turing machine M is defined as $PROG_M(\alpha) = \{\pi \mid M(\pi) = \alpha\}$. The algorithmic probability of α relative to M is defined as $\mathbf{m}_M(\alpha) = \Pr(PROG_M)$. The algorithmic probability of α with respect to the universal Turing machine U is denoted as $\mathbf{m}_U(\alpha) = \mathbf{m}(\alpha)$. $\mathbf{m}$ is known as the Solomonoff-Levin distribution. It is the universal enumerable probability distribution, in that, it multiplicatively dominates all enumerable probability distributions. Thus, for any enumerable probability distribution P there is a constant $c \in \mathcal{N}$ such that for all strings α, $c\,\mathbf{m}(\alpha) \geq P(\alpha)$. The *Coding Theorem* due independently to Schnorr, Levin, and Chaitin (Li & Vitányi, 1997) states that $\exists\eta \in \mathcal{N}$ such that $\forall\alpha\ \mathbf{m}_M(\alpha) \leq 2^{\eta - K(\alpha)}$. Intuitively this means that if there are several programs for a string α on some machine M then there is a short program for α on the universal Turning machine (i.e., α has a low Kolmogorov complexity). By optimality of $\mathbf{m}$ it can be shown that: $\exists\eta \in \mathcal{N}$, such that $\forall\alpha \in \{0,1\}^*$, $2^{-K(\alpha)} \leq \mathbf{m}(\alpha) \leq 2^{\eta - K(\alpha)}$. We see that the universal distribution $\mathbf{m}$ assigns higher probability to simple objects (objects with low Kolmogorov complexity).

The interested reader is referred to (Li & Vitányi, 1997) for a thorough treatment of Kolmogorov complexity, universal distribution, and related topics.

3 The RPNI Algorithm

The *regular positive and negative inference* (RPNI) algorithm (Oncina & García, 1992) identifies a DFA consistent with a given set $S = S^+ \cup S^-$ of labeled examples in time polynomial in the sum of the lengths of the strings in S ($||S||$). Further, if S is a characteristic set for the target DFA then the algorithm is guaranteed to return a canonical representation of the target DFA. Given a labeled set of examples S the algorithm constructs a *prefix tree automaton* (PTA) that accepts exactly the strings in S^+ and nothing else. The states of the PTA are numbered in the standard order of the shortest strings that lead to them from the start state. For instance, the PTA corresponding to the set $S = S^+ \cup S^-$ where $S^+ = \{b, aa, aaaa\}$ and $S^- = \{\lambda, a, aaa, baa\}$ is depicted in Fig. 2. The PTA is consistent with S and is treated as the initial hypothesis.

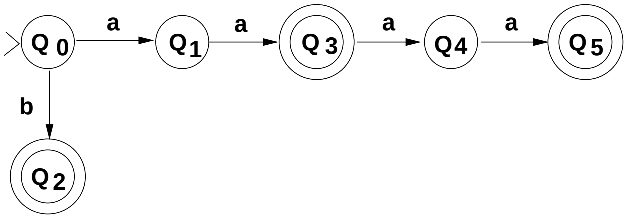

Figure 2: Prefix Tree Automaton.

A quadratic loop is used to find a more general hypothesis by systematically merging the states of the PTA in order (state 1 with state 0, state 2 with state 0, state 2 with state 1, and so on). The derived automaton obtained by merging two states is tested for consistency with the set S^-. If the derived automaton accepts any string from S^- then the merge is rejected and the remaining states are considered for merging in order. Otherwise, the derived automaton that is consistent with S is treated as the new hypothesis and the state merging continues with the states of the new hypothesis. For example, Fig. 3 depicts the derived automaton obtained by merging together the states 0 and 2 of the PTA. This derived automaton accepts the string $\lambda \in S^-$ and hence the merge is rejected.

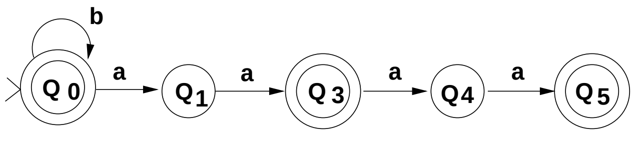

Figure 3: Derived Automaton Obtained by Merging the States 2 and 0 of the PTA.

The algorithm terminates when no further state merges result in a derived automaton that is consistent with S^-. The last consistent hypothesis is then returned as the learned DFA.

The interested reader is referred to (Dupont, 1996a) for a detailed exposition of the RPNI algorithm.

4 LEARNING SIMPLE DFA UNDER THE SIMPLE PAC MODEL

Li and Vitányi have proposed the simple PAC learning model where the class of probability distributions is restricted to *simple* distributions (Li & Vitányi, 1991). A distribution is simple if it is multiplicatively dominated by some enumerable distribution. Simple distributions include all computable distributions. Further, the *simple distribution independent learning theorem* due to Li and Vitanyí states that a concept class is learnable under universal distribution **m** *iff* it is learnable under the entire class of *simple distributions* provided the examples are drawn according to the universal distribution (Li & Vitányi, 1991). Thus, the simple PAC learning model is sufficiently general. Concept classes such as $\log n$*-term DNF* and *simple k-reversible DFA* are learnable under the simple PAC model whereas their PAC learnability in the standard sense is unknown (Li & Vitányi, 1991). We show that the class of *simple* DFA is polynomially learnable under the simple PAC learning model.

A simple DFA is one with low Kolmogorov complexity. More specifically, a DFA A with N states and a standard encoding (or canonical representation) r is simple if $K(r) = O(\lg N)$. For example, a DFA that accepts all strings of length N is a simple DFA. Note that this DFA contains a path for every string of length N and hence it has a path of Kolmogorov complexity N. In general, simple DFA might actually have very random paths. We saw in section 2.3 that a natural learning scenario would typically involve learning from a *simple* and *representative* set of examples for the target concept. We adopt Kolmogorov complexity as a measure of simplicity and define simple examples as those with low Kolmogorov complexity i.e., with Kolmogorov complexity $O(\lg N)$. Further, a characteristic set for the DFA A can be treated as its representative set. We demonstrate that for every simple DFA there exists a characteristic set of simple examples S_c.

Lemma 1 *For any N state simple DFA (with Kolmogorov complexity $O(\lg N)$) there exists a characteristic set of simple examples S_c such that the length of each string in this set is at most $2N-1$.*

Proof: Consider the following enumeration of a characteristic set of examples for a DFA $A = (Q, \delta, \Sigma, q_0, F)$ with N states[1].

1. Fix an enumeration of the shortest paths (in standard order) from the state q_0 to each state in Q except the dead state. This is the set of short prefixes of A. There are at most N such paths and each path is of length at most $N-1$.

2. Fix an enumeration of paths that includes each path identified above and its extension by each letter of the alphabet Σ. From the paths just enumerated retain only those that do not terminate in the dead state of A. This represents the kernel of A. There are at most $N(|\Sigma|+1)$ such paths and each path is of length at most N.

3. Let the characteristic set be denoted by $S_c = S_c^+ \cup S_c^-$.

   (a) For each string α identified in step 2 above, determine the first suffix β in the standard enumeration of strings such that $\alpha\beta \in L(A)$. Since $|\alpha| \leq N$, and β is the shortest suffix in the standard order it is clear that $|\alpha\beta| \leq 2N-1$. Each such $\alpha\beta$ is a member of S_c^+.

[1] Note that this enumeration strategy applies to any DFA and is not restricted to simple DFA alone.

(b) For each pair of strings (α, β) in order where α is a string identified in step 1, β is a string identified in step 2, and α and β lead to different states of A determine the first suffix γ in the standard enumeration of strings such that $\alpha\gamma \in L(A)$ and $\beta\gamma \notin L(A)$ or vice versa. Since $|\alpha| \leq N-1$, $|\beta| \leq N$, and γ is the shortest distinguishing suffix for the states represented by α and β it is clear that $|\alpha\gamma|, |\beta\gamma| \leq 2N-1$. The accepted string from among $\alpha\gamma$ and $\beta\gamma$ is a member of S_c^+ and the rejected string is a member of S_c^-.

Trivial upper bounds on the sizes of S_c^+ and S_c^- are $|S_c^+| \leq N^2(|\Sigma|+1)+N(|\Sigma|)$, $|S_c^-| \leq N^2(|\Sigma|+1)-N$. Thus, $|S_c| = \lceil(2N^2|\Sigma|+2N^2+N|\Sigma|-N)\rceil$ i.e., $|S_c| \leq kN^2$ where k is a constant . Further, the length of each string in S_c is less than $2N-1$.

The strings in S_c can be ordered in some way such that individual strings can be identified by an index of length at most $\lg(kN^2) = O(\lg N)$ bits. There exists a Turing machine M that implements the above algorithm for constructing the set S_c. M can take as input an encoding of a simple DFA of length $O(\lg N)$ bits and an index of length $O(\lg N)$ bits and output the corresponding string α belonging to S_c. Thus, $\forall \alpha \in S_c$,

$$\begin{aligned} K(\alpha) &\leq k_1 \lg N + k_2 \lg N \\ &= O(\lg N) \end{aligned}$$

This proves the lemma. □

Lemma 2 *Suppose a sample S is drawn according to* $\mathbf{m}$. *For $0 < \delta \leq 1$, if $|S| = O(N^k \lg(\frac{1}{\delta}))$ then with probability greater than $1-\delta$, $S_c \subseteq S$ where k is a constant.*

Proof: From lemma 1 we know that $\forall \alpha \in S_c,\ K(\alpha) = O(\lg N)$. Further, $|S_c| = O(N^2)$. By definition, $\mathbf{m}(\alpha) \geq 2^{-K(\alpha)}$. Thus, $\mathbf{m}(\alpha) \geq 2^{-k_1 \lg N}$ or equivalently $\mathbf{m}(\alpha) \geq N^{-k_1}$ where k_1 is a constant.
$\Pr(\alpha \in S_c$ is not sampled in one random draw$) \leq (1 - N^{-k_1})$
$\Pr(\alpha \in S_c$ is not sampled in $|S|$ random draws $\leq (1 - N^{-k_1})^{|S|}$
$\Pr($ some $\alpha \in S_c$ is not sampled in $|S|$ random draws $\leq |S_c|(1 - N^{-k_1})^{|S|}$
$\Pr($ some $\alpha \in S_c$ is not sampled in $|S|$ random draws $\leq k_2N^2(1 - N^{-k_1})^{|S|}$ since $|S_c| = O(N^2)$
$\Pr(S_c \not\subseteq S) \leq k_2N^2(1 - N^{-k_1})^{|S|}$

We want this probability to be less than δ.

$$\begin{aligned} k_2N^2(1 - N^{-k_1})^{|S|} &\leq \delta \\ k_2N^2(e^{-N^{-k_1}})^{|S|} &\leq \delta^2 \\ \ln(k_2) + \ln(N^2) - N^{-k_1}|S| &\leq \ln(\delta) \\ N^{k_1}(\ln(\frac{1}{\delta}) + \ln(k_2) + \ln(N^2)) &\leq |S| \\ O(N^k \lg(\frac{1}{\delta})) &= |S| \\ \text{where } k \text{ replaces } k_1 \end{aligned}$$

Thus, $\Pr(S_c \subseteq S) \geq 1 - \delta$. □

We now prove that the class of simple DFA is polynomially exactly learnable under $\mathbf{m}$.

Theorem 1 *For all N, the class $\mathcal{C}^{\leq N}$ of simple DFA whose canonical representations have at most N states is probably exactly learnable under the simple PAC model.*

Proof: Let A be a simple DFA with at most N states. Let S_c be a characteristic sample of A enumerated as described in lemma 1 above. Recall that the examples in S_c are simple (i.e., each example has Kolmogorov complexity $O(\lg N)$). Now consider the algorithm $\mathcal{A}$ in Fig. 4 that draws a sample S with the following properties:

1. $S = S^+ \cup S^-$ is a set of positive and negative examples corresponding to the target DFA A.

2. The examples in S are drawn at random according to the distribution $\mathbf{m}$.

3. $|S| = O(N^k \lg(\frac{1}{\delta}))$.

```
Algorithm $\mathcal{A}$

Input: $N, 0 < \delta \leq 1$
Output: A DFA $M$

begin
   • Randomly draw a labeled sample $S$
     according to $\mathbf{m}$.
   • Retain only those examples in $S$
     that have length at most $2N-1$.
   • $M = RPNI(S)$
   • return $M$
end
```

Figure 4: A Probably Exact Algorithm for Learning Simple DFA.

Lemma 1 showed that for every simple DFA A there exists a characteristic set of simple examples S_c where each example is of length at most $2N-1$. Lemma 2 showed that if a labeled sample S of size $O(N^k \lg(\frac{1}{\delta}))$ is randomly drawn according to $\mathbf{m}$ then with probability greater than $1-\delta$, $S_c \subseteq S$. The RPNI algorithm is guaranteed to return a canonical representation of the target DFA A if the set of examples S provided is a superset of a characteristic set S_c. Since the size of S is polynomial in N and $1/\delta$ and the length of each string in S is restricted to $2N-1$, the RPNI algorithm, and thus the algorithm $\mathcal{A}$ can be implemented to run in time polynomial in N and $1/\delta$. Thus, with probability greater than $1-\delta$, $\mathcal{A}$ is guaranteed to return a canonical representation of the target DFA A. This proves that the class $\mathcal{C}^{\leq N}$ of simple DFA whose canonical representations have at most N states is exactly learnable with probability greater than $1-\delta$. □

5 DISCUSSION

The problem of exactly learning the target DFA from an arbitrary set of labeled examples and the problem of approximating the target DFA from labeled examples under Valiant's PAC learning framework are both known to be hard problems. Thus, the question as to whether DFA are efficiently learnable under some restricted yet fairly general and practically useful classes of distributions was clearly of interest. In this paper, we have answered this question in the affirmative for the class of *simple* DFA by demonstrating that the class of simple DFA is polynomially learnable under the universal distribution $\mathbf{m}$ (the simple PAC learning model).

The class of simple distributions includes a large variety of probability distributions (including all computable distributions). It has been shown that a concept class is efficiently learnable under the universal distribution if and only if it is efficiently learnable under each simple distribution provided that sampling is done according to the universal distribution (Li & Vitányi, 1991). This raises the possibility of using sampling under the universal distribution to learn under all computable distributions. However, the universal distribution is not computable. Whether one can instead get by with a polynomially computable approximation of the universal distribution remains an open question. It is known that the universal distribution for the class of polynomially-time bounded simple distributions is computable in exponential time (Li & Vitányi, 1991). This opens up a number of interesting possibilities for learning under simple distributions.

Denis *et al* proposed a model of learning (known as the PACS model) where examples are drawn at random according to the universal distribution by a teacher that is knowledgeable about the target concept (Denis *et al.*, 1996). In this model examples that have low Kolmogorov complexity given a canonical representation (r) of the target concept (i.e., $K(x|r) = O(\lg N)$) are treated as *simple* examples. Further, the probability of drawing an example x is given by $\mathbf{m}_r(x) = 2^{-K(x|r)+O(1)}$. In related work we have shown that the entire class of DFA is efficiently learnable in the PACS model (Parekh & Honavar, 1997). Recently, Castro and Guijarro have independently shown that if a concept class is learnable under the PACS learning model then the set of *simple* concepts of that concept class are learnable under the simple PAC model (Castro & Guijarro, 1998). An analysis of the relationship between the PACS and simple PAC learning models and other popular models for learning in *helpful environments* such as learning from example based queries (Angluin, 1988), learning from polynomial teaching sets (Goldman & Mathias, 1993; Gold, 1978), and mistake bounded learning (Littlestone, 1988) appears in (Parekh & Honavar, 1999).

A related question of interest has to do with the nature of environments that can be modeled by simple distributions. In particular, if Kolmogorov complexity is an appropriate measure of the intrinsic complexity of objects in nature and if nature (or the teacher) has a propensity for simplicity, then it stands to reason that the examples presented to the learner by the environment are likely to be generated by a simple distribution. Against this background, empirical evaluation of the performance of the proposed algorithms using examples that come from natural domains is clearly of interest. Also of interest are investigation of other interesting and practically useful concept classes that might be learnable from simple examples.

References

Angluin, D. (1988). Queries and Concept Learning. *Machine Learning*, **2**(4), 319–342.

Carmel, D., & Markovitch, S. (1996). Learning Models of Intelligent Agents. *Pages 62–67 of: Proceedings of the AAAI-96 (vol. 1).* AAAI Press/MIT Press.

Castro, J., & Guijarro, D. (1998). *Query, PACS and simple-PAC Learning.* Tech. rept. LSI-98-2-R. Universitat Polytéctica de Catalunya, Spain.

Chomsky, N. (1956). Three Models for the Description of Language. *PGIT*, **2**(3), 113–124.

Denis, F., D'Halluin, C., & Gilleron, R. (1996). PAC Learning with Simple Examples. *STACS'96 - Proceedings of the* 13^{th} *Annual Symposium on the Theoretical Aspects of Computer Science*, 231–242.

Dupont, P. (1996a). Incremental Regular Inference. *Pages 222–237 of:* Miclet, L., & Higuera, C. (eds), *Proceedings of the Third ICGI-96, Lecture Notes in Artificial Intelligence 1147*. Montpellier, France: Springer.

Dupont, P. (1996b). *Utilisation et Apprentissage de Modèles de Language pour la Reconnaissance de la Parole Continue.* Ph.D. thesis, Ecole Normale Supérieure des Télécommunications, Paris, France.

Dupont, P., Miclet, L., & Vidal, E. (1994). What is the Search Space of the Regular Inference? *Pages 25–37 of: Proceedings of the Second International Colloquium on Grammatical Inference (ICGI'94).*

Feldman, J. A., Lakoff, G., Stolcke, A., & Weber, S. H. (1990). *Miniature Language Acquisition: A Touchtone for Cognitive Science.* Tech. rept. TR-90-009. International Computer Science Institute, Berkeley, CA.

Fu, K. (1982). *Syntactic Pattern Recognition and Applications.* Prentice Hall, N.J.

Gold, E. (1978). Complexity of Automaton Identification from Given Data. *Information and Control*, **37**(3), 302–320.

Goldman, S., & Mathias, H. (1993). Teaching a Smarter Learner. *Pages 67–76 of: Proceedings of the Workshop on Computational Learning Theory (COLT'93).* A. C. M. Press.

Hopcroft, J., & Ullman, J. (1979). *Introduction to Automata Theory, Languages, and Computation.* Addison Wesley.

Kearns, M., & Valiant, L. G. (1989). Cryptographic Limitations on Learning Boolean Formulae and Finite Automata. *Pages 433–444 of: Proceedings of the* 21^{st} *Annual ACM Symposium on Theory of Computing.* ACM, New York.

Lang, K. (1992). Random DFA's can be approximately learned from sparse uniform sample. *Pages 45–52 of: Proceedings of the 5th ACM workshop on Computational Learning Theory.*

Li, M., & Vitányi, P. (1991). Learning Simple Concepts under Simple Distributions. *SIAM Journal of Computing*, **20**(5), 911–935.

Li, M., & Vitányi, P. (1997). *An Introduction to Kolmogorov Complexity and its Applications,* 2^{nd} *edition.* New York: Springer Verlag.

Littlestone, N. (1988). Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm. *Machine Learning*, **2**, 285–318.

Oncina, J., & García, P. (1992). Inferring Regular Languages in Polynomial Update Time. *Pages 49–61 of:* Pérez, N. *et al* (ed), *Pattern Recognition and Image Analysis.* World Scientific.

Pao, T., & Carr, J. (1978). A Solution of the Syntactic Induction-Inference Problem for Regular Languages. *Computer Languages*, **3**, 53–64.

Parekh, R., & Honavar, V. (1993). Efficient Learning of Regular Languages using Teacher Supplied Positive Examples and Learner Generated Queries. *Pages 195–203 of: Proceedings of the Fifth UNB Conference on AI.*

Parekh, R., & Honavar, V. (1997). Learning DFA from Simple Examples. *Pages 116–131 of: Proceedings of the Eighth International Workshop on Algorithmic Learning Theory (ALT'97), Lecture Notes in Artificial Intelligence 1316.* Sendai, Japan: Springer. Also presented at the *Workshop on Grammar Inference, Automata Induction, and Language Acquisition* (ICML'97), Nashville, TN. July 12, 1997.

Parekh, R., & Honavar, V. (1999). *On the Relationship Between Models for Learning in Helpful Environments.* (Submitted for review).

Pitt, L. (1989). Inductive Inference, DFAs and Computational Complexity. *Pages 18–44 of: Analogical and Inductive Inference, Lecture Notes in Artificial Intelligence 397.* Springer-Verlag.

Pitt, L., & Warmuth, M. K. (1988). Reductions among prediction problems: on the difficulty of predicting automata. *Pages 60–69 of: Proceedings of the* 3^{rd} *I.E.E.E. Conference on Structure in Complexity Theory.*

Pitt, L., & Warmuth, M. K. (1989). The minimum consistency DFA problem cannot be approximated within any polynomial. *Pages 421–432 of:*

Proceedings of the 21^{st} *ACM Symposium on the Theory of Computing.* ACM.

Trakhtenbrot, B., & Barzdin, Ya. (1973). *Finite Automata: Behavior and Synthesis.* Amsterdam: North Holland Publishing Company.

Valiant, L. (1984). A Theory of the Learnable. *Communications of the ACM*, **27**, 1134–1142.