PAC-Learnability of Probabilistic Deterministic Finite State Automata in Terms of Variation Distance*

Nick Palmer and Paul W. Goldberg

Dept. of Computer Science, University of Warwick,
Coventry CV4 7AL, U.K.
{npalmer, pwg}@dcs.warwick.ac.uk
http://www.dcs.warwick.ac.uk/research/acrg

Abstract. We consider the problem of PAC-learning distributions over strings, represented by probabilistic deterministic finite automata (PDFAs). PDFAs are a probabilistic model for the generation of strings of symbols, that have been used in the context of speech and handwriting recognition, and bioinformatics. Recent work on learning PDFAs from random examples has used the KL-divergence as the error measure; here we use the variation distance. We build on recent work by Clark and Thollard, and show that the use of the variation distance allows simplifications to be made to the algorithms, and also a strengthening of the results; in particular that using the variation distance, we obtain polynomial sample size bounds that are independent of the expected length of strings.

1 Introduction

A probabilistic deterministic finite automaton (PDFA) is a deterministic finite automaton that has, for each state, a probability distribution over the transitions going out from that state. Thus, a PDFA defines a probability distribution over the set of strings over its alphabet. The topic of PAC-learning of PDFAs was introduced by Ron et al. [10], where they show how to PAC-learn *acyclic* PDFAs, and apply the algorithm to speech and handwriting recognition. Recently Clark and Thollard [3] presented an algorithm that PAC-learns general PDFAs, using the Kullback-Leibler divergence (see Cover and Thomas [4]) as the error measure (the distance between the true distribution defined by the target PDFA, and the hypothesis returned by the algorithm). The algorithm is polynomial in three parameters: the number of states, the "distinguishability" of states, and the expected length of strings generated from any state of the target PDFA.

* This work was supported by EPSRC Grant GR/R86188/01. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

S. Jain, H.U. Simon, and E. Tomita (Eds.): ALT 2005, LNAI 3734, pp. 157–170, 2005.

In this paper we study the same problem, using variation distance instead of Kullback-Leibler divergence. The general message of this paper is that this modification allows some strengthening and simplifications of the resulting algorithms. The main one is that – as conjectured in [3] – a polynomial bound on the sample-size requirement is obtained that does not depend on the length of strings generated by the automaton. We also have no need for a distinguished "final symbol" that must terminate all data strings, or a "ground state" in the automaton constructed by the algorithm.

The variation distance between probability distributions D and D' is the L_1 distance; for a discrete domain X, it is $L_1(D, D') = \sum_{x \in X} |D(x) - D'(x)|$. KL divergence is in a strong sense a more "sensitive" measure than variation distance; this was pointed out in Kearns et al. [8], which introduced the general topic of PAC-learning probability distributions. In Cryan et al. [5] a smoothing technique is given for distributions over the boolean domain — an algorithm that PAC learns distributions using the variation distance can be converted to an algorithm that PAC learns using the KL-divergence. (Abe et al. [1] give a similar result in the context of learning p-concepts.) Over the domain Σ^* (strings of unrestricted length over alphabet Σ) that technique does not apply, which is why we might expect stronger results as a result of switching to the variation distance.

In the context of pattern classification, the variation distance is useful in the following sense. Suppose that we seek to classify labelled data by fitting distributions to each label class, and using the Bayes classifier on the hypothesis distributions. (See [6] for a discussion of the motivation for this general approach.) We show in [9] that PAC learnability using the variation distance implies agnostic PAC classification. The corresponding result for KL-divergence is that the expected negative log-likelihood cost is close to optimum.

Our approach follows [3], in that we divide the algorithm into two parts. The first (Algorithm 1 of Figure 1) finds a DFA that represents the deterministic structure of the hypothesis, and the second (Algorithm 2 of Figure 2) finds estimates of the transition probabilities. Algorithm 1 constructs (with high probability) a DFA whose states and transitions are a subset of those of the target. Algorithm 2 learns the transition probabilities by following the paths of random strings through the DFA constructed by Algorithm 1. We take advantage of the fact that commonly-used transitions can be estimated more precisely.

2 Terms and Definitions

A probabilistic deterministic finite state automaton (PDFA) stochastically generates strings of symbols. The automaton has a finite set of states - one of which is denoted as the initial state. The automaton generates a string by making transitions between states (starting at the initial state), each occurring with a constant probability specifically associated with that transition, and a symbol is output as a function of the transition. The automaton halts when the *final state* is reached.

Definition 1. *A PDFA A is a sextuple $(Q, \Sigma, q_0, q_f, \tau, \gamma)$, where*

- *Q is a finite set of states,*
- *Σ is a finite set of symbols (the alphabet),*
- *$q_0 \in Q$ is the initial state,*
- *$q_f \notin Q$ is the final state,*
- *$\tau : Q \times \Sigma \to Q \cup \{q_f\}$ is the (partial) transition function,*
- *$\gamma : Q \times \Sigma \to [0, 1]$ is the function giving the probability of a symbol occurring from any state.*

Where appropriate, we extend the use of τ and γ to strings:

$$\tau(q, \sigma_1\sigma_2...\sigma_k) = \tau(\tau(q, \sigma_1), \sigma_2...\sigma_k)$$
$$\gamma(q, \sigma_1\sigma_2...\sigma_k) = \gamma(q, \sigma_1).\gamma(\tau(q, \sigma_1), \sigma_2...\sigma_k)$$

We use the pair (q, σ) to denote the transition from state $q \in Q$ labeled with character $\sigma \in \Sigma$. Note that $\gamma(q, \sigma) = 0$ when $\tau(q, \sigma)$ is undefined. It should also be noted that the output probabilities from each state sum to one:

$$\forall q \in Q : \sum_{\sigma \in \Sigma} \gamma(q, \sigma) = 1.$$

We assume that the final state can be reached from any state of the automaton, that is, $\forall q \in Q, \exists s \in \Sigma^* : \tau(q, s) = q_f \wedge \gamma(q, s) > 0$. It follows that the PDFA A defines a probability distribution over all strings in Σ^*. Let $D_A(s)$ denote the probability that A generates $s \in \Sigma^*$, so we have

$$D_A(s) \;=\; \gamma(q_0, s) \text{ for } s \text{ such that } \tau_A(q_0, s) = q_f.$$

We define $D_A(q)$ to be the probability that a random string generated by A uses state $q \in Q$. Thus $D_A(q)$ is the probability that $s \sim D_A$ (i.e. s sampled from distribution D_A) has a prefix p with $\tau(q_0, p) = q$. In a similar way, $D_A(q, \sigma)$ is the probability that a random string generated by A uses transition (q, σ) — the probability that a random string $s \sim D_A$ has a prefix $p\sigma$ with $\tau(q_0, p) = q$.

Suppose D and D' are probability distributions over Σ^*. The variation (L_1) distance between D and D' is $L_1(D, D') = \sum_{s \in \Sigma^*} |D(s) - D'(s)|$. A class $\mathcal{C}$ of probability distributions is PAC-learnable by algorithm $\mathcal{A}$ with respect to the variation distance if the following holds. Given parameters $\epsilon > 0$, $\delta > 0$, and access to samples from $D_A \in \mathcal{C}$, using runtime and sample size polynomial in ϵ^{-1} and δ^{-1}, $\mathcal{A}$ should, with probability $1 - \delta$, output a distribution D_H with $L_1(D_A, D_H) < \epsilon$. If $\mathcal{C}$ is described in terms of additional parameters that represent the complexity of D_A, then we require $\mathcal{A}$ to be polynomial in these parameters as well as ϵ^{-1} and δ^{-1}.

3 Constructing the PDFA

The algorithm is shown in Figure 1. We have the following parameters (in addition to the PAC parameters ϵ and δ):

- $|\Sigma|$: the alphabet size,
- n: an upper bound on the number of states of the target automaton,
- μ: a lower bound on distinguishability, defined below.

In the context of learning using the KL-divergence, a simple class of PDFAs (see [3]) can be constructed to show that the parameters above are insufficient for PAC learnability in terms of just those parameters. In [3], parameter L is also used, denoting the expected length of strings.

We construct a digraph $G = \langle V, E\rangle$ with labelled edges (V is a set of vertices and $E \subseteq V \times \Sigma \times V$ is a set of edges). Each edge is labelled with a letter $\sigma \in \Sigma$. Note that due to the deterministic nature of the automaton, there can be at most one vertex v_q such that $(v_p, \sigma, v_q) \in E$ for any $v_p \in V$ and $\sigma \in \Sigma$.

From the target automaton A we generate a hypothesis automaton H using a variation on the method described by Clark and Thollard [3] utilising *candidate nodes*, where the L_∞ norm between the suffix distributions of states is used to distinguish between them (as studied also in [7, 10]). We define a Candidate Node in the same way as [3]. Suppose G is a graph whose vertices correspond to a subset of the states of A. Initially G will have a single vertex corresponding to the initial state; G is then constructed in a greedy incremental fashion.

Definition 2. *A candidate node in hypothesis graph G is a pair (u, σ) (also denoted $\hat{q}_{u,\sigma}$), where u is a node in the graph and $\sigma \in \Sigma$ where $\tau_G(u, \sigma)$ is undefined. It will have an associated multiset $S_{u,\sigma}$.*

The L_∞-norm is a measure of distance between a pair of distributions, defined as follows.

Definition 3. $L_\infty(D, D') = \max_{s\in\Sigma^*} |D(s) - D'(s)|$.

Let $D_q(s)$ denote the distribution over strings generated using state q as the initial state, so that

$$D_q(s) = \gamma(q, s) \text{ for } s \text{ such that } \tau(q, s) = q_f.$$

As in [10, 3], we say that a pair of nodes (q_1, q_2) are *distinguishable* if $L_\infty(D_{q_1}, D_{q_2}) = \max_{s\in\Sigma^*} |D_{q_1}(s) - D_{q_2}(s)| \geq \mu$. We define as follows the $\hat{L}_\infty$-norm (an empirical version of the L_∞-norm) with respect to multisets of strings S_{q_1} and S_{q_2}, where S_{q_1} and S_{q_2} have been respectively sampled from D_{q_1} and D_{q_2}.

Definition 4. *For nodes q_1 and q_2, with associated multisets S_{q_1} and S_{q_2},*

$$\hat{L}_\infty\left(D_{q_1}, D_{q_2}\right) = \max_{s\in\Sigma^*} \left(\left| \frac{|s \in S_{q_1}|}{|S_{q_1}|} - \frac{|s \in S_{q_2}|}{|S_{q_2}|} \right| \right)$$

where D_q is the empirical distribution over the strings in the multiset S_q associated with q, and where $|s \in S_q|$ is the number of occurrences of string s in multiset S.

Algorithm 1 *Construct Automaton.*

Hypothesis Graph $G = \langle V, E \rangle = \langle \{q_0\}, \emptyset \rangle$
$m_0 = \frac{n|\Sigma|}{\mu^2 \delta'}$
$N = \max\left(\frac{8n^2|\Sigma|^2}{\epsilon^2} \ln\left(\frac{2n^2|\Sigma|^2}{\delta'}\right), \frac{4m_0 n|\Sigma|}{\epsilon}\right)$
complete = false
repeat
  % *create candidate node for each undefined transition from each vertex in* G
  for each vertex $v \in V$
    for each symbol $\sigma \in \Sigma$, where $\tau_G(v, \sigma)$ is undefined
      create a candidate node $\hat{q}_{v,\sigma}$ with associated multiset $S_{v,\sigma} = \emptyset$
  generate a sample S of N strings iid from D_A
  for each string $s \in S$, where $s = r\sigma' t$ and $\hat{q}_{\tau_G(q_0,r),\sigma'}$ is a candidate node
    $S_{\tau(q_0,r),\sigma'} \leftarrow S_{\tau(q_0,r),\sigma'} \cup \{t\}$
  identify candidate node $\hat{q}_{u,\sigma''}$ with the largest multiset, $S_{u,\sigma''}$
  if $(|S_{u,\sigma''}| \geq m_0)$
    replace multiset $S_{u,\sigma''}$ with m_0 suffixes chosen iid from $S_{u,\sigma''}$
    if $\left(\exists v \in V : \hat{L}_\infty\left(D_{\hat{q}_{u,\sigma''}}, D_v\right) \leq \frac{\mu}{2}\right)$ % *candidate "looks like" existing node*
      add edge (u, σ'', v) to E
    else
      add node $\hat{q}_{u,\sigma''}$ to V, with multiset $S_{u,\sigma''}$
      add edge $(u, \sigma'', \hat{q}_{u,\sigma''})$ to E
  else
    complete = true
  delete all candidate nodes $q \notin V$
until(complete)
return G

Fig. 1. Constructing the underlying graph

The algorithm uses two quantities, m_0 and N. m_0 is the number of suffixes required in the multiset of a candidate node for the node to be added as a state (or as a transition) to the hypothesis. It will be shown that m_0 is a sufficiently large number to allow us to establish that the distribution over suffixes in the multiset that begin at state q is likely to approximate the true distribution D_q over suffixes at that state. N is the number of strings generated iid during each iteration of the algorithm. Polynomial expressions for m_0 and N are given in Algorithm 1.

We show that the probability of Algorithm 1 failing to adequately learn the structure of the automaton is upper bounded by δ'. In Section 5 we show that the transition probabilities are learnt by Algorithm 2 with a failure probability of at most δ''. Overall, the probability of the algorithms failing to learn the target PDFA within a variation distance of ϵ is at most δ, for $\delta = \delta' + \delta''$.

Algorithm 1 differs from [3] as follows. We do not introduce a *ground node* - a node to catch any undefined transitions in the hypothesis graph so as to give a probability greater than zero to the generation of any string. Instead, any state q, for which $D_A(q) < \frac{\epsilon}{2n|\Sigma|}$ can be discarded - no corresponding node is formed in our hypothesis graph. There is only a small probability that a string is generated such that our hypothesis automaton rejects it (there is no corresponding path through the graph), which means that the contribution to the overall variation distance is very small.

4 Analysis of PDFA Construction Algorithm

Theorem 1. *Given that for each pair* (q_1, q_2) *of distinct states in* A, $L_\infty(D_{q_1}, D_{q_2}) > \mu$, *the corresponding*[1] *states* $(\hat{q}_1, \hat{q}_2)$ *in hypothesis automaton* H *are distinguished* $(\hat{L}_\infty(D_{\hat{q}_1}, D_{\hat{q}_2}) > \frac{\mu}{2})$ *with probability at least* $1 - \frac{\delta'}{2}$, *if* $m_0 \geq \frac{n|\Sigma|}{\mu^2\delta'}$.

Proof. States q_1 and q_2 are distinguished if there exists a string s' such that:

$$\left| \frac{|s' \in S_{q_1}|}{|S_{q_1}|} - \frac{|s' \in S_{q_2}|}{|S_{q_2}|} \right| \geq \frac{\mu}{2}$$

Due to the assumption that $L_\infty(D_{q_1}, D_{q_2}) > \mu$, a string s'' exists such that:

$$|D_{q_1}(s'') - D_{q_2}(s'')| > \mu$$

We give a sufficient sample size, such that the proportion of each string occurring in the sample is within $\frac{\mu}{4}$ of the expected proportion (with probability at least $1 - \frac{\delta'}{2n^2|\Sigma|^2 m_0}$). From Hoeffding's Inequality (see for example [2]) we obtain that, for state q and string s:

$$\Pr\left(\left| \left(\frac{|s \in \hat{S}_q|}{|\hat{S}_q|} \right) - D_q(s) \right| \geq \frac{\mu}{4} \right) \leq e^{-2m_0\left(\frac{\mu}{4}\right)^2} \tag{1}$$

A value of m_0 is chosen such that for sufficiently large n:

$$e^{-\frac{m_0\mu^2}{8}} \leq \frac{\delta'}{2n^2|\Sigma|^2 m_0} \tag{2}$$

It can be verified that this is satisfied if we choose $m_0 \geq \frac{n^2|\Sigma|^2}{\mu^2\delta'}$.
From Equation 2 it can be seen that (for some string s and some state q):

$$\Pr\left(\left| \left(\frac{|s \in \hat{S}_q|}{|\hat{S}_q|} \right) - D_q(s) \right| \geq \frac{\mu}{4} \right) \leq e^{-2m_0\left(\frac{\mu}{4}\right)^2} \leq \frac{\delta'}{2n^2|\Sigma|^2 m_0} \tag{3}$$

[1] At every interation of the algorithm, a bijection Φ exists between the states of H and candidate states, and a subset of the states of A, such that $\tau_A(u, \sigma) = v \Leftrightarrow \tau_H(\Phi(u), \sigma) = \Phi(v)$.

For state q, a multiset S_q is said to be *representative* of the true distribution with respect to q, if $\forall s \in S_q : \left|\left(\frac{|s \in S_q|}{|S_q|}\right) - D_q(s)\right| \leq \frac{\mu}{4}$. If two states q_1 and q_2 have representative multisets, then given that $L_\infty(D_{q_1}, D_{q_2}) > \mu$, it must be the case that for some string s'':

$$\left|\frac{|s'' \in S_{q_1}|}{|S_{q_1}|} - \frac{|s'' \in S_{q_2}|}{|S_{q_2}|}\right| \geq \frac{\mu}{2}$$

Each multiset is representative (given that it contains m_0 suffixes) with probability at least $1 - \frac{\delta'}{2n^2|\Sigma|^2}$, due to a union bound. There are at most $n|\Sigma|$ candidate nodes in total and candidate nodes are re-generated (as are their multisets) in each iteration of the algorithm (of which there are at most $n|\Sigma|$). Therefore, the probability that a candidate node has a representative multiset at the point when it is converted to a node in the hypothesis graph (or found to be indistinct from another node in the graph) is at least $1 - \left(\frac{\delta'}{2n^2|\Sigma|^2}.n^2|\Sigma|^2\right) = 1 - \frac{\delta'}{2}$. □

Proposition 1. *Let A' be a PDFA whose states and transitions are a subset of those of A. Assume q_0 is a state of A'. Suppose q is a state of A' but $\tau(q, \sigma)$ is not a state of A'. Let S be a sample from D_A, $|S| \geq \frac{8n^2|\Sigma|^2}{\epsilon^2} \ln\left(\frac{2n^2|\Sigma|^2}{\delta'}\right)$. Let $S_{q,\sigma}(A')$ be the number of elements of S of the form $s_1\sigma s_2$ where $\tau(q_0, s_1) = q$ and for all prefixes s_1' of s_1, $\tau(q_0, s_1') \in A$. Then*

$$\Pr\left(\left|\left(\frac{S_{q,\sigma}(A')}{|S|}\right) - E\left[\frac{S_{q,\sigma}(A')}{|S|}\right]\right| \geq \frac{\epsilon}{8n|\Sigma|}\right) \leq \frac{\delta'}{2n^2|\Sigma|^2}.$$

Proof. From Hoeffding's Inequality it can be seen that

$$\Pr\left(\left|\left(\frac{S_{q,\sigma}(A')}{|S|}\right) - E\left[\frac{S_{q,\sigma}(A')}{|S|}\right]\right| \geq \frac{\epsilon}{8n|\Sigma|}\right) \leq e^{-2|S|\left(\frac{\epsilon}{4n|\Sigma|}\right)^2} \tag{4}$$

We need $|S|$ to satisfy $e^{-\frac{|S|\epsilon^2}{8n^2|\Sigma|^2}} \leq \frac{\delta'}{2n^2|\Sigma|^2}$. Equivalently,

$$\frac{8n^2|\Sigma|^2}{\epsilon^2} \ln\left(\frac{2n^2|\Sigma|^2}{\delta'}\right) \leq |S|.$$

So the sample size identified in the statement is indeed sufficiently large.

Theorem 2. *There exists T' a subset of the transitions of A, and Q' a subset of the states of A, such that $\sum_{(q,\sigma)\in T'} D_A(q, \sigma) + \sum_{q \in Q'} D_A(q) \leq \frac{\epsilon}{2}$, and with probability at least $1 - \delta'$, every transition $(q, \sigma) \notin T'$ in target automaton A for which $D_A(q, \sigma) \geq \frac{\epsilon}{4n|\Sigma|}$, has a corresponding transition in hypothesis automaton H, and every state $q \notin Q'$ in target automaton A for which $D_A(q) \geq \frac{\epsilon}{4n|\Sigma|}$, has a corresponding state in hypothesis automaton H.*

Proof. Theorem 1 shows that if a candidate node has a multiset containing at least m_0 suffixes, then there is a probability of at least $1 - \frac{\delta'}{2n^2|\Sigma|^2}$ that the

multiset is representative (as defined in the proof of Theorem 1). Furthermore, it shows that the probability of all candidate nodes having representative multisets (if the multisets contain at least m_0 suffixes) is at least $1-\frac{\delta'}{2}$, from which we can deduce that all candidate nodes can be correctly distinguished from any nodes[2] in the hypothesis automaton.

Proposition 1 shows that with a probability of at least $1-\frac{\delta'}{2n^2|\Sigma|^2}$, the proportion of strings in a sample S (generated iid over D_A, and for $|S| \geq \frac{8n^2|\Sigma|^2}{\epsilon^2}\ln\left(\frac{2n^2|\Sigma|^2}{\delta'}\right)$) reaching candidate node $\hat{q}$ is within $\frac{\epsilon}{8n|\Sigma|}$ of the expected proportion $D_A(\hat{q})$. This holds for each of the candidate nodes (of which there are at most $n|\Sigma|$), in each iteration of the algorithm (of which there are at most $n|\Sigma|$), with a probability of at least $1-\frac{\delta'}{2}$.

If a candidate node (or a *potential candidate node*[3]) $\hat{q}$, for which $D_A(\hat{q}) \geq \frac{\epsilon}{2n|\Sigma|}$, is not included in H, then from the facts above it follows that at least $\frac{\epsilon N}{4n|\Sigma|}$ strings in the sample are not accepted by the hypothesis graph. For each string not accepted by H, a suffix is added to the multiset of a candidate node, and there are at most $n|\Sigma|$ such candidate nodes. From this it can be seen that some candidate node has a multiset containing at least $\frac{\epsilon N}{4}$ suffixes. From the definition of N, $N \geq \frac{4m_0n|\Sigma|}{\epsilon}$. Therefore, some multiset contains at least $m_0n|\Sigma|$ suffixes, which must be at least as great as m_0. This means that as long as there exists some significant transition or state that has not been added to the hypothesis, some multiset must contain at least m_0 suffixes, so the associated candidate node will be added to H, and the algorithm will not halt.

Therefore it has been shown that all candidate nodes which are significant enough to be required in the hypothesis automaton (at least a fraction $\frac{\epsilon}{2n|\Sigma|}$ of the strings generated reach the node) are present with a probability of at least $1-\frac{\delta'}{2}$, and that since all multisets contain m_0 suffixes, the candidate nodes and hypothesis graph nodes are all correctly distinguished from each other (or combined as appropriate) with a probability of at least $1-\frac{\delta'}{2}$. We conclude that with a probability of at least $1-\delta'$, every transition $(q,\sigma) \notin T'$ in target automaton A for which $D_A(q,\sigma) \geq \frac{\epsilon}{2n|\Sigma|}$ and every state $q \notin Q'$ in target automaton A for which $D_A(q) \geq \frac{\epsilon}{2n|\Sigma|}$, has a corresponding transition or state in hypothesis automaton H. □

5 Finding Transition Probabilities

The algorithm is shown in Figure 2. We can assume that we have at this stage found DFA H, whose graph is a subgraph of the graph of target PDFA A. Algorithm 2 finds estimates of the probabilities $\gamma(q,\sigma)$ for each state q in H, $\sigma \in \Sigma$.

[2] Note that due to the deterministic nature of the automaton, distinguishability of transitions is not an issue.

[3] A potential candidate node is any state or transition in the target automaton which has not yet been added to H, and is not currently represented by a candidate node.

If we generate a sample S from D_A, we can trace each $s \in S$ through H, and each visit to a state $q_H \in H$ provides an observation of the distribution over the transitions that leave the corresponding state q_A in A. For string $s = \sigma_1\sigma_2 \ldots \sigma_\ell$, let q_i be the state reached by the prefix $\sigma_1 \ldots \sigma_{i-1}$. The probability of s is $D_A(s) = \prod_{i=0}^{\ell-1} \gamma(q_i, \sigma_{i+1})$. Let $n_{q,\sigma}(s)$ denote the number of times that string s uses transition (q, σ), then

$$D_A(s) = \prod_{q,\sigma} \gamma(q,\sigma)^{n_{q,\sigma}(s)} \tag{5}$$

Let $\hat{\gamma}(q, \sigma)$ denote the estimated probability that is given to transition (q, σ) in H. Provided H accepts s, the estimated probability of string s is given by

$$D_H(s) = \prod_{q,\sigma} \hat{\gamma}(q,\sigma)^{n_{q,\sigma}(s)} \tag{6}$$

We aim to ensure that with high probability, for $s \sim D_A$, if H accepts s then the ratio $D_H(s)/D_A(s)$ is close to 1. This is motivated by the following.

Observation 3. *Suppose that with probability* $1-\frac{1}{4}\epsilon$, *for* $s \sim D_A$, $D_H(s)/D_A(s) \in [1-\frac{1}{4}\epsilon, 1+\frac{1}{4}\epsilon]$. *Then* $L_1(D_A, D_H) \leq \frac{1}{2}\epsilon$.

Proof.

$$L_1(D_A, D_H) = \sum_{s \in \Sigma^*} |D_A(s) - D_H(s)|$$

Let $X = \{s \in \Sigma^* \ : \ D_H(s)/D_A(s) \in [1-\frac{1}{4}\epsilon, 1+\frac{1}{4}\epsilon]\}$. Then

$$L_1(D_A, D_H) = \sum_{s \in X} |D_A(s) - D_H(s)| + \sum_{s \in \Sigma^* \setminus X} |D_A(s) - D_H(s)|$$

The first term of the right-hand side is $\sum_{s \in X} D_A(s)(1 - D_H(s)/D_A(s)) \leq \sum_{s \in X} D_A(s).(\frac{1}{4}\epsilon) \leq \frac{1}{4}\epsilon$.

$D_A(X) \geq 1 - \frac{1}{4}\epsilon$ and $D_H(X) \geq D_A(X) - \frac{1}{4}\epsilon$, hence the second term in the right-hand side is at most $\frac{1}{4}\epsilon$. □

We have so far allowed the possibility that H may fail to accept up to a fraction $\frac{1}{4}\epsilon$ of strings generated by D_A. Of the strings s that are accepted by H, we want to ensure that with high probability $D_H(s)/D_A(s)$ is close to 1, to allow Observation 3 to be used.

Suppose that $n_{q,\sigma}(s)$ is large, so that s uses transition (q, σ) a large number of times. In that case, errors in the estimate of transition probability $\gamma(q, \sigma)$ can have a disproportionately large influence on the ratio $D_H(s)/D_A(s)$. What we show is that with high probability for random $s \sim D_A$, regardless of how many times transition (q, σ) typically gets used, the training sample contains a large enough subset of strings that use that transition more times than s does, so that $\gamma(q, \sigma)$ is nevertheless known to a sufficiently high precision.

We say that $s \in \Sigma^*$ is (q,σ)-good for some transition (q,σ), if s satisfies:

$$\Pr_{s' \sim D_A}(n_{q,\sigma}(s') > n_{q,\sigma}(s)) \leq \frac{\epsilon}{4n|\Sigma|}$$

A (q,σ)-good string is one that is more useful than most in providing an estimate of $\gamma(q,\sigma)$.

Observation 4. *Let $m \geq 1$. Let S be a sample from D_A, $|S| \geq m\left(\frac{32n|\Sigma|}{\epsilon}\right)\ln\left(\frac{2n|\Sigma|}{\delta''}\right)$. With probability $1-\frac{\delta''}{2n|\Sigma|}$, for transition (q,σ) there exist at least $\frac{\epsilon}{8n|\Sigma|}|S|$ (q,σ)-good strings in S.*

Proof. From the definition of (q,σ)-good, the probability that a string generated at random over D_A is (q,σ)-good for transition (q,σ), is at least $\frac{\epsilon}{4n|\Sigma|}$.

Applying a standard Chernoff Bound (see e.g. [2], p360), for any transition (q,σ), given sample S, with high probability the observed number of (q,σ)-good strings in S is at least half the expected number:

$$\Pr\left(|\{s \in S \ : \ s \text{ is } (q,\sigma)-\text{good}\}| < \frac{1}{2}.\frac{\epsilon}{4n|\Sigma|}|S|\right) \leq \exp\left(-\frac{\frac{1}{4}\left(\frac{\epsilon}{4n|\Sigma|}\right)|S|}{2}\right) \tag{7}$$

We wish to bound this probability to be at most $\frac{\delta''}{2n|\Sigma|}$, so from Equation (7),

$$\exp\left(-\frac{\frac{1}{4}\left(\frac{\epsilon}{4n|\Sigma|}\right)|S|}{2}\right) \leq \frac{\delta''}{2n|\Sigma|}$$

$$|S| \geq \left(\frac{32n|\Sigma|}{\epsilon}\right)\ln\left(\frac{2n|\Sigma|}{\delta''}\right)$$

□

Notation. Suppose $S_{q,\sigma}$ is as defined in Algorithm 2. Let $M_{q,\sigma}$ be the largest number with the property that at least a fraction $\frac{\epsilon}{8n|\Sigma|}$ of strings in $S_{q,\sigma}$ use (q,σ) at least $M_{q,\sigma}$ times.

Observation 5. *From Observation 4 (plugging in $m = (\frac{2n|\Sigma|}{\delta''})(\frac{32n|\Sigma|^2}{\epsilon})^2$) it follows that with probability $1-\frac{\delta''}{2n|\Sigma|}$ (over random samples $S_{q,\sigma}$),*

$$\Pr_{s \sim D_A}(n_{q,\sigma}(s) > M_{q,\sigma}) \leq \frac{\epsilon}{4n|\Sigma|} \tag{8}$$

Theorem 6. *Suppose that H is a DFA that differs from A by the removal of a set of transitions that have probability at most $\frac{1}{2}\epsilon$ of being used by $s \sim D_A$. Then Algorithm 2 assigns probabilities $\hat{\gamma}(q,\sigma)$ to the transitions of H such the resulting distribution D_H satisfies $L_1(D_A, D_H) < \epsilon$, with probability $1-\delta''$.*

Algorithm 2 *Finding Transition Probabilities.*

Input: DFA H, a subgraph of A.

For each state $q \in H$, $\sigma \in \Sigma$:
  generate sample $S_{q,\sigma}$ from D_A; $|S_{q,\sigma}| = (\frac{2n|\Sigma|}{\delta''})(\frac{32n|\Sigma|^2}{\epsilon})^2(\frac{32n|\Sigma|}{\epsilon})\ln(\frac{2n|\Sigma|}{\delta''})$;
  repeat
    for strings $s \in S_{q,\sigma}$, trace paths through H;
    Let $N_{q,-\sigma}$ be random variable: number of observations of state q before we observe transition (q,σ) (include observations of q and (q,σ) in rejected strings).
  until(all strings in $S_{q,\sigma}$ have been traced)
  Let $\hat{N}_{q,-\sigma}$ be the mean of the observations of $N_{q,-\sigma}$;
  Let $\hat{\gamma}(q,\sigma) = 1/\hat{N}_{q,\sigma}$.
  Let $n_{q,\sigma}$ be number of observations of (q,σ);
  for all q let $\sigma_{\min}(q) = \arg\min_\sigma(n_{q,\sigma})$.
  Adjust $\hat{\gamma}(q,\sigma_{\min}(q))$ such that $\sum_\sigma \hat{\gamma}(q,\sigma) = 1$.

Fig. 2. Finding Transition Probabilities

Comment. It can be seen that the sample size used by Algorithm 2 is polynomial in the parameters of the problem. It is linear in $\frac{1}{\delta''}$; we believe that a more refined analysis would yield a logarithmic bound, alternatively one could modify the algorithm to obtain a logarithmic bound. The general idea would be to make $O(\log(\frac{1}{\delta''}))$ independent empirical estimates of each $N_{q,-\sigma}$, and take their median.

Proof. Recall Observation 5, that with probability $1 - \frac{\delta''}{2n|\Sigma|}$,

$$\Pr_{s\sim D_A}(n_{q,\sigma}(s) > M_{q,\sigma}) \leq \frac{\epsilon}{4n|\Sigma|}$$

Using Observation 4, the sets $S_{q,\sigma}$ are large enough to ensure that with probability $1 - \frac{\delta''}{2n|\Sigma|}$, there are $M_{q,\sigma}(\frac{2n|\Sigma|}{\delta''})(\frac{32n|\Sigma|^2}{\epsilon})^2$ uses of transition (q,σ). This is because at least $(\frac{2n|\Sigma|}{\delta''})(\frac{32n|\Sigma|^2}{\epsilon})^2$ members of $S_{q,\sigma}$ use (q,σ) at least $M_{q,\sigma}$ times.

Consequently, (again with probability $1 - \frac{\delta''}{2n|\Sigma|}$ over random choice of $S_{q,\sigma}$) the set $S_{q,\sigma}$ generates a sequence of independent observations of state q, which continues until $M_{q,\sigma}(\frac{2n|\Sigma|}{\delta''})(\frac{32n|\Sigma|^2}{\epsilon})^2$ of them resulted in transition (q,σ).

Let $N_{q,-\sigma}$ denote the random variable which is the number of times q is observed before transition (q,σ) is taken. Each time state q is visited, the selection of the next transition is independent of previous history, so we obtain a sequence of independent observations of $N_{q,-\sigma}$. So, with probability $1 - \frac{\delta''}{2n|\Sigma|}$, the number of observations of $N_{q,-\sigma}$ is at least $M_{q,\sigma}(\frac{2n|\Sigma|}{\delta''})(\frac{32n|\Sigma|^2}{\epsilon})^2$.

Recall Chebyshev's inequality, that for random variable X with mean μ and variance σ^2, for positive k,

$$\Pr(|X-\mu|>k) \leq \frac{\sigma^2}{k^2}.$$

$N_{q,-\sigma}$ has a discrete exponential distribution with mean $\gamma(q,\sigma)^{-1}$ and variance $\leq \gamma(q,\sigma)^{-2}$. Hence the empirical mean $\hat{N}_{q,-\sigma}$ is a random variable with mean $\gamma(q,\sigma)^{-1}$ and variance at most $\gamma(q,\sigma)^{-2}(M_{q,\sigma})^{-1}(\frac{2n|\Sigma|}{\delta''})^{-1}(\frac{32n|\Sigma|^2}{\epsilon})^{-2}$. Applying Chebyshev's inequality with $\hat{N}_{q,-\sigma}$ for X, and
$k=\gamma(q,\sigma)^{-1}(\frac{\epsilon}{32n|\Sigma|^2\sqrt{M_{q,\sigma}}})$, we have

$$\Pr\Big(|\hat{N}_{q,-\sigma}-\gamma(q,\sigma)^{-1}|>\gamma(q,\sigma)^{-1}(\frac{\epsilon}{32n|\Sigma|^2\sqrt{M_{q,\sigma}}})\Big) \leq \frac{\delta''}{2n|\Sigma|}.$$

Since $\gamma(q,\sigma)=1/E[N_{q,-\sigma}]$ and $\hat{\gamma}(q,\sigma)=1/\hat{N}_{q,-\sigma}$,

$$\Pr\Big(|\hat{\gamma}(q,\sigma)-\gamma(q,\sigma)|>2\gamma(q,\sigma)(\frac{\epsilon}{32n|\Sigma|^2\sqrt{M_{q,\sigma}}})\Big) \leq \frac{\delta''}{2n|\Sigma|}.$$

The rescaling at the end of Algorithm 2 loses a factor of $|\Sigma|$ from the upper bound on $|\gamma(q,\sigma)-\hat{\gamma}(q,\sigma)|$. Overall, with high probability $1-\frac{\delta''}{2n|\Sigma|}$,

$$|\hat{\gamma}(q,\sigma)-\gamma(q,\sigma)| \leq \Big(\frac{\epsilon\gamma(q,\sigma)}{16n|\Sigma|\sqrt{M_{q,\sigma}}}\Big) \tag{9}$$

For $s\in\Sigma^*$ let $n_q(s)$ denote the number of times the path of s passes through state q. By definition of $M_{q,\sigma}$, with high probability $1-\frac{\epsilon}{4n|\Sigma|}$ for $s\sim D_A$,

$$M_{q,\sigma} > n_q(s).\gamma(q,\sigma). \tag{10}$$

For $s\sim D_A$ we upper bound the expected log-likelihood ratio,

$$\log\Big(\frac{D_H(s)}{D_A(s)}\Big)=\sum_{i=1}^{|s|}\frac{\hat{\gamma}(q_i,\sigma_i)}{\gamma(q_i,\sigma_i)}$$

where σ_i is the i-th character of s and q_i is the state reached by the prefix of length $i-1$.

Suppose A generates a prefix of s and reaches state q. Let random variable X_q be the contribution to $\log(\frac{D_H(s)}{D_A(s)})$ when A generates the next character.

$$\begin{aligned} E[X_q] &= \sum_\sigma \gamma(q,\sigma)\log\Big(\frac{\hat{\gamma}(q,\sigma)}{\gamma(q,\sigma)}\Big) \\ &= \sum_\sigma \gamma(q,\sigma)[\log(\hat{\gamma}(q,\sigma))-\log(\gamma(q,\sigma))] \end{aligned}$$

We claim that (with high probability $1 - \frac{\delta''}{2n|\Sigma|}$)

$$\log(\hat{\gamma}(q,\sigma)) - \log(\gamma(q,\sigma)) \leq |\hat{\gamma}(q,\sigma) - \gamma(q,\sigma)| \frac{1}{\gamma(q,\sigma)} . A_{q,\sigma} \tag{11}$$

for some $A_{q,\sigma} \in [1 - \frac{\epsilon}{8n|\Sigma|\sqrt{M_{q,\sigma}}}, 1 + \frac{\epsilon}{8n|\Sigma|\sqrt{M_{q,\sigma}}}]$. The claim follows from (9) and the inequality, for $|\xi| < x$, that $\log(x+\xi) - \log(x) \leq \xi . \frac{1}{x}(1 + \frac{2\xi}{x})$ (plug in $\gamma(q,\sigma)$ for x). Consequently,

$$\begin{aligned} E[X_q] &\leq \sum_\sigma \gamma(q,\sigma)\Big(\frac{1}{\gamma(q,\sigma)}\Big) A_{q,\sigma}[\hat{\gamma}(q,\sigma) - \gamma(q,\sigma)] \\ &= \sum_\sigma A_{q,\sigma}[\hat{\gamma}(q,\sigma) - \gamma(q,\sigma)] \\ &= \sum_\sigma [\hat{\gamma}(q,\sigma) - \gamma(q,\sigma)] + \sum_\sigma B_{q,\sigma}[\hat{\gamma}(q,\sigma) - \gamma(q,\sigma)] \end{aligned}$$

for some $B_{q,\sigma} \in [-\frac{\epsilon}{8n|\Sigma|\sqrt{M_{q,\sigma}}}, \frac{\epsilon}{8n|\Sigma|\sqrt{M_{q,\sigma}}}]$. The first term vanishes, so we have

$$\begin{aligned} E[X_q] &\leq \sum_\sigma B_{q,\sigma}[\hat{\gamma}(q,\sigma) - \gamma(q,\sigma)] \\ &= \frac{\epsilon}{8n|\Sigma|} \sum_\sigma (\frac{1}{\sqrt{M_{q,\sigma}}})[\hat{\gamma}(q,\sigma) - \gamma(q,\sigma)] \\ &\leq \frac{\epsilon}{8n|\Sigma|} \sum_\sigma \frac{\gamma(q,\sigma)}{M_{q,\sigma}} \end{aligned}$$

using (9). For $s \sim D_A$, given values $n_q(s)$, the expected contribution to $\log(\frac{D_A(s)}{D_A(s)})$ from all $n_q(s)$ usages of state q is, using (10), at most

$$n_q(s) \frac{\epsilon}{8n|\Sigma|} \sum_\sigma \frac{1}{n_q(s)} = \frac{\epsilon n_q(s)}{8n|\Sigma|} |\Sigma| \frac{1}{n_q(s)} = \frac{\epsilon}{8n}$$

The total contribution from all n states q, each being used $n_q(s)$ times is

$$\sum_q \frac{\epsilon}{8n} = \frac{\epsilon}{8}. \tag{12}$$

So the expected difference between the likelihood of string s using the $\hat{\gamma}(q,\sigma)$ values in place of the $\gamma(q,\sigma)$ values, is small. Using (11),

$$\begin{aligned} Var[X_q] &\leq \sum_\sigma \frac{A_{q,\sigma}^2}{\gamma(q,\sigma)} [\hat{\gamma}(q,\sigma) - \gamma(q,\sigma)]^2 \\ &\leq \sum_\sigma \frac{A_{q,\sigma}^2}{\gamma(q,\sigma)} \frac{\gamma(q,\sigma)^2}{M_{q,\sigma}} \Big(\frac{\epsilon}{8n|\Sigma|}\Big)^2 \end{aligned}$$

$$\leq \left(\frac{\epsilon}{8n|\Sigma|}\right)^2 \sum_{\sigma} \frac{A_{q,\sigma}^2 \gamma(q,\sigma)}{M_{q,\sigma}}$$
$$\leq \left(\frac{\epsilon}{8n|\Sigma|}\right)^2 \sum_{\sigma} \frac{2}{n_q(s)}$$

Hence the variance of the total contribution to the error $\log(\frac{D_H(s)}{D_A(s)})$ from all $n_q(s)$ uses of state q, is at most $(\frac{\epsilon}{8n|\Sigma|})^2$. Using (12), with high probability for $s \sim D_A$, all the states contribute at most $\frac{1}{8}\epsilon$ to $\log(\frac{D_H(s)}{D_A(s)})$.

Finally, to use Observation 3, note that $\frac{D_H(s)}{D_A(s)} \in [1-\frac{1}{4}\epsilon, 1+\frac{1}{4}\epsilon]$ follows from $\log(\frac{D_H(s)}{D_A(s)}) \in [-\frac{1}{8}\epsilon, \frac{1}{8}\epsilon]$.

□

References

[1] N. Abe, J. Takeuchi and M. Warmuth. Polynomial Learnability of Stochastic Rules with respect to the KL-divergence and Quadratic Distance. *IEICE Trans. Inf. and Syst., Vol E84-D(3) pp. 299-315 (2001).*

[2] M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations.* Cambridge University Press (1999)

[3] A. Clark and F. Thollard. PAC-learnability of Probabilistic Deterministic Finite State Automata. Journal of Machine Learning Research 5 pp. 473-497 (2004)

[4] T.M. Cover and J.A. Thomas. *Elements of Information Theory* Wiley Series in Telecommunications. John Wiley & Sons (1991).

[5] M. Cryan and L. A. Goldberg and P. W. Goldberg. Evolutionary Trees can be Learnt in Polynomial Time in the Two-State General Markov Model. *SIAM Journal on Computing 31(2) pp. 375-397 (2001)*

[6] P.W. Goldberg When Can Two Unsupervised Learners Achieve PAC Separation? *Procs. of COLT/EUROCOLT, LNAI 2111, pp. 303-319 (2001)*

[7] C. de la Higuera and J. Oncina. Learning Probabilistic Finite Automata. *tech. rept. EURISE, Université de Saint-Etienne and Departamento de Lenguajes y Sistemas Informaticos (2002)*

[8] M. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R.E. Schapire and L. Sellie. On the Learnability of Discrete Distributions. *Procs. of STOC, pp. 273-282 (1994).*

[9] Nick Palmer and Paul. W. Goldberg. PAC Classification via PAC Estimates of Label Class Distributions. *Tech rept. 411, Dept. of Computer Science, University of Warwick (2004)*

[10] D. Ron, Y. Singer and N. Tishby. On the Learnability and Usage of Acyclic Probabilistic Finite Automata. *Journal of Computer and System Sciences, 56(2), pp. 133-152 (1998).*