# Probabilistic Automata and Distributions over Sequences

Omri Guttman

A thesis submitted for the degree of
Doctor of Philosophy at
The Australian National University

September 1st, 2006

# Declaration

This thesis is an account of research undertaken between March 2003 and August 2006 at The Research School of Information Sciences and Engineering, Faculty of Science, The Australian National University, Canberra, Australia.

Except where acknowledged in the customary manner, the material presented in this thesis is, to the best of my knowledge, original and has not been submitted in whole or part for a degree in any university.

The work described in this thesis has been carried out under the supervision of Prof. Robert C. Williamson, Dr. Alex Smola and Dr. S.V.N. Vishwanathan. However, the majority of the work, approximately 70%, is my own.

Omri Guttman

Omri Guttman
2006

# Acknowledgements

I would like to express my gratitude to all the people whose help, advice and support made significant contributions to this thesis. First, I would like to warmly acknowledge the guidance and support of my primary supervisor, Professor Robert C. Williamson. His wide perspective and insight were instrumental in steering my research over the course of my studies. I would like to thank my supervisor Dr. Alex Smola, for contributing numerous sharp insights and helpful references from his endless knowledge base. And I thank my supervisor Dr. S.V.N. Vishwanathan for his scientific and personal support, his pure and tireless pursuit of scientific discovery, and for many pleasant hours of brainstorming.

I would like to express my deep gratitude to Dr. Shahar Mendelson, my good friend and guiding light in all matters mathematical, for whom all problems are trivial. His deep insight, brilliant intuition and generous support were key to my research.

I would like to thank Dr. Robert Orsi, who provided me with support and insight into the optimization aspects of my problems.

I extend my gratitude to all staff, students and visitors I've had the pleasure to meet over the years. In particular I would like to thank my fellow PhD students whose company and thoughtful advice were a great source of support: Tim Sears, Petra Phillips, Evan Greensmith, and Cheng Soon Ong.

I greatly appreciate the opportunity I've had to study at the Australian National University and the National ICT Australia (NICTA). This opportunity would not have been made possible without their funding.

Finally, my deepest thanks go to my beloved wife Tali and son Guy. Their love and support provided the balance I so greatly needed during these years. My life could not be complete without them.

# Abstract

This thesis considers probabilistic finite automata (PFA), distributions of sequences over finite alphabets, the links between them and the learnability thereof. Pervasive in scientific fields ranging from computer science to electrical engineering to information theory, PFA models also find numerous practical applications in speech recognition, bioinformatics and natural language processing.

PFA models are the most general among the myriad of syntactic objects providing probabilistic extensions of finite state machines. Closely related to hidden Markov models (HMMs), PFAs have been the focus of extensive research, but continue to pose interesting theoretical and practical problems to this day.

The thesis presents geometric insights into the PFA learning problem, a characterization theorem for the family of distributions induced by PFA models, as well as a number of applications of this theorem. For a subclass of PFA called probabilistic deterministic finite automata (PDFA), a number of learnability results are presented. These results place limits on the PDFA subclasses which are learnable using a class of algorithms collectively known as state merging.

The sample complexity of learning general distributions over countable sets is considered, and lower and upper bounds, which asymptotically match up to a logarithmic factor are developed. An example is constructed exhibiting a class of PDFA models which is efficiently learnable using state merging. It is demonstrated that distributions induced by this class are not efficiently learnable by direct estimation (making no assumptions on the distribution's source) in the sense that the sample complexity is bounded below by an exponential in the number of states.

# Contents

# List of Figures

# List of Algorithms

Chapter 1

# Introduction

The concept of a *state machine* is fundamental to the field of computer science. The subject of extensive investigation tracing back to the 1940's and 1950's, the state machine continues to pose interesting and open problems to this day.

The research presented in this thesis revolves around an extension of the state machine concept to the probabilistic framework, namely the *probabilistic finite automaton* (or PFA). The PFA is the most general among several syntactic objects providing probabilistic extensions to the finite state machine concept[1]. The hidden Markov model (or HMM), probabilistic (or stochastic) deterministic finite automata (PDFA), Markov chains, $n$-grams and probabilistic suffix trees are among the numerous special cases of PFA which have been proposed to model and generate distributions over sequences.

The PFA concept is widely used and studied in a number of theoretical as well as practical communities. In the information theory literature, PFA models (typically under the name *finite state sources*) are frequently used to model communication channels, alongside their hidden Markov model (HMM) counterparts (sometimes referred to as *hidden Markov processes*). A comprehensive survey paper providing an overview of hidden Markov processes from the statistical and information-theoretic viewpoints was written by Ephraim and Merhav [2002].

In a recent pair of survey papers [Vidal et al., 2005a], [Vidal et al., 2005b], the theoretical machine learning aspects of PFA were outlined. Aiming to enhance the thesis' readability, we will attempt to adhere to the notation used in these papers where relevant and introduce new notation only where necessary.

From the practical perspective, many engineering disciplines including speech recognition, language modeling, machine translation and bioinformatics utilize the probabilistic finite state machine concept.

The remainder of this thesis is organized as follows:

- In the subsequent sections of this chapter we will rigorously define the concepts

[1] The term *deterministic finite automaton* (DFA) is sometimes used instead of finite state machine. We generally make no distinction between "machine" and "automaton" in this context.

relating to probabilistic finite state machines, and provide detailed descriptions of their inter-relationships.

- In Chapter 2 we will describe the intrinsic geometry of the PFA models. We will examine the geometry of the interplay between the PFA's parameterization, the mapping into distribution space, and the resulting distribution. Our focus will be on convexity properties, which will highlight the difficulty of the PFA learning problem.

- In Chapter 3 we will extend the well-known Myhill-Nerode theorem of finite state machines to PFA. Subsequently, we will show how the extension theorem can be used to prove certain distributions cannot be induced by PFA or PDFA models. The chapter will be concluded with applications of the Myhill-Nerode extension, placing bounds on the relative expressive power of the PFA and PDFA models in approximating arbitrary probability distributions over bounded-length strings.

- In Chapter 4 we will summarize and extend the theoretical results regarding the inference of PFA and PDFA models from data. Focusing on the PDFA models' learnability, we will present a novel extension to a class of algorithms collectively known as *state merging*. We will then formulate and prove extended positive and negative learnability results. We will provide analysis showing learnability of novel PDFA subclasses that strictly include families already known to be learnable. We will conclude with a negative result, proving that state merging algorithms have a high likelihood of failure on another PDFA subclass.

- The sample complexity of PFA and discrete distribution learning is discussed in Chapter 5. We will summarize known results in the field, present and provide a novel analysis of a baseline algorithm against which any other algorithm's sample complexity can be measured. Our analysis comprises a pair of sufficient conditions on the sample size required for learnability, as well as a sufficient condition for failure mode. The conditions are tight up to a logarithmic factor. We will perform a critical comparison between our analysis and the existing state-of-the-art results. In conclusion, we will construct an example on which the state-merging algorithm requires polynomial computational complexity, while the baseline algorithm's complexity is exponential in the number of states.

- In Chapter 6 we present our conclusions and a discussion of possibilities for further research.

- In Appendix A we provide proofs for a number of technical lemmas.

- Appendix B contains a number of results repeated from [Clark and Thollard, 2004], serving to make the document self-contained. In order to enhance clarity, we have translated the results to the notation used throughout the thesis.

## 1.1 Notational Conventions

Throughout the thesis, standard symbols will assume the usual meaning, unless otherwise stated. The symbols $O$ and $o$ will have the usual meaning:

$$f(z) = O(g(z)) \quad \text{as} \quad z \to w \quad \text{means} \quad f(z)/g(z) \quad \text{is bounded as } z \to w;$$
$$f(z) = o(g(z)) \quad \text{as} \quad z \to w \quad \text{means} \quad f(z)/g(z) \to 0 \quad \text{as } z \to w.$$

When an asymptotic relation is stated for an integer variable $n$, it will implicitly be taken to apply only for integer values of $n$, and the limit will always be $\infty$. Logarithms will be natural (to base $e$) by default, or to base 2 (when explicitly stated).

Given a random variable denoted by $X$, the expected value of $X$ is denoted $\mathbb{E}\,X$, the variance of $X$ is denoted $\mathrm{var}(X)$, and the probability of some statistical event $f(X)$ is denoted as $\Pr(f(X))$. Given some probability distribution $D$, the notation $X \sim D$ means $X$ is randomly drawn according to the distribution $D$.

The *delta distribution* $\delta(x)$ where $x \in \mathcal{X}$ ($\mathcal{X}$ denoting some set) will denote a distribution with all probability concentrated on the (single) element $x$:

$$\forall y \in \mathcal{X}, \quad \Pr_{y \sim \delta(x)}(y) = \begin{cases} 1 & \text{if } y = x, \\ 0 & \text{otherwise.} \end{cases}$$

We denote by $\Sigma$ a finite alphabet and by $\Sigma^*$ the set of all sequences (or strings) of characters taken from $\Sigma$, including the empty string denoted by $\epsilon$. The set of all strings of length $n$ (resp. less than, at most $n$) will be denoted by $\Sigma^n$ (resp. $\Sigma^{<n}, \Sigma^{\leq n}$). The length of a string $s \in \Sigma^*$ is denoted by $|s|$. The substring of $s$ from position $i$ to position $j$ is written $s_i \ldots s_j$. Given two strings $x, y \in \Sigma^*$, their *concatenation* $xy \in \Sigma^*$ is the string obtained by appending the suffix $y$ to the string $x$.

A *sample multiset* $S$ is a multiset of strings. Since they are typically obtained by sampling, a particular string may appear more than once. The total number of strings in a sample $S$ is denoted by $|S|$, and the empirical distribution associated with $S$ is $\widehat{S}(s) := \#\{s \in S\}/|S|$ (with $\#\{s \in S\}$ denoting the number of occurances of a specific string $s$ in the multiset $S$).

We will (for the most part, unless explicitly stated otherwise) follow the symbol conventions summarized in the Glossary of Symbols (page 117).

## 1.2 Basic Tools of Probability

The following basic probabilistic results are used in the thesis:

**Theorem 1.1 (Markov's inequality)** *Let $X$ be a nonnegative random variable. Then for every $t > 0$,*

$$\Pr(X \geq t) \leq \frac{\mathbb{E}\,X}{t}.$$

**Theorem 1.2 (Chebyshev's inequality)** *Let $X$ be a random variable with finite variance. Then for every $t > 0$,*

$$\Pr(|X - \mathbb{E}\,X| \geq t) \leq \frac{var(X)}{t^2}.$$

## 1.3 Stochastic Languages

A *language* is a subset of $\Sigma^*$. A *stochastic language* $L$ is a probability distribution over $\Sigma^*$. We denote by $\Pr(s|L)$ the probability of the string $s$ given the language $L$. The distribution defined by $L$ must satisfy $\sum_{s\in\Sigma^*}\Pr(s|L) = 1$. The probability of any set (i.e. not multiset) $X \subseteq \Sigma^*$ given $L$ is given by:

$$\Pr(X|L) = \sum_{x\in X}\Pr(x|L).$$

Given a set of probability distributions $\{D_i\}_{i=1}^n$, the *convex hull* of $\{D_i\}_{i=1}^n$ is the set of distributions of the form $D = \sum_{i=1}^n \lambda_i D_i$, where the coefficients $\lambda_1, \ldots, \lambda_n$ are such that $\lambda_i \geq 0$ for every $i \in \{1, \ldots, n\}$ and $\sum_{i=1}^n \lambda_i = 1$. For such a set of *convex coefficients* it follows immediately that $D$ is indeed a distribution. We will use the notation $\mathrm{co}(D_1, \ldots, D_n)$ to denote such a set of distributions.

In the following definitions we (roughly) adopt the notation of [Carrasco and Oncina, 1999]. Some of the symbols below are also used to denote other concepts, but any confusion is avoided by the context.

**Definition 1.3** *A* stochastic regular grammar *is a 5-tuple $G = \langle \Sigma, V, S, R, p\rangle$, with $\Sigma$ a finite alphabet, $V$ a finite set of variables, $S$ a* starting symbol, *and $R$ a finite set of derivation rules*[2] *with either of the following structures:*

$$\begin{aligned} X &\longrightarrow aY \\ X &\longrightarrow \epsilon \end{aligned}$$

[2] *See e.g. [Hopcroft and Ullman, 1979] for an explanation of the concept of derivation rules.*

*where $a \in \Sigma$, $X, Y \in V$, and a real-valued function $p : R \rightarrow [0,1]$ giving the probability of derivation. The sum of the probabilities for all derivations from a given variable $X$ must equal to one.*

**Definition 1.4** *A stochastic regular grammar $G$ is* deterministic *if for all $X \in V$ and for all $a \in \Sigma$ there is at most one $Y \in V$ such that $p(X \rightarrow aY) \neq 0$.*

Every stochastic deterministic regular grammar $G$ defines a *stochastic deterministic regular language* (SDRL), $L_G$, through the probabilities $p(w|L_G) = p(S \Rightarrow w)$, where the probability $p(S \Rightarrow w)$ that the grammar $G$ generates the string $w \in \Sigma^*$ is defined recursively:

$$\begin{aligned} p(X \Rightarrow \epsilon) &= p(X \rightarrow \epsilon) \\ p(X \Rightarrow aw) &= p(X \rightarrow aY)p(Y \Rightarrow w), \end{aligned}$$

where $Y$ is the only variable satisfying $p(X \rightarrow aY) \neq 0$ (if such a variable does not exist then $p(X \rightarrow aY) = 0$).

**Definition 1.5** *The* quotient language $x^{-1}L$ *is the stochastic language defined by the probabilities of the strings in $L$ starting with $x$, properly normalized:*

$$p(w|x^{-1}L) := \frac{p(xw|L)}{p(x\Sigma^*|L)} := \frac{p(xw|L)}{\sum_{z \in \Sigma^*} p(xz|L)}.$$

*If $\sum_{z \in \Sigma^*} p(xz|L) = 0$ then by convention $x^{-1}L = \emptyset$ and $p(w|x^{-1}L) = 0$. Note that $\epsilon^{-1}L = L$.*

Esposito et al. [2002] used *residual language* to denote the quotient language.

## 1.4 Finite State Machines

We now introduce the finite state machine (FSM), a fundamental concept to the theory of computer science. Also referred to as the *deterministic finite automaton* (DFA), it is formally defined as follows[3]:

**Definition 1.6** *A DFA $\mathcal{A}$ is a five-tuple $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$, with:*

- *$Q$ a finite set of states,*
- *$\Sigma$ an alphabet or a finite set of symbols,*

[3] We adhere to the notation of Hopcroft and Ullman [1979].

- $\delta : Q \times \Sigma \rightarrow Q$ *a transition function,*
- $q_0 \in Q$ *a start state,*
- $F \subseteq Q$ *a set of* final *or* accepting *states.*

The transition function admits a recursive extension $\hat{\delta}$ using the recursion:

$$\begin{aligned}\hat{\delta}(q, \epsilon) &= q \quad \forall q \in Q, \\ \hat{\delta}(q, (w_1 \dots w_k)) &= \delta(\hat{\delta}(q, w_1 \dots w_{k-1}), w_k) \quad \forall w = w_1 w_2 \dots w_k \in \Sigma^*.\end{aligned}$$

The DFA $\mathcal{A}$ *accepts* a (possibly empty) subset of $\Sigma^*$, defining the *language* $L_{\mathcal{A}}$:

$$L_{\mathcal{A}} := \{w \in \Sigma^* \quad | \quad \hat{\delta}(q_0, w) \in F\}.$$

For a complete introduction to the theory of DFA, the reader is referred to [Hopcroft and Ullman, 1979].

## 1.5 Probabilistic Finite Automata

The probabilistic finite automaton (or *PFA*) model has been extensively investigated in a number of scientific fields. Due to this, discussions of PFA models use widely differing terminology and notation[4].

In the machine learning community, the PFA terminology is the most commonly used, occasionally creating confusion with the closely related *hidden Markov model* (or *HMM*, see Section 1.8.1 for a more detailed discussion of the relationship between the two models). In the information theory literature, the term *finite state source* (or *FSS*) is typically used (see [Ephraim and Merhav, 2002] for a thorough survey).

We now formally define the PFA model:

**Definition 1.7** *A PFA is a tuple* $\mathcal{A} = \langle Q_{\mathcal{A}}, \Sigma, \delta_{\mathcal{A}}, I_{\mathcal{A}}, F_{\mathcal{A}}, P_{\mathcal{A}} \rangle$*, where:*

- $Q_{\mathcal{A}}$ *is a finite set of* states*,*
- $\Sigma$ *is a finite alphabet,*
- $\delta_{\mathcal{A}} \subseteq Q_{\mathcal{A}} \times \Sigma \times Q_{\mathcal{A}}$ *is a set of* transitions*,*
- $I_{\mathcal{A}} : Q_{\mathcal{A}} \rightarrow [0, 1]$ *are the initial state probabilities,*
- $P_{\mathcal{A}} : \delta_{\mathcal{A}} \rightarrow [0, 1]$ *are the transition probabilities,*

[4] Parts of the exposition in the following sections include segments repeated from the survey papers [Vidal et al., 2005a] and [Vidal et al., 2005b].

- $F_{\mathcal{A}} : Q_{\mathcal{A}} \to [0,1]$ *are the final state probabilities.*

The transition set $\delta_{\mathcal{A}}$ should not be confused with the delta distribution due to the different number of arguments. When understood from context, the subscript $\mathcal{A}$ will be dropped.

The transition probabilities of non-existing transitions are null, i.e. $P_{\mathcal{A}}(q,a,q') = 0$ for all $(q,a,q') \notin \delta_{\mathcal{A}}$. The initial state probabilities $I_{\mathcal{A}}$ have $\sum_{q \in Q_{\mathcal{A}}} I_{\mathcal{A}}(q) = 1$, while for the transition and final state probabilities we have:

$$F_{\mathcal{A}}(q) + \sum_{\substack{a \in \Sigma \\ q' \in Q_{\mathcal{A}}}} P_{\mathcal{A}}(q,a,q') = 1, \qquad \forall q \in Q_{\mathcal{A}}.$$

PFA are often represented as directed labeled graphs. Figure 1.1 shows a graphic depiction of a PFA with four states, $Q = \{q_0, q_1, q_2, q_3\}$ and a single initial state $q_0$ (i.e. a state $q$ with $I(q) > 0$) using a four-symbol alphabet, $\Sigma = \{a,b,c,d\}$. The numbers below each state and above each arrow are the final-state and transition probabilities respectively.

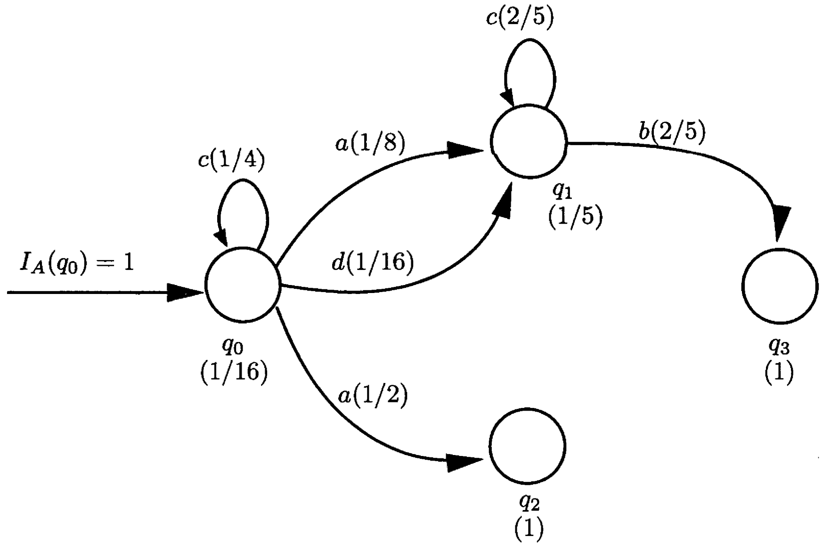

**Figure 1.1**: Graphical representation of a PFA.

We define a concept introduced by Abe and Warmuth [1992] and used in PFA training scenarios where a subset of the PFA's transition probabilities are forced to zero:

**Definition 1.8** *A* PFA constraint *is a tuple $C := \langle I, G \rangle$ where $I$ is the* initial state set[5] *and $G \subseteq Q_{\mathcal{A}} \times \Sigma \times Q_{\mathcal{A}}$ is a subset of all possible transitions. We denote the* size *of the constraint, written $|C|$ by: $|C| = |I| + |G|$.*

[5] In this context, any state *not* in $I$ is constrained to zero initial probability.

## 1.6 Probabilistic Finite Automata as Distribution Generators

Probabilistic finite automata can be viewed as mechanisms for generating strings randomly from a probability distribution. Given a PFA $\mathcal{A}$, the process of generating a string proceeds as follows:

- Initialization: Choose (with respect to the distribution $I_\mathcal{A}$) one state $q_0 \in Q$ as the initial state. Define $q_0$ as the current state.
- Generation: Let $q$ be the current state. Decide whether to *stop*, with probability $F(q)$, or to produce a *move* $(q, a, q')$ with probability $P(q, a, q')$, where $a \in \Sigma \cup \{\epsilon\}$ and $q' \in Q$. Output $a$ and set the current state to $q'$. Repeat.

Formally, the PFA $\mathcal{A}$ induces a probability distribution over $\Sigma^*$ in the following manner. Let $\theta = (s_0, x_1, s_1, x_2, s_2, \ldots, s_{k-1}, x_k, s_k)$ denote a *path* traversed while passing through the states $(s_0, \ldots, s_k)$ and producing the string $x = (x_1 x_2 \ldots x_k)$. Equivalently, there is a sequence of transitions $(s_0, x_1, s_1), (s_1, x_2, s_2), \ldots, (x_{k-1}, x_k, s_k) \in \delta_\mathcal{A}$. The probability of generating such a path is:

$$\Pr_\mathcal{A}(\theta) = I_\mathcal{A}(s_0) \cdot \left(\prod_{j=1}^{k} P_\mathcal{A}(s_{j-1}, x_j, s_j)\right) \cdot F_\mathcal{A}(s_k). \tag{1.1}$$

**Definition 1.9** *Given a string $x \in \Sigma^*$, a* valid path *for $x$ is a path with probability greater than zero.*

In general, a string $x$ can be generated by $\mathcal{A}$ through multiple valid paths. The set of all valid paths for $x$ in $\mathcal{A}$ is denoted $\Theta_\mathcal{A}(x)$. The set of all paths in $\mathcal{A}$ which are valid for some string $x \in \Sigma^*$ will be denoted by $\Theta_\mathcal{A}$.

The probability of generating $x$ with $\mathcal{A}$ is:

$$\Pr_\mathcal{A}(x) = \sum_{\theta \in \Theta_\mathcal{A}(x)} \Pr_\mathcal{A}(\theta). \tag{1.2}$$

A natural question is "under which conditions do the probabilities of Equation (1.2) sum up to 1?". A PFA for which this holds is called *consistent.* A sufficient condition for PFA consistency established in [McAllester and Schapire, 2000] is the following:

**Definition 1.10** *A state of a PFA $\mathcal{A}$ is* useful *if it appears in at least one valid path of $\Theta_\mathcal{A}$.*

**Proposition 1.11** *A PFA is consistent if all its states are useful.*

Note that the condition of Proposition 1.11 is sufficient but not necessary: a non useful state is harmless if it is *inaccessible*; i.e., if no string can reach it with probability greater than zero. In the remainder of the thesis we will limit our discussion to consistent PFAs (unless stated otherwise).

**Definition 1.12** *A distribution is* regular *if it can be generated by some PFA.*

The distribution over $\Sigma^*$ induced by the PFA $\mathcal{A}$ will be denoted $D_{\mathcal{A}}$.

A concept closely related to (and which indeed usually assumes the same name as) the PFA is the *non-terminating PFA* (see e.g. [Abe and Warmuth, 1992]):

**Definition 1.13** *A non-terminating PFA is a PFA $\mathcal{A}$ with $F_{\mathcal{A}}(q) = 0$ for all states $q \in Q_{\mathcal{A}}$.*

In this model, the termination probabilities are null for every state, and therefore the model cannot generate any finite-length string. However, for any chosen integer $\ell \in \mathbb{N}$ the model defines a distribution over $\Sigma^\ell$. Specifically, given a length $\ell$ string $x \in \Sigma^\ell$, the individual paths' probabilities are calculated by:

$$\Pr_{\mathcal{A}}(\theta) = I_{\mathcal{A}}(s_0) \cdot \left( \prod_{j=1}^{\ell} P_{\mathcal{A}}(s_{j-1}, x_j, s_j) \right), \tag{1.3}$$

and the string's probability is again given by (1.2):

$$\Pr_{\mathcal{A}}(x) = \sum_{\theta \in \Theta_{\mathcal{A}}(x)} \Pr_{\mathcal{A}}(\theta). \tag{1.4}$$

Sometimes, use of a non-terminating PFA allows a clearer exposition.

**Example 1.1**

In the PFA of Figure 1.1, there is only one valid path for the string *accb*:

$$\Theta_{\mathcal{A}}(accb) = \{(q_0, a, q_1, c, q_1, c, q_1, b, q_3)\}.$$

The probability of *accb* is:

$$\begin{aligned} \Pr_{\mathcal{A}}(accb) &= I(q_0) \cdot P(q_0, a, q_1) \cdot P(q_1, c, q_1) \cdot P(q_1, c, q_1) \cdot P(q_1, b, q_3) \cdot F(q_3) \\ &= 1 \cdot 0.125 \cdot 0.4 \cdot 0.4 \cdot 0.4 \cdot 1 \\ &= 0.008. \end{aligned}$$

For the string $a$, there are two *valid paths*:

$$\Theta_{\mathcal{A}}(a) = \{(q_0, a, q_1), (q_0, a, q_2)\}.$$

Therefore, the probability of $a$ is:

$$\begin{aligned} \Pr_{\mathcal{A}}(a) &= I(q_0) \cdot P(q_0, a, q_1) \cdot F(q_1) + I(q_0) \cdot P(q_0, a, q_2) \cdot F(q_2) \\ &= 1 \cdot 0.125 \cdot 0.2 + 1 \cdot 0.5 \cdot 1 \\ &= 0.525. \end{aligned}$$

**Definition 1.14** *If a PFA's underlying graph is acyclic, the model is called an* acyclic probabilistic finite automaton *or APFA.*

The set $\Theta_{\mathcal{A}}(x)$ can potentially have cardinality proportional to $|Q_{\mathcal{A}}|^{|x|}$, exponential in the length of the string being generated. This precludes any attempt at direct calculation of $\Pr_{\mathcal{A}}(x)$. However, a simple dynamic programming recursion known as the *forward* algorithm was discovered by Chang and Hancock [1966] and later rediscovered by Baum et al. [1970], which reduces the computational complexity to $O(|x| \cdot |\delta|)$, where $|x|$ is the length of $x$ and $|\delta|$ is the number of transitions in $\mathcal{A}$. We now describe their recursion.

### 1.6.1 The Forward and Backward Algorithms

In describing the forward algorithm we follow the notation of [Vidal et al., 2005a] and repeat the relevant parts of their exposition. Let $\alpha_x(i,q)$, for $q \in Q$ and $0 \leq i \leq |x|$, denote the probability of generating the prefix $x_1 \dots x_i$ and reaching state $q$:

$$\alpha_x(i,q) := \sum_{(s_0,s_1,\dots,s_i)\in\Theta_{\mathcal{A}}(x_1\dots x_i)} I(s_0) \cdot \prod_{j=1}^{i} P(s_{j-1}, x_j, s_j) \cdot \mathbf{1}_{\{q=s_i\}}, \tag{1.5}$$

where $\mathbf{1}_{\{q=q'\}} = 1$ if $q = q'$ and 0 if $q \neq q'$. Equation (1.5) can be calculated using the following recursion (usually referred to as the *forward recursion* or *Baum recursion*):

$$\alpha_x(0,q) = I(q), \tag{1.6a}$$

$$\alpha_x(i,q) = \sum_{q'\in Q} \alpha_x(i-1,q') \cdot P(q', x_i, q), \quad 1 \leq i \leq |x|. \tag{1.6b}$$

Given a string $x \in \Sigma^*$, it is straightforward to see that:

$$\Pr_{\mathcal{A}}(x) = \sum_{q\in Q} \alpha_x(|x|,q) \cdot F(q). \tag{1.7}$$

A closely related recursion called the *backward algorithm* calculates the probability (1.2) by introducing the *backward density* $\beta_x(i,q)$ as the probability of generating the suffix $x_{i+1} \dots x_{|x|}$ from the state $q$:

$$\beta_x(i,q) := \sum_{(s_i,\dots,s_{|x|}) \in \Theta_{\mathcal{A}}(x_{i+1} \dots x_{|x|})} \mathbf{1}_{\{q=s_i\}} \cdot \left( \prod_{j=i+1}^{|x|} P(s_{j-1}, x_j, s_j) \right) \cdot F(s_{|x|}). \tag{1.8}$$

The backward density can be calculated using the backward recursion:

$$\begin{aligned} \beta_x(|x|,q) &= F(q), & (1.9a) \\ \beta_x(i,q) &= \sum_{q' \in Q} \beta_x(i+1,q') \cdot P(q,x_i,q'), \quad 0 \le i \le |x|-1. & (1.9b) \end{aligned}$$

For a string $x \in \Sigma^*$ we have:

$$\Pr_{\mathcal{A}}(x) = \sum_{q \in Q} I(q) \cdot \beta_x(0,q). \tag{1.10}$$

The calculation of both $\alpha$ and $\beta$ can be performed with a time complexity of $O(|x| \cdot |\delta|)$.

### 1.6.2 The Viterbi Algorithm

In Equation (1.2), the probability of generating $x$ by the PFA $\mathcal{A}$ is given by the sum of the probabilities over all valid paths that generate $x$. However, in some applications it is desirable to search for a valid path $\tilde{\theta}$ that generates $x$ with the maximal probability,

$$\tilde{\theta}_x := \operatorname{argmax}_{\theta \in \Theta_{\mathcal{A}}(x)} \Pr_{\mathcal{A}}(\theta). \tag{1.11}$$

The probability of this *optimal path* $\tilde{\theta}_x$ will be denoted $\widetilde{\Pr}_{\mathcal{A}}(x)$. In practice, often the probability given by (1.2) is mainly distributed among a few paths close to the optimal one, in which case (1.11) is an adequate approximation. The optimal path $\tilde{\theta}$ is of practical interest in many pattern recognition applications, since useful information can be attached to the states, and in many cases the problem is to search for the information that is in the optimal path. This path is also useful for an efficient estimation of the parameters of the model from a training sample (see e.g. [Ephraim and Merhav, 2002, Section VII]). The computation of $\tilde{\theta}$ can be efficiently performed by defining a function $\gamma_x(i,q)$ $\forall q \in Q, 0 \le i \le |x|$, as the probability of generating the prefix $x_1, \dots, x_i$ through the best path and reaching state $q$:

$$\gamma_x(i,q) := \max_{(s_0,s_1,\dots,s_i) \in \Theta_{\mathcal{A}}(x_1,\dots,x_i)} I(s_0) \cdot \prod_{j=1}^{i} P(s_{j-1}, x_j, s_j) \cdot \mathbf{1}_{\{q=s_i\}}. \tag{1.12}$$

An efficient algorithm for calculating the above path is given by the following well-known recursion due to Viterbi [1967]:

$$\begin{aligned}\gamma_x(0,q) &= I(q),\\ \gamma_x(i,q) &= \max_{q'\in Q}\gamma_x(i-1,q')\cdot P(q',x_i,q), \quad 1\leq i\leq |x|,\end{aligned}$$

with $\widetilde{\Pr}_{\mathcal{A}}(x)$ determined by:

$$\widetilde{\Pr}_{\mathcal{A}}(x) = \max_{q\in Q}\gamma_x(|x|,q)\cdot F(q). \tag{1.13}$$

The calculation of $\gamma$ has the same asymptotic time complexity as the calculation of $\alpha$ or $\beta$. In practice, however, the implementation becomes simpler and the running time faster, as the complicated floating point operations are no longer required.

## 1.7 Probabilistic Deterministic Finite Automata

A special case of the PFA occurs when the following restrictions are imposed:

- $\exists q_0 \in Q$ (initial state), such that $I_{\mathcal{A}}(q_0) = 1$,
- $\forall q \in Q,\ \forall\sigma\in\Sigma,\ |\{q' : (q,\sigma,q') \in \delta_{\mathcal{A}}\}| \leq 1$.

In words, the restrictions amount to requiring a single initial state, and at most a single outgoing edge from any given state, emitting a given alphabet letter. These conditions ensure that for any given string, at most a single path generating the string exists in the automaton.

The resulting model is termed a *probabilistic deterministic finite automaton*, or *PDFA*, also referred to as the *deterministic probabilistic finite automaton* (*DPFA*) or the *stochastic deterministic finite automaton* (*SDFA*). The corresponding term in the information theory literature is the *unifilar finite-state source* or *unifilar source.*

In the PDFA model, the *single* path through states can be deterministically recovered, given the generated string. This fact enables the following recursive notation, which will prove useful in the sequel. Given a PDFA $\mathcal{A} = \langle Q_{\mathcal{A}}, \Sigma, \delta_{\mathcal{A}}, I_{\mathcal{A}}, F_{\mathcal{A}}, P_{\mathcal{A}}\rangle$, the transition function and transition probabilities can be recursively defined. Given a state $q \in Q_{\mathcal{A}}$ and a string $s = s_1 \ldots s_\ell$,

- $\delta_{\mathcal{A}}(q,s) := \delta_{\mathcal{A}}(\delta_{\mathcal{A}}(q,s_1), s_2\ldots s_\ell)$,
- $P_{\mathcal{A}}(q,s) := P_{\mathcal{A}}(q,\delta_{\mathcal{A}}(q,s_1),s_1)\cdot P_{\mathcal{A}}(\delta_{\mathcal{A}}(q,s_1), s_2\ldots s_\ell)$.

The probability that $\mathcal{A}$ will generate a $s$ and *terminate*, given the current state of the PDFA is $q$ will be denoted by:

$$P^q_{\mathcal{A}}(s) := P_{\mathcal{A}}(q, s) \cdot F_{\mathcal{A}}(\delta_{\mathcal{A}}(q, s)).$$

**Definition 1.15** *A distribution is* regular deterministic *if it can be generated by a PDFA.*

The regular deterministic distributions are a strict subset of the regular distributions[6]. This observation was proved by a simple counterexample in [Vidal et al., 2005a, chapter IV], which we reproduce in Figure 1.2 below.

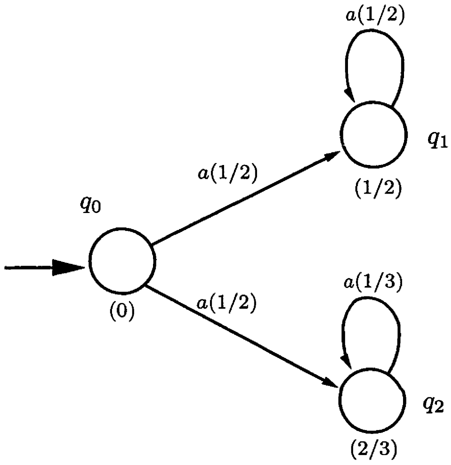

**Figure 1.2:** A simple PFA inducing a distribution which cannot be generated by a PDFA, using the *single-letter* alphabet $\Sigma = \{a\}$.

In Section 3.3.1 we will discuss this example again and offer an alternative proof.

We mention that the concept of a PFA constraint naturally extends to the so-called *deterministic constraint*, the situation (corresponding to a PDFA subclass with a fixed structure but variable transition probabilities) wherein a PFA has one initial state and for every given state $q_i$ and alphabet letter $\sigma \in \Sigma$, there is at most one transition from $q_i$ labeled with $\sigma$ in the transitions probabilities matrix.

## 1.8 Connections to Related Probabilistic Models

We now mention a number of related probabilistic models, and briefly discuss their relation to the PFA. The discussion is arranged in an increasing order of the models' expressivity.

[6] Which correspond to the stochastic deterministic regular grammars introduced in Section 1.4.

We repeat parts of the exposition of Vidal et al. [2005b, Section II], where a more complete exposition is offered.

The class of *probabilistic residual finite state automata* (PRFA) was introduced in [Esposito et al., 2002] and shown to have expressive capability (strictly) greater than PDFA and (strictly) less than PFA models. We defer the definition of PRFA to Section 3.1, as it relies on technical concepts introduced in that section (and is discussed solely in a context relevant to Chapter 3).

### 1.8.1 Connections to the Hidden Markov Model

PFA models are closely related to the well-known hidden Markov models (HMMs), which are used in numerous practical applications including speech recognition [Rabiner, 1989], handwritten text recognition [Raviv, 1967], machine translation [Jelinek, 1998], and bioinformatics [Abe and Mamitsuka, 1997][7]. We define the HMM below and briefly discuss its relation to the PFA.

**Definition 1.16** *A HMM is a 6-tuple $\mathcal{M} = \langle Q, \Sigma, I, q_f, T, E \rangle$, where:*

- *$Q$ is a finite set of states,*
- *$\Sigma$ is a finite alphabet of symbols,*
- *$T : (Q \setminus \{q_f\}) \times Q \to \mathbb{R}^+$ is a state to state transition probability function,*
- *$I : Q \setminus \{q_f\} \to \mathbb{R}^+$ is an initial state probability function,*
- *$E : (Q \setminus \{q_f\}) \times \Sigma \to \mathbb{R}^+$ is a state-based symbol emission probability function,*
- *$q_f \in Q$ is a special (final) state,*

*subject to the following normalization conditions:*

$$\begin{aligned}
\sum_{q \in Q \setminus \{q_f\}} I(q) &= 1, \\
\sum_{q \in Q} T(q, q') &= 1, \ \forall q \in Q \setminus \{q_f\}, \\
\sum_{a \in \Sigma} E(q, a) &= 1, \ \forall q \in Q \setminus \{q_f\}.
\end{aligned}$$

The main distinction between the PFA and the HMM is the manner in which the models' output is generated. Specifically, in the HMM, the output is generated at the

[7] See [Vidal et al., 2005a] for additional references.

states, while in the PFA the output is generated upon traversal of the edges[8]. Keeping this distinction in mind, the following propositions proven in [Vidal et al., 2005b] are intuitive ($D_{\mathcal{M}}$ denotes the distribution generated by $\mathcal{M}$):

**Proposition 1.17** *Given a PFA $\mathcal{A}$ with $m$ transitions and* $\Pr_{\mathcal{A}}(\epsilon) = 0$, *there exists an HMM $\mathcal{M}$ with at most $m$ states such that $D_{\mathcal{A}} = D_{\mathcal{M}}$.*

**Proposition 1.18** *Given an HMM $\mathcal{M}$ with $n$ states there exists a PFA $\mathcal{A}$ with at most $n$ states such that $D_{\mathcal{A}} = D_{\mathcal{M}}$.*

Figure 1.3 summarizes the relative expressive capabilities of the probabilistic models mentioned above. Note that each class is unbounded in the number of states, and no attempts are made at approximation of any class by another.

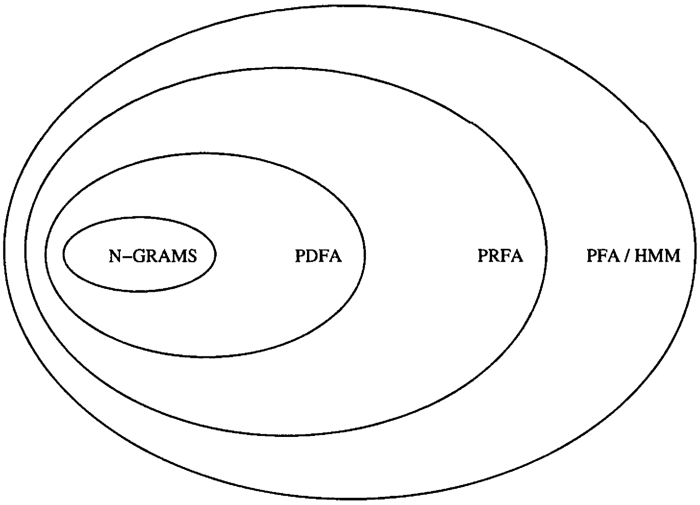

**Figure 1.3:** A hierarchy of probabilistic models. The models are arranged in order of expressive capabilities, with the assumption that the number of states in each class is unbounded.

Many practical applications involving HMMs generalize the alphabet to (typically high dimensional) continuous spaces. The discussion of the ensuing conceptual and algorithmic modifications lies beyond the scope of this thesis.

[8] There are (uncommon) exceptions to this rule, wherein HMMs are defined to produce emissions on transitions instead of states, see e.g. [Casacuberta, 1990] and [Bahl et al., 1983].

## 1.9 Distance Functions Between Distributions

Given two distributions $D_1$ and $D_2$ over $\Sigma^*$, we will use the following notions of distance: For $1 \leq p < \infty$, the $L_p$ distance between $D_1$ and $D_2$ is defined as:

$$\|D_1 - D_2\|_p := \left[ \sum_{s \in \Sigma^*} |D_1(s) - D_2(s)|^p \right]^{1/p}.$$

In the limit $p \to \infty$, we obtain the $L_\infty$ distance:

$$\|D_1 - D_2\|_\infty := \max_{s \in \Sigma^*} |D_1(s) - D_2(s)|.$$

The distance between a given distribution $D'$ and a class of distributions $\mathcal{D}$ is defined as:

$$\| \mathcal{D} - D' \| := \inf_{D \in \mathcal{D}} \|D - D'\|.$$

An additional notion of proximity between distributions with deep roots in the information theory literature is the KL-divergence, defined as:

$$\mathrm{KL}(D_1 \parallel D_2) := \sum_{s \in \Sigma^*} D_1(s) \log \left( \frac{D_1(s)}{D_2(s)} \right).$$

The KL-divergence can be interpreted as the expected extra message-length per datum that must be communicated if a code that is optimal for a given (wrong) distribution $D_2$ is used, compared to using a code based on the (correct) distribution $D_1$. Note that although frequently used as a criterion for proximity between distributions, the KL-divergence is *not* a metric. Pinsker's inequality (see e.g. [Cover and Thomas, 1991]) states that for any pair of distributions $D_1$ and $D_2$ the following holds:

$$\mathrm{KL}(D_1 \| D_2) \geq \frac{1}{2 \ln 2} \|D_1 - D_2\|_1^2,$$

which in turn upper-bounds all $L_p$-distances, making this notion of distance stronger. We mention that certain refinements of Pinsker's inequality have been proposed. These refinements are usually not dependent on the distributions $D_1$ and $D_2$, with the exception of a result discussed in [Weissman et al., 2003].

Chapter 2

# The Geometry of Probabilistic Automata

In this chapter we discuss aspects of the geometry underlying probabilistic finite automata models, especially convexity, which is relevant for learnability. As we will discuss in Chapter 4, a number of hardness results show that the general problem of PFA learning is a difficult one. In this chapter we will try to illuminate some of the geometrical aspects behind PFA learning, providing insight into the general problem's difficulty.

## 2.1 Basic Geometric Properties of PFA

In this section, we will discuss the geometric relationship between a PFA's parameters and its induced distribution. We will focus on the transient (as opposed to stationary) behaviour induced by the model. At the end of the section, we will relate our observations to some existing results describing stationary behaviour.

Following Definition 1.13 of Section 1.6, we will use the ($n$-state) *non-terminating* PFA model in order to streamline the exposition. Denoted by $\mathcal{A}$, the model is parameterized by its initial state probabilities $I_{\mathcal{A}}$ ($n$ parameters[1]) and its transition probabilities $P_{\mathcal{A}}$ ($|\Sigma|n^2$ parameters[2]). Given these, the set of transitions $\delta_{\mathcal{A}}$ can be inferred. Graphically, the transition matrix can be pictured as a "stack" composed of $|\Sigma|$ matrices of size $n \times n$ , as illustrated in Figure 2.1. This is the most common parameterization of PFA, which we will refer to as the *usual parameterization.*

### 2.1.1 Extreme Points of the $n$-State PFA Set

**Definition 2.1** *Let $C$ be a convex subset of a vector space $X$. A point $x \in C$ is called an* extreme point *if it is not an interior point of any line segment in $C$. That is, $x$ is extreme if and only if $x = \lambda y + (1-\lambda)z$, $\lambda \in (0,1)$ implies either $y \notin C$ or $z \notin C$.*

[1] More precisely $(n-1)$ *free* parameters, due to the sum-to-one constraint.

[2] Rather $|\Sigma|n(n-1)$ free parameters via similar reasoning. We will henceforth neglect the (inconsequential) effects of the sum-to-one constraints on the number of parameters.

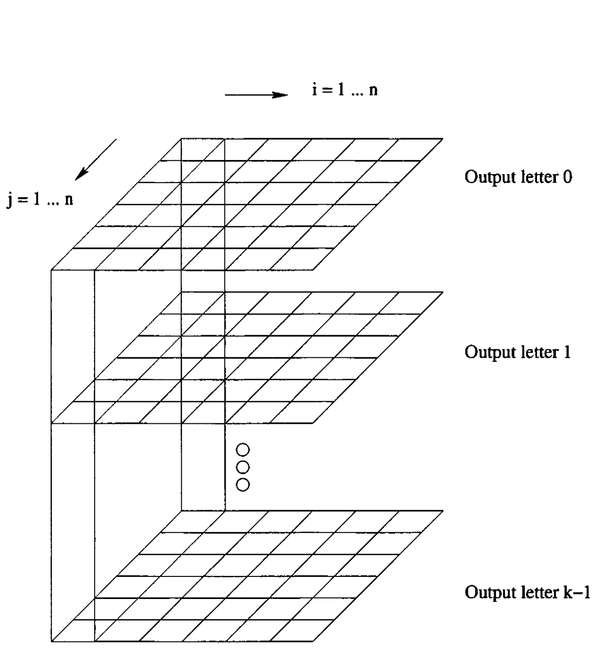

**Figure 2.1:** Matrix stack representing a PFA transition function. The red "slice" on the left-hand side denotes all outgoing edges from state $i = 1$, and its entry sum is 1.

**Proposition 2.2** *The number of extreme points of the n-state PFA set (in the usual parameterization) is* $n(|\Sigma|n)^n$.

**Proof** For each $\sigma \in \Sigma$, the $n \times n$ matrix $[P_{\mathcal{A}}(q_i, \sigma, q_j)]_{i,j=1}^{n}$ is a *row stochastic* matrix (i.e. all entries nonnegative with rows summing to one). An easy observation in nonnegative matrix theory [Bapat and Raghavan, 1997] shows that the set of row stochastic matrices of order $n$ is isomorphic to a polytope in $\mathbb{R}^{n \times n}$ . This polytope has $n^n$ "vertices" or *extreme points*, specifically the matrices with a single entry of 1 in each row.

For the transition parameters $P_{\mathcal{A}}$, the number of extreme points is $(|\Sigma|n)^n$, specifically the transition matrices with exactly one entry 1 on each $|\Sigma| \times n$ "slice" (see red highlight in Figure 2.1) with all other entries 0's. In other words, the location of the 1 entries can be chosen independently on $n$ slices, each offering $|\Sigma|n$ possible positions, hence $(|\Sigma|n)^n$ combinations. The proof that these are indeed the extreme points of the relevant set follows immediately from the non-negativity and sum-to-one properties. The additional degree of freedom stemming from the initial state distribution $I_{\mathcal{A}}$ adds a multiplicative factor of $n$ to the number of extreme points, arriving at a total of $n(|\Sigma|n)^n$. ■

We will next use the PFA as a means of generating a probability distribution over $\Sigma^{\ell}$, using Equation (1.4) as a *mapping* from parameter space onto the $|\Sigma|^{\ell}$ probability simplex. Note that in the space of probability distributions over $\Sigma^{\ell}$, the number of parameters required to fully characterize an arbitrary distribution is $|\Sigma|^{\ell}$, while the number of parameters characterizing a PFA is merely $|\Sigma|n^2 + n$. Note also that the PFA's states may always be permuted without affecting the induced distribution. This ambiguity can be removed by ordering the states.

The extreme points described above denote *extreme PFA* which are "deterministic" in the sense that given a specific starting state, the ensuing state transitions as well as the automaton's output are uniquely determined. However, they should not be confused either with finite state machines, with deterministic PFA constraints, or with PDFAs. Rather, they are maximally degenerate cases of PFA. In general, finite state machines (aside from the extreme PFA special cases) generate (generally infinite) *languages* as opposed to (valid) distributions. For any $\ell > 0$, an extreme PFA generates a delta distribution $\delta(x),\ x \in \Sigma^\ell$.

### 2.1.2 Geometry of the n-State PDFA Subset

**Proposition 2.3** *Assuming $n \geq |\Sigma|$, the PDFA subset (in the n-state PFA parameter set) occupies a union of $\left(\prod_{i=0}^{|\Sigma|-1}(n-i)\right)^n$ convex hulls of $|\Sigma|n$ extreme points each.*

**Proof** Assuming $n \geq |\Sigma|$, the largest set of PFA extreme points whose convex hull still includes only PDFAs contains $|\Sigma|n$ points. Such a set is readily constructed by taking at each $n \times |\Sigma|$ slice any $|\Sigma|$ extreme points with only a single "1" on each row, and repeating the process $n$ times independently along the direction $i$.

The number of ways such a maximal set may be chosen is $\left(\prod_{i=0}^{|\Sigma|-1}(n-i)\right)^n$. Thus, the PDFA subset occupies a union of $\left(\prod_{i=0}^{|\Sigma|-1}(n-i)\right)^n$ convex hulls of $|\Sigma|n$ extreme points each. ■

The PDFA subset includes many (intersecting) linear sections of the complete PFA polytope, and is non-convex. For a *single* linear section defining a deterministic constraint, however, an efficient method for determining the optimal set of parameters exists. This is due to the fact that the optimization problem involved *is* convex. We will discuss this fact again in Section 4.1.1.

### 2.1.3 Geometric Properties of the PFA Mapping

We now develop some intuition regarding the PFA parameters' mapping onto the $|\Sigma|^\ell$-dimensional probability simplex (Equation (1.4)). In order to avoid degeneracies in our presentation, we restrict our attention to the case $\ell > n$. We will present a series of propositions regarding the PFA mapping onto the probability simplex. A visualization is given in Figure 2.2 below:

In the figure, the left hand side corresponds to the convex polytope describing the PFA parameterization (of dimensionality $|\Sigma|n^2 + n$). The right hand side describes the

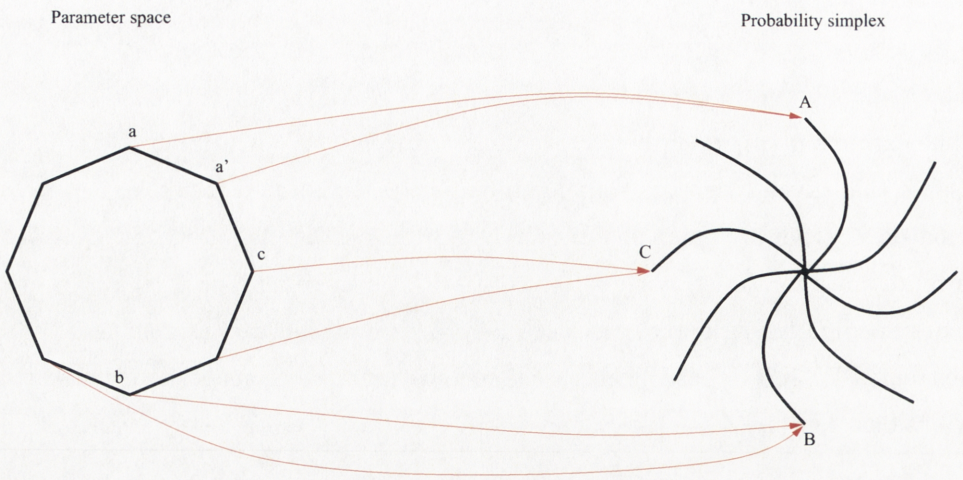

**Figure 2.2:** Visualization of the interplay between convex combinations of row stochastic matrix stacks parameterizing PFA (on left) and the induced distributions over $\Sigma^\ell$ (on right).

resulting set on the ($|\Sigma|^\ell$-dimensional) probability simplex. For $n = 1$, $\Sigma = \{0, 1\}$ and $\ell = 2$, a visualization of the probability simplex is provided in Figure 2.3.

### 2.1.4 Extreme PFA and the PFA Mapping

We now consider the interplay between extreme points in parameter space and on the probability simplex.

**Proposition 2.4** *Each extreme point in the parameter space is mapped to an extreme point on the probability simplex. However, assuming $\ell > n$, not all strings in $\Sigma^\ell$ can be induced by* extreme PFA *(i.e. corresponding to extreme points in parameter space).*

**Proof** An extreme point in the parameter space always induces a delta distribution on $\Sigma^\ell$, which is an extreme point on the probability simplex. In the example of Figure 2.3, the strings 00 and 11 can be induced by extreme single-state PFA, while the strings 01 and 10 cannot, proving the proposition's second statement. ■

**Proposition 2.5** *The mapping from matrix stack space to the probability simplex is not one-to-one. Furthermore, the set of extreme PFA which induce identical delta distributions on $\Sigma^\ell$ are not only those with permuted states, as the example in Figure 2.4 shows.*

This proposition is graphically depicted in Figure 2.2 by the mappings $a, a' \to A$.

**Proof** The automata (a), (b) and (c) of Figure 2.4, all of which are extreme points of the set of 3-state automata over $\Sigma = \{0, 1\}$, induce an identical distribution for any $\ell$. ■

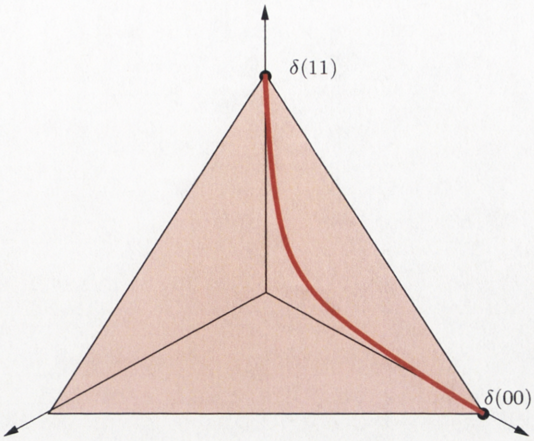

**Figure 2.3:** Visualization of the probability simplex for automata of the family $n = 1, \Sigma = \{0, 1\}$ and $\ell = 2$, graphically depicted in Figure 2.5(d). The space of all possible probability distributions over $\Sigma^2$ is depicted by the pink region (which should actually be a 3-dimensional hyperplane in a 4-dimensional space), while the distributions which can be induced by one-state automata are depicted by the curved red line.

We now establish bounds on the number of *distinct* images of extreme PFA on the probability simplex.

**Proposition 2.6** *The number of distinct delta distributions induced by the set of extreme $n$-state PFA is lower-bounded by $|\Sigma|^n$ and upper-bounded by $n|\Sigma|^n$.*

**Proof** In this context, we restrict our attention to extreme PFA for which all states are accessible[3]. An immediate lower bound on the number of distinct produceable strings is $|\Sigma|^n$, the number of different length-$n$ prefixes of length-$\ell$ strings.

All extreme PFA have only one outgoing edge from each state. In order to ensure no inaccessible states, all states must point to an as-yet unseen state, *except the last state*, which is free to point to any one of the $n$ states. Thus, an upper bound on the number of distinct images of extreme PFA is $n|\Sigma|^n$.

■

[3] It is easy to show that any distribution generated by an extreme PFA with inaccessible states may also be generated by an extreme PFA with no inaccessible states.

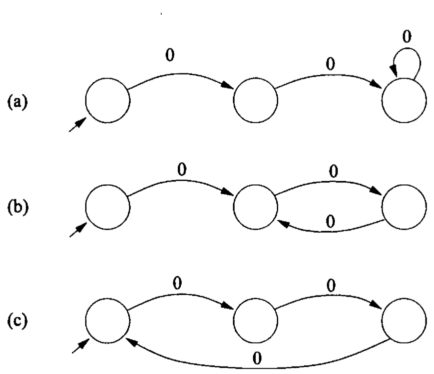

**Figure 2.4**: A set of extreme PFA inducing identical delta distributions.

## 2.2 Convexity Properties

It seems natural in this context to ask when the *image set* under the PFA mapping (1.4) of a particular PFA constraint is convex. A convex combination of two extreme points in parameter space results in a valid PFA, a fact which follows immediately from the definitions. An example based on the PFA displayed in Figure 2.5 is instructive. We consider the space of 1-state PFA over the alphabet $\Sigma = \{0, 1\}$. In parameter space, this set has 2 extreme points, namely the automata shown in (a) and (b) of Figure 2.5. A convex combination of the two extreme points *in parameter space* takes the form of Figure 2.5(d), which induces the distribution:

$$\Pr(x) = \lambda^{n_0(x)} \cdot (1 - \lambda)^{n_1(x)}, \tag{2.1}$$

where $n_0(x)$ and $n_1(x)$ denote the number of zeros and ones respectively in the word $x$. The function is continuous in $\lambda$ (as is depicted in Figure 2.3). For any fixed $\ell \in \mathbb{N}$, a convex combination of the two automata (a) and (b) of Figure 2.5 on the probability simplex will induce a distribution of the form:

$$x = \begin{cases} 0^\ell & \text{w.p.} \quad \lambda \\ 1^\ell & \text{w.p.} \quad 1 - \lambda. \end{cases}$$

This distribution cannot be induced by a single-state PFA (as it does not match Equation (2.1) for any value of $\lambda$), and requires a two-state automaton, depicted in Figure 2.5(c). This example shows that the image under the PFA mapping of $n$-state PFAs is (in general) not convex. In general, a convex combination (on the probability simplex) of $k$ PFAs of $n$ states each requires a $(kn)$-state PFA to realize (by a trivial construction).

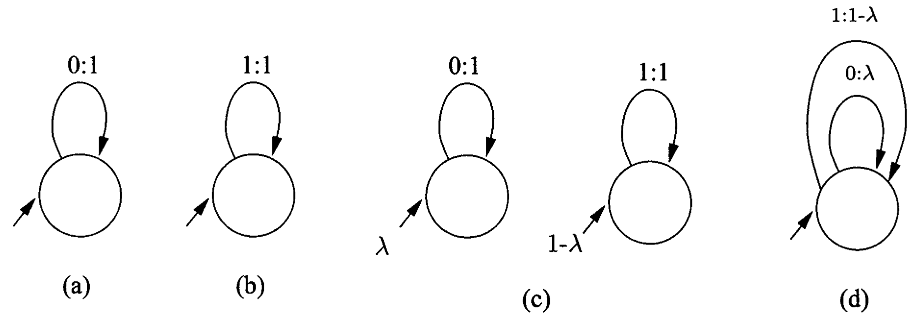

**Figure 2.5:** Extreme points and convex combinations of single state PFA over $\Sigma = \{0,1\}$. The automata (a) and (b) denote the two extreme points, the automaton (c) shows the resulting convex combination on the probability simplex, while the automaton (d) shows the convex combination induced in parameter space. The parameters $\lambda$ and $1-\lambda$ of (c) denote the initial probabilities of each state.

### 2.2.1 Convexity of Distributions Induced by PFA Constraints

We now present two sufficient conditions for a PFA constraint set to be mapped to a non-convex set. We require the following definition:

**Definition 2.7** *Let $C$ be a PFA constraint, and let $\mathcal{C}$ be the set of PFA allowable by the constraint $C$. If a transition parameter can assume all values in $[0,1]$ given the constraint $C$, then it is called a* free parameter.

The first condition shows that free parameters on cycles immediately imply non-convexity.

**Proposition 2.8** *Let $C$ be a PFA constraint with a cycle of strictly positive probability containing a free parameter. Let $\mathcal{C}$ be the set of PFA parameters allowable by the constraint $C$. Then the image (on the probability simplex) of $\mathcal{C}$ is non-convex.*

**Proof** Let $\mathcal{D}^{\text{ext}}(\mathcal{C},\ell)$ be the set of extreme distributions over $\Sigma^\ell$ induced by all $\mathcal{A} \in \mathcal{C}$. Let $V = q_{i_1} \cdots q_{i_c}$ denote a cycle with a free parameter. Let $v \in \Sigma^*$ be the sequence of letters emitted while traversing $V$. Fix all free parameters, and let $xvz \in \Sigma^\ell$ be some positive probability string emitted by $\mathcal{A}$, as depicted in Figure 2.6.

This implies that for all $xv^k \in \Sigma^{\leq \ell}$, $k \in \mathbb{N}$, there exist strings with prefixes $xv^k$ which $\mathcal{A}$ generates with strictly positive probabilities. Therefore, $\mathcal{A}$ generates an unbounded number of strings with positive probability.

The number of distinct extreme distributions induced by $\mathcal{A}$, however, is bounded. Appealing to Proposition 2.6, we obtain the upper bound

$$|\mathcal{D}^{\text{ext}}(\mathcal{C},\ell)| \leq n|\Sigma|^n.$$

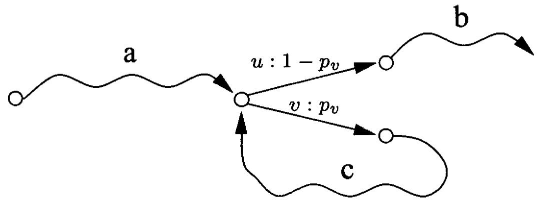

**Figure 2.6:** Construction used for showing all PFA constraints with free parameters on cycles induce non-convex distribution sets.

This upper bound does not depend on $\ell$. Thus, for some large enough $\ell$, there exists a string $z \notin \mathcal{D}^{\text{ext}}(C, \ell)$ such that $\Pr_{\mathcal{A}}(z) > 0$, implying $D_{\mathcal{A}} \notin \operatorname{co}\left(\mathcal{D}^{\text{ext}}(C, \ell)\right)$, implying the image set is non-convex. ■

The second condition illustrates a different situation, showing that cycles are not the only culprits responsible for non-convexity.

**Proposition 2.9** *Let $C$ be a* deterministic *PFA constraint with no free parameters on cycles. Let $|C|$ denote the number of free parameters in $C$. Let $\mathcal{C}$ be the set of PFA allowable by the constraint $C$ and let $\mathcal{D}^{ext}(C, \ell)$ be the set of extreme distributions over $\Sigma^\ell$ induced by all $\mathcal{A} \in \mathcal{C}$. Then if for some $\ell \in \mathbb{N}$*

$$|C| < |\mathcal{D}^{ext}(C, \ell)|,$$

*then the image (on the probability simplex) of $\mathcal{C}$ is non-convex.*

**Proof** By the assumption, for some $\ell \in \mathbb{N}$, the number of free parameters is strictly smaller than the number of extreme distributions. The PFA constraint $C$ is deterministic, and therefore each string generated with positive probability by $\mathcal{A}$ follows a single path. Therefore, by the pigeonhole principle there exists a free parameter $p_f$ and (at least) two strings $s_1, s_2 \in \Sigma^\ell$ (both extreme distribution) such that $p_f = 0 \Rightarrow \Pr_{\mathcal{A}}(s_1) = \Pr_{\mathcal{A}}(s_2) = 0$. A convex combination with $\Pr_{\mathcal{A}}(s_1) = 0$ and $\Pr_{\mathcal{A}}(s_2) > 0$ can therefore not be obtained by $\mathcal{A}$, proving the proposition. ■

Despite the results above, the problem of training a PFA with a deterministic constraint[4] *is* a convex optimization problem, and is indeed efficiently solvable. This fact was proved in [Abe and Warmuth, 1992, Chapter 4][5]. However, the general PDFA learning

[4] Namely, setting the PFA's parameters to maximize the likelihood of a given sample multiset, as defined rigorously in Section 4.1.1.

[5] The original proof, however, predates the paper.

problem (i.e. over the set of *all* deterministic constraints) is readily shown to be non-convex, even when the PDFA structure is constrained to be acyclic. Indeed, the acyclic $n$-state PDFA learning problem is widely believed to be hard, as implied by the reduction of Kearns et al. [1994] which we will discuss in detail in Section 4.3.1).

### 2.2.2 Convex Optimization in Distribution Space

The following result shows that direct optimization[6] of PFAs in distribution space cannot amount to useful learning.

**Proposition 2.10** *Let $\mathcal{R} = \{r_1, \ldots, r_z\}$ be an arbitrary finite set of distributions over $\Sigma^\ell$. Let $\mathcal{D}^{ext}(\ell) = \{d_1^{ext}, \ldots, d_t^{ext}\}$ be the set of* all *extreme distributions induced by $n$-state PFA over $\Sigma^\ell$.*

*Then for all possible sets $\mathcal{R}$ such that $z = |\,\mathcal{R}\,| = c_0|\Sigma|^n$ for some $c_0 < 1$, there exists a distribution $d_b^{ext} \in \mathcal{D}^{ext}(\ell)$ such that:*

$$\left\| co(\mathcal{R}) - d_b^{ext} \right\|_1 \geq 1 - \frac{c_0}{|\Sigma|^{\ell-n}}.$$

**Proof** Using Proposition 2.6, the number of distinct delta distributions induced by the set of extreme $n$-state PFA is lower-bounded by $|\Sigma|^n$. The fact that $z = c_0|\Sigma|^n$ with $c_0 < 1$ has the following implication. All distributions $r \in \mathcal{R}$ are over $\Sigma^\ell$, and we assume $\ell > n$. The distribution $d_i^{\text{ext}} \in \mathcal{D}^{\text{ext}}(\ell)$ is a delta distribution inducing the string $s_i \in \Sigma^\ell$. Therefore, there exists at least one distribution $d_b^{\text{ext}} \in \mathcal{D}^{\text{ext}}(\ell)$ such that for all distributions $r \in \mathcal{R}$, the following holds:

$$\max_{r \in \mathcal{R}} r(s_b) \leq \frac{z}{|\Sigma|^\ell} = \frac{c_0}{|\Sigma|^{\ell-n}}.$$

As a direct consequence, we have:

$$\left\| \text{co}(\mathcal{R}) - d_b^{\text{ext}} \right\|_1 \geq |\max_{r \in \mathcal{R}} r(s_b) - 1| \geq 1 - \frac{c_0}{|\Sigma|^{\ell-n}},$$

proving the proposition. ■

It follows that even for $n = \ell$, an exponentially large number of distributions would be needed to directly approximate the *entire* PFA set on the probability simplex.

## 2.3 Related Results

We briefly mention a number of relevant results from the information theory and statistics literature, placing the chapter's results in a larger context. The results mentioned below

[6] Namely, an approximation of the target distribution using a mixture of distributions.

were formulated for HMMs, but can be straightforwardly adapted to PFA.

### Equivalence Classes of Identifiable PFAs

The *identifiable* HMM subclass was defined to resolve some of the degeneracies discussed above. This class, however, is defined only for *stationary* HMMs (those for which the likelihood of occupying a given state is independent of the time). An adaptation of the definition for PFA follows:

**Definition 2.11** *A* stationary *PFA $\mathcal{A}^0$ with initial and state transition parameters $I^0_{\mathcal{A}}$ and $P^0_{\mathcal{A}}$ is said to be* identifiable *if for every PFA $\mathcal{A}$ such that $(I_{\mathcal{A}}, P_{\mathcal{A}}) \neq (I^0_{\mathcal{A}}, P^0_{\mathcal{A}})$, there exists some $\ell > 0$ for which the distributions over $\Sigma^\ell$ induced by $\mathcal{A}^0$ and $\mathcal{A}$ are not identical.*

Leroux [1992, Lemma 2] showed that the equivalence class of the parameters of an identifiable HMM comprises all parameters obtained by permutation of the HMM's states. This result trivially carries over to the PFA model.

### Exponential Forgetting

A property related to stationarity termed *exponential forgetting* was formulated and shown for HMMs with *primitive* transition matrices in [LeGland and Mevel, 2000, Theorem 2.2]. In non-technical terms, the likelihood function associated with PFAs exhibiting this property has an exponential rate of forgetting of the initial conditions.

### Lipschitz Continuity of the Forward Mapping

We also mention a relevant Lipschitz continuity result shown in [LeGland and Mevel, 2000, Theorem 2.1], who showed that for an HMM with a *primitive* state transition matrix[7], a Lipschitz continuity property holds for the forward recursion (1.6). Their result places an upper bound on the Lipschitz constant.

## 2.4 Discussion: Implications for Automata Learning

General PFA or PDFA learning is a non-convex problem of exponential dimensionality (in the number of states), and therefore need be approached with due caution. Indeed, in their comprehensive survey paper Ephraim and Merhav [2002] comment that:

> Algorithms for global maximization of the likelihood function $p(y^n; \phi)$ over $\phi \in \Phi$ [the usual HMM parameterization] are not known for most interesting HMMs.

[7] A stochastic matrix $Q$ is primitive if there exists an integer $r$ such that the matrix $Q^r$ is positive

We will briefly discuss a number of established local optimization methods suited to PFA learning in Section 4.1.1. Moreover, in subsequent sections of Chapter 4 we will discuss and analyze a number of algorithms dealing with learning (certain subclasses of) PDFA under certain (restrictive) frameworks. The positive learnability results which will be discussed there are not geometric (i.e. relying on convex optimization), but are rather due to efficient branching algorithms utilizing provably accurate statistical tests.

An interesting avenue for extending the research presented in this chapter involves the construction of distribution families which enable efficient learnability (and evaluation) on one hand, while providing good approximation to (interesting subsets of) PFA on the other.

Chapter 3

# The Myhill-Nerode Theorem for PFA

Characterization of distribution families is an important step toward gaining an understanding of the families' expressive capabilities. In the context of formal languages, the classical Myhill-Nerode characterization theorem provides a useful tool for proving certain languages *do not* belong to the *regular language* class. In the present chapter we extend the Myhill-Nerode theorem to PFA, and present a number of applications.

After reviewing related work (Section 3.1), we present an extension of the well-known Myhill-Nerode theorem of finite state machines to the PFA family in Section 3.2. We then show how the Myhill-Nerode extension theorem provides a tool for proving that certain distributions cannot be modelled by PDFA or PFA models (Section 3.3), and show how it implies bounds on the relative expressive power of PFA and PDFA models in approximating arbitrary probability distributions over bounded length strings (Section 3.4).

## 3.1 Related Work

We begin by stating the original Myhill-Nerode theorem [Hopcroft and Ullman, 1979], followed by an explanation of the terms and the theorem's significance:

**Theorem 3.1 (Myhill-Nerode)** *The following three statements are equivalent:*

1. *A formal language $L \subseteq \Sigma^*$ is accepted by some finite automaton.*

2. *$L$ is the union of some of the equivalence classes of an extension invariant equivalence relation of finite index.*

3. *The equivalence relation on $\Sigma^*$ induced by $L$ is of finite index.*

The Myhill-Nerode theorem is concerned with a natural equivalence relation on strings induced by a DFA. Given a DFA $\mathcal{A}$ and two strings $x, y \in \Sigma^*$, we say that $x$ and $y$

are *equivalent* modulo $\mathcal{A}$ if after their generation the DFA reaches the same state. The resulting relation between strings constitutes an *equivalence relation.*

An equivalence relation induces *equivalence classes*, (possibly infinite) sets of strings which are equivalent. The (possibly infinite) number of equivalence classes in an equivalence relation is called its *index.* An equivalence relation $E$ on $\Sigma^*$ is called *extension invariant* iff for all $x, y, z \in \Sigma^*$, $xEy \Rightarrow xzEyz$.

The Myhill-Nerode theorem states that a DFA is characterized by two properties: that the equivalence relation it induces is of finite index and that it is invariant under the extension of the strings by the same characters.

A related DFA result which is also sometimes called the Myhill-Nerode theorem, is described in the following [Hopcroft and Ullman, 1979]:

**Theorem 3.2** *The minimum state automaton accepting L is unique up to an isomorphism (i.e. a permutation of the states).*

### 3.1.1 A PDFA Extension to the Myhill-Nerode Theorem

For the class of PDFA models, Theorem 3.2 was generalized and proved by Carrasco and Oncina [1999]. We repeat their result here (using somewhat different terminology):

**Theorem 3.3** *If L is a SDRL*[1]*, then a* canonical generator, *or a* minimal *PDFA generating L exists.*

We repeat the construction of the minimal PDFA generating a stochastic deterministic regular language (termed the *canonical generator*), which was shown in [Carrasco and Oncina, 1999]. This construction is based on the definition of an equivalence relation between strings on the one hand and between states on the other. Given an SDRL $L$, the minimal PDFA generating $L$ is given by $\mathcal{M} = \langle Q_\mathcal{M}, \Sigma, \delta_\mathcal{M}, q_{0_\mathcal{M}}, P_\mathcal{M}, F_\mathcal{M} \rangle$, where:

$$
\begin{aligned}
Q_\mathcal{M} &= \{x^{-1}L \neq \emptyset : x \in \Sigma^*\} \\
\delta_\mathcal{M}(x^{-1}L, a) &= (xa)^{-1}L, \\
q_{0_\mathcal{M}} &= \epsilon^{-1}L \\
P_\mathcal{M}(x^{-1}L, a) &= p(a\Sigma^*|x^{-1}L).
\end{aligned}
$$

A characterization theorem complementing the result in [Carrasco and Oncina, 1999] and completing the Myhill-Nerode extension for PDFA models was formulated (using somewhat different notation) and proved by Esposito et al. [2002, Theorem 3]:

[1] *I.e., stochastic deterministic regular language, see definition in Section 1.3.*

**Theorem 3.4 (Myhill-Nerode Extension for PDFA Models)** *Let $\Sigma$ be a finite alphabet. The following two statements are equivalent:*

1. *A distribution $D$ over $\Sigma^*$ can be induced by an $n$-state PDFA.*
2. *There exists a set of $n$ fixed distributions $\mathcal{D} = \{D_1, \ldots, D_n\}$ over $\Sigma^*$ such that all quotient languages of $D$ are members of $\mathcal{D}$.*

### 3.1.2 A PRFA Extension to the Myhill-Nerode Theorem

Esposito et al. [2002] also formulated and proved a characterization theorem for the class of *probabilistic residual finite automata* (PRFA)[2], which we define now.

Let $\mathcal{A}$ be a PFA, and let $e_k$ be a vector with the $k^{th}$ entry equal to 1 and all other entries 0. Define the PFA $\mathcal{A}_{q_k}$ $(k = 1, \ldots, n)$ by $\mathcal{A}_{q_k} = \langle Q_{\mathcal{A}}, \Sigma, \delta_{\mathcal{A}}, e_k, F_{\mathcal{A}}, P_{\mathcal{A}} \rangle$ (i.e. a PFA identical to $\mathcal{A}$, except for the initial state distribution which is concentrated on the $k^{th}$ state, $q_k$), and let $L_{q_k}$ be the stochastic language induced by $\mathcal{A}_{q_k}$.

**Definition 3.5** *A PRFA is a PFA $\mathcal{A} = \langle Q_{\mathcal{A}}, \Sigma, \delta_{\mathcal{A}}, I_{\mathcal{A}}, F_{\mathcal{A}}, P_{\mathcal{A}} \rangle$ such that*

$$\forall q \in Q_{\mathcal{A}}, \quad \exists u \in \Sigma^* \text{ such that } L_q = u^{-1} L_{\mathcal{A}},$$

*where $L_{\mathcal{A}}$ is the stochastic language induced by $\mathcal{A}$ and $L_q$ is the stochastic language induced by $\mathcal{A}_q$.*

In other words, a PRFA is a PFA such that every state defines a quotient language.

In order to describe the PRFA characterization theorem, we define a number of additional concepts. Let $L$ be a stochastic language on $\Sigma$ and let $U$ be a finite subset of $\Sigma^*$. The set of *linearly generated residual languages* of $L$ associated with $U$ is:

$$LG_L(U) := \left\{ l \in SL(\Sigma) : l = \sum_{u \in U} \lambda_u \cdot u^{-1} L \right\},$$

where $SL(\Sigma)$ is the set of all stochastic languages on $\Sigma$ and $\{\lambda_u\}_{u \in U}$ is a set of convex coefficients (i.e. non-negative, sum-to-one). The set $U$ is a *finite residual generator of $L$* if every quotient language of $L$ belongs to $LG_L(U)$. The class $L_{frg}(\Sigma)$ is defined as the class of stochastic languages on $\Sigma$ having a finite residual generator. The PRFA characterization result [Esposito et al., 2002, Theorem 4] states that the class of all stochastic languages induced by PRFA is the class of languages having finite residual generators:

[2] In their paper, the term *residual language* was used for a quotient language, shedding light on the naming of the PRFA.

**Theorem 3.6**

$$L_{frg}(\Sigma) = L_{PRFA}(\Sigma).$$

## 3.2 A PFA Extension to the Myhill-Nerode Theorem

In this section we formulate and prove an extension of the classical Myhill-Nerode theorem to PFA models. We begin by formally defining the notions of a *suffix distribution*, $D_{[z]}$ (the distributional analogue to the quotient language $z^{-1}L$, defined mainly for enhancing the clarity of exposition) and the *suffix set*:

**Definition 3.7** *Given a distribution $D$ over $\Sigma^*$ and a string $z \in \Sigma^*$ such that $D(z) > 0$, the suffix distribution $D_{[z]}$ is defined by:*

$$D_{[z]}(x) := \frac{D(zx)}{D(z)} \qquad \forall x \in \Sigma^*. \tag{3.1}$$

*If $D(z) = 0$ then $D_{[z]}(x)$ is undefined.*

**Definition 3.8** *The set of all suffix distributions of a given distribution $D$ is denoted by suff($D$):*

$$\textit{suff}(D) := \left\{D_{[z]}\right\}_{z\in\Sigma^*}.$$

Note that $D \in \textit{suff}(D)$.

We are now in a position to present the following *characterization* theorem for PFA distributions:

**Theorem 3.9 (Myhill-Nerode Extension for PFA Models)** *The following two statements are equivalent:*

1. *A distribution $D$ over $\Sigma^*$ can be induced by an $n$-state PFA.*

2. *There exist $n$ fixed distributions $\{D_1, \ldots, D_n\}$ over $\Sigma^*$ such that:*

$$\textit{suff(D)} \subseteq \textit{co}(D_1, \ldots, D_n).$$

**Proof** **1 $\Rightarrow$ 2:** Assuming the distribution $D$ is induced by some PFA $\mathcal{A} = \langle Q_\mathcal{A}, \Sigma, \delta_\mathcal{A}, I_\mathcal{A}, F_\mathcal{A}, P_\mathcal{A}\rangle$ with $|Q_\mathcal{A}| \le n$, we show that the set of its suffix distributions is contained in the convex hull of (at most) $n$ fixed distributions. For simplicity we assume $|Q_\mathcal{A}| = n$. In order to clarify the proof, we will use $s_k$ to denote the $k^{\text{th}}$ state in the context of it being used as an *initial state*, and the notation $q_k$ to denote the $k^{\text{th}}$ state in other contexts.

Let $z \in \Sigma^*$ be a prefix, $x \in \Sigma^*$ a suffix, and let $D_k,\ k = 1, \dots, n$, denote the distribution induced by $\mathcal{A}_{q_k}$. In analogy to Equation (1.5), the forward density for $\mathcal{A}_{q_k}$ is given by:

$$\alpha_x^k(i,q) = I_{\mathcal{A}}(s_k) \cdot \sum_{(s_1,\dots,s_i)\in\Theta_{\mathcal{A}}(x_1\dots x_i)} \prod_{j=1}^{i} P_{\mathcal{A}}(s_{j-1}, x_j, s_j) \cdot \mathbf{1}(q, s_i),$$

or in other words, $\alpha_x^k(i,q)$ denotes the probability of generating the prefix $x_1 \dots x_i$ and reaching state $q$ after *starting* from state $s_k$. As there exist only $n$ possible states to reach, we have the following equalities:

$$D_k(z) = \sum_{\ell=1}^{n} \alpha_z^k(|z|, q_\ell)\ ; \qquad D_k(zx) = \sum_{\ell=1}^{n} \alpha_z^k(|z|, q_\ell) D_\ell(x), \tag{3.2}$$

where $D_k(z)$ denotes the probability of generating a string $z \in \Sigma^*$ conditioned on the fact that we started from state $s_k$. Since the probability of starting in the state $s_k$ is given by $I_{\mathcal{A}}(s_k)$, the overall probability of generating $x$ becomes $D(x) = \sum_{k=1}^{n} I_{\mathcal{A}}(s_k) D_k(x)$. Using Definition 3.7 and plugging in (3.2), we get:

$$\begin{aligned} D_{[z]}(x) &= \frac{D(zx)}{D(z)} = \frac{\sum_{k=1}^{n} I_{\mathcal{A}}(s_k) D_k(zx)}{\sum_{k=1}^{n} I_{\mathcal{A}}(s_k) D_k(z)} \\ &= \frac{\sum_{k=1}^{n} I_{\mathcal{A}}(s_k) \cdot \sum_{\ell=1}^{n} \alpha_z^k(|z|, q_\ell) D_\ell(x)}{\sum_{k=1}^{n} I_{\mathcal{A}}(s_k) \cdot \sum_{\ell'=1}^{n} \alpha_z^k(|z|, q_{\ell'})} \\ &= \sum_{\ell=1}^{n} \left[ \frac{\sum_{k=1}^{n} I_{\mathcal{A}}(s_k) \cdot \alpha_z^k(|z|, q_\ell)}{\sum_{k=1}^{n} I_{\mathcal{A}}(s_k) \cdot \sum_{\ell'=1}^{n} \alpha_z^k(|z|, q_{\ell'})} \right] D_\ell(x). \end{aligned} \tag{3.3}$$

The bracketed coefficients are nonnegative, sum to 1, and do not depend on $x$, so the resulting expression is a convex combination of the distributions $\{D_\ell(\cdot)\}_{\ell=1}^{n}$, proving the claim.

**2 ⇒ 1:** The distribution $D$ is in *suff*$(D)$ and therefore in *co*$(D_1, \dots, D_n)$ (the convex hull of $(D_1, \dots, D_n)$). Thus, there exist nonnegative, sum-to-one (*i.e.*, convex) coefficients $\{\pi_1, \dots, \pi_n\}$ such that $D = \sum_{i=1}^{n} \pi_i D_i$.

For every $\sigma \in \Sigma$ the distribution $(D_i)_{[\sigma]}$ is in *co*$(D_1, \dots, D_n)$, and hence can be written as $(D_i)_{[\sigma]}(x) = \sum_j \lambda_{\sigma,i}^j D_j(x)$ for some set of convex coefficients $\lambda_{\sigma,i}^j$. When generating a string $z = z_1 z_2 \dots z_k$, we first pick $D_i$ with probability $\pi_i$. We then pick $D_j$ with probability $\lambda_{z_1,i}^j$, output $z_1$, and proceed to generate $z_2 \dots z_k$. But since $(D_j)_{[z_2]}$ is itself in $co(D_1, \ldots, D_n)$, the process is recursively repeated. Written formally, we have:

$$\begin{aligned} D(z) &= \sum_{i=1}^{n} \pi_i D_i(z_1 z_2 \ldots z_k) = \sum_{i=1}^{n} \pi_i (D_i)_{[z_1]}(z_2 \ldots z_k) \\ &= \sum_{i=1}^{n} \pi_i \sum_{j_1=1}^{n} \lambda_{z_1,i}^{j_1} D_{j_1}(z_2 \ldots z_k) = \sum_{i=1}^{n} \pi_i \sum_{j_1=1}^{n} \lambda_{z_1,i}^{j_1} \sum_{j_2=1}^{n} \lambda_{z_2,j_1}^{j_2} D_{j_2}(z_3 \ldots z_k) \\ &= \ldots = \sum_{i=1}^{n} \pi_i \sum_{j_1=1}^{n} \lambda_{z_1,i}^{j_1} \sum_{j_2=1}^{n} \lambda_{z_2,j_1}^{j_2} \cdots \sum_{j_k=1}^{n} \lambda_{z_k,j_{(k-1)}}^{j_k} D_{j_k}(\epsilon). \end{aligned}$$

In order to construct an $n$-state PFA $\mathcal{A} = \langle Q_{\mathcal{A}}, \Sigma, \delta_{\mathcal{A}}, I_{\mathcal{A}}, F_{\mathcal{A}}, P_{\mathcal{A}} \rangle$ which induces the above distribution, we proceed as follows: identify each $D_i$ with a state $q_i$ of $\mathcal{A}$ and set $I_{\mathcal{A}} = (\pi_1, \ldots, \pi_n)$. The transition probabilities $P_{\mathcal{A}}(q_i, \sigma, q_j)$ are set to $\lambda_{\sigma,i}^{j}$, and the final state probabilities are set to $F_{\mathcal{A}}(q_i) = D_i(\epsilon)$. Using Equation (1.7) and writing out the recursion in (1.6), we have $\Pr_{\mathcal{A}}(z) = D(z)$ for any string $z \in \Sigma^*$. ■

Theorem 3.4 follows as a special case. In the first direction, the forward densities and initial state probabilities of (3.3) reduce to delta distributions, implying the suffix probabilities are in the set $\{D_1, \ldots, D_n\}$. In the second direction, the PDFA case has a delta initial distribution, setting the initial state. The parameters $\{\lambda_{\sigma,i}^{j}\}_{i,j=1}^{n}$ satisfy the PDFA constraints, and the constructed automaton $\mathcal{A}$ reduces to a PDFA.

### 3.2.1 Connections to PRFA

We now elaborate on the distinction between PFAs and PRFAs, in light of their respective characterization theorems. The example presented in [Esposito et al., 2002] and repeated in Figure 3.1 is instructive.

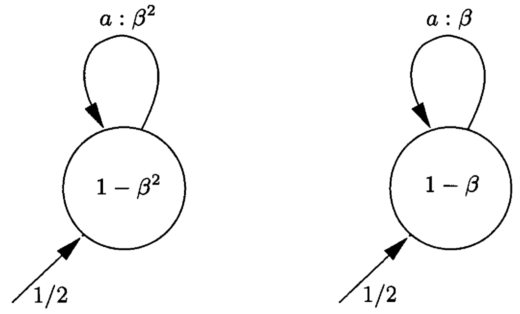

**Figure 3.1:** A simple PFA using the *single-letter* alphabet $\Sigma = \{a\}$ and inducing a distribution which cannot be generated by a PRFA. The initial state probabilities are 1/2 for each state, and the termination probabilities are denoted within the states.

In their paper, the authors showed that the distribution induced by the depicted PFA cannot be induced by a PRFA. For this example, the probabilities of termination after emitting the string $a^\ell$ were shown to be:

$$\Pr\left(\epsilon|(a^\ell)^{-1}L\right) = 1 - \beta^2 - \frac{\beta - \beta^2}{\beta^\ell + 1}.$$

Assuming $\beta < 1$, the expression tends to $1 - \beta$ strictly monotonically as $\ell \to \infty$. This implies that for a given value of $\ell$, the probability $\Pr\left(\epsilon|(a^{\ell+1})^{-1}L\right)$ cannot be expressed as a convex combination of the preceding probabilities $\left\{\Pr\left(\epsilon|(a^i)^{-1}L\right)\right\}_{i=1}^{\ell}$. In turn, this implies the same for the suffix distributions, implying further that the distribution cannot be modelled by a PRFA.

A PFA, in contrast, can model the above suffix distributions by a convex combination of the two fixed distributions $D_1$ and $D_2$, defined in this case by:

$$D_1(a^\ell) = \beta^{2\ell}(1 - \beta^2); \qquad D_2(a^\ell) = \beta^\ell(1 - \beta).$$

## 3.3 Applications of the PDFA and PFA Myhill-Nerode Extension Theorems

In this section we give two examples where applications of the Myhill-Nerode extension theorems provide immediate proofs that certain stochastic languages are either not regular or not deterministic regular. The first example was discussed in [Vidal et al., 2005a] (and mentioned in Section 1.7), while the second is novel.

### 3.3.1 A Stochastic Non-Deterministic Regular Language

We recall the example presented in Section 1.7 (page 15), which we repeat below in Figure 3.2.

We now provide an alternative proof of the fact that no PDFA can induce an identical distribution:

**Proposition 3.10** *No PDFA can generate the stochastic language induced by the PFA of Figure 3.2.*

**Proof** Define the stochastic language $L$ to be that induced by the PFA of Figure 3.2. For the string $x = a^k$, the quotient $x^{-1}L$ is calculated as follows:

$$\Pr\left(w|(a^k)^{-1}L\right) = \frac{\Pr(a^k w|L)}{\sum_{z\in\Sigma^*}\Pr(a^k z|L)} = \frac{\frac{1}{2}\left[\left(\frac{1}{2}\right)^{k-1} D^{q_1}(w) + \left(\frac{1}{3}\right)^{k-1} D^{q_2}(w)\right]}{\frac{1}{2}\left[\left(\frac{1}{2}\right)^{k-1} + \left(\frac{1}{3}\right)^{k-1}\right]}. \qquad (3.4)$$

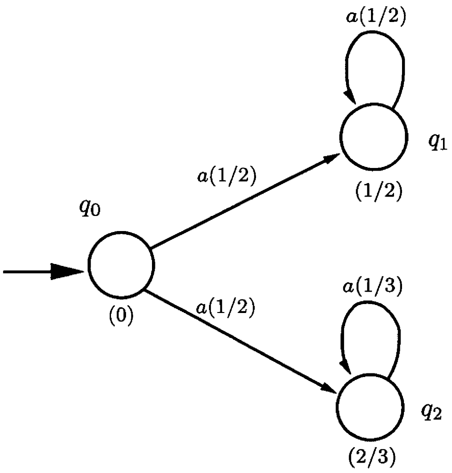

**Figure 3.2:** A simple PFA using the *single-letter* alphabet $\Sigma = \{a\}$ and inducing a distribution which cannot be generated by a PDFA.

For the specific string $w = a$, we have $D_1(w) = 1/2 \cdot 1/2 = 1/4$, while $D_2(w) = 1/3 \cdot 2/3 = 2/9$. Plugging this specific case into (3.4), we get:

$$\Pr\left(a|(a^k)^{-1}L\right) = \frac{\frac{1}{4}\cdot\left(\frac{1}{2}\right)^{k-1} + \frac{2}{9}\cdot\left(\frac{1}{3}\right)^{k-1}}{\left(\frac{1}{2}\right)^{k-1} + \left(\frac{1}{3}\right)^{k-1}}. \tag{3.5}$$

Expression (3.5) assumes different values for different values of $k$, implying the number of quotient languages is not finite, which by Theorem 3.4 (the PDFA extension to the Myhill-Nerode theorem) implies the distribution cannot be induced by a PDFA. ■

### 3.3.2 A Stochastic Non-Regular Language

We now construct a distribution over $\{0,1\}^*$ that cannot be induced by any PFA.

**Proposition 3.11** *The distribution specified by:*

$$\Pr(x) = \begin{cases} 2^{-n} & x = 0^n 1^n, \ n \geq 1, \\ 0 & \textit{otherwise} \end{cases}$$

*cannot be induced by any PFA model.*

**Proof** Corresponding to the set of prefixes $\{0^k 1\}_{k=1}^{\infty}$, we obtain the following set of quotient languages: $\{\delta(1^{k-1})\}_{k=1}^{\infty}$ (with $\delta(\cdot)$ denoting the delta function). For any finite $n > 0$, the set $\{\delta(1^{k-1})\}_{k=1}^{\infty}$ cannot be contained in the convex hull of any $n$ distributions. Indeed, for any set of distributions $\{D_1, \ldots, D_n\}$ over $\Sigma^*$, there exists an $\ell \leq n+1$ for which:

$$\max_{D_i \in \{D_1, \dots D_n\}} D_i(1^\ell) < 1,$$

implying $\delta(1^\ell) \notin \text{co}(D_1, \dots D_n)$. ■

## 3.4 Approximation of Distributions over Bounded Length Strings

Little is known about the relative power of PFA and PDFA models in approximating arbitrary probability distributions. In this section we provide bounds which answer the following question: how well can ($n$-state) PFA and PDFA models approximate arbitrary distributions over *bounded length* strings. We also show that the task of approximating arbitrary distributions over strings with a given *expected length* is (in some strong sense) hard.

### 3.4.1 Upper Bound on PDFA Approximation of Bounded Length Distributions

We begin by presenting an immediate upper bound on the number of PDFA states required to represent a bounded length distribution:

**Lemma 3.12** *Let $D$ be a distribution with length bounded by $L$ (i.e. $w \sim D$ implies $|w| \leq L$). Then a PDFA $\mathcal{A}$ with $|\Sigma|^{L+1} - 1$ states can be constructed which induces the distribution $D$.*

**Proof** We construct a $|\Sigma|$-ary tree of depth (at most) $L+1$ and set $\mathcal{A}$'s initial state $q_0$ to be the tree's root. For $q_i$ denoting one of the tree's internal nodes, we let $w(q_i)$ denote the sequence of letters that had been traversed while reaching $q_i$ from $q_0$. For each alphabet letter $\sigma \in \Sigma$ and internal node $q_i$ we label one outgoing edge from $q_i$ with $\sigma$, thus defining the state transition function $\delta_{\mathcal{A}}$. For a pair of nodes $(q_i, q_j)$ such that $(q_i, \sigma, q_j) \in \delta_{\mathcal{A}}$, we set the transition probability $P_{\mathcal{A}}(q_i, \sigma, q_j)$ to:

$$P_{\mathcal{A}}(q_i, \sigma, q_j) = \frac{\sum_{x \in \Sigma^*} D(w(q_j)x)}{\sum_{x \in \Sigma^*} D(w(q_i)x)}$$

if $\sum_{x \in \Sigma^*} D(w(q_i)x) > 0$ and remove the transition otherwise. The final state probability for state $q_i$ is set to:

$$F_{\mathcal{A}}(q_i) = \frac{D(w(q_i))}{\sum_{x \in \Sigma^+} D(w(q_i)x)}, \tag{3.6}$$

where $\Sigma^+ = \Sigma^* \setminus \{\epsilon\}$.

We now show that the PDFA $\mathcal{A}$ thus constructed induces the distribution $D$. For a string $z = z_1 \dots z_\ell$ with $\ell \leq L$, there exists a unique state $q(z) \in Q_\mathcal{A}$ such that $w(q(z)) = z$. Calculating the probability of the (unique) path in $\mathcal{A}$ from $q_0$ to $q(z)$, we get:

$$\begin{aligned}
&\prod_{i=1}^{\ell} P_\mathcal{A}\left(q(z_1 \dots z_{i-1}), z_i, q(z_1 \dots z_i)\right) \\
=\; & \frac{\sum_{x\in\Sigma^*} D(z_1 x)}{\sum_{x\in\Sigma^*} D(x)} \cdot \frac{\sum_{x\in\Sigma^*} D(z_1 z_2 x)}{\sum_{x\in\Sigma^*} D(z_1 x)} \cdots \frac{\sum_{x\in\Sigma^*} D(z_1 z_2 \dots z_\ell x)}{\sum_{x\in\Sigma^*} D(z_1 z_2 \dots z_{\ell-1} x)} \\
=\; & \sum_{x\in\Sigma^*} D(z_1 z_2 \dots z_\ell x) = \sum_{x\in\Sigma^*} D(zx),
\end{aligned}$$

where the first term in the product (i.e. $q(z_1 \dots z_0)$) is taken to mean $q_0$. The final state probability for $q(z)$ is (via (3.6)):

$$F_\mathcal{A}(q(z)) = \frac{D(z_1 z_2 \dots z_\ell)}{\sum_{x\in\Sigma^*} D(z_1 z_2 \dots z_\ell x)} = \frac{D(z)}{\sum_{x\in\Sigma^*} D(zx)}.$$

Pulling the calculated expressions together, we obtain:

$$\Pr_\mathcal{A}(z) = \prod_{i=1}^{\ell} P_\mathcal{A}\left(q(z_1 \dots z_{i-1}), z_i, q(z_1 \dots z_i)\right) \cdot F_\mathcal{A}(q(z)) = D(z).$$

■

### 3.4.2 Lower Bounds on PDFA and PFA Approximations of Bounded Length Distributions

We will now present lower bounds for approximation of bounded length distributions by PDFA / PFA models. The techniques used for obtaining the lower bounds are illustrated in Figure 3.3. In both cases, the target distribution is composed of a uniform prefix distribution followed by a "symmetry-breaking" set of suffixes.

In the case of the PFA, we require each suffix to be be poorly approximable by the convex hull of all other suffixes, so all the chosen suffixes are distinct delta distributions. Thus, the $L_1$-distance between each suffix distribution and the convex hull of all other suffix distributions equals 2. In the PDFA case, we require a weaker condition, namely that the $L_1$-distance between each suffix pair is at least 1/2. Therefore, (exponentially) shorter suffix lengths suffice.

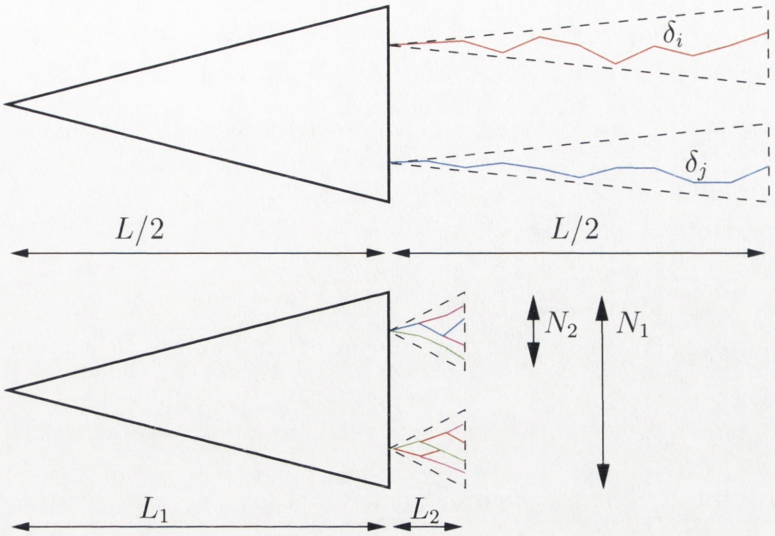

**Figure 3.3:** An illustration of the techniques used for proving lower bounds on the approximation ability of PFA (top) and PDFA (bottom) models. The triangles on the left illustrate uniform distributions, while the suffix distributions on the right hand side serve as "symmetry-breakers" (with $\delta_i, \delta_j$ denoting two different delta distributions).

### Lower bound for PFA Approximation of Bounded Length Distributions

In this section we apply the Myhill-Nerode theorem for PFA to derive a lower bound on the models' approximation ability.

**Theorem 3.13** *There exists a distribution $D^*$ of bounded length $L$ such that for any distribution $\widehat{D}$ induced from a PFA with no more than $|\Sigma|^{\left(\frac{L}{2}-1\right)}$ states, $\|D^* - \widehat{D}\|_1 \geq 1/2$.*

Before proving the theorem, we define the notion of *suffix mass*:

**Definition 3.14** *The* suffix mass *of a family of probability distributions $\mathcal{D}$ is defined by:*

$$\mathrm{SM}(\mathcal{D}) := \max_{D \in \mathcal{D}} \sum_{w \in \Sigma^*} \max_{S \in \mathrm{suff(D)}} S(w).$$

For the family of distributions induced by $n$-state PFA we now show:

**Lemma 3.15** *Let $\mathcal{D}_n = \{D : D$ is induced by an $n$-state PFA $\}$. Then $SM(\mathcal{D}_n) \leq n$.*

**Proof** Denote the $n$-dimensional probability simplex by $\Delta^n = \{\alpha \in \mathbb{R}^n : \alpha_i \geq 0,\ i = 1, \ldots, n,\ \sum_{i=1}^n \alpha_i = 1\}$. By the Myhill-Nerode extension for PFA (Theorem 3.9), all suffix distributions $S \in$ suff(D) in Definition 3.14 reside in the convex hull of at most $n$ distributions, which we denote by $\{D_1, \ldots, D_n\}$. We thus have:

$$\sum_{w \in \Sigma^*} \max_{S \in \mathrm{suff(D)}} S(w) = \sum_{w \in \Sigma^*} \max_{\alpha \in \Delta^n} \sum_{i=1}^{n} \alpha_i D_i(w).$$

However, for each $w \in \Sigma^*$ there exists an index $i(w) \in \{1,\dots,n\}$ such that $D_{i(w)}(w) = \max_{\alpha\in\Delta^n}\sum_{i=1}^n \alpha_i D_i(w)$. Observing that $\{i(w) : w \in \Sigma^*\} \subseteq \{1,\dots,n\}$, we have:

$$\sum_{w\in\Sigma^*} \max_{\alpha\in\Delta^n} \sum_{i=1}^{n} \alpha_i D_i(w) = \sum_{w\in\Sigma^*} D_{i(w)}(w) \leq \sum_{i=1}^{n} 1 = n,$$

and the proof immediately follows. ■

We will also require the following technical lemma, for which a proof is supplied in Appendix A:

**Lemma 3.16** *Let $D$ denote a distribution over $\{1,\dots,N\}$ with individual probabilities $(d_1,\dots,d_N)$, such that $d_i \geq 0$ and $\sum_{i=1}^N d_i = 1$. Let $T = (t_1,\dots,t_N)$ be some sequence such that $0 \leq t_i \leq 1$. Then the following inequality holds:*

$$\sum_{i=1}^{N} \left|\frac{1}{N} - d_i t_i\right| \geq 1 - \frac{1}{N}\sum_{i=1}^{N} t_i.$$

We are now in a position to prove the negative result mentioned earlier:

**Proof (Theorem 3.13)**

Define the target distribution $D^*$ as follows:

$$D^*(w) := \begin{cases} |\Sigma|^{-L/2} & w = w'w', \quad w' \in \Sigma^{L/2} \\ 0 & \text{otherwise.} \end{cases}$$

Let $\mathcal{D}$ be some family of distributions with $\mathrm{SM}(\mathcal{D}) \leq |\Sigma|^{(\frac{L}{2}-1)}$, let $\widehat{D} \in \mathcal{D}$ be an approximating distribution and enumerate the set $\Sigma^{L/2}$ as $w_1,\dots,w_N$ with $N = |\Sigma|^{L/2}$. We proceed to lower-bound the $L_1$-distance:

$$\begin{aligned} \|D^* - \widehat{D}\|_1 &= \sum_{w\in\Sigma^*} |D^*(w) - \widehat{D}(w)| \geq \sum_{w\in\Sigma^L} |D^*(w) - \widehat{D}(w)| \\ &= \sum_{i=1}^{N} |D^*(w_i w_i) - \widehat{D}(w_i w_i)| + \sum_{i=1}^{N} \sum_{\substack{w_j\in\Sigma^L \\ w_j\neq w_i}} |D^*(w_i w_j) - \widehat{D}(w_i w_j)| \\ &\geq \sum_{i=1}^{N} \left|\frac{1}{N} - \widehat{D}(w_i w_i)\right|. \end{aligned}$$

Plugging Lemma 3.15 above into Lemma 3.16, we have shown $\|D^* - \widehat{D}\|_1 \geq 1/2$. ■

The preceding analysis is loose in the sense that (asymptotically) a square factor in the number of states separates the upper and the lower bounds of Lemma 3.12 and Theorem 3.13 respectively. A tighter analysis for PDFA models follows.

### Lower bound for PDFA Approximation of Bounded Length Distributions

We now present a sharper lower bound when the approximating class is restricted to $n$-state PDFA models. We assume for simplicity that the alphabet's cardinality is 2, and accordingly all logarithms used below are to base 2. Before presenting the bound we state the following lemma regarding the packing number on the probability simplex:

**Lemma 3.17** *Let $\Delta^d$ denote the d-dimensional probability simplex. Then the number of distributions on $\Delta^d$ which are at least $\varepsilon$-separated in the $L_1$-norm (denoted $M(\varepsilon, \Delta^d, \|\cdot\|_1)$) is lower-bounded by:*

$$M(\varepsilon, \Delta^d, \|\cdot\|_1) \geq \left(\frac{1}{2\varepsilon}\right)^d .$$

The proof of Lemma 3.17 is provided in Appendix A.

**Theorem 3.18** *Suppose $\Sigma = \{0,1\}$. For any positive integer $L$, there exists a distribution $D^*$ over $\Sigma^{L+\log L}$ such that for any distribution $\widehat{D}$ induced by a PDFA with no more than $2^L$ states, $\|D^* - \widehat{D}\|_1 \geq 1/16$.*

**Proof** We set all $L$-length prefixes of $D^*$ to be equiprobable (i.e. of probability $2^{-L}$ each) and let $N_1 = 2^L$. For $i \in \{1, \dots, N_1\}$, we denote the target suffix distribution following prefix $i$ by $D_i^*$. For the same prefix, we denote the approximating distribution's prefix probability by $\widehat{d_i}$ and its corresponding suffix distribution by $\widehat{D}_i$.

We construct $D^*$ such that each suffix distribution $D_i^*$ is composed solely of strings of length $L_2$. We wish to construct a set of suffix distributions of cardinality $N_1$, such that each pair is at least (1/4)-separated (i.e. $\|D_{i_1}^* - D_{i_2}^*\|_1 \geq 1/4$, $i_1 \neq i_2$). Denoting $N_2 = 2^{L_2}$ (the number of possible length-$L_2$ strings), we appeal to Lemma 3.17. In this case, the dimensionality $d$ of the simplex is $N_2$. Setting $L_2 = \log L$ and using Lemma 3.17, we see that the number of possible distributions conforming to the demands is lower-bounded by $N_1$, as desired.

We now proceed to lower-bound $\|D^* - \widehat{D}\|_1$. Given that the approximating PDFA has (at most) $N_1/2$ states, there exist (at least) $N_1/2$ "coupled" pairs $(i_1, i_2)$ such that $\widehat{D}_{i_1} = \widehat{D}_{i_2}$. Writing out $\|D^* - \widehat{D}\|_1$, we get:

$$\begin{aligned}
\|D^* - \widehat{D}\|_1 &= \sum_{w\in\Sigma^*} |D^*(w) - \widehat{D}(w)| \geq \sum_{w\in\Sigma^{(L+L_2)}} |D^*(w) - \widehat{D}(w)| \\
&\geq \sum_{i=1}^{N_1} \left\| \frac{1}{N_1} D_i^* - \widehat{d}_i \widehat{D}_i \right\|_1 .
\end{aligned}$$

Using the standard norm inequality $\|x - y\| \geq \big|\|x\| - \|y\|\big|$, we find that each term in the sum is lower-bounded by $|\frac{1}{N_1} - \widehat{d}_i|$. For the coupled pair $(i_1, i_2)$, we therefore have:

$$\begin{aligned}
t &:= \left\| \frac{1}{N_1} D_{i_1}^* - \widehat{d}_{i_1} \widehat{D}_{i_1} \right\|_1 + \left\| \frac{1}{N_1} D_{i_2}^* - \widehat{d}_{i_2} \widehat{D}_{i_2} \right\|_1 \\
&\geq \left| \frac{1}{N_1} - \widehat{d}_{i_1} \right| + \left| \frac{1}{N_1} - \widehat{d}_{i_2} \right| \geq \left| \widehat{d}_{i_1} - \widehat{d}_{i_2} \right| .
\end{aligned}$$

However, as $\widehat{D}_{i_1} = \widehat{D}_{i_2}$, we also have that:

$$\begin{aligned}
t &= \left\| \frac{1}{N_1} D_{i_1}^* - \widehat{d}_{i_1} \widehat{D}_{i_1} \right\|_1 + \left\| \frac{1}{N_1} D_{i_2}^* - \widehat{d}_{i_2} \widehat{D}_{i_2} \right\|_1 \\
&\geq \left\| \frac{1}{N_1} \left( D_{i_1}^* - D_{i_2}^* \right) - \left( \widehat{d}_{i_1} - \widehat{d}_{i_2} \right) \widehat{D}_{i_1} \right\|_1 \\
&\geq \left\| \frac{1}{N_1} \left( D_{i_1}^* - D_{i_2}^* \right) \right\|_1 - \left| \widehat{d}_{i_1} - \widehat{d}_{i_2} \right| \\
&\geq \frac{1}{4N_1} - \left| \widehat{d}_{i_1} - \widehat{d}_{i_2} \right| .
\end{aligned}$$

Hence for all coupled pairs $(i_1, i_2)$, $t \geq \max(\frac{1}{4N_1} - b,\ b)$ where $b = |\hat{d}_{i_1} - \hat{d}_{i_2}|$. Since $1/(4N_1) - b > b$ for $b < 1/(8N_1)$, we have $t > 1/(8N_1)$ for any possible value of $b$. In other words, for all possible values of $(\widehat{d}_{i_1}, \widehat{d}_{i_2})$ we have:

$$\left\| \frac{1}{N_1} D_{i_1}^* - \widehat{d}_{i_1} \widehat{D}_{i_1} \right\|_1 + \left\| \frac{1}{N_1} D_{i_2}^* - \widehat{d}_{i_2} \widehat{D}_{i_2} \right\|_1 \geq \frac{1}{8N_1} .$$

Summing over (at least) $N_1/2$ coupled pairs, we obtain $\|D^* - \widehat{D}\|_1 \geq \frac{1}{16}$. ■

### 3.4.3 Approximation of Bounded Expected Length Distributions

A natural question in this context regards the PFA / PDFA models' ability to approximate the class of bounded *expected* length distributions. The difficulty of such PFA approximation is shown in the following lemma:

**Lemma 3.19** *Given any $\varepsilon > 0$ and $L \in \mathbb{N}$, there exists a distribution $D^{**}$ of expected length bounded by $L$ (i.e. $\mathbb{E}_{w \sim D^{**}} |w| \leq L$), such that for any PFA $\mathcal{A}$ with no more than*

$|\Sigma|^{\left(\frac{L-1}{8\varepsilon}-1\right)}$ *states, the following will hold:*

$$\|D^{**} - D_{\mathcal{A}}\|_1 \geq \varepsilon.$$

**Proof** The proof relies on the lower bound for PFA approximation presented in Section 3.4.2. Let the distribution $D^*$ be as defined in Section 3.4.2 with $L$ replaced by $L_0$. We define $D^{**}$ as follows:

$$D^{**}(w) := \begin{cases} 1 - 4\varepsilon & w = 0. \\ 4\varepsilon \cdot D^*(w') & w = 1w',\ \forall w' \in \Sigma^*. \end{cases}$$

It follows from the definition that $\mathbb{E}_{w \sim D^{**}} |w| = 1 - 4\varepsilon + 4\varepsilon(1 + L_0) = 1 + 4\varepsilon L_0$. Selecting $L_0 \leq \frac{L-1}{4\varepsilon}$ guarantees $\mathbb{E}_{w \sim D^{**}} |w| \leq L$. An $\varepsilon$-approximation of $D^{**}$ can only be achieved if $D^*$ is approximated to accuracy 1/2, which by Lemma 3.13 cannot be achieved using a PFA of less than $|\Sigma|^{\left(\frac{L_0}{2}-1\right)}$ states. Substituting $L_0 = \frac{L-1}{4\varepsilon}$ concludes the proof. ■

An analogous result for PDFA-based approximation can also be formulated, based on the lower bound shown in Section 3.4.2.

## 3.5 Discussion and Conclusion

The lower bounds presented in Section 3.4.2 are conceptually loose in the following sense: we made only *partial* use of the Myhill-Nerode extension theorems. Namely, when constructing the PFA lower bound, we only used Theorem 3.9 indirectly via Lemma 3.15. The lemma does not utilize the fact that all PFA suffix distributions must *recursively* and exclusively contain suffix distributions which also conform to the conditions of the theorem. A proof technique utilizing this additional information could potentially provide a tighter result. In the PDFA case, a similar criticism holds true, but the result obtained is tight to a logarithmic factor, leaving little room for improvement.

Our main goal for extending the research presented in this chapter is to attain a complete understanding of the relationship between distributions induced by PFA and PDFA families. Specifically, we seek to understand how well (and under which circumstances) PDFA models can approximate distributions induced by PFA models. This problem was addressed in [Zeitouni et al., 1992], where assuming a certain condition (a lower bound on *all* the approximated PFA's transition probabilities), an $L_\infty$-approximation result was shown. In this result, however, the number of states required grows exponentially with the (inverse of the) accuracy parameter; the approximation is in the (weak) $L_\infty$ sense, and the condition assumed may be unnecessarily strong. A more complete understanding would be theoretically desirable, and could potentially have practical implications.

Chapter 4

# Computational Complexity of PFA Learning

In this chapter we consider the computational complexity of PDFA learning. Given a finite multiset of samples drawn from a *target* distribution generated by a PDFA, under which conditions can we guarantee "learnability", and under which conditions is a learning algorithm likely to fail? Learnability of PFA and PDFA models has been widely investigated. Characterization of the family of PDFA which can be learned efficiently is a deep question for which only partial answers are known so far. There are indications that the general PDFA learning problem is hard. For instance, Kearns et al. [1994] showed that KL-PAC learnability of PDFA implies the computability of the *noisy parity function*, thus violating the *noisy parity assumption*, widely believed to be true in the cryptography community (see e.g. [Kearns, 1993]). We seek to understand the criteria and algorithms which enable efficient PDFA learning, and the results presented in this chapter form a step in that direction.

The chapter is composed of the following sections:

- In Sections 4.1 and 4.2 we discuss a number of learning frameworks relevant to our discussion, and mention key results regarding PFA / PDFA learnability within each framework.

- In Section 4.3 we present a thorough overview of existing PFA / PDFA learnability results, and introduce the notation used when necessary. We discuss both negative and positive existing results, set the stage for our novel results, and highlight the incomplete aspects of the current understanding.

- In Section 4.4 we present an extension to a central negative result, justifying our selection of learning framework in the subsequent sections.

- In Section 4.5 we discuss and generalize the most important class of PDFA learning algorithms, namely the state merging (SM) algorithms.

- In Section 4.6 we present a novel analysis and an accompanying algorithm which improves on (and asymptotically tightens) a recent result on testing $L_2$ proximity between distributions. Using this result and the generalized SM algorithm of Section 4.5, we extend the class of efficiently learnable PDFAs to the *$\mu_2$-distinguishable* family.

- In Section 4.7 we present a novel analysis of the SM algorithm, further extending the efficiently learnable PDFA family to the *$\rho$-distinguishable* subclass.

- In Section 4.8 we summarize the results presented in the chapter and discuss avenues for extending the research.

## 4.1 PFA Learning Models

Reflecting the extensive applicability of the models, numerous PFA "learning" algorithms have been proposed, alongside a host of learning frameworks. In this section we discuss the leading learning frameworks, the most commonly used algorithms within each framework, and present some relevant learnability results.

The learning frameworks discussed below all pertain to the *non-terminating* PFA model, and assume all sample strings have the same length $\ell$. This streamlines the presentation. However, all the algorithms discussed have general (variable-length) versions, and all hardness results carry over to the normal PFA setting trivially.

To enhance the clarity of our discussion, we will use the following notation for the probability of generating a string $s \in \Sigma^*$ induced by $\mathcal{A}$:

$$\mathcal{A}(s) = \mathrm{Pr}_{\mathcal{A}}(s).$$

### 4.1.1 Parameter Estimation

The simplest and most commonly used PFA learning framework is parameter estimation, typically in the maximum likelihood (ML) setting[1]. In this framework, the PFA's underlying graph structure has been preselected, and the models' transition parameters and initial distribution are to be inferred from a given data sample.

Formally, the ML PFA parameter estimation problem is posed as follows:

**Definition 4.1 (Maximum Likelihood PFA Parameter Estimation)** *Given a finite multiset $S = \{s_1, \ldots, s_m\}$ of strings $s_i \in \Sigma^\ell$, find an $n$-state PFA $\mathcal{A}^{opt}$ which, among the set of all $n$-state (non-terminating) PFA having a given underlying graph structure,*

[1] A number of Bayesian PFA parameter estimation results are discussed in [Ephraim and Merhav, 2002].

*assigns the maximum generation probability (i.e. maximum likelihood) to S:*

$$\prod_{i=1}^{m} \mathcal{A}^{opt}(s_i) \doteq \max\left\{\prod_{i=1}^{m} \mathcal{A}(s_i) : |Q_{\mathcal{A}}| = n\right\}. \tag{4.1}$$

Lacking precision ($\varepsilon$) and confidence ($\delta$) parameters, the definition above cannot directly be utilized to quantitatively study learning algorithms. To facilitate a quantitative discussion, the "approximate ML PFA parameter estimation" problem is defined, following [Abe and Warmuth, 1992]$^2$.

**Definition 4.2 (Approximate ML PFA Parameter Estimation)** *A randomized algorithm $T$ is said to approximate the ML PFA parameter estimation problem for the class of n-state PFA within factor $K$ (possibly a function of various parameters of the problem) in time t, if given a multiset $S = (s_1, \ldots, s_m)$, $s_i \in \Sigma^\ell$ for some $\ell > 0$, $T$ terminates in t steps and outputs an n-state PFA $\mathcal{A}$, which with probability at least* 1/2 *satisfies:*

$$\frac{\prod_{i=1}^{m} \mathcal{A}^{opt}(s_i)}{\prod_{i=1}^{m} \mathcal{A}(s_i)} \leq K,$$

*where $\mathcal{A}^{opt}$ is an n-state PFA satisfying* (4.1).

This framework, however, is still not a probably approximately correct (PAC) definition. The following PAC model has been shown in [Abe and Warmuth, 1992] to match Definition 4.2 (in the sense defined in Theorem 4.4 below):

**Definition 4.3 (Distribution Free PAC Computational Complexity)** *Let $S = \{s_1, \ldots, s_m\}$, (where $s_i \in \Sigma^\ell$), be a multiset and let $\mathcal{Q}$ over $\Sigma^\ell$ be a class of distributions. Let $|\,\mathcal{Q}\,|$ denote some measure of complexity associated with the class $\mathcal{Q}$.*

*A (possibly randomized) algorithm $T$* trains *$\mathcal{Q}$ if there exists some $M_0 = q(\varepsilon^{-1}, \delta^{-1}, \ell, |\,\mathcal{Q}\,|)$ such that for an arbitrary distribution $D$ over $\Sigma^*$, if $m \geq M_0$ then with probability at least $1 - \delta$, $T$ outputs a distribution $Q \in \mathcal{Q}$ such that either of the following criteria holds:*

$$KL(D \,\|\, Q) - KL(D \,\|\, Q^{opt}) \leq \varepsilon, \tag{4.2}$$

$$\|D - Q\|_1 - \left\|D - Q^{opt}\right\|_1 \leq \varepsilon, \tag{4.3}$$

*where $Q^{opt} \in \mathcal{Q}$ is a distribution satisfying (by the respective context):*

$$KL(D \,\|\, Q^{opt}) = \min\{KL(D \,\|\, P) : P \in \mathcal{Q}\},$$

$$\left\|D - Q^{opt}\right\|_1 = \min\{\|D - P\|_1 : P \in \mathcal{Q}\}.$$

[2] In that paper, the problem is called the "approximate sample MLM problem".

*When* (4.2) *is the chosen optimization criterion (and assuming* $\min\{KL(D \parallel P) : P \in \mathcal{Q}\}$ *exists), the runtime of* $T$ *defines the (distribution free)* KL computational complexity, *while for* (4.3) *it defines the* $L_1$ computational complexity. *If the runtime of* $T$ *is polynomial, the computational complexity is said to be polynomial.*

A polynomial computational complexity immediately implies the function $q(\cdot)$ (which will be referred to as the *sample complexity* in the sequel) is also polynomial in all its arguments.

In practice, aside from limiting the number of states, further constraints are usually placed on the optimization process. Typically, a set of transition probabilities are forced to zero, leaving the maximization process to deal with the complementary set (see definition of *PFA constraint* in Section 1.5).

The following result linking Definitions 4.2 and 4.3 was established in [Abe and Warmuth, 1992, Theorem 4.1]:

**Theorem 4.4** *Let* $t$ *denote the size of the input constraint to be trained,* $m$ *the sample size, and* $\ell$ *the length of each sample string. For an arbitrary PFA constraint* $C$*, the following three statements are equivalent:*

1. *There exists a training algorithm for* $C$ *with sample complexity polynomial in* $\varepsilon^{-1}, \delta^{-1}, t$ *and* $\ell$*, running in time polynomial in the total sample length.*
2. *There exists a training algorithm for* $C$ *with sample complexity polynomial in* $\varepsilon^{-1}, \log(\delta^{-1}), t$ *and* $\ell$*, running in time polynomial in the total sample length.*
3. *The ML parameter estimation problem (Definition 4.2) for* $C$ *is approximable within a factor* $1 + \varepsilon$*, with probability at least* $1/2$*, in time polynomial in* $\varepsilon^{-1}, t, \ell$ *and* $m$*.*

In most theoretical contributions within this framework, the ML problem discussed is usually the degenerate case where the input consists of a single string (i.e. $m = 1$). In most of the problems studied[3], the complexity associated with the single-string ML estimation problem equals the general problem's (i.e. multiple string).

We briefly discuss the key results in these frameworks.

### Parameter Estimation Algorithms

Two common algorithms for PFA parameter estimation are the (iterative) Baum algorithm (sometimes referred to as the Baum-Welch algorithm), which is a particular instance of the EM algorithm of Dempster et al. [1977], and the closely related Baum-Viterbi algorithm, essentially an approximation to the Baum algorithm. For a complete discussion of these

[3] With the exception of the result of Farago and Lugosi [1989] mentioned below.

algorithms, we refer the reader to [Ephraim and Merhav, 2002]. These algorithms provably converge to a local optimum, but convergence to the global optimum is not guaranteed.

A non-iterative, provably globally convergent maximum likelihood parameter estimation algorithm has been developed for the special case of a *left-to-right* PFA[4] [Farago and Lugosi, 1989]. This algorithm, however, is valid only when presented with a single sequence, limiting its practical applicability.

Abe and Warmuth [1992, Corollary 4.2] show that for any *deterministic* constraint, the computational complexity associated with learning is polynomial. This result, however, does not qualify as a PDFA learning algorithm, rather as a degenerate case of PFA parameter estimation (of limited practical value).

### Hardness Results

Abe and Warmuth [1992] proved that for the class of 2-state PFA, the computational complexity of Definitions 4.2 or 4.3 cannot be a polynomial in $|\Sigma|$, unless $\mathbf{RP} = \mathbf{NP}$[5].

Despite the hardness results mentioned, the parameter estimation framework (coupled with additional heuristics) lies at the heart of practical applications such as speech recognition and handwritten character recognition.

## 4.1.2 Structure Estimation

When a PFA's structure is not assumed to be known in advance, the models' structure has to be estimated as well as its parameters. The hardness result of Abe and Warmuth [1992] regarding PFA parameter estimation is immediately inherited by the (more general) structure estimation problem. Approaches for dealing with the difficulty inherent to structure estimation range from the use of heuristic algorithms (with either asymptotic or no performance guarantees) to the adoption of various restrictions on the learning model, as we discuss in the next sections.

We mention that the related problem of *order estimation*, i.e. the estimation of the number of states in the PFA, has received significant theoretical attention. A number of information-theoretic approaches (of limited practical applicability) are described in [Ephraim and Merhav, 2002, Section VIII].

[4] The class of left-to-right PFA consists of all acyclic PFA with self-loops on any (and possibly all) states. This class is particularly useful in speech recognition applications, where PFA (or HMMs) are used to model temporal behaviour.

[5] The conjecture that **RP** is strictly contained in **NP** is widely held in the theoretical computer science community.

## 4.2 PDFA Learning Frameworks

Although technically a subfield of PFA learning, PDFA learning is most often discussed separately, and consists of a largely separate toolbox of learning frameworks, theoretical results and algorithms. Most notably, in PDFA learning, the rule (rather than the exception) is that both structure and parameters must be inferred from the data.

### 4.2.1 Identification in the Limit With Probability One

In this early learning paradigm due to E. M. Gold [Gold, 1967], [Gold, 1978], there is an infinite source of examples that are generated following the distribution induced by the (hidden) target. The learning algorithm is expected to return after each new example some *hypothesis*. The class is said to be *identifiable in the limit with probability one* if for all targets within the class, the algorithm identifies the target (i.e. there is a point from which all hypotheses are equivalent to the target) with probability one. The paradigm bears some obvious drawbacks:

- it does not entail complexity constraints,
- one typically does not know if the amount of data required by the algorithm has already been supplied,
- an algorithm can be proven to identify in the limit, but might return arbitrarily bad answers if the required amount of data is not provided.

Despite these drawbacks, the identification in the limit paradigm can be seen as a necessary condition for the potential learnability of a given class of models. If this condition is not met, the target class is surely not learnable.

Identification in the limit with probability one of the PDFA structure was shown in [Carrasco and Oncina, 1999], and the proof was extended to include identification of the probabilities in [Higuera and Thollard, 2000].

### 4.2.2 PAC Frameworks for PDFA Learning

The following PAC framework was proposed in [Ron et al., 1995] and more recently used in [Clark and Thollard, 2004]:

**Definition 4.5 (KL-PAC Learning)** *Given a class of distributions $\mathcal{D}$ over $\Sigma^*$, an algorithm* KL-PAC *learns $\mathcal{D}$ if there is a polynomial $q(\cdot)$ such that for all $D \in \mathcal{D}$, $\varepsilon > 0$ and $\delta > 0$, the algorithm is given a sample multiset $S$ of size $m$ drawn from $D$, and produces a hypothesis $\widehat{D}$, such that* $\Pr\left[KL(D \parallel \widehat{D}) > \varepsilon\right] < \delta$ *whenever $m > q(1/\varepsilon, 1/\delta, |D|)$. By $|D|$ we denote some measure of the complexity of the target. The algorithm's running time is bounded by a polynomial in $m$ plus the total length of the strings in $S$.*

The definition of $L_p$-PAC learnability is analogous, with the corresponding change in distance measure. We mention that the novel results presented in this chapter all pertain to the $L_1$-PAC learning framework, for reasons which we will establish below.

In this learning framework, the samples are drawn from a target distribution which is *known* to be regular deterministic, and therefore the framework differs from the usual (distribution-free) PAC setting. Results obtained within this learning framework do not provide performance guarantees for situations wherein the samples drawn are from an *arbitrary* distribution.

## 4.3 Related Work

We now present related work in the field, focusing on central results which will set the stage for our novel contributions.

### 4.3.1 Negative Results for PDFA Learnability

In this section we discuss the main hardness result for PAC-learning of PDFA, due to Kearns et al. [1994]. In this result, a reduction is constructed showing that KL-PAC learnability of PDFA implies learnability of *noisy parity functions*, thus violating the *noisy parity assumption*, widely believed to be true in the cryptography community (see e.g. [Kearns, 1993]). We repeat the reduction here, and will extend it to the $L_1$-PAC learning model in Section 4.4.

The reduction is demonstrated by showing how by KL-PAC learning a specific family of (acyclic) PDFA, one can learn the class of noisy parity functions. Placing the concepts on formal ground, we now define the parity function:

**Definition 4.6 (Parity Function)** *Let $a \in \{0,1\}^n$ be a fixed but unknown binary vector. For any $x \in \{0,1\}^n$, define $f_a(x) = \sum_{i=1}^{n} a_i x_i (mod\, 2)$.*

**Definition 4.7 (Noisy Parity Problem)** *Let $\eta < 0.5$ denote a "bit flipping" noise parameter, and let $\widetilde{f_a^\eta}(x)$ denote a random variable which with probability $1-\eta$ equals $f_a(x)$, and with probability $\eta$ equals $\neg f_a(x)$. Given a set of examples $(x, \widetilde{f_a^\eta}(x))$, drawn uniformly at random, the* noisy parity problem *is to infer the vector a within the PAC framework. It is widely believed that the noisy parity problem is hard (see e.g. [Kearns, 1993]).*

A *noisy parity PDFA* (see example in Figure 4.1) can be used to generate a noisy parity function. It consists of $n$ layers, each of which encodes one bit of the vector $a$. The PDFA comprises of two parallel, upper and lower *tracks*. A layer where all transitions remain on the same track encodes a "0" bit, while a *crossover* layer (for which the "1" transition crosses tracks) encodes a "1". The last layer of the PDFA introduces *noise*, parameterized by $\eta < 0.5$.

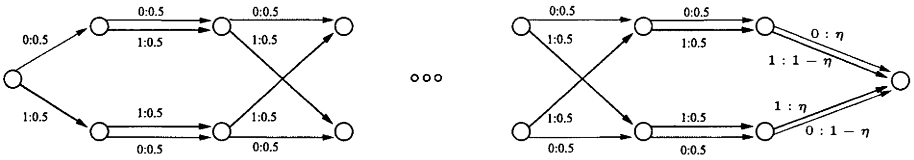

**Figure 4.1:** Noisy parity PDFA family. In the specific example above, the first layer encodes a "0", the second a "1", and so forth, encoding the binary vector 01…10. The last layer introduces noise via the parameter $\eta < 0.5$.

The distribution (over $\{0,1\}^n$) induced by such a PDFA will be denoted $D_a$. Given $D_a$ represented as a PDFA and a vector $x \in \{0,1\}^{n-1}$, we can compare between $D_a(x0)$ and $D_a(x1)$ and subsequently determine $f_a(x)$.

Kearns et al. [1994] show that if one could efficiently KL-PAC learn the noisy parity PDFA, one could then compute $f_a(x)$ with arbitrary precision, violating the hardness assumption.

### 4.3.2 Positive Results for PDFA Learnability

In this section we discuss positive PAC-framework PDFA learnability results, due to Ron et al. [1995], Clark and Thollard [2004] and Palmer and Goldberg [2005].

As shown in Kearns et al. [1994], the general PDFA PAC-learning problem is hard, and therefore additional conditions must be imposed before attempting to provide an efficient PAC learning algorithm. The concept of $\mu$-distinguishability was first introduced by Ron et al. [1995], where it was shown to be sufficient for KL-PAC learnability of *acyclic* PDFA. Namely, (a variant of) the state merging algorithm (defined below) was shown to require sample and computational complexities polynomial in $\mu^{-1}$ (in addition to $\varepsilon^{-1}, \delta^{-1}, n$ and $|\Sigma|$), where $\mu$ is defined as follows[6]:

**Definition 4.8 ($\mu$-distinguishability)** *Let $\mu > 0$. Given a PDFA $\mathcal{A} = \langle Q_{\mathcal{A}}, \Sigma, \delta_{\mathcal{A}}, q_0, F_{\mathcal{A}}, P_{\mathcal{A}} \rangle$, the state pair $(q_i, q_j) \in Q_{\mathcal{A}} \times Q_{\mathcal{A}}$ is said to be* $\mu$-distinguishable *if*

$$\left\| D_{\mathcal{A}}^{q_i} - D_{\mathcal{A}}^{q_j} \right\|_{\infty} > \mu.$$

*A PDFA $\mathcal{A}$ is $\mu$-distinguishable if each of its state pairs is $\mu$-distinguishable.*

[6] We use the notation $D_{\mathcal{A}}^{q_i}$ to denote the distribution induced by $\mathcal{A}_{q_i}$.

Clark and Thollard [2004] extended the result to general PDFA learning, while imposing an additional condition, namely an upper bound on the expected string length from *all* states. The theorem's formal statement is:

**Theorem 4.9 (Clark, Thollard)** *Suppose $\mathcal{A}$ is an $n$-state $\mu$-distinguishable PDFA such that the expected length of strings generated from every state is upper-bounded by $L$. Then for every $\delta > 0$ and $\varepsilon > 0$, Algorithm 4.1 outputs a hypothesis PDFA $\widehat{\mathcal{A}}$ such that with probability greater than $1 - \delta$, $\mathrm{KL}(\mathrm{D}_{\mathcal{A}} \parallel \mathrm{D}_{\widehat{\mathcal{A}}}) < \varepsilon$.*

*The sample and computational complexities of Algorithm 4.1 are polynomial in $\{\mu^{-1}, \varepsilon^{-1}, \delta^{-1}, |\Sigma|, n, L\}$.*

We relegate a number of definitions of concepts used in Algorithm 4.1 to Section 4.5, where a more general algorithm is discussed in detail.

**Algorithm 4.1**: State Merging

```
Input: $\varepsilon, \delta$ (accuracy, confidence parameters), $\mu$ (lower bound on the target's
       distinguishability), $n_{\max}$ (upper bound on the number of states of the
       target) and $L$ (upper bound on the expected length of strings generated
       from any state of the target). The algorithm is also supplied with a random
       source of strings generated independently by $\mathcal{A}$, the target PDFA.
Output: $\widehat{\mathcal{A}}$, a hypothesis PDFA such that $\mathrm{KL}(\mathrm{D}_{\mathcal{A}} \parallel \mathrm{D}_{\widehat{\mathcal{A}}}) < \varepsilon$ with probability at
        least $1 - \delta$.
Compute $m_0$, a threshold on the size of a multiset required for statistical testing;
$M$, the size of the sample we draw at each step of the algorithm, and $p_{\min}$, a small
smoothing constant.
repeat
    Draw $M$ strings from $\mathcal{A}$
    foreach $u \in V$ and $\sigma \in \Sigma$ such that $\delta_{\widehat{\mathcal{A}}}(u, \sigma)$ is undefined do
        Compute $S_{u,\sigma}$ (suffix multiset of $(u, \sigma)$)
        if $|S_{u,\sigma}| \geq m_0$ then  foreach $v \in V$ do
            if $\left\| \widehat{S}_{u,\sigma} - \widehat{S}_v \right\|_\infty < \mu/2$ then
                Add arc labeled with $\sigma$ from $u$ to $v$.
            end
        end
        else if $\left\| \widehat{S}_{u,\sigma} - \widehat{S}_v \right\|_\infty \geq \mu/2 \quad \forall v \in V$ then
            Create new node in graph $G$
            Add an edge labeled with $\sigma$ from $u$ to the new node
        end
    end
until no candidate node has a suffix multiset of cardinality (at least) $m_0$.
Complete $G$ by adding a ground node which represents low frequency states.
Add a final state $\widehat{q}_f$ and transitions labeled with $\zeta$ from each state to $\widehat{q}_f$.
```

In Section 4.5 we will discuss aspects of the theorem's proof in detail and generalize the result. Specifically, we will relax the $\mu$-distinguishability condition and extend the result to a strictly larger PDFA class called $\mu_2$-distinguishable. In Section 4.7 we will further extend the class of SM-learnable PDFA to the so-called *ρ-distinguishable* class.

Clark and Thollard [2004] gave a counterexample showing that when the upper bound ($L$) on the expected string length from each state is lifted, KL-PAC learnability cannot be guaranteed. Moreover, their counterexample specifically shows that a bound on the PDFA's *overall* expected string length cannot guarantee KL-PAC learnability.

In contrast, Palmer and Goldberg [2005] showed that in the (weaker) $L_1$-PAC learning framework, the upper bound $L$ can be lifted altogether. Their result uses Algorithm 4.1, with the modification that the graph completion and final state addition (lines 17 and 18 respectively) are no longer required.

## 4.4 An Extended Negative PDFA Learning Result

We presently show that even in the *weaker* $L_1$-PAC learning model, general PDFA learnability still violates the noisy parity assumption. This implies that the difficulty is inherent to PDFA learning, and not merely an artifact of the KL-divergence. In the following we use the notation of Murphy [1996]. We repeat the noisy parity PDFA graphic for convenience in Figure 4.2:

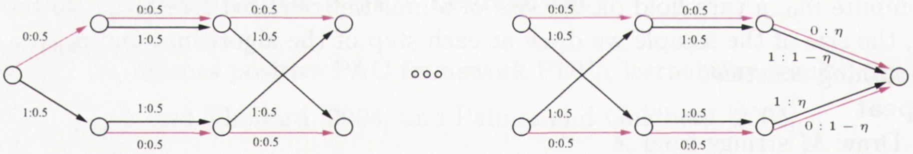

**Figure 4.2:** Noisy parity PDFA family. In the specific example above, the first layer encodes a "0", the second a "1", and so forth. The last layer introduces noise via the parameter $\eta < 0.5$.

**Theorem 4.10** *$L_1$-PAC learnability of the noisy parity PDFA family violates the noisy parity assumption.*

**Proof** As before, let $D_a$ denote the distribution induced by the noisy parity PDFA. To prove the theorem it is enough to show that given an $L_1$-approximation to $D_a$, we can determine $f_a(x)$ to arbitrary precision.

Let $\widehat{D}$ be the estimated distribution, such that $\|D_a - \widehat{D}\|_1 < \varepsilon$. Given the specific architecture of noisy parity PDFAs, all *correct parity* strings $xp$ such that $p = f_a(x)$ are assigned probability $\eta \cdot 2^{-n}$, while all *incorrect parity* strings $xn$ (such that $n = \neg f_a(x)$) are assigned probability $(1 - \eta) \cdot 2^{-n}$.

For any $x \in \{0,1\}^n$, an error in the evaluation of $f_a(x)$ is encountered when $\widehat{D}(xn) > \widehat{D}(xp)$. Each such error contributes at least $(1 - 2\eta) \cdot 2^{-n}$ to $\|D_a - \widehat{D}\|_1$, and thus the total number of errors cannot exceed $\frac{2^n \cdot \varepsilon}{1-2\eta}$. Reconciling this with the fact that there are $2^n$ strings $x \in \{0,1\}^n$, we have that the probability of error is bounded above by $\frac{\varepsilon}{1-2\eta}$. Thus, by varying $\varepsilon$ we can approximate $f_a(x)$ to arbitrary precision. ■

We mention that this particular method of reduction does not extend to the $L_p$-PAC learning framework for $p > 1$. To show this, we set the approximating distribution $\widehat{D}$ to be the "opposite" of $D_a$ in the following manner: $\widehat{D}(xn) = D_a(xp)$, $\widehat{D}(xp) = D_a(xn)$ for every $x \in \{0,1\}^n$. Calculating the consequences, we find:

$$\begin{aligned} \left\| D_a - \widehat{D} \right\|_p &= \\ &= \left[ 2^n \left[ (1-2\eta)2^{-n} \right]^p \right]^{1/p} \\ &= 2^{\frac{n}{p}} (1-2\eta) 2^{-n} \\ &= (1-2\eta) 2^{(\frac{n}{p}-n)}. \end{aligned}$$

For a large enough number of layers (i.e. $n$), however, the expression above can be made arbitrarily small. This example depicts a situation in which a good approximation in the $L_p$-PAC framework does not amount to useful learning[7]. Thus motivated, we will consider only the KL-PAC and $L_1$-PAC learning frameworks throughout the remainder of this thesis.

## 4.5 Learnability of PDFA via Oracles

As discussed in Section 4.3.2, efficient learnability using the state merging algorithm is known for the $\mu$-distinguishable PDFA subclass. In this section, we show that state merging algorithms can be extended to efficiently learn a larger subclass of PDFAs called $\mu_2$-*distinguishable*[8]. We will draw heavily on the positive result of Clark and Thollard [2004], and our result can indeed be seen as a direct extension of their work.

As discussed in Section 3.1.1, Carrasco and Oncina [1999] showed that corresponding to every distribution induced by a PDFA there exists a canonical PDFA with the minimal number of states which induces the same distribution. Furthermore, the suffix distributions of the states of the canonical PDFA are unique. Therefore, if we are given an oracle which can distinguish between two suffix distributions, we can learn the PDFA.

In this section we prove that for the $L_2$ distance, a simple statistical test can efficiently distinguish between sample multisets drawn from identical distributions and mul-

[7] Indeed, in a prominent density estimation textbook [Devroye and Lugosi, 2001, Section 6.5], a complete section is titled *"$L_2$ Distances Are To Be Avoided"*.

[8] The work described in this section was published in [Guttman et al., 2005].

tisets drawn from (sufficiently) distant distributions. We consequently show how the state merging algorithm can use an oracle to learn $\mu_2$-distinguishable PDFA. Our definition of $\mu_p$ distinguishability is a generalization of $\mu$-distinguishability. Namely, the suffix distributions of any two states of a $\mu_p$-distinguishable PDFA are at least $\mu$ apart in the $L_p$ distance for some $1 \leq p \leq \infty$.

### 4.5.1 The Generalized State Merging Algorithm: Introduction

The general PDFA learning problem is hard, and therefore additional conditions are required before attempting to provide an efficient learning algorithm. We seek to understand which criteria indeed enable efficient testing for discrimination between distributions. To this end, we present the SM algorithm in a more general setting using the concept of an *oracle*, which will be rigorously defined below.

In order to describe how state merging algorithms can use oracles to learn PDFA distributions, we first provide a modular analysis of the proof due to Clark and Thollard [2004], and then extend it to deal with oracles. In particular, we show that the state merging algorithm may be decoupled into two parts:

- A construction algorithm which iteratively builds the PDFA graph and sets the transition probabilities.

- An oracle, providing an accurate test for deciding whether or not two sample multisets were drawn from two distinct suffix distributions.

Given such an oracle, the state merging algorithm will induce a PDFA such that with high probability, the KL-divergence between target and induced distributions can be made arbitrarily small.

#### Generalized State Merging: Detailed Description

Pseudocode for *generalized state merging* (GSM) is given in Algorithm 4.2 below. The learning algorithm is given the following parameters as input: an alphabet $\Sigma$, an upper bound $L$ on the expected length of strings generated from *any* state of the target, an upper bound $n$ on the number of states in the target, a confidence parameter $\delta$ and a precision parameter $\varepsilon$. We will show that given a *matching oracle* (defined below), the algorithm will (with high probability) learn a PDFA class $\mathcal{H}$.

The algorithm maintains a digraph $G = (V, E)$ with labeled edges, $V$ being the set of vertices (or nodes) and $E \subseteq V \times \Sigma \times V$ a set of edges. The graph holds a current *hypothesis* about the structure of the target PDFA. A particular vertex $v_0 \in V$ corresponds to the initial state of the hypothesis. Each arc in the graph is labeled with an alphabet letter,

**Algorithm 4.2**: Generalized State Merging

```
Input: $\varepsilon, \delta$ (accuracy, confidence parameters), $n_{\max}$ (upper bound on the number
       of states of the target) and $L$ (upper bound on the expected length of
       strings generated from any state of the target). The algorithm is also
       supplied with $\mathcal{O}_{\mathcal{H}}$ (a $(\delta_1, m_1)$-matching oracle for $\mathcal{H}$), and a random source
       of strings generated independently by $\mathcal{A}$, the target PDFA.
Output: $\widehat{\mathcal{A}}$, a hypothesis PDFA such that $\mathrm{KL}(D_{\mathcal{A}} \parallel D_{\widehat{\mathcal{A}}}) < \varepsilon$ with probability at
        least $1 - \delta$.
Data: The algorithm maintains a graph $G = (V, E)$ with labeled edges (i.e.
      $E \subseteq V \times \Sigma \times V$), which holds the current hypothesis about the structure of
      the target automaton.
repeat
   Draw $M$ strings from $\mathcal{A}$
   foreach $u \in V$ and $\sigma \in \Sigma$ which does not yet label an edge out of $u$ do
      Hypothesize a candidate node, referred to as $(u, \sigma)$
      Compute $S_{u,\sigma}$ (suffix multiset of candidate node $(u, \sigma)$ )
      if $|S_{u,\sigma}| \geq m_0$ then
         foreach $v \in V$ do
            Query $\mathcal{O}_{\mathcal{H}}$ to compare $S_{u,\sigma}$ with $S_v$
            if $\mathcal{O}_{\mathcal{H}}$ returns ACCEPT then
               Add arc labeled with $\sigma$ from $u$ to $v$.
            end
         end
         if $\mathcal{O}_{\mathcal{H}}$ returns REJECT on all comparisons then
            Create new node to graph $G$
            Add an edge labeled with $\sigma$ from $u$ to the new node
         end
      end
   end
until no candidate node has a suffix multiset of cardinality (at least) $m_0$.
Complete $G$ by adding a ground node which represents low frequency states
Add a final state $\widehat{q}_f$ and transitions labeled with $\zeta$ from each state to $\widehat{q}_f$
```

and there is at most one edge labeled with a particular letter from each node. For some node $v \in V$, the notation $\delta_G(v, \sigma)$ refers to the node reached by the arc from $v$ labeled with $\sigma$, if such a node exists.

If $S$ is a multiset of strings from $\Sigma^*$, for each $s \in \Sigma^*$, $S(s)$ denotes the multiplicity of $s$ in $S$, the cardinality of $S$ is defined as $|S| := \sum_{s \in \Sigma^*} S(s)$, and for every $\sigma \in \Sigma$, $S(\sigma) := \sum_{s \in \Sigma^*} S(\sigma s)$.

Initially, the graph $G$ consists of a single node representing the initial state $q_0$ of the target PDFA, and an accompanying multiset that is a sample of strings generated by the target PDFA. At any given moment in the course of the algorithm's run, (with high probability) the graph is isomorphic to a subgraph of the target PDFA. For each node $v \in V$, the *suffix multiset* $S_v$ represents the multiset of suffixes of strings incident on $v$.

At each iteration (line 2) the algorithm is supplied with $M$ strings generated independently by the target PDFA. For each node $u$ in the graph and each letter $\sigma \in \Sigma$ which does not yet label an arc out of $u$, a *candidate node* is hypothesized, referred to by the pair $(u, \sigma)$ (always in a relevant context, avoiding confusion). The algorithm computes $S_{u,\sigma}$, the multiset of suffixes associated with $(u, \sigma)$. This is performed by deleting from each string incident on $u$ the prefix emitted before arrival at $u$. If a suffix begins with the symbol $\sigma$, the symbol is deleted and the resulting string is added to the multiset $S_{u,\sigma}$. Intuitively, this sample represents the suffix distribution of the relevant state.

If at any point in the algorithm (line 6) a suffix multiset's size, $|S_{u,\sigma}|$, achieves the threshold $m_0$, the oracle $\mathcal{O}_{\mathcal{H}}$ is queried, comparing $S_{u,\sigma}$ to all multisets $S_v$, $v \in V$. If all comparisons to existing nodes in the graph are negative, a new node is created, and an arc from $u$ to the new node is added, labeled with $\sigma$. When a node is added to the graph, its accompanying multiset is kept and remains unchanged for the remainder of the algorithm.

After each iteration, all candidate nodes are deleted. The algorithm terminates when a sample had been drawn where no candidate node has a sufficiently large multiset. Subsequently, the hypothesis PDFA graph is completed. If there are strings not accepted by the graph, a new node called the *ground node* is added, representing all low frequency states. The graph is then completed by adding all possible arcs from all states leading to the ground node, including from the ground node to itself. Since the hypothesis PDFA must accept every string, every state must have an arc leading out of it for each letter in the alphabet.

The transition probabilities are estimated using a simple additive smoothing scheme introduced in [Ron et al., 1995]. For a state $u \in Q_{\hat{\mathcal{A}}}$ and a symbol $\sigma \in \Sigma$, the transition probability is set to:

$$\begin{aligned} P_{\widehat{\mathcal{A}}}(\widehat{q},\sigma) &= \left(\frac{S_u(\sigma)}{|S_u|}\right) \cdot (1-(|\Sigma|+1)p_{\min}) + p_{\min}, \quad \text{with} \\ p_{\min} &:= \frac{\varepsilon}{4(L+1)(|\Sigma|+1)}. \end{aligned} \tag{4.4}$$

A quick analysis shows that the number of oracle queries performed at each step of the GSM algorithm is upper bounded by $n^2|\Sigma|$, as there are at most $n$ nodes in the graph at any time and at most $n|\Sigma|$ candidate nodes. When the algorithm runs correctly there are at most $n|\Sigma|+2$ iterations. Therefore, over the course of a complete *successful* run, the number of oracle calls is at most $n^2|\Sigma|(n|\Sigma|+2)$.

The main distinction between Algorithms 4.1 and 4.2 lies in GSM's decoupling of the oracle (in the pseudocode, and more importantly in the ensuing analysis) from the remaining parts of the algorithm.

### Formal Definition of Oracle

For our purposes, an oracle is a black-box which can distinguish between suffix distributions. More formally, we have:

**Definition 4.11 (Oracle)** *Given a class $\mathcal{H}$ of PDFA, an oracle $\mathcal{O}_{\mathcal{H}}$ is said to $(\delta,m)$-*match *the class $\mathcal{H}$ if for any PDFA $\mathcal{A} \in \mathcal{H}$ and for any pair of states $q,q'$ in $\mathcal{A}$, given sample multisets of at least $m$ samples drawn as suffixes from $q$ and $q'$, the oracle can determine with probability at least $1-\delta$ whether or not the two multisets were drawn from suffix distributions of the same state.*

The definition provides for an accurate test for deciding whether or not two distributions over strings are similar (i.e. drawn from the same suffix distribution). We argue in Section 4.5.2 below that such accurate testing can in fact be achieved efficiently under relaxed conditions compared to [Clark and Thollard, 2004], which in turn still guarantee KL-PAC learnability.

## 4.5.2 The Generalized State Merging Algorithm: Analysis

The learning algorithm we use is analogous to the state merging algorithm described in [Clark and Thollard, 2004], with the oracle $\mathcal{O}_{\mathcal{H}}$ testing whether to merge a hypothesized *candidate* state (see Definition 4.13) with an existing one, or to construct a new state. Our main result is:

**Theorem 4.12** *Let $\mathcal{H}$ be a class of PDFAs over the alphabet $\Sigma$, $\varepsilon > 0$, $\delta > 0$, $L$ and $n$ positive integers, and $\delta_1, \delta_2, \varepsilon_1, m_2$ as defined in* (4.9) *and* (4.10) *below.*

*Suppose $\mathcal{O}_{\mathcal{H}}$ is a $(\delta_1, m_1)$-matching oracle for $\mathcal{H}$. For every n-state PDFA $\mathcal{A} \in \mathcal{H}$ such that the expected length of the string generated from every state is upper-bounded by L, with probability at least $1 - \delta$ over a random draw of $\max(m_1, m_2)$ samples generated by $\mathcal{A}$, Algorithm 4.2 produces an hypothesis PDFA $\widehat{\mathcal{A}}$ such that $\mathrm{KL}(D_{\mathcal{A}} \parallel D_{\widehat{\mathcal{A}}}) < \varepsilon$.*

Our proof closely follows the proof in [Clark and Thollard, 2004], with two key changes:

- In the original proof, distinguishability between two states' suffix distributions (with high probability) is inherent to the sample size bounds. In our case, the oracle provides the distinguishing test, so the size of the multiset drawn at each step of the algorithm is reformulated to reflect this (see (4.9)).
- In our case, two sources of randomness are present: the randomly drawn multisets and the oracle. To bound the probability of error, both sources need to be accounted for.

With the exceptions noted above, the original proof is directly transferable to our setting. In order to make the exposition self-contained, we repeat the statements and proofs of lemmas and theorems from [Clark and Thollard, 2004] in Appendix B, translated to the notation conventions used throughout the thesis.

We decompose our proof into the following *modules*:

(i) Given sufficiently many samples randomly drawn from $\mathcal{A}$, a multiset is likely to be "good". Namely, *every* string will appear with an empirical probability that is close to its actual probability.

(ii) Assuming an oracle which matches the PDFA family under consideration, at each step the hypothesis graph will be isomorphic to a subgraph of the target with high probability.

(iii) With high probability, when the algorithm stops drawing samples, there will be in the hypothesis graph a state representing each frequent state in the target. In addition, all frequent transitions will also have a representative edge in the graph.

(iv) After the algorithm terminates, (again with high probability) all transition probability estimates will be close to their correct target values.

(v) A KL-PAC result between target and hypothesis PDFAs follows.

### Additional Notation

The following definitions quantify the notion of *weight* for the various elements of a PDFA:

- For all $q \in Q_{\mathcal{A}}$ and $\ell \in \mathbb{N}$, define

$$W_\ell(q) := \sum_{\substack{s \in \Sigma^\ell: \\ \delta_{\mathcal{A}}(q_0, s) = q}} P_{\mathcal{A}}(q_0, s).$$

This quantity is the probability that $\mathcal{A}$ will generate at least $\ell$ characters and be in the state $q$ after having generated the first $\ell$ characters. Note that the definition addresses prefix probabilities rather than string probabilities, as the final state probabilities are not included in the summation. This quantity may be expressed recursively:

$$W_\ell(q) = \sum_{b \in Q_{\mathcal{A}}} W_{\ell-1}(b) \sum_{\sigma : \delta_{\mathcal{A}}(b,\sigma) = q} P_{\mathcal{A}}(b, \sigma). \tag{4.5}$$

- For $\ell \in \mathbb{N}$, define the probability that $\mathcal{A}$ will generate a string of length at least $\ell$,

$$W_\ell := \sum_{q \in Q_{\mathcal{A}}} W_\ell(q).$$

- The *weight* of a state $q \in Q_{\mathcal{A}}$ is defined as:

$$W(q) := \sum_{\ell=0}^{\infty} W_\ell(q) = \sum_{\substack{s \in \Sigma^*: \\ \delta_{\mathcal{A}}(q_0, s) = q}} P_{\mathcal{A}}(q_0, s).$$

The expected length of strings generated from any state can be defined using these terms. The (assumed) bound $L$ on the expected string length generated from any state $q \in Q_{\mathcal{A}}$ is expressed as:

$$\sum_{s \in \Sigma^*} P_{\mathcal{A}}(q, s) \leq L + 1.$$

Using this bound, we can establish that for any string length $k$,

$$\sum_{\ell > k} W_\ell = \sum_{q \in Q_{\mathcal{A}}} W_\ell(q) \sum_{\ell > 0} \sum_{s \in \Sigma^\ell} P_{\mathcal{A}}(q, s) \leq L W_k.$$

This enables the following bound, which will be useful in the sequel:

$$\sum_{\ell=0}^{\infty} \ell W_\ell(q) \leq \sum_{\ell=0}^{\infty} \ell W_\ell = \sum_{\ell=0}^{\infty} \sum_{k > \ell} W_k \leq \sum_{\ell=0}^{\infty} L W_\ell \leq L(L+1). \tag{4.6}$$

The following definitions quantify the joint weight of state pairs in the target and hypothesis PDFAs, $\mathcal{A}$ and $\widehat{\mathcal{A}}$ respectively. Formally, given two PDFA $\mathcal{A}, \widehat{\mathcal{A}}$ and a pair of states $q \in Q_{\mathcal{A}}, \widehat{q} \in Q_{\widehat{\mathcal{A}}}$, the *joint weight* $W(q, \widehat{q})$ is the expected number of times the automata are simultaneously in the states $q \in Q_{\mathcal{A}}$ and $\widehat{q} \in Q_{\widehat{\mathcal{A}}}$, when strings are being generated by $\mathcal{A}$ and parsed by $\widehat{\mathcal{A}}$. The definitions proceed along similar lines to the single state weights.

For a string $s \in \Sigma^*$, define

$$W_\ell(q,\widehat{q}) := \sum_{\substack{s:\delta_\mathcal{A}(q_0,s)=q \\ \delta_{\widehat{\mathcal{A}}}(\widehat{q}_0,s)=\widehat{q} \\ |s|=\ell}} P_\mathcal{A}(q_0,s).$$

Letting $W_0(q_0,\widehat{q}_0) = 1$, the definition can be expressed recursively:

$$W_\ell(q,\widehat{q}) = \sum_{b\in Q_\mathcal{A}} \sum_{\widehat{b}\in Q_{\widehat{\mathcal{A}}}} W_{\ell-1}(b,\widehat{b}) \sum_{\substack{\sigma:\delta_\mathcal{A}(b,\sigma)=q \\ \delta_{\widehat{\mathcal{A}}}(\widehat{b},\sigma)=\widehat{q}}} P_\mathcal{A}(b,\sigma).$$

We now define the expected number of times the first automaton will be in state $q$ and the second in state $\widehat{q}$,

$$\begin{aligned} W(q,\widehat{q}) &:= \sum_{\ell\in\mathbb{N}} W_\ell(q,\widehat{q}), \quad \text{noting that} \\ W(q) &= \sum_{\widehat{q}\in Q_{\widehat{\mathcal{A}}}} W(q,\widehat{q}). \end{aligned} \tag{4.7}$$

Given these quantities we can now use the following decomposition of the KL-divergence shown in [Carrasco, 1997]:

$$\mathrm{KL}(\mathcal{A} \,\|\, \widehat{\mathcal{A}}) = \sum_{q\in\mathcal{A}} \sum_{\widehat{q}\in\widehat{\mathcal{A}}} \sum_{\sigma\in\Sigma} W(q,\widehat{q}) P_\mathcal{A}(q,\sigma) \log \frac{P_\mathcal{A}(q,\sigma)}{P_{\widehat{\mathcal{A}}}(\widehat{q},\sigma)}.$$

The following notions define relationships between PDFA states to sets of strings:

- The set of strings that reach $q$ for the first time (i.e. have no proper prefix that reaches $q$),

$$R_\mathcal{A}(q) := \{s \in \Sigma^* \;:\; \delta_\mathcal{A}(q_0,s) = q \wedge (\nexists x,y \in \Sigma^* \text{ s.t. } y \neq \epsilon \;\wedge\; xy = s \;\wedge\; \delta(q_0,x) = q)\}.$$

- The probability of $\mathcal{A}$ being in state $q$ at least once,

$$P(q) := \sum_{s\in R_\mathcal{A}(q)} P_\mathcal{A}(q_0,s).$$

- The *exit probability* of a graph $G$ with respect to a PDFA $\mathcal{A}$ defines the probability that for a randomly generated string $s$, there exists no node $v$ such that $\delta_G(v_0, s) = v$.

$$P_{\text{exit}}(G) := \sum_{\substack{s \in \Sigma^* \\ s \text{ exits } G}} P_{\mathcal{A}}(q_0, s).$$

### Definition Modifications

The following definitions from [Clark and Thollard, 2004] have been modified for the purposes of our proof. Definition B.1 of a *candidate node* is generalized to the following:

**Definition 4.13 (Candidate node)** *A* candidate node *is a pair $(u, \sigma)$ where $u$ is a node in the graph $G$ underlying the current hypothesis PDFA, $\sigma \in \Sigma$, and $\delta_G(u, \sigma)$ is undefined. It will have an associated suffix multiset $S_{u,\sigma}$. A candidate node $(u, \sigma)$ and a node $v$ in a hypothesis graph $G$ are* similar *if and only if the matching oracle $\mathcal{O}_{\mathcal{H}}$ returns* ACCEPT *when queried with $\widehat{S}_{u,\sigma}$ and $\widehat{S}_v$, where $\widehat{S}$ denotes the empirical distribution induced by a multiset $S$.*

Definition B.2 of a *good multiset* is relaxed, with the original distinguishability condition $\left\|\widehat{S} - P^q_{\mathcal{A}}\right\|_\infty < \mu/4$ lifted:

**Definition 4.14 (good multiset)** *A multiset $S$ is $\varepsilon_1$-good for a state $q$ if for every $\sigma \in \Sigma$, $\left|(S(\sigma)/|S|) - P^q_{\mathcal{A}}(\sigma)\right| < \varepsilon_1$.*

The concept of a good hypothesis graph (Definition B.3) has been relaxed accordingly, with the (only) modification inherited from Definition 4.14.

**Definition 4.15 (good hypothesis graph)** *A hypothesis graph $G$ for a PDFA $\mathcal{A}$ is* good *if there is a bijective function $\Phi$ from a subset of states of $\mathcal{A}$ to all the nodes of $G$ such that $\Phi(q_0) = v_0$, and if $\delta_G(u, \sigma) = v$ then $\delta_{\mathcal{A}}(\Phi^{-1}(u), \sigma) = \Phi^{-1}(v)$, and for every node $u$ in $G$, the multiset $S_u$ attached to $u$ is $\varepsilon_1$-good for the state $\Phi^{-1}(u)$.*

Finally, the concept of a good sample (Definition B.4) has been relaxed in a similar manner, with the modification inherited from Definition 4.14.

**Definition 4.16 (good sample)** *Given a good hypothesis graph $G$, a sample of size $M$ is* good *if for every candidate node $(u, \sigma)$ such that $|S_{u,\sigma}| > m_0$, $S_{u,\sigma}$ is $\varepsilon_1$-good for the state $\delta_{\mathcal{A}}(\Phi^{-1}(u), \sigma)$ and if $P_{exit}(G) > \varepsilon_6$ then the number of strings that exit the graph is more than $\frac{1}{2}NP_{exit}(G)$.*

Note that if there are no candidate nodes with multisets larger than $m_0$, then the total number of strings that exited the graph must be less than $n|\Sigma|m_0$ (since there are at most $n$ nodes in a good graph, and therefore at most $n|\Sigma|$ candidate nodes). Therefore in this circumstance, if the samples are good, we can conclude that either $P_{\text{exit}}(G) \leq \varepsilon_6$ or $P_{\text{exit}}(G) < 2n|\Sigma|m_0/M$.

We proceed to analyze the algorithm, adhering to the module structure defined above.

### Module (i)

The threshold $m_0$ on the minimal multiset size required for testing distribution proximity in Algorithm 4.2 is set to

$$m_0 = \max(m_1, m_2), \tag{4.8}$$

where $m_1$ is the number of samples required by the matching oracle $\mathcal{O}_{\mathcal{H}}$ to guarantee an error probability of at most $\delta_1$, and $m_2$ is the multiset size defined below, shown in Lemma B.5 to guarantee a good sample (according to Definition 4.14 above) with probability at least $1 - \frac{\delta_2}{n|\Sigma|}$:

$$\begin{aligned} m_2 &:= \frac{1}{2\,\varepsilon_1^2} \log\left(\frac{24n|\Sigma|(|\Sigma|+1)(n|\Sigma|+2)}{\delta_2}\right), \quad \text{with} \\ \varepsilon_1 &:= \frac{\varepsilon^2}{16(|\Sigma|+1)(L+1)^2}. \end{aligned} \tag{4.9}$$

The parameters $\delta_1$ and $\delta_2$ are related to the confidence parameter $\delta$ by Equations (4.10), and the number of samples drawn at each iteration of the algorithm is:

$$\begin{aligned} M &:= \frac{4n|\Sigma|L^2(L+1)^3}{\varepsilon_3^2} \max\left(2n|\Sigma|m_0, 4\log\frac{2(n|\Sigma|+2)}{\delta}\right), \quad \text{with} \\ \varepsilon_3 &:= \frac{\varepsilon}{2(n+1)\log\left(4(L+1)(|\Sigma|+1)/\varepsilon\right)}. \end{aligned}$$

### Module (ii)

We use the matching oracle to prove that with high probability, at all times over the course of the algorithm's run, the hypothesis graph is good, as specified in Definition 4.15.

**Lemma 4.17** *Let $\mathcal{H}$ be a PDFA class and let $\mathcal{O}_{\mathcal{H}}$ be a $(\delta_1, m_1)$-matching oracle. Assume $G_i$, the hypothesis graph at the $i^{th}$ iteration, is good. Assume further that for every candidate node $(u, \sigma)$ such that $|S_{u,\sigma}| \geq m_0$, the multiset $S_{u,\sigma}$ is $\varepsilon_1$-good for the state $\delta_G(\Phi^{-1}(u), \sigma)$, and there exists at least one such candidate node. Then with probability at least $1 - \delta_1 n^2 |\Sigma|$, the hypothesis graph $G_{i+1}$ is also good.*

**Proof** Consider a candidate node $(u, \sigma)$ and a node $v$. If both are representative of the same state, the oracle errs with probability at most $\delta_1$, and otherwise the (good) hypothesis graph $G_i$ remains unchanged. If no node $v$ representative of the same state exists, the algorithm constructs a new node $v$ and sets $\Phi^{-1}(v)$ to $\delta_G(\Phi^{-1}(u), \sigma)$, in which case the new graph $G_{i+1}$ is also good. Additionally, since the candidate node multisets are $\varepsilon_1$-good, the multiset of this node will also be good.

The algorithm queries $\mathcal{O}_{\mathcal{H}}$ at most $n^2|\Sigma|$ times at each iteration. By applying the union bound and using the definition of a matching oracle we obtain the lemma. ■

### Module (iii)

This module's statements and proofs are given in Lemmas B.7 and B.8. Aside from notation, the two lemmas and their proofs remain unchanged relative to [Clark and Thollard, 2004].

### Module (iv)

Lemma B.9 rigorously formulates the relevant claim and presents its proof, both of which require no changes relative to [Clark and Thollard, 2004].

### Module (v)

We now derive a KL-PAC bound. In comparison to the result in [Clark and Thollard, 2004], our framework contains an additional degree of randomness due to the probabilistic nature of the oracle. However, if this probability of error is controlled, the same KL-divergence bound between target and hypothesis PDFA (namely $\varepsilon$) follows. Setting:

$$\delta_1 = \frac{\delta}{2n^2|\Sigma|(n|\Sigma| + 2)}, \tag{4.10a}$$

$$\delta_2 = \delta/2, \tag{4.10b}$$

using multisets of size $m_0 = \max(m_1, m_2)$, and applying the union bound, we bound the probability of error by $\delta$.

In order to prove the desired $\varepsilon$ approximation accuracy, we appeal to Theorem B.10. The proof of Theorem 4.12 follows.

## 4.6 Learnability of $\mu_2$-Distinguishable PDFA

### 4.6.1 Related Work

We now discuss a recent result on testing distribution proximity in $L_2$ due to [Batu et al., 2000], which was used in [Guttman et al., 2005] to show efficient learnability of the $\mu_2$-distinguishable PDFA class. The result applies to distributions over finite sets, and the computational complexity does not depend on the cardinality of the set.

**Theorem 4.18 ([Batu et al., 2000])** *Let $\delta > 0$ be a given confidence parameter and let $\mathbb{A}$ denote a finite set of cardinality $|\mathbb{A}|$. Let $D_1$ and $D_2$ be two distributions over $\mathbb{A}$. Given $m = O\left(\varepsilon^{-4} \log\left(\frac{1}{\delta}\right)\right)$ samples drawn from $D_1$ and $D_2$, if $\|D_1 - D_2\|_2 \leq \varepsilon/2$, Algorithm 4.3 will output* CLOSE *with probability at least $1-\delta$. If $\|D_1 - D_2\|_2 \geq \varepsilon$ then the* *algorithm outputs* FAR *with probability at least* $1 - \delta$*. The running time of the algorithm is* $O\left(\varepsilon^{-4}\log\left(\frac{1}{\delta}\right)\right)$.

In Algorithm 4.3, $r_D$ denotes the number of self-collisions in the multiset $F_D$, namely the count of $i < j$ such that the $i^{\text{th}}$ sample in $F_D$ is same as the $j^{\text{th}}$ sample in $F_D$. Similarly, $c_{D_1D_2}$, the number of collisions between $D_1$ and $D_2$ is the count of $(i, j)$ such that the $i^{\text{th}}$ sample in $D_1$ is same as the $j^{\text{th}}$ sample in $D_2$.

```
Algorithm 4.3: $L_2$-Distance-Test
  Input: Samples from $D_1$ and $D_2$, parmeters $m, \varepsilon, \delta$
  Result: CLOSE or FAR
repeat
    Draw $F_{D_1}$, a multiset of $m$ samples from $D_1$
    Draw $F_{D_2}$, a multiset of $m$ samples from $D_2$
    Let $r_{D_1} = |F_{D_1} \cap F_{D_1}|$ (the number of self-collisions in $F_{D_1}$)
    Let $r_{D_2} = |F_{D_2} \cap F_{D_2}|$
    Draw $Q_{D_1}$, a multiset of $m$ samples from $D_1$
    Draw $Q_{D_2}$, a multiset of $m$ samples from $D_2$
    Let $c_{D_1D_2} = |Q_{D_1} \cap Q_{D_2}|$
    Let $r = \frac{2m}{m-1}(r_{D_1} + r_{D_2})$
    Let $s = 2c_{D_1D_2}$
    if $r - s > m^2\,\varepsilon^2\,/2$ then declare trial FAR else declare trial CLOSE
until $O\left(\log\left(\frac{1}{\delta}\right)\right)$ iterations
if majority of trials declared FAR then return FAR else return CLOSE
```

Note that Algorithm $L_2$-Distance-Test's running time is independent of the cardinality $|\mathbb{A}|$.

### 4.6.2 A Novel Distribution Proximity Testing Result

We now propose a novel analysis and an accompanying algorithm for testing distribution proximity in the $L_2$ distance, improving the sample and computational complexities of Theorem 4.18 from $O(\varepsilon^{-4})$ to $O(\varepsilon^{-2})$. We use the trivial empirical proximity test formally described in Algorithm 4.4, and our analysis involves a symmetrization inequality. Formally, we prove the following theorem:

**Theorem 4.19** *Let* $\varepsilon, \delta > 0$ *be accuracy and confidence parameters and let* $\mathbb{A}$ *denote a finite set of cardinality* $|\mathbb{A}|$*. Let* $D_1$ *and* $D_2$ *be two distributions over* $\mathbb{A}$*. Given* $m = 8192\,\varepsilon^{-2}\log\left(\frac{1}{\delta}\right)$ *samples drawn from each of the distributions* $D_1$ *and* $D_2$*, if* $D_1 = D_2$*, Algorithm 4.4 will output* CLOSE *with probability at least* $1 - \delta$*. If* $\|D_1 - D_2\|_2 \geq \varepsilon$ *then the algorithm outputs* FAR *with probability at least* $1 - \delta$*. The running time of the algorithm is* $O\left(\varepsilon^{-2}\log\left(\frac{1}{\delta}\right)\right)$.

We will need the following (symmetrization) lemma, proved in Appendix A.

**Algorithm 4.4**: Improved $L_2$-Distance-Test

```
Input: Samples from $D_1, D_2, \varepsilon, \delta$
Result: CLOSE or FAR
Draw $1024\varepsilon^{-2}\log\left(\frac{1}{\delta}\right)$ samples from the distributions $D_1$ and $D_2$
repeat
    Draw $m$ samples from $D_1$, obtaining empirical distribution $\widehat{D}_1$
    Draw $m$ samples from $D_2$, obtaining empirical distribution $\widehat{D}_2$
    if $\|\widehat{D}_1 - \widehat{D}_2\|_2 > \varepsilon/2$ then declare trial FAR else declare trial CLOSE
until $8\log\left(\frac{1}{\delta}\right)$ iterations
if majority of trials declared FAR then return FAR else return CLOSE
```

**Lemma 4.20** *Let $\{X_i\}_{i=1}^m$ be i.i.d. random variables. Then:*

$$\mathbb{E}\left|\frac{1}{m}\sum_{i=1}^m X_i - \mathbb{E}X_1\right| \leq \frac{2}{m}\mathbb{E}_X\mathbb{E}_\varepsilon\left|\sum_{i=1}^m \varepsilon_i X_i\right|,$$

*where $\{\varepsilon_i\}_{i=1}^m$ are Rademacher random variables, which assume the values $-1$ and $1$ with probability $1/2$ each. The expectations $\mathbb{E}X$ are with respect to the (random) draw of the random variables, while $\mathbb{E}_\varepsilon$ are expectations with respect to the Rademacher random variables.*

We will also use the following lemma.

**Lemma 4.21** *Let $\{X_i\}_{i=1}^m$ be i.i.d. Bernoulli random variables with $\mathbb{E}X_1 < p$, $p < 1/2$. Let $Y = \sum_{i=1}^m X_i$. Then for $m = c\log\frac{1}{\delta}$ with $c = \frac{2}{(1-2p)^2}$,*

$$\Pr\left\{\frac{1}{m}\sum_{i=1}^m X_i > \frac{1}{2}\right\} < \delta.$$

**Proof** By the Hoeffding bound [Motwani and Raghavan, 1995], for $m$ i.i.d. Bernoulli random variables $\{X_i\}_{i=1}^m$ and any $t \geq 0$,

$$\Pr\left\{\sum_{i=1}^m X_i \geq \sum_{i=1}^m \mathbb{E}X_1 + t\right\} \leq e^{-2t^2/m}.$$

Substituting $t = \left(\frac{1}{2} - p\right)m$, $\delta = e^{-2t^2/m}$ and solving for $m$, we obtain the desired expression. ∎

**Proof** (Theorem 4.19)

Let $\{X_j\}_{j=1}^m$ be i.i.d. random variables such that $X_j \sim D$. Let $\widehat{d_i^m} = \frac{1}{m}\sum_{j=1}^m \mathbf{1}_{\{X_j=i\}}$ denote the estimates of $d_i$, and $\widehat{D}^m = \{\widehat{d_i^m}\}_{i=1}^m$ the corresponding distribution. We first bound the expected $L_2$-deviation between $\widehat{D}^m$ and $D$:

$$\begin{aligned}
& \mathbb{E}\left\|\widehat{D}^m - D\right\|_2 = \\
& \qquad = \mathbb{E}\left[\sum_{i=1}^{|\mathbb{A}|}\left|\frac{1}{m}\sum_{j=1}^{m}\mathbf{1}_{\{X_j=i\}} - \mathbb{E}\,\mathbf{1}_{\{X_j=i\}}\right|^2\right]^{1/2} \\
\text{(Lemma 4.20)} & \qquad \leq 2\,\mathbb{E}\left[\sum_{i=1}^{|\mathbb{A}|}\mathbb{E}_\varepsilon\left|\frac{1}{m}\sum_{j=1}^{m}\varepsilon_j\,\mathbf{1}_{\{X_j=i\}}\right|^2\right]^{1/2} \\
& \qquad = 2\,\mathbb{E}\left[\frac{1}{m^2}\sum_{i=1}^{|\mathbb{A}|}\mathbb{E}_\varepsilon\left|\sum_{j=1}^{m}\varepsilon_j\,\mathbf{1}_{\{X_j=i\}}\right|^2\right]^{1/2} \\
& \qquad = 2\,\mathbb{E}\left[\frac{1}{m^2}\sum_{i=1}^{|\mathbb{A}|}\mathbb{E}_\varepsilon\left(\sum_{j=1}^{m}\varepsilon_j^2\,\mathbf{1}^2_{\{X_j=i\}} + \sum_{j=1}^{m}\sum_{k=1}^{m}\varepsilon_j\,\varepsilon_k\,\mathbf{1}_{\{X_j=i\}}\,\mathbf{1}_{\{X_k=i\}}\right)\right]^{1/2} \\
(\mathbb{E}_\varepsilon\,\varepsilon_j\,\varepsilon_k = 0) & \qquad = 2\,\mathbb{E}\left[\sum_{i=1}^{|\mathbb{A}|}\frac{1}{m^2}\sum_{j=1}^{m}\mathbf{1}^2_{\{X_j=i\}}\right]^{1/2} \\
\text{(Jensen's inequality)} & \qquad \leq \frac{1}{\sqrt{m}}\left[\sum_{i=1}^{|\mathbb{A}|}\mathbb{E}\,\mathbf{1}^2_{\{X_j=i\}}\right]^{1/2} \\
& \qquad = \frac{1}{\sqrt{m}}\left[\sum_{i=1}^{|\mathbb{A}|}\mathbb{E}\,\mathbf{1}_{\{X_j=i\}}\right]^{1/2} \\
& \qquad = \frac{1}{\sqrt{m}}. \qquad (4.11)
\end{aligned}$$

This result is *independent* of the distribution $D$ and of the cardinality $|\mathbb{A}|$. For any $k \in \mathbb{N}$, plugging $m = \frac{k^2}{\varepsilon^2}$ into (4.11), we get:

$$\mathbb{E}\left\|\widehat{D}^m - D\right\|_2 \leq \frac{\varepsilon}{k}.$$

Using the Markov inequality, we have the upper-bound:

$$\Pr\left\{\left\|\widehat{D}^m - D\right\|_2 \geq \frac{8\varepsilon}{k}\right\} \leq \frac{1}{8}. \qquad (4.12)$$

The analysis of $\left\|\widehat{D}_1^m - \widehat{D}_2^m\right\|_2$ proceeds along standard lines. Let $\{X_j\}_{j=1}^m$ and $\{Y_j\}_{j=1}^m$ be i.i.d. random variables such that $X_j \sim D_1$ and $Y_j \sim D_2$. Applying the triangle inequality, we establish the following:

$$\begin{aligned}
\left[\sum_{i=1}^{n}\left|\frac{1}{m}\sum_{j=1}^{m}\mathbf{1}_{\{X_j=i\}}-\mathbf{1}_{\{Y_j=i\}}\right|^2\right]^{1/2} & = \\
& = \left\|\widehat{D}_1^m-\widehat{D}_2^m\right\|_2 \\
& = \left\|\widehat{D}_1^m-D_1+D_1-\widehat{D}_2^m+D_2-D_2\right\|_2 \\
& \geq \|D_1-D_2\|_2-\left\|\widehat{D}_1^m-D_1\right\|_2-\left\|\widehat{D}_2^m-D_2\right\|_2 \\
& \geq \varepsilon-\left(\left\|\widehat{D}_1^m-D_1\right\|_2+\left\|\widehat{D}_2^m-D_2\right\|_2\right).
\end{aligned}$$

Using (4.12) and applying the union bound, we have:

$$\Pr\left\{\left\|\widehat{D}_1^m-\widehat{D}_2^m\right\|_2\leq\varepsilon\left(1-\left(\frac{8}{k}+\frac{8}{k}\right)\right)\right\}\leq\frac{1}{8}+\frac{1}{8}=\frac{1}{4}. \tag{4.13}$$

Plugging $k=32$ into (4.13),

$$\Pr\left\{\left\|\widehat{D}_1^m-\widehat{D}_2^m\right\|_2\leq\frac{3\varepsilon}{4}\right\}\leq 1/4, \quad \text{for } m=\frac{1024}{\varepsilon^2}. \tag{4.14}$$

Let $\widehat{D}_1^m$ and $\widetilde{D}_1^m$ denote two independent empirical distributions, each using $m$ samples drawn from $D_1$. Using similar reasoning for $\left\|\widehat{D}_1^m-\widetilde{D}_1^m\right\|_2$, we get:

$$\Pr\left\{\left\|\widehat{D}_1^m-\widetilde{D}_1^m\right\|_2\geq\frac{\varepsilon}{4}\right\}\leq 1/4. \tag{4.15}$$

Thus, the correctness of the statistical test described above can be considered a Bernoulli random variable, with probability of failure bounded above by 1/4. Algorithm 4.4 uses a majority vote, so by Lemma 4.21, repeating the statistical test $8\log\left(\frac{1}{\delta}\right)$ times, the probability of error is bounded by $\delta$, proving the theorem. ■

The sample complexity proved above matches the lower-bound of $\Omega(\varepsilon^{-2})$ shown in [Batu et al., 2000, Theorem 24], establishing the asymptotic tightness of our analysis.

### 4.6.3 Efficient Learnability of $\mu_2$-Distinguishable PDFA

We will now use Theorem 4.12 to prove efficient learnability of a new class of PDFA strictly larger than the $\mu$-distinguishable class. We begin by generalizing the notion of $\mu$-distinguishability:

**Definition 4.22 ($\mu_p$-distinguishability)** *Let $\mu>0$ and $1\leq p\leq\infty$. Given a PDFA $\mathcal{A}=\langle Q_{\mathcal{A}},\Sigma,\delta_{\mathcal{A}},q_0,F_{\mathcal{A}},P_{\mathcal{A}}\rangle$, the state pair $(q_i,q_j)\in Q_{\mathcal{A}}\times Q_{\mathcal{A}}$ is said to be $\mu_p$-*

distinguishable *if*

$$\left\|D_{\mathcal{A}}^{q_i} - D_{\mathcal{A}}^{q_j}\right\|_p > \mu.$$

*A PDFA $\mathcal{A}$ is $\mu_p$-distinguishable if each of its state pairs is $\mu_p$-distinguishable.*

Note that for $p = \infty$ we recover the original $\mu$-distinguishability condition. Moreover, for any distribution $D$ over $\Sigma^*$, if $1 \leq p_1 < p_2 \leq \infty$ then we have $\|D\|_{p_1} \geq \|D\|_{p_2}$; hence the $\mu_{p_1}$-distinguishable class *properly* contains the $\mu_{p_2}$-distinguishable class.

As a consequence of Theorem 4.18, the $L_2$ proximity test of Algorithm 4.4 can serve as a $\left(\delta, \frac{C_2}{\varepsilon^2}\right)$-matching oracle for the $\mu_2$-distinguishable PDFA class, where $C_2$ is a constant hidden in the asymptotic notation. Thus, as a direct consequence of Theorem 4.12 and the $L_2$ matching oracle, we have the following theorem:

**Theorem 4.23** *The $\mu_2$-distinguishable class is efficiently learnable.*

### A Case for $\mu_2$-Distinguishability

A natural question in this context involves the relative efficiency of generalized state merging (Algorithm 4.2) and "standard" state merging (Algorithm 4.1). We now show that for the noisy parity PDFA family described in Section 4.3.1, the GSM algorithm outperforms SM by an arbitrary polynomial factor.

In the $n$-state noisy parity PDFA family, the $\mu_1$-distinguishability is a constant, while the $\mu_2$-distinguishability is $O(2^{-n/2})$ and the $\mu_\infty$-distinguishability is $O(2^{-n})$. Setting $n = \alpha \log t$, we obtain a $\mu_2$-distinguishability of $O(t^{-\alpha/2})$, and a $\mu_\infty$-distinguishability of $O(t^{-\alpha})$.

For this example, comparing between $\mu_\infty$ and $\mu_2$ and using Theorem 4.19, we have exhibited a PDFA learning problem for which GSM outperforms the SM by an arbitrary polynomial factor.

### Learnability of $\mu_p$-distinguishable Automata

For the case of $\mu_p$-distinguishable PDFA with $1 < p < 2$, a modification of Algorithm 4.4 with $\|\cdot\|_2$ replaced by $\|\cdot\|_p$ and an accompanying analysis similar to that of Theorem 4.19, the sample size required can be shown to be $m = O\left(|\mathbb{A}|^{\frac{2}{p}-1} \varepsilon^{-2} \log(\frac{1}{\delta})\right)$, which is no longer independent of $|\mathbb{A}|$. We conjecture that for $1 \leq p < 2$, no algorithm can guarantee a sample size that is independent of $|\mathbb{A}|$, suggesting that the class of $\mu_p$-distinguishable PDFA is not efficiently learnable.

For the specific case $p = 1$, the result in [Batu et al., 2000, Theorem 19] places a lower-bound of $\Omega(|\mathbb{A}|^{2/3})$ samples in order to distinguish between two (specific) distributions which have an $L_1$ distance of 1. This result is in line with the reduction to the noisy parity problem discussed in Section 4.3.1.

## 4.7 Learnability of $\rho$-Distinguishable PDFA

As mentioned earlier, the positive PDFA learnability result of Theorem 4.9 [Clark and Thollard, 2004] requires a constraint on the expected string length of all states' suffix distributions. In [Palmer and Goldberg, 2005], this constraint was lifted, and the algorithm's correctness was shown for the (weaker) $L_1$-PAC learning framework. In the present section we will prove that a PDFA subclass called ***ρ-distinguishable***, which strictly includes the $\mu$-distinguishable class, maintains learnability using the state merging algorithm.

The example shown in Figure 4.3 motivates the novel sufficient condition for learnability presented in this section. In the figure, the target PDFA's true distinguishability ($\mu$) tends to 0 as $\varepsilon$ tends to 0. Running SM$_\nu$ (defined in Algorithm 4.5 below) on (sufficiently many) samples drawn from the target PDFA will (for most values of $\nu$) result in a hypothesis PDFA with the top and bottom states merged. The distribution induced by the hypothesis will, however, constitute a good approximation of the target distribution. By relaxing the distinguishability condition we show that even when two significant states are wrongly merged, if their suffix distributions are sufficiently similar the resulting approximation remains good.

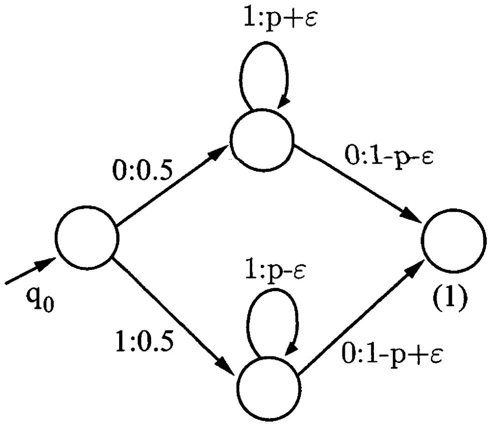

**Figure 4.3:** a PDFA which can be learned using SM$_\nu$ with $\nu > \mu$. In the example, $\mu = \|D^{q_1} - D^{q_2}\|_\infty \to 0$ as $\varepsilon \to 0$. However, (given enough samples) running SM$_\nu$ will result in a good approximation for many values of $\nu$.

### 4.7.1 Effects of Merging States on Induced Distribution

Let $\mathcal{A}$ be a PDFA inducing the distribution $D_\mathcal{A}$. Assume that some state $q_i \in Q_\mathcal{A}$ has been deleted from $Q_\mathcal{A}$, and that all edges pointing to $q_i$ have been diverted to another state $q_j$. Denoting the resulting PDFA $\widehat{\mathcal{A}}^{[q_i,q_j]}$ (and noting that the definition is not symmetric in its arguments), we seek to calculate the distances between distributions induced by the original and merged PDFAs.

We now calculate the KL-divergence and the $L_1$-distance between the original PDFA distribution, $D_{\mathcal{A}}$, and the post-merge distribution $D_{\widehat{\mathcal{A}}}^{[q_i,q_j]}$.

**Lemma 4.24** *Let $\mathcal{A}$ be a PDFA, and let $\widehat{\mathcal{A}}^{[q_i,q_j]}$ be the PDFA obtained by deleting the state $q_i \in Q_{\mathcal{A}}$ and diverting all edges incident on $q_i$ to $q_j \in Q_{\mathcal{A}}$. Then the $L_1$-distance between the distributions induced by pre-merge and post-merge PDFAs can be expressed as:*

$$\left\|D_{\mathcal{A}} - D_{\widehat{\mathcal{A}}}^{[q_i,q_j]}\right\|_1 = P_{\mathcal{A}}(q_i) \cdot \left\|D_{\mathcal{A}}^{q_i} - D_{\mathcal{A}}^{q_j}\right\|_1, \tag{4.16}$$

*while the KL-divergence can be expressed as:*

$$KL\left(D_{\mathcal{A}} \parallel D_{\widehat{\mathcal{A}}}^{[q_i,q_j]}\right) = P_{\mathcal{A}}(q_i) \cdot KL\left(P_{\mathcal{A}}^{q_i} \parallel P_{\mathcal{A}}^{q_j}\right) \tag{4.17}$$

**Proof** We use the definition of $R_{\mathcal{A}}(q)$, the set of strings reaching state $q$ for the first time, which we repeat for convenience, noting $P_{\mathcal{A}}(q) = \sum_{s \in R_{\mathcal{A}}(q)} P_{\mathcal{A}}(q_0, s)$:

$$R_{\mathcal{A}}(q) = \{s \in \Sigma^* \ : \ \delta_{\mathcal{A}}(q_0, s) = q \ \wedge \ (\nexists x, y \in \Sigma^* \text{ s.t. } y \neq \epsilon \ \wedge \ xy = s \ \wedge \ \delta(q_0, x) = q)\}.$$

Calculating the $L_1$-distance,

$$\begin{aligned}
&\left\|D_{\mathcal{A}} - D_{\widehat{\mathcal{A}}}^{[q_i,q_j]}\right\|_1 = \\
&= \sum_{s \in \Sigma^*} \left|D_{\mathcal{A}}(s) - D_{\widehat{\mathcal{A}}}^{[q_i,q_j]}(s)\right| \\
&= \sum_{s_1 \in R_{\mathcal{A}}(q_i)} \sum_{s_2 \in \Sigma^*} \left|P_{\mathcal{A}}(q_0, s_1) P_{\mathcal{A}}^{q_i}(s_2) - P_{\mathcal{A}}(q_0, s_1) P_{\mathcal{A}}^{q_j}(s_2)\right| \\
&= \sum_{s_1 \in R_{\mathcal{A}}(q_i)} P_{\mathcal{A}}(q_0, s_1) \sum_{s_2 \in \Sigma^*} \left|P_{\mathcal{A}}^{q_i}(s_2) - P_{\mathcal{A}}^{q_j}(s_2)\right| \\
&= P_{\mathcal{A}}(q_i) \cdot \left\|P_{\mathcal{A}}^{q_i} - P_{\mathcal{A}}^{q_j}\right\|_1.
\end{aligned}$$

Performing a similar calculation for the KL-divergence,

$$\begin{aligned}
&\mathrm{KL}\left(D_{\mathcal{A}} \parallel D_{\widehat{\mathcal{A}}}^{[q_i,q_j]}\right) = \\
&= \sum_{s \in \Sigma^*} D_{\mathcal{A}}(s) \log \frac{D_{\mathcal{A}}(s)}{D_{\widehat{\mathcal{A}}}^{[q_i,q_j]}(s)} \\
&= \sum_{s_1 \in R_{\mathcal{A}}(q_i)} \sum_{s_2 \in \Sigma^*} P_{\mathcal{A}}(q_0, s_1) P_{\mathcal{A}}^{q_i}(s_2) \log \frac{P_{\mathcal{A}}(q_0, s_1) P_{\mathcal{A}}^{q_i}(s_2)}{P_{\mathcal{A}}(q_0, s_1) P_{\mathcal{A}}^{q_j}(s_2)} \\
&= \sum_{s_1 \in R_{\mathcal{A}}(q_i)} P_{\mathcal{A}}(q_0, s_1) \sum_{s_2 \in \Sigma^*} P_{\mathcal{A}}^{q_i}(s_2) \log \frac{P_{\mathcal{A}}^{q_i}(s_2)}{P_{\mathcal{A}}^{q_j}(s_2)} \\
&= P_{\mathcal{A}}(q_i) \cdot \mathrm{KL}\left(P_{\mathcal{A}}^{q_i} \parallel P_{\mathcal{A}}^{q_j}\right).
\end{aligned}$$

■

The expressions show that in both cases, the operation is not symmetric in the roles of the states $q_i$ and $q_j$. For the $L_1$-distance, we see that if either $P_\mathcal{A}(q_1)$ or $\left\|P_\mathcal{A}^{q_1} - P_\mathcal{A}^{q_2}\right\|_1$ is negligibly small, the effect of the merge will be negligible (as the first term is bounded by 1 and the second by 2).

For the KL-divergence this is not generally the case, due to the unboundedness of $\mathrm{KL}\left(P_\mathcal{A}^{q_i} \parallel P_\mathcal{A}^{q_j}\right)$. Moreover, if two successive merges are performed then lacking a triangle inequality for the KL-divergence, the analysis is more involved. However, given the bounded expected suffix lengths from all states and the smoothing procedure inherent to the induction algorithm, a similar statement holds true as is detailed below.

### 4.7.2 Definition of $\rho$-Distinguishability

We now prove that a larger class of PDFA is efficiently learnable by using a generalization of Algorithm 4.1. The ensuing analysis is a further refinement of the results of Sections 4.5 and 4.6.

**Definition 4.25 ($\rho$-distinguishability)** *A PDFA $\mathcal{A}$ is said to be $\rho$-distinguishable if for every state pair $(q_i, q_j)$ such that $q_i \neq q_j$, the following holds:*

$$\frac{\max\{P_\mathcal{A}(q_i), P_\mathcal{A}(q_j)\} \cdot KL\left(P_\mathcal{A}^{q_i} \parallel P_\mathcal{A}^{q_j}\right)}{\left\|P_\mathcal{A}^{q_i} - P_\mathcal{A}^{q_j}\right\|_2} \leq \rho.$$

In Algorithm 4.5, we present a version of state merging which includes two additional free parameters $m_0$ and $\nu$, used to obtain our positive result. The algorithm uses the $L_2$ distribution proximity test described in Section 4.6.2.

The algorithm's inputs include the usual accuracy ($\varepsilon$) and confidence ($\delta$) parameters, as well as a *free* distinguishability parameter ($\nu$) and a minimal sample size for distribution proximity testing ($m_0$). The distinguishability parameter $\nu$ supplied to the algorithm may be different from the target PDFA's true distinguishability $\mu$. We will use the notation $\mathrm{SM}_\nu$ to denote the SM algorithm run with parameter $\nu$.

We state our result:

**Theorem 4.26** *Let $\nu$ and $m_0$ be as defined in* (4.19). *Let $\mathcal{A}$ be a $\rho$-distinguishable PDFA such that the expected length of strings generated from every state is upper-bounded by $L$. Then Algorithm 4.5 run with the parameters $\nu$ and $m_0$ will output a PDFA $\widehat{\mathcal{A}}$ such that $KL\left(D_\mathcal{A} \parallel D_{\widehat{\mathcal{A}}}\right) < \varepsilon$ with probability at least $1 - \delta$.*

Before proving the theorem, we define a number of new concepts and modify a number of existing definitions. An (ordered) state pair $(q_i, q_j)$ is called *first category* if

**Algorithm 4.5**: State Merging with Free Parameters $\nu, m_0$

```
Input: $\varepsilon, \delta, \nu$ (accuracy, confidence and distinguishability parameters), $m_0$
       (minimal sample size for distribution proximity testing). The algorithm is
       also supplied with a random source of strings generated independently by
       $\mathcal{A}$, the target PDFA.
Output: $\widehat{\mathcal{A}}$, a hypothesis PDFA such that $\mathrm{KL}(\mathrm{D}_{\mathcal{A}} \parallel \mathrm{D}_{\widehat{\mathcal{A}}}) < \varepsilon$ with probability at
        least $1 - \delta$.
Data: The algorithm maintains a graph $G = (V, E)$ with labeled edges (i.e.
      $E \subseteq V \times \Sigma \times V$), which holds the current hypothesis about the structure of
      the target automaton.
repeat
    Draw $M$ strings from $\mathcal{A}$
    foreach $u \in V$ and $\sigma \in \Sigma$ which does not yet label an edge out of $u$ do
        Hypothesize a candidate node $(u, \sigma)$
        Compute $S_{u,\sigma}$ (suffix multiset of candidate node $(u, \sigma)$ )
        if $|S_{u,\sigma}| \geq m_0$ then  foreach  $v \in V$ do
            if $\left\| \widehat{S}_{u,\sigma} - \widehat{S}_v \right\|_2 < \nu/2$ then
                Add arc labeled with $\sigma$ from $u$ to $v$.
            end
        end
        else if $\left\| \widehat{S}_{u,\sigma} - \widehat{S}_v \right\|_2 \geq \nu/2 \quad \forall v \in V$ then
            Create new node to graph $G$
            Add an edge labeled with $\sigma$ from $u$ to the new node
        end
    end
    Complete $G$ by adding a ground node which represents low frequency states
    Add a final state $\widehat{q}_f$ and transitions labeled with $\zeta$ from each state to $\widehat{q}_f$
until no candidate node has a suffix multiset of cardinality (at least) $m_0$.
```

$\mathrm{KL}\left(P_{\mathcal{A}}^{q_i} \parallel P_{\mathcal{A}}^{q_j}\right) \leq \varepsilon_{10}$, and *second category* otherwise[9]. For a state $q \in Q_{\mathcal{A}}$, we define the set $Z(q)$ of all states $q'$ such that $(q, q')$ is a first category pair:

$$Z(q) := \left\{q' \in Q_{\mathcal{A}} \ : \quad \mathrm{KL}\left(P_{\mathcal{A}}^{q} \parallel P_{\mathcal{A}}^{q'}\right) \leq \varepsilon_{10}\right\}.$$

Note that $q \in Z(q)$.

In the analysis, Definition 4.15 of a good hypothesis graph is relaxed to the following:

**Definition 4.27** *A hypothesis graph $G$ for a PDFA $\mathcal{A}$ is* good *if there is a bijective function $\Phi$ from a subset of states of $\mathcal{A}$ to all the nodes of $G$ such that $\Phi(q_0) = v_0$ and if $\delta_G(u, \sigma) = v$ then $\delta_{\mathcal{A}}\left(\Phi^{-1}(u), \sigma\right) \in Z\left(\Phi^{-1}(v)\right)$. Moreover, for every node $u$ in $G$, the multiset $S_u$ attached to $u$ is $\varepsilon_1$-good for the state $\Phi^{-1}(u)$.*

The relaxed version of good hypothesis graph (Definition 4.27) necessitates restating Lemmas B.7 and B.9. The lemmas' proofs, however, are identical, as they accommodate the relaxed definition of a good hypothesis graph (Definition 4.27).

### Restatement of Lemma B.7

**Lemma 4.28** *For any state $q \in Q_{\mathcal{A}}$ of the target PDFA such that $W(q) > \varepsilon_2$, if all sample multisets are good, then there will be a node $u$ in the final hypothesis graph such that $\Phi^{-1}(u) = q'$, with $q' \in Z(q)$. Furthermore, for such a state $q$ and any $\sigma \in \Sigma$ such that $P_{\mathcal{A}}(q, \sigma) > \varepsilon_5$, the node $\delta_G(u, \sigma)$ is defined and is equal to $\Phi(\delta_{\mathcal{A}}(q', \sigma))$.*

### Restatement of Lemma B.9

**Lemma 4.29** *Let $q \in Q_{\mathcal{A}}$ be a state with $W(q) > \varepsilon_2$. Then there exists a state $\widehat{q} \in \widehat{\mathcal{A}}$ such that $\widehat{q} = \Phi(q')$ with $q' \in Z(q)$, and*

$$W(q') - W(q', \widehat{q}) \leq \varepsilon_3 .$$

Note that $\Phi(q)$ may be undefined, in which case there exists a *single* $q' \in Q_{\mathcal{A}}$ to the above specification.

**Proof** (Theorem 4.26)

Our analysis refines that of Clark and Thollard [2004] in that we consider the target PDFA's *state pairs*, providing additional control over the approximation error. We first present a proof sketch. As shown in Theorem B.10, given the bound on expected string length ($L$) and the smoothing procedure inherent to the algorithm ($p_{\min}$), all negligible-weight states may be effectively disregarded. For the remaining states, we show that if the

[9] We use double-digit indices to avoid confusion with earlier sections.

pair $(q_i, q_j)$ is first category then assuming $\Phi(q_j)$ exists, the hypothesis PDFA $\mathcal{A}$ may have $\Phi(q_j)$ substituted for the (undefined) $\Phi(q_i)$, again causing no significant approximation error. For the remaining state pairs, the $\rho$-distinguishability condition implies a large distinguishability parameter $\mu_2$, in turn implying a correct merge / non-merge decision with high probability.

### Confidence Bound and Sample Complexity

Let $(q_i, q_j)$ be a second category state pair (i.e. KL $\left(P_{\mathcal{A}}^{q_i} \parallel P_{\mathcal{A}}^{q_j}\right) \geq \varepsilon_{10}\right)$ , with both states' weights greater than $\varepsilon_2$. Therefore, by the $\rho$-distinguishability condition,

$$\left\|P_{\mathcal{A}}^{q_i} - P_{\mathcal{A}}^{q_j}\right\|_2 \geq \frac{\varepsilon_2 \varepsilon_{10}}{\rho}.$$

Appealing to Theorem 4.19, the number of samples required to achieve a confidence of $\delta_1$ is:

$$m_0 = \frac{8192\rho^2}{\varepsilon_2^2 \varepsilon_{10}^2} \log\left(\frac{1}{\delta_1}\right). \tag{4.18}$$

This expression corresponds to the sample complexity required for a single $L_2$ proximity test. For a complete (successful) run of the state merging algorithm, the required number of samples is therefore upper-bounded by $m_0 n^2 |\Sigma|(n|\Sigma| + 2)$, as shown in Section 4.5. Proceeding in an identical manner to Section 4.5, we use the expressions for $\delta_1$ and $\delta_2$ provided in Equations (4.10):

$$\delta_1 = \frac{\delta}{2n^2|\Sigma|(n|\Sigma| + 2)}, \qquad \delta_2 = \delta/2.$$

Running Algorithm 4.5 with the parameters $\nu$ and $m_0$ set to

$$\nu = \frac{\varepsilon_2 \varepsilon_{10}}{2\rho}, \qquad m_0 = \frac{8192\rho^2}{\varepsilon_2^2 \varepsilon_{10}^2} \log\left(\frac{1}{\delta_1}\right), \tag{4.19}$$

and appealing to Theorem 4.12, we obtain a confidence of $\delta$, while the algorithm's computational and sample complexities are polynomial in $\{n, |\Sigma|, \rho, \varepsilon^{-1}, \delta^{-1}\}$.

### Approximation Error Bound

We now bound the approximation error, using the decomposition of Carrasco [1997]:

$$\mathrm{KL}(\mathcal{A} \parallel \widehat{\mathcal{A}}) = \sum_{q \in \mathcal{A}} \sum_{\widehat{q} \in \widehat{\mathcal{A}}} \sum_{\sigma \in \Sigma} W(q, \widehat{q}) P_{\mathcal{A}}(q, \sigma) \log \frac{P_{\mathcal{A}}(q, \sigma)}{P_{\widehat{\mathcal{A}}}(\widehat{q}, \sigma)}.$$

Defining

$$D(q, \widehat{q}) = W(q, \widehat{q}) \sum_{\sigma \in \Sigma} P_{\mathcal{A}}(q, \sigma) \log \frac{P_{\mathcal{A}}(q, \sigma)}{P_{\widehat{\mathcal{A}}}(\widehat{q}, \sigma)},$$

we decompose the summation into four non-intersecting parts:

$$\begin{aligned}
D_1 &= \sum_{\substack{q\in Q_{\mathcal{A}}:\\ W(q)>\varepsilon_2}} \sum_{\substack{\widehat{q}\in Q_{\widehat{\mathcal{A}}}:\\ \Phi(q)=\widehat{q}}} D(q,\widehat{q}),\\
D_2 &= \sum_{\substack{q\in Q_{\mathcal{A}}:\\ W(q)>\varepsilon_2}} \sum_{\substack{\widehat{q}\in Q_{\widehat{\mathcal{A}}}:\\ \Phi(q)\neq\widehat{q},\\ \widehat{q}\in Z(q)}} D(q,\widehat{q}),\\
D_3 &= \sum_{\substack{q\in Q_{\mathcal{A}}:\\ W(q)>\varepsilon_2}} \sum_{\substack{\widehat{q}\in Q_{\widehat{\mathcal{A}}}:\\ \Phi(q)\neq\widehat{q},\\ \widehat{q}\notin Z(q)}} D(q,\widehat{q}),\\
D_4 &= \sum_{\substack{q\in Q_{\mathcal{A}}:\\ W(q)\leq\varepsilon_2}} \sum_{\widehat{q}\in Q_{\widehat{\mathcal{A}}}} D(q,\widehat{q}), \quad \text{such that}\\
\mathrm{KL}(\mathcal{A}\,\|\,\widehat{\mathcal{A}}) &= D_1+D_2+D_3+D_4.
\end{aligned}$$

The terms $D_1$, $D_3$ and $D_4$ above are analogous to $D_1$, $D_2$ and $D_3$ respectively in the analysis of Theorem B.10, and their respective bounds remain valid, as shown below.

Using Lemma B.8 and recalling that $W(q) \geq W(q,\widehat{q})$,

$$D_1 \leq \sum_{\substack{q\in Q_{\mathcal{A}}:\\ W(q)>\varepsilon_2}} W(q)\log(1+\varepsilon_4) \leq (L+1)\log(1+\varepsilon_4) \leq (L+1)\,\varepsilon_4\,.$$

For a state $q$ with $W(q) > \varepsilon_2$ for which $\Phi(q)$ is undefined, Lemma 4.29 ensures the existence of a (single) state $q' \in Z(q)$ such that $\Phi(q')$ is defined and $W(q') - W(q',\Phi(q')) \leq \varepsilon_3$. This situation corresponds to the elements in the sum $D_2$. Consider a pair $(q_i, q_j)$ in the sum $D_2$ above. Writing out the pair's contribution to the KL-divergence,

$$\begin{aligned}
&\sum_{\sigma\in\Sigma} W(q_i,\widehat{q}_j)P_{\mathcal{A}}(q_i,\sigma)\log\frac{P_{\mathcal{A}}(q_i,\sigma)}{P_{\widehat{\mathcal{A}}}(\widehat{q}_j,\sigma)}\\
&= W(q_i,\widehat{q}_j)\sum_{\sigma\in\Sigma} P_{\mathcal{A}}(q_i,\sigma)\log\frac{P_{\mathcal{A}}(q_i,\sigma)}{P_{\mathcal{A}}(q_j,\sigma)}\cdot\frac{P_{\mathcal{A}}(q_j,\sigma)}{P_{\widehat{\mathcal{A}}}(\widehat{q}_j,\sigma)}\\
(W(q_i,\widehat{q}_j \leq 1)) &\leq \sum_{\sigma\in\Sigma} P_{\mathcal{A}}(q_i,\sigma)\log\frac{P_{\mathcal{A}}(q_i,\sigma)}{P_{\mathcal{A}}(q_j,\sigma)}\\
&\quad +\sum_{\sigma\in\Sigma} P_{\mathcal{A}}(q_i,\sigma)\log\frac{P_{\mathcal{A}}(q_j,\sigma)}{P_{\widehat{\mathcal{A}}}(\widehat{q}_j,\sigma)}\\
(\text{by Lemma B.8}) &\leq \varepsilon_{10}+\sum_{\sigma\in\Sigma} P_{\mathcal{A}}(q_j,\sigma)\log(1+\varepsilon_4)\\
&\leq \varepsilon_{10}+\varepsilon_4\,.
\end{aligned}$$

The number of merges is upper-bounded by $n$, implying:

$$D_2 \leq n(\varepsilon_{10} + \varepsilon_4).$$

As a consequence of Lemma 4.29, $\sum_{\substack{\widehat{q}\in Q_{\widehat{\mathcal{A}}}:\\ \Phi(q)\neq\widehat{q}}} W(q,\widehat{q}) \leq \varepsilon_3$. Therefore,

$$\begin{aligned} D_3 &\leq \sum_{\substack{q\in Q_{\mathcal{A}}:\\ W(q)>\varepsilon_2}} \sum_{\substack{\widehat{q}\in Q_{\widehat{\mathcal{A}}}:\\ \Phi(q)\neq\widehat{q}}} W(q,\widehat{q}) \sum_{\sigma\in\Sigma} P_{\mathcal{A}}(q,\sigma) \log\frac{1}{p_{\min}} \\ &\leq \sum_{\substack{q\in Q_{\mathcal{A}}:\\ W(q)>\varepsilon_2}} \varepsilon_3 \log\frac{1}{p_{\min}} \leq n\,\varepsilon_3 \log\frac{1}{p_{\min}}. \end{aligned}$$

The sum $D_4$ is bounded using (4.7) and the condition on $W(q)$,

$$D_4 \leq \sum_{\substack{q\in Q_{\mathcal{A}}:\\ W(q)\leq\varepsilon_2}} \sum_{\widehat{q}\in Q_{\widehat{\mathcal{A}}}} W(q,\widehat{q}) \sum_{\sigma\in\Sigma} P_{\mathcal{A}}(q,\sigma) \log\frac{1}{p_{\min}} \leq n\,\varepsilon_2 \log\frac{1}{p_{\min}}.$$

Summing up the bounds, we have

$$\mathrm{KL}(\mathcal{A} \parallel \widehat{\mathcal{A}}) \leq D_1 + D_2 + D_3 + D_4 \leq (L+1)\,\varepsilon_4 + n(\varepsilon_{10} + \varepsilon_4) + n\,\varepsilon_3 \log\frac{1}{p_{\min}} + n\,\varepsilon_2 \log\frac{1}{p_{\min}}.$$

Setting

$$\varepsilon_4 = \varepsilon_{10} = \frac{\varepsilon}{2n(L+1)}$$

and using the existing values for all other $\varepsilon_i$, $i \in \{1,2,3,5,6\}$, we have $\mathrm{KL}(\mathcal{A} \parallel \widehat{\mathcal{A}}) \leq \varepsilon$, as desired. ■

The definition of $\rho$ can be varied in the following manners:

- Substituting $\|D^{q_i} - D^{q_j}\|_1$ in the numerator for $\mathrm{KL}\left(P_{\mathcal{A}}^{q_i} \parallel P_{\mathcal{A}}^{q_j}\right)$. In this case, the result becomes a (strict) generalization of the result of Palmer and Goldberg [2005], and the upper-bound condition ($L$) on the expected string length from each state can be lifted. We mention without proof that in this case, an $L_1$-PAC learnability result follows.

- Substituting $\left\|P_{\mathcal{A}}^{q_i} - P_{\mathcal{A}}^{q_j}\right\|_\infty$ for $\left\|P_{\mathcal{A}}^{q_i} - P_{\mathcal{A}}^{q_j}\right\|_2$ in the denominator, a (strict) generalization of the result in [Clark and Thollard, 2004] follows.

## 4.8 Discussion

In this chapter we presented a number of positive PDFA learnability results. We have discussed the relevance of PDFA learning frameworks, and the learnability of PDFA families using state merging algorithms. Figures 4.4 and 4.5 provide a graphical summary of our current understanding in these regards.

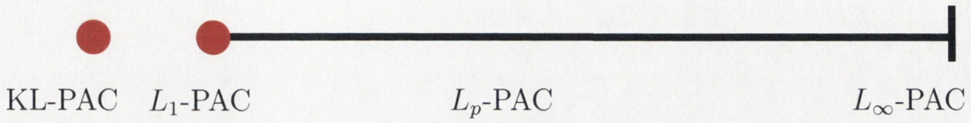

**Figure 4.4:** PAC-learning framework ordering displaying the relative difficulty and relevance of PDFA learning frameworks (hardest on left). In both the KL-PAC and the $L_1$-PAC frameworks, general PDFA learnability implies a contradiction to the noisy parity assumption. For $L_p$-PAC ($p > 1$), the reduction to noisy parity does not apply, and successful learning does not amount to "useful" information.

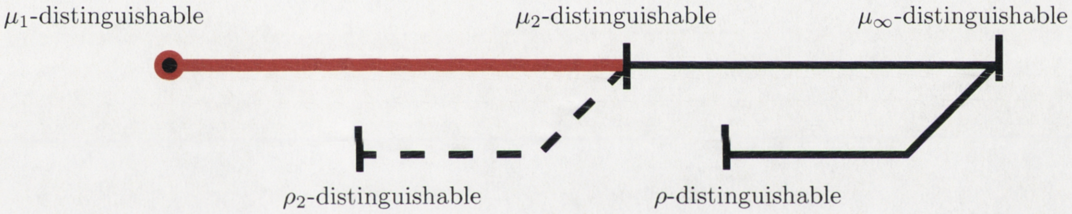

**Figure 4.5:** An ordering of distinguishability conditions (strongest on right). For $p \geq 2$, the $\mu_p$-distinguishability condition guarantees ($L_1$-PAC) learnability, as is the case for $\rho_2$-distinguishability (and for $\rho_\infty$-distinguishability by a straightforward extension). As implied by the noisy parity counterexample, $\mu_1$-distinguishability does not imply learnability, as is likely the case for $\mu_p$-distinguishability, $1 \leq p < 2$.

The general question of which PDFA subclasses are efficiently learnable (and which algorithm enables learnability) is not fully answered, but we feel the contributions in this chapter are a step in that direction.

A natural extension to the research presented in this chapter involves gaining an understanding of the behaviour of the SM algorithm when presented with samples drawn from non-PDFA distributions. This would be highly desirable theoretically, and may point the way to practical algorithmic modifications.

Chapter 5

# Sample Complexity of PFA and Discrete Distribution Learning

In this chapter we consider the sample complexity of PFA and discrete distribution learning. In contrast to the computational complexity, even the most general PFA learning problem has only polynomial sample complexity (as detailed in Section 5.2).

We introduce and formally define the problems (Section 5.1), and provide an overview of known results (Section 5.2). We establish a pair of *distribution dependent* upper and lower sample complexity bounds for (general) discrete distribution learning (Section 5.3). We then present two instructive examples which compare our novel sample complexity bounds to the existing state-of-the-art. We construct (Section 5.4) an example on which the memorizer algorithm provably fails to learn a distribution (i.e. has a sample complexity lower-bounded by an exponential function in the number of states), while the state merging algorithm provably succeeds (i.e. has polynomial sample and computational complexities).

## 5.1 Sample Complexity Frameworks and Notation

We will (mostly) adhere to notation introduced in previous chapters, with a number of additions. $D$ and $Q$ will denote probability distributions, while $\mathcal{D}$ and $\mathcal{Q}$ will denote classes of probability distributions. $\mathbb{A}$ will denote a finite discrete set of $a$ elements: $\mathbb{A} = \{1, \ldots, a\}$, and $\mathbb{F}$ will denote a countably infinite set.

A multiset $X = \{X_j\}_{j=1}^m$ will denote a collection of independent identically distributed random variables $X_j$ drawn from some arbitrary set. The multiset $S = \{s_j\}_{j=1}^m$ is a string multiset, i.e. $s_j \in \Sigma^*$.

Given a distribution $D$ over $\mathbb{F}$ and a sample multiset $X = \{X_j\}_{j=1}^m \in \mathbb{F}^m$, $X_j \sim D$, the empirical estimate (or *memorizer*) to $D$ is defined as:

$$\widehat{D}^m = \left\{\hat{d}_i^m\right\}_{i\in\mathbb{F}}, \qquad \hat{d}_i^m = \frac{1}{m}\sum_{j=1}^{m} 1_{\{X_j=i\}}\,. \tag{5.1}$$

We will also refer to the empirical estimate (5.1) as the *memorizer algorithm.*

We will use the following notion of learning for distributions:

**Definition 5.1 (Distribution learning)** *Let $D = \{(i, d_i)\}_{i\in\mathbb{F}}$ be an arbitrary probability distribution over $\mathbb{F}$:*

$$0 \le d_i \le 1 \quad \forall i \in \mathbb{F} \quad ; \quad \sum_{i\in\mathbb{F}} d_i = 1.$$

*For $1 \le p \le \infty$, an algorithm $T$ is said to $(\varepsilon, \delta, m_0)_p$-*learn *the distribution $D$ if for any $m \ge m_0$, given $m$ samples $T$ will output a distribution $\widehat{D}$ such that:*

$$\Pr\left\{\|\widehat{D} - D\|_p \ge \varepsilon\right\} \le \delta.$$

For the specific situation of PFA learning, the closely related notion of *sample complexity* is defined as:

**Definition 5.2 (Sample Complexity)** *Let $S = \{s_1, \ldots, s_m\}$, $s_i \in \Sigma^*$ denote a sample multiset, let $\mathcal{Q}$ denote a distribution class over $\Sigma^*$, and let $|\,\mathcal{Q}\,|$ denote some measure of complexity associated with the class $\mathcal{Q}$. For an arbitrary distribution $D$ over $\Sigma^*$, let $Q^{opt} \in \mathcal{Q}$ be a distribution satisfying either of the two conditions $(1 \le p \le \infty)$:*

$$KL(D \,\|\, Q^{opt}) = \min\{KL(D \,\|\, P) : P \in \mathcal{Q}\},$$
$$\|D - Q^{opt}\|_p = \min\{\|D - P\|_p : P \in \mathcal{Q}\}.$$

*An algorithm $T$* learns *$\mathcal{Q}$ if there exists some $M = f(\varepsilon^{-1}, \delta^{-1}, |\,\mathcal{Q}\,|)$ such that for an arbitrary distribution $D$ over $\Sigma^*$, if $m \ge M$ then with probability at least $1 - \delta$, $T$ outputs a distribution $Q \in \mathcal{Q}$ such that either of the following criteria holds:*

$$KL(D \,\|\, Q) - KL(D \,\|\, Q^{opt}) \le \varepsilon, \tag{5.2}$$
$$\|D - Q\|_p - \|D - Q^{opt}\|_p \le \varepsilon\,. \tag{5.3}$$

*When (5.2) is the chosen optimization criterion, the function $f(\cdot)$ defines the* KL sample complexity, *while for (5.3) it defines the $L_p$* sample complexity. *If the function $f(\cdot)$ is polynomial in all its arguments, the sample complexity is termed polynomial.*

## 5.2 Overview of Existing Results

Polynomial KL sample complexity for the class of distributions over $\Sigma^\ell$ defined by *non-terminating* PFA was shown by Abe and Warmuth [1992, Corollary 3.1]:

**Theorem 5.3** *The class of distributions over $\Sigma^\ell$ induced by n-state non-terminating PFA is learnable with KL sample complexity:*

$$O\left(\left(\frac{\ell}{\varepsilon}\right)^2 n^2|\Sigma| \cdot \log^3 \frac{\ell n|\Sigma|}{\varepsilon} \cdot \log \frac{1}{\delta} \cdot \log^2 \log \frac{1}{\delta}\right). \tag{5.4}$$

Theorem 5.3 shows that the PFA and PDFA learnability *hardness* results of Sections 4.1.1 and 4.3.1 are due to the *computational*, rather than the sample complexity aspects of the problem. Abe and Warmuth [1992] conjecture that the bound (5.4) may be loose, and discuss various possible sources of looseness which could potentially be tightened.

### 5.2.1 Discrete Distribution Learning Results

Weissman et al. [2003, Theorem 2.1] presented a distribution-dependent bound for the $L_1$ deviation between the true and empirical distributions. We now repeat the relevant part of their result. For $0 \le p_1, p_2 \le 1$, let $D_B(p_1 \parallel p_2)$ denote the *binary divergence*, defined as:

$$D_B(p_1 \parallel p_2) = p_1 \log \frac{p_1}{p_2} + (1 - p_1) \log \frac{1 - p_1}{1 - p_2}.$$

For $p \in (0, 1/2)$, define:

$$\varphi(p) = \frac{1}{1 - 2p} \log \frac{1 - p}{p},$$

setting $\varphi(1/2) = 2$. A graph of the function $\varphi(\cdot)$ is shown in Figure 5.1:

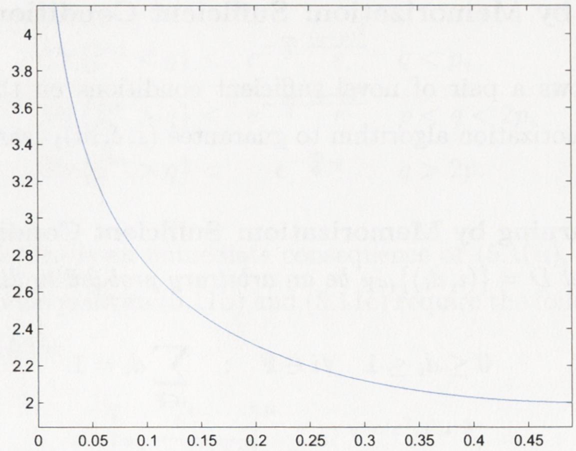

**Figure 5.1**: The function $\varphi(p)$ graphed over the interval $p \in [0, 1/2)$.

For a probability distribution $D$ on $\mathbb{A}$, define:

$$\pi_D = \max_{A \subseteq \mathbb{A}} \min\{D(A), 1 - D(A)\},$$

noting that $\pi_D \leq 1/2$ for all $D$.

**Theorem 5.4 (Weissman et al.)** *Let $D$ be a probability distribution on the finite set $\mathbb{A} = \{1, \ldots, a\}$. Let $X = \{X_1, \ldots, X_m\}$, $X_j \sim D$. Then for all $\varepsilon > 0$,*

$$\begin{aligned} \Pr\left\{\|D - \widehat{D}^m\|_1 \geq \varepsilon\right\} &\leq (2^a - 2)e^{-m[\min_{A \subseteq \mathbb{A}} D_B(D(A) + \frac{\varepsilon}{2} \| D(A))]} && \text{(5.5a)} \\ &\leq (2^a - 2)e^{-m\varphi(\pi_D)\,\varepsilon^2\,/4}. && \text{(5.5b)} \end{aligned}$$

Rearranging Inequality (5.5b), we arrive at the following *sufficient* condition on the number of samples guaranteeing $(\varepsilon, \delta, m)_1$-learnability:

$$m \geq \frac{4\left((\log_e 2) \cdot a + \log\left(\frac{1}{\delta}\right)\right)}{\varphi(\pi_D)\,\varepsilon^2}. \tag{5.6}$$

## 5.3 Sample Complexity of Learning Discrete Distributions

In this section we develop a novel *distribution dependent* sample complexity bound associated with memorization, and show that up to a logarithmic term, the bound is (asymptotically) *tight.* Therefore, the bound can serve as a *baseline* against which one can compare other results and algorithms.

### 5.3.1 Learning by Memorization: Sufficient Conditions

The next result shows a pair of novel sufficient conditions on the number of samples required by the memorization algorithm to guarantee $(\varepsilon, \delta, m)_1$-learnability:

**Theorem 5.5 (Learning by Memorization: Sufficient Conditions)** *Let $\mathbb{F}$ be a countable set, and let $D = \{(i, d_i)\}_{i \in \mathbb{F}}$ be an arbitrary probability distribution over $\mathbb{F}$:*

$$0 \leq d_i \leq 1 \quad \forall i \in \mathbb{F} \quad ; \quad \sum_{i \in \mathbb{F}} d_i = 1.$$

*Let $\varepsilon, \delta > 0$ be arbitrary precision and confidence parameters. Define the set $R \subseteq \mathbb{F}$ by:*

$$R = \min_{|V|} V \subseteq \mathbb{F} \quad s.t. \quad \sum_{i \in V} d_i \geq 1 - \varepsilon/2,$$

*set* $a = |R|$ *and* $d_{\min} = \min_{i \in R} d_i$. *Let*

$$\begin{aligned} m_1 &= 12 \log\left(\frac{a}{\delta}\right) \frac{\left(\sum_{i \in R} \sqrt{d_i}\right)^2}{\varepsilon^2}, && (5.7) \\ m_2 &= \frac{3}{d_{\min}} \log\left(\frac{1}{\delta}\right), && (5.8) \\ m_3 &= \frac{4 \left(\sum_{i \in R} \sqrt{d_i}\right)^2}{\delta^2 \, \varepsilon^2}. && (5.9) \end{aligned}$$

*If* $m \geq \min\{\max\{m_1, m_2\}, m_3\}$ *then the memorization algorithm is guaranteed to* $(\varepsilon, \delta, m)_1$*-learn the distribution:*

$$\Pr\left\{\|\widehat{D}^m - D\|_1 \geq \varepsilon\right\} \leq \delta.$$

In the theorem's proof, we will use the following form of Chernoff's bound [Motwani and Raghavan, 1995]:

**Theorem 5.6** *Let* $\{X_1, X_2, \ldots, X_m\}$ *be independent Bernoulli random variables with* $\Pr(X_i = 1) = p_i$, *let* $X = \sum_{i=1}^m X_i$, *and let* $\mu = \mathbb{E}(\sum_{i=1}^m X_i)$. *Then*

$$\begin{aligned} \Pr\{X < (1-\delta)\mu\} &< e^{-\mu\delta^2/2}, \quad 0 < \delta \leq 1, && (5.10a) \\ \Pr\{X > (1+\delta)\mu\} &< \left[\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right]^\mu, \quad \delta > 0. && (5.10b) \end{aligned}$$

If $p_i = p$, $i = 1, \ldots, m$, we have the following corollary:

**Corollary 5.7** *Let* $\{X_1, X_2, \ldots, X_m\}$ *be independent Bernoulli random variables with* $\Pr(X_i = 1) = p$, *and let* $\hat{p}^m = \frac{1}{m} \sum_{i=1}^m \mathbf{1}_{\{X_i = 1\}}$. *Then:*

$$\begin{aligned} \Pr\{\hat{p}^m < q\} &< e^{-\frac{m}{2} \cdot \frac{(q-p)^2}{p}} \quad q < p, && (5.11a) \\ \Pr\{\hat{p}^m > q\} &< e^{-\frac{m}{3} \cdot \frac{(q-p)^2}{p}} \quad p < q < 2p, && (5.11b) \\ \Pr\{\hat{p}^m > q\} &< e^{-\frac{m}{6} \cdot q} \quad q > 2p. && (5.11c) \end{aligned}$$

**Proof** Inequality (5.11a) is an immediate consequence of (5.10a), with the substitution $q = (1-\delta)p$. The two inequalities (5.11b) and (5.11c) require the following basic inequality [Hagerup and Rub, 1990]:

$$\left[\frac{e^\varepsilon}{(1+\varepsilon)^{1+\varepsilon}}\right]^\mu \leq e^{-\min(\varepsilon^2, \varepsilon)\mu/3}, \qquad (5.12)$$

Using (5.12) for $q < 2p$ and plugging $q = (1+\delta)p$ into (5.10b), we readily obtain (5.11b). Using the same technique for $q > 2p$, we obtain $\Pr(\hat{p}^m > q) < e^{-\frac{m}{3}(q-p)}$, which can trivially be relaxed to (5.11c). ■

**Proof (Theorem 5.5)**

For each $i \in R$, we set the number of samples to guarantee that with probability at least $1 - \delta_0$, the quantity $|d_i - \hat{d}_i^m|$ is not greater than $\sqrt{\left(\frac{3}{m}\right) d_i \log\left(\frac{1}{\delta_0}\right)}$. We then apply the union bound and show that with probability at least $1 - a\delta_0$, this bound applies to *all* $d_i$, $i \in R$.

Let $Z \sim B(p, m)$ be a binomially distributed random variable (*i.e.,* the sum of $m$ Bernoulli random variables, each with probability of success $p$) with $p < 0.5$, and let $X = \frac{1}{m}Z$ be its associated normalized random variable. It is well known (see e.g. [Bertsekas and Tsitsiklis, 2002]) that for binomial random variables, $\mathbb{E}(Z - \mathbb{E}\,Z)^2 = mp(1-p)$. It follows immediately that for the normalized variable $X = \frac{1}{m}Z$,

$$\mathbb{E}(X - \mathbb{E}\,X)^2 = \frac{p(1-p)}{m}.$$

The Chernoff bound (5.11b) for the probability $d_i$ gives

$$\Pr\left\{\hat{d}_i^m > q_i\right\} < e^{-\frac{m}{3}\cdot\frac{(q_i - d_i)^2}{d_i}}.$$

Equating the right-hand side of the inequality to $\delta_0$ and solving for $q_i$, we get:

$$q_i = d_i + \sqrt{\left(\frac{3}{m}\right) d_i \log\left(\frac{1}{\delta_0}\right)}.$$

The condition $q_i < 2d_i$ of (5.11b) now becomes $m > \frac{3}{d_i}\log\left(\frac{1}{\delta}\right)$, which is included in the theorem's statement as the condition on $m_2$, (5.8). Using (5.11a) we obtain an analogous expression for the situation $q_i < d_i$, namely

$$q_i = d_i - \sqrt{\left(\frac{2}{m}\right) d_i \log\left(\frac{1}{\delta_0}\right)}.$$

Collecting the inequalities, we have

$$\Pr\left\{|d_i - \hat{d}_i^m| \geq \sqrt{\left(\frac{3}{m}\right) d_i \log\left(\frac{1}{\delta_0}\right)}\right\} \leq \delta_0, \quad i \in R.$$

Applying the union bound over $d_i$, $i \in R$, $|R| = a$, we obtain

$$\Pr\left\{\sum_{i\in R} |d_i - \hat{d}_i^m| \geq \sum_{i\in R} \sqrt{\left(\frac{3}{m}\right) d_i \log\left(\frac{1}{\delta_0}\right)}\right\} \leq a\delta_0.$$

Setting

$$\delta = a\delta_0, \quad \sum_{i\in R}\sqrt{\left(\frac{3}{m}\right) d_i \log\left(\frac{a}{\delta}\right)} = \frac{\varepsilon}{2},$$

and solving for $m$, the theorem's condition on $m_1$ follows:

$$m_1 \geq 12\log\left(\frac{a}{\delta}\right)\frac{\left(\sum_{i\in R}\sqrt{d_i}\right)^2}{\varepsilon^2}.$$

Since $\|D - \widehat{D}^m\|_1 \leq \sum_{i\in R}|d_i - \hat{d}_i^m| + \frac{\varepsilon}{2}$, pulling the results above together we have established the first part of the theorem.

To obtain the condition on $m_3$ we bound the expectation $\mathbb{E}\,|X - \mathbb{E}\,X|$ using the Cauchy-Schwarz inequality:

$$\mathbb{E}\,|X - \mathbb{E}\,X| \leq \sqrt{\mathbb{E}\,(X - \mathbb{E}\,X)^2} = \sqrt{\frac{p(1-p)}{m}} \leq \sqrt{\frac{p}{m}}.$$

Therefore:

$$\mathbb{E}\left\|\widehat{D}^m - D\right\|_1 \leq \sum_{i\in R}\sqrt{\frac{d_i}{m}} + \frac{\varepsilon}{2}.$$

Using the Markov inequality, for all $k \in \mathbb{N}$:

$$\Pr\left\{\sum_{i\in R}\left|\hat{d}_i^m - d_i\right| \geq k\left(\sum_{i\in R}\sqrt{\frac{d_i}{m}}\right)\right\} \leq 1/k.$$

Setting $k = \lceil\frac{1}{\delta}\rceil$ and $m = 4\delta^{-2}\varepsilon^{-2}\left(\sum_{i\in R}\sqrt{d_i}\right)^2$, we get:

$$\Pr\left\{\sum_{i\in R}\left|\hat{d}_i^m - d_i\right| \geq \frac{\varepsilon}{2}\right\} \leq \delta, \text{ implying } \Pr\left\{\mathbb{E}\left\|\widehat{D}^m - D\right\|_1 \geq \varepsilon\right\} \leq \delta,$$

showing the condition on $m_3$ in (5.9) is sufficient. ■

### 5.3.2 Learning by Memorization: a Failure Mode

We now present a sufficient condition for *failure* of learning discrete distributions using memorization. Namely, we show that when the number of samples provided to the algorithm is upper-bounded, a non-approximability situation ensues. Up to a logarithmic term (or an inverse-square confidence term), the number of samples in this condition asymptotically matches that of Theorem 5.5, implying that the sum of probabilities' square roots is the key quantity controlling the difficulty of learning discrete distributions.

**Theorem 5.8 (Learning by Memorization: a Failure Mode)** *Let $\mathbb{F}$ be a countable set, let $D = \{(i, d_i)\}_{i \in \mathbb{F}}$ be an arbitrary probability distribution over $\mathbb{F}$, and assume $d_i \leq 1/2 \quad \forall i \in \mathbb{F}$. Let $\varepsilon > 0$ and define the set $R \subseteq \mathbb{F}$ by:*

$$R = \min_{|V|} V \subseteq \mathbb{F} \quad s.t. \quad \sum_{i \in V} d_i \geq 1 - \varepsilon/2.$$

*Let $d_{\min} = \min_{i \in R} d_i$. If*

$$\frac{20}{d_{\min}} < m \leq \frac{1}{640000} \cdot \frac{\left(\sum_{i \in R} \sqrt{d_i}\right)^2}{\varepsilon^2} \tag{5.13}$$

*then $\Pr\left\{\|\widehat{D}^m - D\|_1 \geq \varepsilon\right\} \geq 2\varepsilon$.*

The condition $m > 20/d_{\min}$ may seem counterintuitive, as it places a *lower bound* on the number of samples needed to establish a *non-approximability* result. However, the requirement is a result of the techniques used in proving the theorem. Specifically, our use of the Chernoff bound requires $m$ (at least) on the order of magnitude of $d_{\min}^{-1}$. It can be shown in a straightforward manner that for $m = o\left(d_{\min}^{-1}\right)$ (e.g. $m < 20/d_{\min}$), the approximation error for $d_{\min}$ is a constant fraction of $d_{\min}$, implying a relative error at least as large for all $d_i$, $i \notin R$, in turn guaranteeing a non-approximability result. We will not treat this case, however, as it does not illuminate any particular distribution dependent bound.

Before proving Theorem 5.8, we need two lemmas.

**Lemma 5.9 (Converse to Markov's inequality)** *Suppose $X$ is a nonnegative random variable bounded above by $B$: $0 \leq X \leq B$ and $\mu = \mathbb{E}X$. Then for all $0 \leq t \leq \mu$,*

$$\Pr\{X \geq t\} \geq \frac{\mu - t}{B}.$$

**Proof**

$$\begin{aligned} \mu &= \mathbb{E}\left(X\,\mathbf{1}_{\{0 \leq X \leq B\}}\right) = \mathbb{E}\left(X\,\mathbf{1}_{\{0 \leq X \leq t\}}\right) + \mathbb{E}\left(X\,\mathbf{1}_{\{t \leq X \leq B\}}\right) \\ &\leq t\Pr\{X \leq t\} + B\Pr\{X \geq t\} \leq t + B\Pr\{X \geq t\}. \end{aligned}$$

Rearranging terms, the lemma follows. ■

**Lemma 5.10** *Let $Z \sim B(p, m)$ be a binomially distributed random variable with $p < 0.5$, and let $X = \frac{1}{m}Z$ be its associated normalized random variable. Then if $m > 20/p$, there exist absolute constants $c_1 > 0$ and $c_2 > 0$ such that at least one of the following two* *inequalities holds:*

$$\mathbb{E}\left[(X-\mathbb{E}X)^2\,\mathbf{1}_{\left\{p+c_1\sqrt{\frac{p}{m}}\le X\le p+c_2\sqrt{\frac{p}{m}}\right\}}\right]\ge\frac{1}{4}\mathbb{E}(X-\mathbb{E}X)^2 \tag{5.14a}$$

$$\mathbb{E}\left[(X-\mathbb{E}X)^2\,\mathbf{1}_{\left\{p-c_2\sqrt{\frac{p}{m}}\le X\le p-c_1\sqrt{\frac{p}{m}}\right\}}\right]\ge\frac{1}{4}\mathbb{E}(X-\mathbb{E}X)^2 \tag{5.14b}$$

**Proof** We show that for specific values of $c_1$ and $c_2$, the second moment's contribution in each of the intervals $\left[0,p-c_2\sqrt{\frac{p}{m}}\right]$, $\left[p-c_1\sqrt{\frac{p}{m}},p\right]$, $\left[p,p+c_1\sqrt{\frac{p}{m}}\right]$ and $\left[p+c_2\sqrt{\frac{p}{m}},\infty\right)$ amounts to not more than $\frac{p(1-p)}{8m}$, *i.e.*, less than a constant fraction of $\mathbb{E}(X-\mathbb{E}X)^2$. We prove the assertion for the interval $\left[p,p+c_1\sqrt{\frac{p}{m}}\right]$, and the result follows for the interval $\left[p-c_1\sqrt{\frac{p}{m}},p\right]$ using the exact same reasoning (since $\mathbb{E}X=p$):

$$\mathbb{E}\left[(X-\mathbb{E}X)^2\,\mathbf{1}_{\left\{p\le X\le p+c_1\sqrt{\frac{p}{m}}\right\}}\right]\le\frac{c_1^2p}{m}\Pr\left\{p\le X\le p+c_1\sqrt{\frac{p}{m}}\right\}\le\frac{c_1^2p}{m}.$$

Since $p<0.5$ by assumption, choosing $c_1=0.25$ leads to:

$$\frac{c_1^2p}{m}<\frac{p(1-p)}{8m}.$$

We proceed to bound the contribution of the tail $\left[p+c_2\sqrt{\frac{p}{m}},\infty\right)$, using the Chernoff bounds (5.11b) and (5.11c). We first deal with the interval $\left[p+c_2\sqrt{\frac{p}{m}},2p\right]$:

$$\begin{aligned}
&\mathbb{E}\left[(X-\mathbb{E}X)^2\,\mathbf{1}_{\left\{p+c_2\sqrt{\frac{p}{m}}\le X<2p\right\}}\right]\\
<\;&\mathbb{E}\left[(X-\mathbb{E}X)^2\left(\sum_{n=1}^{\frac{\sqrt{4mp}}{c_2}}\mathbf{1}_{\left\{p+nc_2\sqrt{\frac{p}{m}}<X<p+(n+1)c_2\sqrt{\frac{p}{m}}\right\}}\right)\right]\\
\le\;&\sum_{n=1}^{\frac{\sqrt{4mp}}{c_2}}\frac{((n+1)c_2)^2p}{m}\mathbb{E}\left[\mathbf{1}_{\left\{p+nc_2\sqrt{\frac{p}{m}}<X<p+(n+1)c_2\sqrt{\frac{p}{m}}\right\}}\right]\\
<\;&\frac{pc_2^2}{m}\sum_{n=1}^{\frac{\sqrt{4mp}}{c_2}}(n+1)^2\Pr\left\{X>p+nc_2\sqrt{\frac{p}{m}}\right\}.
\end{aligned}$$

Using inequality (5.11b), we have $\Pr\left\{X > p + nc_2\sqrt{\frac{p}{m}}\right\} < e^{-\frac{(nc_2)^2}{3}}$. Plugging this back into the tail calculation above:

$$\begin{aligned}
&\mathbb{E}\left[(X - \mathbb{E}X)^2 \mathbf{1}_{\left\{p+c_2\sqrt{\frac{p}{m}} \le X < 2p\right\}}\right] \\
&< \frac{pc_2^2}{m} \sum_{n=1}^{\frac{\sqrt{4mp}}{c_2}} (n+1)^2 e^{-\frac{(nc_2)^2}{3}} \\
&< \frac{pc_2^2}{m} \sum_{n=1}^{\infty} (n+1)^2 e^{-\frac{(nc_2)^2}{3}}.
\end{aligned}$$

In this infinite series the polynomial term is dominated by the decaying exponential. For $c_2 = 5$, we obtain by a straightforward calculation:

$$c_2^2 \sum_{n=1}^{\infty} (n+1)^2 e^{-\frac{(nc_2)^2}{3}} < 1/32,$$

giving the bound:

$$\mathbb{E}\left[(X - \mathbb{E}X)^2 \ \mathbf{1}_{\left\{p+c_2\sqrt{\frac{p}{m}} \le X < 2p\right\}}\right] < \frac{p}{32m} < \frac{p(1-p)}{16m}. \tag{5.15}$$

The exact same argument (using Chernoff's bound (5.11a)) also applies to the interval $\left[0, p - c_2\sqrt{\frac{p}{m}}\right]$.

Next, we need to bound $\mathbb{E}\left[(X - \mathbb{E}X)^2 \ \mathbf{1}_{\{2p \le X < \infty\}}\right]$. To this end, we employ inequality (5.11c) and proceed in a similar manner:

$$\begin{aligned}
&\mathbb{E}\left[(X - \mathbb{E}X)^2 \ \mathbf{1}_{\{2p \le X < \infty\}}\right] \\
&< \sum_{n=2}^{\infty} (n+1)^2 p^2 e^{-\frac{mnp}{6}} \qquad (\text{define } c_0 = e^{-\frac{mp}{6}}) \\
&= p^2 \sum_{n=2}^{\infty} (n+1)^2 c_0^n \\
&= p^2 \left[\frac{9c_0^2}{1-c_0} + \frac{7c_0^3}{(1-c_0)^2} + \frac{2c_0^4}{(1-c_0)^3}\right] \qquad (\text{by a standard infinite series sum}).
\end{aligned}$$

Performing the arithmetic, we see that for $m > \frac{20}{p}$:

$$\mathbb{E}\left[(X - \mathbb{E}X)^2 \ \mathbf{1}_{\{2p \le X < \infty\}}\right] \le \frac{p}{32m} \le \frac{p(1-p)}{16m}. \tag{5.16}$$

Collecting (5.15) and (5.16), we have the lemma. ■

**Proof** (of Theorem 5.8)

Using Lemma 5.10 and assuming without loss of generality that of the two inequalities (5.14) inequality (5.14a) holds, we have (conditioned on $m > \frac{20}{p}$):

$$\begin{aligned} &\frac{1}{4}\mathbb{E}(X-\mathbb{E}X)^2 \\ &\le \mathbb{E}\left[(X-\mathbb{E}X)^2\,\mathbf{1}_{\{p+c_1\sqrt{\frac{p}{m}}\le X\le p+c_2\sqrt{\frac{p}{m}}\}}\right] \\ &\le \frac{c_2^2 p}{m}\Pr\left\{p+c_1\sqrt{\frac{p}{m}} < X < p+c_2\sqrt{\frac{p}{m}}\right\}. \end{aligned} \tag{5.17}$$

On the other hand:

$$\begin{aligned} &\mathbb{E}\left|X-\mathbb{E}X\right| \\ &\ge\ \mathbb{E}\left[\left|X-\mathbb{E}X\right|\mathbf{1}_{\{p+c_1\sqrt{\frac{p}{m}}\le X\le p+c_2\sqrt{\frac{p}{m}}\}}\right] \\ &\ge\ c_1\sqrt{\frac{p}{m}}\Pr\left\{p+c_1\sqrt{\frac{p}{m}} < X < p+c_2\sqrt{\frac{p}{m}}\right\}. \end{aligned}$$

Combining the results, we have:

$$\begin{aligned} &\mathbb{E}\left|X-\mathbb{E}X\right| \\ &\ge\ \frac{c_1}{4c_2^2}\sqrt{\frac{m}{p}}\,\mathbb{E}(X-\mathbb{E}X)^2 \\ &=\ \frac{c_1}{4c_2^2}\sqrt{\frac{p}{m}}(1-p) \ge \frac{c_1}{4c_2^2}\sqrt{\frac{p}{m}}. \end{aligned}$$

Applying this inequality to the set of probabilities $\{d_i\}_{i\in R}$, we get:

$$\mathbb{E}\left\|\widehat{D}^m-\tilde{D}\right\|_1 \ge \sum_{i\in R}\mathbb{E}\left|\hat{d}_i^m-d_i\right| \ge \frac{c_1}{4c_2^2\sqrt{m}}\sum_{i\in R}\sqrt{d_i}.$$

Defining $\mu_{\min} = \frac{c_1}{4c_2^2\sqrt{m}}\sum_{i\in R}\sqrt{d_i}$ and using Lemma 5.9 leads to:

$$\Pr\left\{\left\|\widehat{D}^m-D\right\|_1 \ge t\right\} \ge \frac{1}{2}\left(\mu_{\min}-t\right), \qquad \forall t<\mu_{\min}.$$

Setting the number of samples to $m = \frac{c_1^2}{64c_2^4}\cdot\frac{\left(\sum_{i\in R}\sqrt{d_i}\right)^2}{\varepsilon^2}$, using the values of $c_1$ and $c_2$ determined above and plugging in $t = \mu_{\min}/2$, we have proved Theorem 5.8. ■

An inspection of the first sufficient condition (5.7) and the failure conditions (5.13) shows they asymptotically match, up to a factor $O\left(\log\left(\frac{a}{\varepsilon}\right)\right)$. The second sufficient condition (5.9) asymptotically matches (5.13) up to a factor $O\left(\delta^{-2}\right)$.

### 5.3.3 Critical Comparison Between Results

In this section we carry out a critical comparison between our results and the results of Weissman et al. [2003]. We present two instructive examples which serve to highlight the differences between the bounds of Theorems 5.4 and 5.5. We then end the chapter with a discussion of the results' significance.

**Example 5.1**

We compare the bounds (5.5b) and (5.7) for the uniform distribution over the finite set $\mathbb{A} = \{1, \dots, a\}$, i.e. $\{p_i\}_{i=1}^{a} = 1/a$. We perform no comparison to the bound (5.9), as it includes a $\delta^{-2}$ term that cannot directly be compared to the bound (5.5b).

In this case, assuming $a$ is an even number, we have $\pi_P = 1/2$ and therefore $\varphi(\pi_P) = 2$ (in the case of $a$ odd, $\varphi(\pi_P)$ may become slightly larger, but still bounded by 2.5 for all $a > 5$). Inequality (5.5b) therefore implies the following sufficient condition for $(\varepsilon, \delta, m)_1$-learnability:

$$m \geq \frac{2\log 2 \cdot a + \log\left(\frac{1}{\delta}\right)}{\varepsilon^2}. \tag{5.18}$$

The sufficient condition implied by (5.7) is:

$$m \geq \frac{12a\log\left(\frac{a}{\delta}\right)}{\varepsilon^2} = \frac{12a\log a + 12a\log\left(\frac{1}{\delta}\right)}{\varepsilon^2}. \tag{5.19}$$

In this example, the bound implied by (5.18) is tighter, by an asymptotic term of $\log a$. Our second example portrays a converse situation.

**Example 5.2**

Let $\varepsilon_0 < \varepsilon$, and consider a distribution with a small number of large probabilities spread uniformly about $k$ of the $a$ elements of $\mathbb{A}$, with small probabilities on all other elements:

$$p_i = \begin{cases} \frac{1}{k} - \frac{\varepsilon_0}{a-k} & i = \ 1, \dots, k \\ \frac{\varepsilon_0}{a-k} & i = \ k+1, \dots, a. \end{cases}$$

Assuming an even $k$, Inequality (5.5b) implies the same sufficient condition as in Example 5.1, namely

$$m \geq \frac{2\log 2 \cdot a + \log\left(\frac{1}{\delta}\right)}{\varepsilon^2}. \tag{5.20}$$

As $\varepsilon_0 < \varepsilon$, Inequality (5.7) implies the sufficient condition

$$m \geq \frac{12k\log\left(\frac{a}{\delta}\right)}{\varepsilon^2}. \tag{5.21}$$

Holding the parameter $\delta$ fixed and assuming $k = o\left(\frac{a}{\log a}\right)$, the condition of Inequality (5.21) is tighter than that of (5.20) by an asymptotic multiplicative term of $\frac{a}{k \log a}$.

The two examples illustrate that for "flat" distributions, the condition of Inequality (5.5b) is tighter, while for even moderately "peaked" distributions, i.e. where a large proportion of the probability mass is concentrated on a subset of cardinality $k = o\left(\frac{a}{\log a}\right)$, Inequality (5.7) is tighter.

We mention that even in case (the tighter) Inequality (5.5a) is used, an immediate calculation of $D_B\left(\frac{1}{2} + \frac{\varepsilon}{2} \parallel \frac{1}{2}\right)$ shows that the same asymptotic result, leading to the same conclusions.

## 5.4 Sample Complexity of Learning $\mu$-Distinguishable PDFA

We now construct a family of PDFA for which learning by memorization provably requires an exponential number of samples (in the number of states), while the sample complexity associated with state merging remains polynomial (in the number of states). Stated formally, we prove the following theorem

**Theorem 5.11** *There exists a family of PDFA $\{A_n\}_{n=1}^{\infty}$ such that $A_n$ has $\Theta(n)$ states, the sample and computational complexities of learning $A_n$ using state merging are polynomial in $n$, but the sufficient condition for a failure mode using memorization (Theorem 5.8) is exponential in $n$.*

The theorem's proof is constructive and uses the family depicted in Figure 5.2.

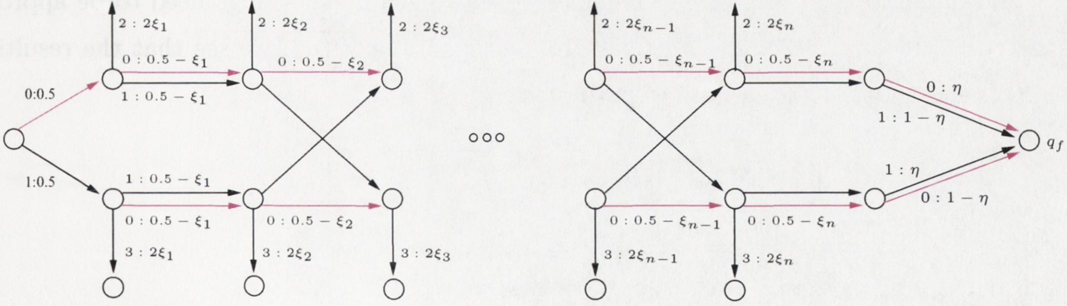

**Figure 5.2:** A modification of the noisy parity PDFA family for which the inverse distinguishability ($\mu^{-1}$) is a *polynomial* function of $n$ (as opposed to exponential, as in the "original" family in Figure 4.1). The alphabet has been augmented to include the symbols $\{2, 3\}$, which serve to "distinguish" between the upper and lower tracks. The parameters $\xi_i = \frac{1}{2(n+i)}$.

We prove two lemmas, the conjunction of which implies Theorem 5.11:

**Lemma 5.12** *The computational and sample complexities of learning $A_n$ via state merging are polynomial in $n$.*

**Proof** Let $A_n$ be an $n$-layer member of the PDFA family depicted in Figure 5.2. $A_n$ uses the alphabet $\Sigma = \{0,1,2,3\}$, and the parameters $\{\xi_i\}_{i=1}^{n}$ are set to $\xi_i = \frac{1}{2(n+i)}$. We will calculate $A_n$'s distinguishability parameter ($\mu$) and appeal to Theorem 4.9 to establish polynomial computational and sample complexities.

From the upper and lower matching states on the $i^{th}$ layer, the probabilities of emitting the (single character) string $x = \langle 2 \rangle$ are $2\xi_i = 1/(n+i)$ and 0 respectively. Thus, the distinguishability between these state pairs is $1/(n+i)$. Between any two top (or bottom) layer pairs, using similar reasoning, the distinguishability is lower-bounded by $1/(4n^2)$. Thus, $A_n$ is $(\frac{1}{4n^2})$-distinguishable. Plugging into Theorem 4.9, we see that the computational complexity of learning $A_n$ is polynomial in $n$. ■

**Lemma 5.13** *The sample complexity of learning $A_n$ using memorization is lower-bounded by an exponential in $n$.*

**Proof** For a given $n$, we show that the probability of reaching the rightmost state $q_f$ (see Figure 5.2) is $1/2$, and therefore the set of length $n+2$ strings has a cumulative probability of $1/2$:

$$\begin{aligned}\Pr_{x\sim A}\{x \text{ reaches } q_f\} &= \left(1-\frac{1}{n+1}\right)\cdot\left(1-\frac{1}{n+2}\right)\cdots\left(1-\frac{1}{2n}\right)\\ &= \left(\frac{n}{n+1}\right)\left(\frac{n+1}{n+2}\right)\cdots\left(\frac{2n-1}{2n}\right)=1/2.\end{aligned}$$

Assuming (e.g.) $\varepsilon < 1/4$, at least half of the length $n+2$ strings need to be approximated. Calculating the sum of square roots $\sum_{w\in\Sigma^{n+2}}\sqrt{D(w)}$, we see that the resulting expression grows exponentially with $n$:

$$\sum_{w\in\Sigma^{n+2}}\sqrt{D(w)} = 2^n\cdot\sqrt{\frac{1}{2}\cdot\frac{1}{2}\cdot 2^{-n}} = 2^{\frac{n}{2}-2}.$$

Appealing to Theorem 5.8, the result follows. ■

## 5.5 Discussion

We have established a novel pair of sample complexity conditions for success and failure modes of the memorization algorithm. The conditions asymptotically match up to a logarithmic factor. Together, the two results can serve as a baseline against which any distribution learning algorithm can be compared.

Using these "memorization" bounds, we constructed an example showing that when provided with the knowledge that a distribution can be generated by a PDFA, the computational and sample complexities can be reduced from exponential to polynomial.

The PFA sample complexity result of [Abe and Warmuth, 1992] mentioned in Section 5.2 may be loose. An analysis using more modern tools of probability may yield a tighter result. Such an analysis would make an interesting avenue for further research.

Chapter 6

# Conclusions

In this thesis we have studied a number of problems relating to probabilistic automata, distributions of sequences over discrete alphabets, the links between and the learnability thereof. By studying the geometry of PFA models we have obtained insights into the factors which conspire to make the PFA learning problem hard.

Owing in part to the PFA extension to the Myhill-Nerode theorem presented in Chapter 3, the distribution families induced by PFA, PRFA and PDFA models have been fully characterized. These insights have enabled us to place bounds on the PFA and PDFA models' ability to approximate arbitrary distributions over bounded-length strings.

Although still incomplete, our understanding of the sample complexity of PFA learning is at a more advanced stage than the problem's computational complexity. Indeed, as a number of hardness results show, the general PFA learning problem's richness is a consequence of purely computational difficulties.

The field of probabilistic automata learning remains a rich field with many open questions that are interesting from the theoretical as well as practical perspective. The rich structure offered by probabilistic automata models belies their practical attractiveness, but presents great difficulties in their inference. We feel that despite the considerable effort spent on the theoretical study of PFA learning, the most practically relevant contributions to the field remain heuristic.

## 6.1 Novel Contributions

Our novel contributions in this thesis include the following:

- A PFA extension to the classical Myhill-Nerode theorem (Section 3.2), and an application thereof, bounding the PFA and PDFA's approximation ability of general distributions over bounded-length strings (Section 3.4).
- An extension of a negative PDFA learnability result (the noisy parity reduction) from the KL-PAC to the $L_1$-PAC learning framework (Section 4.4).
- A decomposition of the state merging algorithm and its analysis (Section 4.5).

- A tight analysis and an accompanying algorithm for $L_2$-proximity testing between discrete distributions, leading to a positive ($\mu_2$-distinguishable PDFA) learnability result (Section 4.6).

- A relaxed condition and an accompanying analysis proving efficient learnability of $\rho$-distinguishable PDFA using the state merging algorithm (Section 4.7).

- A pair of *distribution dependent* upper and lower sample complexity bounds for (general) discrete distribution learning (Section 5.3). The bounds asymptotically match up to a logarithmic factor.

## 6.2 Ideas for Future Research

Throughout the thesis we have pointed out a number of open questions which we feel are important. These include the following:

- The construction of distribution families which enable efficient learnability (and evaluation) and which also provide good approximation to (interesting subsets of) PFA. This question is intimately related to the problem of determining which PFA subfamilies are efficiently learnable.

- An understanding of how well (and under which circumstances) PDFA models can approximate distributions induced by PFA models. This problem was addressed in previous research, but is still not well understood.

- An understanding of the state merging algorithm's behaviour when presented with samples taken from non-PDFA distributions is still lacking. Such an understanding may also be desirable from the practical perspective.

- Tight sample complexity bounds for the PFA and PDFA learning.

- The relatively simple case of PFA learning using a finite alphabet is still not well understood. The analysis of PFA learning with continuous, high dimensional alphabets, while theoretically daunting, may lead to novel insights as well as practically appealing algorithms.

Appendix A

# Proofs of Technical Lemmas

## A.1 Proof of Lemma 3.16

We follow a two step approach. The first step is a reduction to an equivalent problem. The second step shows a bound on the equivalent problem, proving the inequality.

We start by partitioning the summands $d_1t_1, \ldots, d_Nt_N$ into two mutually exclusive sets $I_1, I_2$ according to the following rule:

$$\begin{aligned} i \in I_1 \quad &\text{if} \quad d_it_i \leq \frac{1}{N}, \\ i \in I_2 \quad &\text{if} \quad d_it_i > \frac{1}{N}. \end{aligned}$$

Adding and subtracting $\sum_{i \in I_2}(1/N - d_it_i)$, we get:

$$\sum_{i=1}^{N} |1/N - d_it_i| = 1 - \sum_{i=1}^{N} d_it_i + 2\sum_{i \in I_2} \left(d_it_i - \frac{1}{N}\right).$$

On the right hand side of the equality above, any term with $d_it_i > 1/N$ is "doubly penalized" by the last sum. Indeed, we now show that given a specific pair $\{D, T\}$ we can construct an alternative pair $\{\widehat{D}, \widehat{T}\}$ such that $\widehat{d_i}\widehat{t_i} \leq 1/N$ for all $i$, and $\sum_{i=1}^{N} |1/N - \widehat{d_i}\widehat{t_i}| \leq \sum_{i=1}^{N} |1/N - d_it_i|$. The constructive proof is detailed in algorithm 6.

To prove the algorithm's correctness we note that no operation performed over the algorithm's run will increase $\sum_{i=1}^{N} |1/N - d_it_i|$, and that the algorithm terminates after a finite number of steps.

If at any stage of the while loop of line 1 a suitable $i$ is found, the existence of a suitable $j$ in line 2 is assured ($d_it_i > 1/N \;\Rightarrow\; d_i > 1/N \;\Rightarrow \exists d_j < 1/N$). The number of steps performed is bounded by the number of pairs $(i, j)$ and is therefore finite. In all reassignment operations, the values of both $|1/N - d_it_i|$ and $|1/N - d_jt_j|$ are not increased, assuring that upon termination we have $\sum_{i=1}^{N} |1/N - \widehat{d_i}\widehat{t_i}| \leq \sum_{i=1}^{N} |1/N - d_it_i|$. We have

**Algorithm A.1**: Reduction to a Uniformly Bounded Distribution $\widehat{d}$

**Input**: A distribution $D = \{d_1, \ldots, d_N\}$ and a set $T = \{t_1, \ldots, t_N\}$ with $0 \leq t_i \leq 1$.
**Output**: A distribution $\widehat{d}$ and a set $\widehat{t}$ with $0 \leq \widehat{t_i} \leq 1$, $\sum_{i=1}^{N} \widehat{t_i} \leq \sum_{i=1}^{N} t_i$, such that $\widehat{d_i}\widehat{t_i} \leq 1/N$ for all $i$, and $\sum_{i=1}^{N} |1/N - \widehat{d_i}\widehat{t_i}| \leq \sum_{i=1}^{N} |1/N - d_i t_i|$.

```
while $\exists i \in \{1, \ldots, N\}$ such that $d_i t_i > 1/N$ do
    Find $j \in \{1, \ldots, N\}$ such that $d_j < 1/N$
    if $d_i t_i - 1/N \geq 1/N - d_j$ then
        Set $d_i = d_i - (1/N - d_j)$
        Set $d_j = 1/N$
    else
        Set $d_i = 1/N$
        Set $d_j = d_j - (1/N - d_i t_i)$
    end
end
Set $\widehat{d} = D, \widehat{t} = T$.
```

thus shown that:

$$\min_{\{D,T\}} \sum_{i=1}^{N} |1/N - d_i t_i| = \min_{\{D,T\}} \sum_{i=1}^{N} (1/N - d_i t_i) \quad \text{s.t.} \quad d_i t_i \leq 1/N \quad \forall i.$$

We now examine how the selections of $d_1$ and $t_1$ affect both $(1/N - d_1 t_1)$ and $\sum_{i=2}^{N}(1/N - d_i t_i)$, under the condition $d_i t_i \leq 1/N$. If $d_1 = 1/N - \varepsilon$, we immediately have $1/N - d_1 t_1 \geq \varepsilon$, due to $t_1 \leq 1$. If $d_1 = 1/N + \varepsilon$ we have:

$$\sum_{i=2}^{N} (1/N - d_i t_i) \geq \sum_{i=2}^{N} (1/N - d_i) \geq \sum_{i=2}^{N} 1/N - [(N-1)/N - \varepsilon] = \varepsilon.$$

Writing $t_1$ as $1 - \delta_1$, $0 \leq \delta_1 \leq 1$, we again examine the effect of the choice on $IP_1 = (1/N - d_1 t_1)$ (denoting "immediate penalty") and on $SP_1 = \sum_{i=2}^{N} (1/N - d_i t_i)$ (denoting "subsequent penalty"):

$$\begin{aligned} IP_1 &= 1/N - d_1 t_1 = 1/N - d_1(1 - \delta_1), \\ SP_1 &= \sum_{i=2}^{N} (1/N - d_i t_i) \geq \max\{d_1 - 1/N,\ 0\}. \end{aligned}$$

We can now deduce $IP_1 + SP_1 \geq \delta_1/N$ by considering the following cases:

$$\begin{cases} d_1 < 1/N & IP_1 > \delta_1/N. \\ 1/N \leq d_1 \leq \frac{1}{(1-\delta_1)N} & SP_1 \geq d_1 - 1/N \\ & \Rightarrow IP_1 + SP_1 \geq 1/N - d_1 t_1 + d_1 - 1/N \\ & = d_1(1-t_1) \geq \delta_1/N. \\ d_1 > \frac{1}{(1-\delta_1)N} & \text{Violates the constraint } t_1 d_1 \leq 1/N. \end{cases}$$

Summing over $i = 1, \ldots, N$ we get:

$$\sum_{i=1}^{N}(1/N - d_i t_i) = \sum_{i=1}^{N} IP_i + SP_i \geq \frac{1}{N}\sum_{i=1}^{N}\delta_i = \frac{1}{N}\sum_{i=1}^{N}(1-t_i) = 1 - \frac{1}{N}\sum_{i=1}^{N} t_i,$$

proving the lemma. ■

## A.2 Proof of Lemma 3.17

The $d$-dimensional probability simplex, $\Delta^d$, is a subset of the $\ell_1^d$ sphere, which we denote by $B_1^d$. The probability simplex consists of the positive quadrant of $B_1^d$, and therefore the following volume relation holds:

$$\text{vol}(\Delta^d) = \frac{\text{vol}(B_1^d)}{2^d}. \tag{A.1}$$

Let $A$ be a *maximal packing* of $\varepsilon$-volume $\ell_1^d$ balls in $\Delta^d$. Due to $A$'s maximality it is also a cover of $B_1^d$ (otherwise we would be able to fit in an additional $\varepsilon$-volume ball). We therefore have:

$$\begin{aligned} \Delta^d &\subseteq \bigcup_{a\in A} a + \varepsilon B_1^d, \text{ implying} \\ \text{vol}(\Delta^d) &\leq \text{vol}\left(\bigcup_{a\in A} a + \varepsilon B_1^d\right) \leq \sum_{a\in A}\text{vol}(a + \varepsilon B_1^d) = |A|\,\varepsilon^d \text{vol}(B_1^d). \end{aligned}$$

Plugging in (A.1), we have:

$$|A| \geq \left(\frac{1}{2\varepsilon}\right)^d.$$

## A.3 Proof of Lemma 4.20

Let $\{Y_i\}_{i=1}^m$ be i.i.d. random variables distributed identically to the random variables $X_i$. The Rademacher random variables $\{\varepsilon_i\}_{i=1}^m$ are symmetric binary valued, so for the random variable $X_i - Y_i$ we have:

$$X_i - Y_i \sim \varepsilon_i(X_i - Y_i).$$

Therefore,

$$\begin{aligned}
\mathbb{E}\left|\tfrac{1}{m}\textstyle\sum_{i=1}^m X_i - \mathbb{E}\,X\right| &= \mathbb{E}_X\left|\frac{1}{m}\sum_{i=1}^m X_i - \mathbb{E}_Y\,\frac{1}{m}\sum_{i=1}^m Y_i\right| \\
\text{(by the triangle inequality)} &\le \mathbb{E}_X\,\mathbb{E}_Y\left|\frac{1}{m}\sum_{i=1}^m (X_i - Y_i)\right| \\
&= \mathbb{E}_X\,\mathbb{E}_Y\left|\frac{1}{m}\sum_{i=1}^m \varepsilon_i(X_i - Y_i)\right|.
\end{aligned}$$

The inequality holds for all selections $\{\varepsilon_i\}_{i=1}^m$, and therefore holds also when taking the expectation with respect to $\{\varepsilon_i\}_{i=1}^m$:

$$\begin{aligned}
& \mathbb{E}_X\,\mathbb{E}_Y\left|\frac{1}{m}\sum_{i=1}^m \varepsilon_i(X_i - Y_i)\right| \\
& = \mathbb{E}_\varepsilon\,\mathbb{E}_X\,\mathbb{E}_Y\left|\frac{1}{m}\sum_{i=1}^m \varepsilon_i(X_i - Y_i)\right| \\
\text{(by Fubini's theorem)}\quad & = \mathbb{E}_X\,\mathbb{E}_Y\,\mathbb{E}_\varepsilon\left|\frac{1}{m}\sum_{i=1}^m \varepsilon_i(X_i - Y_i)\right| \\
\text{(by the triangle inequality)}\quad & \le \mathbb{E}_X\,\mathbb{E}_Y\,\mathbb{E}_\varepsilon\left(\left|\frac{1}{m}\sum_{i=1}^m \varepsilon_i X_i\right| + \left|\frac{1}{m}\sum_{i=1}^m \varepsilon_i Y_i\right|\right) \\
& = \frac{2}{m}\,\mathbb{E}_X\,\mathbb{E}_\varepsilon\left|\sum_{i=1}^m \varepsilon_i X_i\right|.
\end{aligned}$$

Appendix B

# Excerpts from [Clark and Thollard, 2004]

In order to make the document self-contained, we repeat a number of results from [Clark and Thollard, 2004] which are used in the thesis. In order to enhance clarity, we have translated the original notation to that used throughout the thesis.

## B.1 Definitions

The following definitions of Clark and Thollard [2004] have been modified in the thesis. We repeat the original definitions' statements using our notation, pointing to the relevant definitions in the paper.

**Definition B.1** *[Clark and Thollard, 2004, Definition 4]*
A candidate node *is a pair $(u, \sigma)$ where $u$ is a node in the graph and $\sigma \in \Sigma$, where $\delta_G(u, \sigma)$ is undefined. It will have an associated multiset $S_{u,\sigma}$. A candidate node $(u, \sigma)$ and a node $v$ in a hypothesis graph are said to be* similar *if and only if for all strings $s \in \Sigma^*$,*

$$\left\| \widehat{S}_{u,\sigma} - \widehat{S}_v \right\|_\infty \leq \mu/2.$$

**Definition B.2** *[Clark and Thollard, 2004, Definition 7]*
*A multiset $S$ is $\mu$-$\varepsilon_1$-*good *for a state $q \in Q_\mathcal{A}$ if and only if $\left\| \widehat{S} - P^q_\mathcal{A} \right\|_\infty < \mu/4$ and for every $\sigma \in \Sigma$, $|S(\sigma)/|S| - P_\mathcal{A}(q, \sigma)| < \varepsilon_1$.*

**Definition B.3** *[Clark and Thollard, 2004, Definition 8]*
*A hypothesis graph $G$ for a PDFA $\mathcal{A}$ is* good *if there is a bijective function $\Phi$ from a subset of states of $\mathcal{A}$ to all the nodes of $G$ such that $\Phi(q_0) = v_0$, and if $\delta_G(u, \sigma) = v$ then $\delta_G(\Phi^{-1}(u), \sigma) = \Phi^{-1}(v)$, and for every node $u$ in $G$, the multiset $S_u$ attached to $u$ is $\mu$-$\varepsilon_1$-good for the state $\Phi^{-1}(u)$.*

**Definition B.4** *[Clark and Thollard, 2004, Definition 11]*
*Given a good hypothesis graph $G$, a sample of size $M$ is* good *if for every candidate*

*node $(u, \sigma)$ such that $|S_{u,\sigma}| > m_0$, $S_{u,\sigma}$ is $\mu$-$\varepsilon_1$-good for the state $\delta_{\mathcal{A}}(\Phi^{-1}(u), \sigma)$ and if $P_{exit}(G) > \varepsilon_6$ then the number of strings that exit the graph is more than $\frac{1}{2} N P_{exit}(G)$.*

## B.2 Lemmas and Theorems

We repeat the statements and proofs of lemmas and theorems from [Clark and Thollard, 2004] used in the thesis, with the relevant revisions in notation.

**Lemma B.5** *[Clark and Thollard, 2004, Section 6.1]*
*Let:*

$$\begin{aligned} m_2 &= \frac{1}{2\,\varepsilon_1^2} \log\left(\frac{24n|\Sigma|(|\Sigma|+1)(n|\Sigma|+2)}{\delta_2}\right), \\ \varepsilon_1 &= \frac{\varepsilon^2}{16(|\Sigma|+1)(L+1)^2}. \end{aligned}$$

*Drawing a multiset of $m_2$ samples from a PDFA $\mathcal{A}$, the probability of generating a good multiset (according to Definition 4.14) is at least $1 - \frac{\delta_2}{n|\Sigma|}$.*

**Proof** We need to show that for every $\sigma \in \Sigma$,

$$\left| P_{\mathcal{A}}(q, \sigma) - \frac{S_u(\sigma)}{|S_u|} \right| \leq \varepsilon_1 .$$

Using the Chernoff bound, the probability of this occurring is less that $e^{-2m_2\varepsilon_1^2}$. Plugging in the expressions for $m_2$ and $\varepsilon_1$, we have:

$$e^{-2m_2\,\varepsilon_1^2} = \frac{\delta_2}{24n|\Sigma|(|\Sigma|+1)(n|\Sigma|+2)},$$

and therefore:

$$(|\Sigma|+1)\, e^{-2m_2\,\varepsilon_1^2} < \frac{\delta_2}{n|\Sigma|}.$$

Applying the union bound, the lemma follows. ■

**Lemma B.6** *[Clark and Thollard, 2004, Lemma 6]*
*Let $\mathcal{A}$ be a PDFA such that the expected length from any state is at most $L$. Then for all $q \in Q_{\mathcal{A}}$, $P_{\mathcal{A}}(q) \geq W(q)/(L+1)$.*

**Proof** Intuitively, after reaching the state $q$, the expected number of times we reach $q$ again will be at most the expected number of times we reach any state after this, which is bounded by $L$. Formally:

$$\begin{aligned}
W(q) &= \sum_{\substack{s\in S(q):\\ \delta_{\mathcal{A}}(q_0,s)=q}} P_{\mathcal{A}}(q_0,s)\\
&= \sum_{r\in R_{\mathcal{A}}(q)} \sum_{\substack{s\in\Sigma^*:\\ \delta_{\mathcal{A}}(q_0,rs)=q}} P_{\mathcal{A}}(q_0,rs)\\
&= \sum_{r\in R_{\mathcal{A}}(q)} \sum_{\substack{s\in\Sigma^*:\\ \delta_{\mathcal{A}}(q_0,rs)=q}} P_{\mathcal{A}}(q_0,r)P_{\mathcal{A}}(q,s)\\
&= \sum_{r\in R_{\mathcal{A}}(q)} P_{\mathcal{A}}(q_0,r) \sum_{\substack{s\in\Sigma^*:\\ \delta_{\mathcal{A}}(q_0,rs)=q}} P_{\mathcal{A}}(q,s)\\
&= P_{\mathcal{A}}(q) \sum_{\substack{s\in\Sigma^*:\\ \delta_{\mathcal{A}}(q_0,rs)=q}} P_{\mathcal{A}}(q,s)\\
&\le P_{\mathcal{A}}(q) \sum_{s\in\Sigma^*} P_{\mathcal{A}}(q,s)\\
&\le P_{\mathcal{A}}(q)(L+1).
\end{aligned}$$

■

**Lemma B.7** *[Clark and Thollard, 2004, Lemma 12]*
*Define the following accuracy bounds:*

$$\varepsilon_2 := \frac{\varepsilon_3}{2nL(L+1)}, \qquad \varepsilon_5 := \frac{\varepsilon_3}{2|\Sigma|L(L+1)}, \qquad \varepsilon_6 := \frac{\varepsilon_2\,\varepsilon_5}{L+1}.$$

*Then for any state $q \in Q_{\mathcal{A}}$ of the target PDFA such that $W(q) > \varepsilon_2$, if all sample multisets are good, then there will be a node $u$ in the final hypothesis graph such that $\Phi(q) = u$. Furthermore, for such a state $q$ and any $\sigma \in \Sigma$ such that $P_{\mathcal{A}}(q,\sigma) > \varepsilon_5$, the node $\delta_G(u,\sigma)$ is defined and is equal to $\Phi(\delta_{\mathcal{A}}(q,\sigma))$.*

**Proof** Since $W(q) > \varepsilon_2$, by Lemma B.6 it follows that $P_{\mathcal{A}}(q) > \varepsilon_2/(L+1)$. From the definitions of $M$ and $\varepsilon_6$

$$\frac{2n|\Sigma|m_0}{M} < \varepsilon_6 < P_{\mathcal{A}}(q)\,\varepsilon_5 < P_{\mathcal{A}}(q).$$

If there were no representative node $u$ for the state $q$, then all strings that reached the state would exit the graph, and thus we would have $P_{\text{exit}}(G) \ge P_{\mathcal{A}}(q)$. Similarly, if there were no edge labelled $\sigma$ from the node, then $P_{\text{exit}}(G) \ge P_{\mathcal{A}}(q)\,\varepsilon_5$. By the goodness of the sample we know that either $P_{\text{exit}}(G) \le \varepsilon_6$ or $P_{\text{exit}}(G) < 2n|\Sigma|m_0/M$, and in both cases $P_{\text{exit}}(G) < P_{\mathcal{A}}(q)\,\varepsilon_5$. Therefore, there is a suitable state $u$ and edge in the graph, and since the final hypothesis graph is $\varepsilon_1$-good, $\delta_G$ will have the correct value. ■

Assuming that all the samples drawn are good, the following lemma shows that the final smoothed transition probabilities will be close to the correct values.

**Lemma B.8** *[Clark and Thollard, 2004, Section 4.3]*
*Let $q \in Q_{\mathcal{A}}$ be a state with $W(q) > \varepsilon_2$ such that $u = \Phi(q)$, and define*

$$\varepsilon_4 := \frac{\varepsilon}{2(L+1)} \quad ; \quad p_{\min} := \frac{\varepsilon_4}{2(|\Sigma|+1)}.$$

*Then for every symbol $\sigma \in \Sigma$,*

$$\frac{P_{\mathcal{A}}(q,\sigma)}{\widehat{P}_{\mathcal{A}}(q,\sigma)} \leq 1 + \varepsilon_4 .$$

**Proof** By the goodness of the multisets we have:

$$\left| P_{\mathcal{A}}(q,\sigma) - \frac{S_u(\sigma)}{|S_u|} \right| \leq \varepsilon_1 .$$

Plugging in the definitions of $p_{\min}$ and $\varepsilon_4$, the claim is easily verified. ■

The following lemma shows that for every frequent state $q \in Q_{\mathcal{A}}$ and its corresponding state in the hypothesis $\widehat{q} = \Phi(q)$ (which exists with high probability), the expected number of times $\mathcal{A}$ "visits" $q$ is close to the expected number of times $\mathcal{A}$ visits $q$ *and* $\widehat{\mathcal{A}}$ visits $\widehat{q}$.

**Lemma B.9** *[Clark and Thollard, 2004, Lemma 13]*
*Let $q \in Q_{\mathcal{A}}$ be a state with $W(q) > \varepsilon_2$. Then there exists a state $\widehat{q} \in \widehat{\mathcal{A}}$ such that $\widehat{q} = \Phi(q)$ and*

$$W(q) - W(q,\widehat{q}) \leq \varepsilon_3 .$$

**Proof** The lemma is proved by induction on $\ell$. For $\ell = 0$ the claim is clearly true. Assume by induction that the claim holds for $\ell - 1$. Using the symbols $b \in Q_{\mathcal{A}}$ and $\widehat{b} \in Q_{\widehat{\mathcal{A}}}$ to denote states, we rewrite the joint weight as

$$W_\ell(q,\widehat{q}) = \sum_{b \in Q_{\mathcal{A}}} \sum_{\widehat{b} \in Q_{\widehat{\mathcal{A}}}} W_{\ell-1}(b,\widehat{b}) \sum_{\substack{\sigma : \delta_{\mathcal{A}}(b,\sigma) = q \\ \delta_{\widehat{\mathcal{A}}}(\Phi(b),\sigma) = \widehat{q}}} P_{\mathcal{A}}(b,\sigma).$$

Considering only the cases where $W(b) > \varepsilon_2$ and $\Phi(b) = \widehat{b}$, we see that

$$W_\ell(q,\widehat{q}) \geq \sum_{\substack{b \in Q_{\mathcal{A}}: \\ W(b) > \varepsilon_2}} W_{\ell-1}(b,\Phi(b)) \sum_{\substack{\sigma : \delta_{\mathcal{A}}(b,\sigma) = q \\ \delta_{\widehat{\mathcal{A}}}(\Phi(b),\sigma) = \widehat{q}}} P_{\mathcal{A}}(b,\sigma).$$

By the inductive assumption, for these frequent states:

$$W_\ell(q,\widehat{q}) \geq \sum_{\substack{b\in Q_{\mathcal{A}}:\\ W(b)>\varepsilon_2}} W_{\ell-1}(b)\left(1-\frac{\varepsilon_3(\ell-1)}{L(L+1)}\right)\sum_{\substack{\sigma:\delta_{\mathcal{A}}(b,\sigma)=q\\ \delta_{\widehat{\mathcal{A}}}(\Phi(b),\sigma)=\widehat{q}}} P_{\mathcal{A}}(b,\sigma).$$

Using the fact that $W_\ell \leq 1$ for all $\ell \in \mathbb{N}$ and using the recursive definition of $W_\ell$, the previous expression can be written as:

$$\begin{aligned}
W_\ell(q,\widehat{q}) &\geq \sum_{\substack{b\in Q_{\mathcal{A}}:\\ W(b)>\varepsilon_2}} W_{\ell-1}(b) \sum_{\substack{\sigma:\delta_{\mathcal{A}}(b,\sigma)=q\\ \delta_{\widehat{\mathcal{A}}}(\Phi(b),\sigma)=\widehat{q}}} P_{\mathcal{A}}(b,\sigma) - \frac{\varepsilon_3(\ell-1)}{L(L+1)} \\
&\geq \sum_{\substack{b\in Q_{\mathcal{A}}:\\ W(b)>\varepsilon_2}} W_{\ell-1}(b) \sum_{\sigma:\delta_{\mathcal{A}}(b,\sigma)=q} P_{\mathcal{A}}(b,\sigma) \\
&\quad - \sum_{\substack{b\in Q_{\mathcal{A}}:\\ W(b)>\varepsilon_2}} W_{\ell-1}(b) \sum_{\substack{\sigma:\delta_{\mathcal{A}}(b,\sigma)=q\\ \delta_{\widehat{\mathcal{A}}}(\Phi(b),\sigma)\neq\widehat{q}}} P_{\mathcal{A}}(b,\sigma) \\
&\quad - \frac{\varepsilon_3(\ell-1)}{L(L+1)}. \qquad \text{(B.1)}
\end{aligned}$$

We next show that most of the weight from $W_{\ell-1}$ must be from states $b$ with $W(b) \geq \varepsilon_2$. This enables changing from $W_{\ell-1}(b)$ to $W_\ell(b)$.

Using Equation (4.5), we have:

$$\begin{aligned}
W_\ell(q) &= \sum_{\substack{b\in Q_{\mathcal{A}}:\\ W(b)\leq\varepsilon_2}} W_{\ell-1}(b) \sum_{\sigma:\delta_{\mathcal{A}}(b,\sigma)=b} P_{\mathcal{A}}(b,\sigma) + \sum_{\substack{b\in Q_{\mathcal{A}}:\\ W(b)>\varepsilon_2}} W_{\ell-1}(b) \sum_{\sigma:\delta_{\mathcal{A}}(b,\sigma)=b} P_{\mathcal{A}}(b,\sigma) \\
&\leq n\varepsilon_2 + \sum_{\substack{b\in Q_{\mathcal{A}}:\\ W(b)>\varepsilon_2}} W_{\ell-1}(b) \sum_{\sigma:\delta_{\mathcal{A}}(b,\sigma)=b} P_{\mathcal{A}}(b,\sigma).
\end{aligned}$$

Using this expression to replace the right-hand side of Equation (B.1), we get

$$W_\ell(b,\widehat{b}) \geq W_\ell(q) - n\varepsilon_2 - \frac{\varepsilon_3(\ell-1)}{L(L+1)} - \sum_{\substack{b\in Q_{\mathcal{A}}:\\ W(b)>\varepsilon_2}} W_{\ell-1}(b) \sum_{\substack{\sigma:\delta_{\mathcal{A}}(b,\sigma)=q\\ \delta_{\widehat{\mathcal{A}}}(\Phi(b),\sigma)\neq\widehat{q}}} P_{\mathcal{A}}(b,\sigma).$$

Since by Lemma B.7 all of the transitions from frequent states with probability greater than $\varepsilon_5$ must go to the correct states, we know that the values of $P_{\mathcal{A}}(b,\sigma)$ in the final term must be less than $\varepsilon_5$.

$$\begin{aligned}
W_\ell(q,\widehat{q}) &\geq W_\ell(q) - n\varepsilon_2 - \frac{\varepsilon_3(\ell-1)}{L(L+1)} - \sum_{\substack{b\in Q_{\mathcal{A}}:\\ W(b)>\varepsilon_2}} W_{\ell-1}(b) \sum_{\sigma\in\Sigma} \varepsilon_5 \\
&\geq W_\ell(q) - n\varepsilon_2 - \frac{\varepsilon_3(\ell-1)}{L(L+1)} - |\Sigma|\,\varepsilon_5\,.
\end{aligned}$$

By the definitions of $\varepsilon_2$ and $\varepsilon_5$,

$$n\,\varepsilon_2 + |\Sigma|\,\varepsilon_5 \le \frac{\varepsilon_3}{L(L+1)}, \quad \text{and therefore}$$
$$W_\ell(q) - W_\ell(q,\widehat{q}) \le \frac{\varepsilon_3\,\ell W_\ell(q)}{L(L+1)}.$$

By Equation (4.6), $\sum_{\ell=0}^{\infty} \ell W_\ell(q) \le L(L+1)$. For any $q \in Q_{\mathcal{A}}$, $W(q) = \sum_{\ell=0}^{\infty} W_\ell(q)$. Summing over all $\ell \in \mathbb{N}$:

$$W(q) - W(q,\widehat{q}) = \sum_{\ell=0}^{\infty} W_\ell(q) - W_\ell(q,\widehat{q}) \le \sum_{\ell=0}^{\infty} \frac{\varepsilon_3\,\ell W_\ell(q)}{L(L+1)} \le \varepsilon_3\,.$$

■

Note that as a direct consequence of Lemma B.9,

$$\sum_{\widehat{q}:\Phi(q)\neq\widehat{q}} W(q,\widehat{q}) \le \varepsilon_3\,. \tag{B.2}$$

We now duplicate the proof of the main result, showing approximation of $\mathcal{A}$ by $\widehat{\mathcal{A}}$.

**Theorem B.10** *[Clark and Thollard, 2004, Section 5] Assuming all samples drawn are good, KL($\mathcal{A}$ || $\widehat{\mathcal{A}}$) < $\varepsilon$.*

**Proof** In order to bound the approximation error, we recall the decomposition of the KL-divergence presented in [Carrasco, 1997]:

$$\mathrm{KL}(\mathcal{A} \,\|\, \widehat{\mathcal{A}}) = \sum_{q\in\mathcal{A}} \sum_{\widehat{q}\in\widehat{\mathcal{A}}} \sum_{\sigma\in\Sigma} W(q,\widehat{q}) P_{\mathcal{A}}(q,\sigma) \log \frac{P_{\mathcal{A}}(q,\sigma)}{P_{\widehat{\mathcal{A}}}(\widehat{q},\sigma)}.$$

The summation is divided into three (non-overlapping) parts. Define

$$D(q,\widehat{q}) = W(q,\widehat{q}) \sum_{\sigma\in\Sigma} P_{\mathcal{A}}(q,\sigma) \log \frac{P_{\mathcal{A}}(q,\sigma)}{P_{\widehat{\mathcal{A}}}(\widehat{q},\sigma)},$$

and split the state pairs into three categories:

$$\begin{aligned}
D_1 &= \sum_{\substack{q\in Q_{\mathcal{A}}:\\ W(q)>\varepsilon_2}} \sum_{\substack{\widehat{q}\in Q_{\widehat{\mathcal{A}}}:\\ \Phi(q)=\widehat{q}}} D(q,\widehat{q}),\\
D_2 &= \sum_{\substack{q\in Q_{\mathcal{A}}:\\ W(q)>\varepsilon_2}} \sum_{\substack{\widehat{q}\in Q_{\widehat{\mathcal{A}}}:\\ \Phi(q)\neq\widehat{q}}} D(q,\widehat{q}),\\
D_3 &= \sum_{\substack{q\in Q_{\mathcal{A}}:\\ W(q)\leq\varepsilon_2}} \sum_{\widehat{q}\in Q_{\widehat{\mathcal{A}}}} D(q,\widehat{q}), \text{ such that}\\
\mathrm{KL}(\mathcal{A}\,\|\,\widehat{\mathcal{A}}) &= D_1+D_2+D_3.
\end{aligned}$$

Note that for the frequent states with $W(q) > \varepsilon_2$, $\Phi$ will be well-defined.

The bound on $D_1$ uses Lemma B.8, recalling that $W(q) \geq W(q,\widehat{q})$:

$$D_1 \leq \sum_{\substack{q\in Q_{\mathcal{A}}:\\ W(q)>\varepsilon_2}} W(q)\log(1+\varepsilon_4) \leq (L+1)\log(1+\varepsilon_4) \leq (L+1)\,\varepsilon_4\,.$$

Bounding $D_2$ is achieved by using Equation (B.2), the fact that $P_{\mathcal{A}}(q,\sigma) \leq 1$ and that $P_{\widehat{\mathcal{A}}}(\widehat{q},\sigma) \geq p_{\min}$ by the smoothing technique.

$$\begin{aligned}
D_2 &\leq \sum_{\substack{q\in Q_{\mathcal{A}}:\\ W(q)>\varepsilon_2}} \sum_{\substack{\widehat{q}\in Q_{\widehat{\mathcal{A}}}\\ \Phi(q)\neq\widehat{q}}} W(q,\widehat{q}) \sum_{\sigma\in\Sigma} P_{\mathcal{A}}(q,\sigma)\log\frac{1}{p_{\min}}\\
&\leq \sum_{\substack{q\in Q_{\mathcal{A}}:\\ W(q)>\varepsilon_2}} \varepsilon_3\log\frac{1}{p_{\min}} \leq n\,\varepsilon_3\log\frac{1}{p_{\min}}.
\end{aligned}$$

With regard to $D_3$, using (4.7) and the bound on $W(q)$ we can see that

$$D_3 \leq \sum_{\substack{q\in Q_{\mathcal{A}}:\\ W(q)\leq\varepsilon_2}} \sum_{\widehat{q}\in Q_{\widehat{\mathcal{A}}}} W(q,\widehat{q}) \sum_{\sigma\in\Sigma} P_{\mathcal{A}}(q,\sigma)\log\frac{1}{p_{\min}} \leq n\,\varepsilon_2\log\frac{1}{p_{\min}}.$$

Substituting in the definitions of $\varepsilon_2$ and $\varepsilon_5$ (Lemma B.7) and assuming $L > 1$,

$$\begin{aligned}
\mathrm{KL}(\mathcal{A}\,\|\,\widehat{\mathcal{A}}) &< (L+1)\,\varepsilon_4 + \left(n\,\varepsilon_3 + \frac{\varepsilon_3}{2nL(L+1)}\right)\log\frac{1}{p_{\min}}\\
&< (L+1)\,\varepsilon_4 + (n+1)\,\varepsilon_3\log\frac{1}{p_{\min}}.
\end{aligned}$$

Substituting in the values of $p_{\min}$, $\varepsilon_3$ and $\varepsilon_4$ gives

$$\mathrm{KL}(\mathcal{A}\,\|\,\widehat{\mathcal{A}}) < \varepsilon\,.$$

■

# Bibliography

N. Abe and Hiroshi Mamitsuka. Predicting protein secondary structure using stochastic tree grammars. *Machine Learning*, 29:275–301, 1997.

N. Abe and M. K. Warmuth. On the computational complexity of approximating distributions by probabilistic automata. *Machine Learning*, 9:205–260, 1992. Special issue for COLT90.

L. Bahl, F. Jelinek, and R. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2): 179–190, 1983.

R. B. Bapat and T. E. S. Raghavan. *Nonnegative Matrices and Applications.* Encyclopedia of Mathematics and its Applications. Cambridge University Press, May 1997.

T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White. Testing that distributions are close. In *Proceedings of the* $41^{st}$ *Annual Symposium on Foundations of Computer Science*, pages 259–269, Washington, DC, USA, 2000. IEEE Computer Society.

L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Annals of Mathematical Statistics*, 41:164–171, 1970.

Dimitri P. Bertsekas and John N. Tsitsiklis. *Introduction to Probability.* Athena Scientific, June 2002.

R. C. Carrasco. Accurate computation of the relative entropy between stochastic regular grammars. *RAIRO (Theoretical Informatics and Applications*, 31(5):437–444, 1997.

R. C. Carrasco and J. Oncina. Learning deterministic regular grammars from stochastic samples in polynomial time. *RAIRO (Theoretical Informatics and Applications)*, 33(1): 1–20, 1999.

F. Casacuberta. Some relations among stochastic finite state networks used in automatic speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):691–695, 1990.

R. W. Chang and J. C. Hancock. On receiver structures for channels having memory. *IEEE Transactions on Information Theory*, IT-12:463–468, October 1966.

Alexander Clark and Franck Thollard. PAC-learnability of probabilistic deterministic finite state automata. *J. Mach. Learn. Res.*, 5:473–497, 2004.

T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley and Sons, New York, 1991.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society B*, 39(1):1–22, 1977.

Luc Devroye and Gàbor Lugosi. *Combinatorial Methods in Density Estimation.* Springer, 2001.

Yariv Ephraim and Neri Merhav. Hidden Markov processes. *IEEE Transactions on Information Theory*, 48(6):1518–1569, June 2002.

Yann Esposito, Aurelien Lemay, Francois Denis, and Pierre Dupont. Learning probabilistic residual finite state automata. In Pieter W. Adriaans, Henning Fernau, and Menno van Zaanen, editors, *Grammatical Inference: Algorithms and Applications, 6th International Colloquium: ICGI*, volume 2484 of *Lecture Notes in Computer Science*, pages 77–91. Springer, 2002.

A. Farago and G. Lugosi. An algorithm to find the global optimum of left-to-right hidden markov model parameters. *Probl. Contr. Inform. Theory*, 18(6):435–444, 1989.

E. M. Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.

E. M. Gold. Complexity of automaton identification from given data. *Information and Control*, 37(3):302–320, 1978.

O. Guttman, S. V. N Vishwanathan, and R. C. Williamson. Probabilistic automata learning via oracles. In Sanjay Jain, Hans Ulrich Simon, and Etsuji Tomita, editors, *Proceedings of ALT 2005*, number 3734 in Lecture Notes in Artificial Intelligence, pages 171 – 182, Singapore, October 2005. Springer-Verlag.

T. Hagerup and C. Rub. A guided tour of Chernov bounds. *Inform. Proc. Lett.*, 33: 305–308, 1990.

Colin De La Higuera and Franck Thollard. Identification in the limit with probability one of stochastic deterministic finite automata. In *Grammatical Inference: Algorithms and Applications, 5th International Colloquium, ICGI 2000, Lisbon, Portugal, September 11 - 13, 2000 ; Proceedings*, volume 1891 of *Lecture Notes in Artificial Intelligence*, pages 141–156. Springer, Berlin, 2000.

J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, Reading, Massachusetts, first edition, 1979.

F. Jelinek. *Statistical Methods for Speech Recognition.* The MIT Press, 1998.

M. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R. Schapire, and L. Sellie. On the learnability of discrete distributions. In *Proc. of Twenty-sixth ACM Symposium on Theory of Computing*, pages 273–282, 1994.

Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. In *Proceedings of the 25th ACM Symposium on the Theory of Computing*, pages 392–401. ACM Press, 1993.

F. LeGland and L. Mevel. Exponential forgetting and geometric ergodicity in hidden Markov models. *Math. Contr. Signals Syst.*, 13:63–93, 2000.

B. G. Leroux. Maximum-likelihood estimation for hidden Markov models. *Stochastic Processes and Their Applications*, 40:127–143, 1992.

David McAllester and Robert E. Schapire. On the convergence rate of Good-Turing estimators. In *Proc. 13th Annu. Conference on Comput. Learning Theory*, pages 1–6. Morgan Kaufmann, San Francisco, 2000.

Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995.

K. Murphy. Passively learning finite automata. Technical report, Santa Fe Institute, 1996.

Nick Palmer and Paul W. Goldberg. PAC-learnability of probabilistic deterministic finite state automata in terms of variation distance. In Sanjay Jain, Hans Ulrich Simon, and Etsuji Tomita, editors, *Proceedings of ALT 2005*, number 3734 in Lecture Notes in Artificial Intelligence, pages 157–170, Singapore, October 2005. Springer-Verlag.

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, February 1989.

J. Raviv. Decision making in Markov chains applied to the problem of pattern recognition. *IEEE Transactions on Information Theory*, IT3:536–551, 1967.

Dana Ron, Yoram Singer, and Naftali Tishby. On the learnability and usage of acyclic probabilistic finite automata. In *Proc. 8th Annu. Conf. on Comput. Learning Theory*, pages 31–40. ACM Press, New York, NY, 1995.

E. Vidal, F. Thollard, C. de la Higuera, F. Casacuberta, and R. C. Carrasco. Probabilistic finite-state machines – Part I. *IEEE Trans. on Pattern analysis and Machine Intelligence*, 27(7):1013–1025, July 2005a.

E. Vidal, F. Thollard, C. de la Higuera, F. Casacuberta, and R. C. Carrasco. Probabilistic finite-state machines – Part II. *IEEE Trans. on Pattern analysis and Machine Intelligence*, 27(7):1026–1039, July 2005b.

A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, IT-13:260–269, April 1967.

Tsachy Weissman, Erick Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J. Weinberger. Inequalities for the $L_1$ deviation of the empirical distribution. Technical Report HPL-2003-97(R.1), HP Labs, HP Laboratories, Palo Alto, June 2003.

Ofer Zeitouni, Jacob Ziv, and Neri Merhav. When is the generalized likelihood ratio test optimal? *IEEE Transactions on Information Theory*, 38(5):1597–1602, 1992.

# Glossary of Symbols

| | |
|---|---|
| $\mathbb{N}$ | the set of natural numbers |
| $\mathbb{R}$ | the set of real numbers |
| $\varepsilon$ | accuracy parameter |
| $\delta$ | confidence parameter |
| $\Sigma$ | finite alphabet of symbols (letters) |
| $\sigma$ | a single symbol (letter) of the alphabet $\Sigma$ |
| $w, u, v, w, x, y, z$ | strings over the alphabet $\Sigma$ |
| $\epsilon$ | the empty string |
| $\ell$ | length of a string |
| $L$ | upper-bound on length of a string |
| Pr | probability with respect to all random variables |
| $\Pr_X$ | probability with respect to the distribution of random $X$ |
| $\mathbb{E}$ | expectation with respect to all random variables |
| $\mathbb{E}_\mu$ | expectation with respect to the measure $\mu$ |
| $\mathbb{E}(X)$ | expectation of the random variable $X$ |
| $\mathrm{var}(X)$ | variance of the random variable $X$ |
| $\mathbf{1}_{\{A\}}$ | indicator of the event $A$ |
| $D$ | distribution |
| $\mathcal{D}$ | class of distributions |
| $\delta(x)$ | distribution with all probability concentrated on $x$ |
| $L$ | stochastic language |
| $x^{-1}L$ | quotient language, defined by the probabilities of the strings in $L$ starting with $x$, properly normalized |
| $\mathcal{L}$ | family of stochastic languages |

| | |
|---|---|
| $\mathcal{A}$ | a probabilistic automaton (PFA or PDFA) |
| $Q_{\mathcal{A}}$ | the set of states of a PFA or PDFA $\mathcal{A}$ |
| $\delta_{\mathcal{A}}$ | set of transitions in $\mathcal{A}$ |
| $I_{\mathcal{A}}$ | initial state probabilities of $\mathcal{A}$ |
| $P_{\mathcal{A}}$ | transition probabilities of $\mathcal{A}$ |
| $F_{\mathcal{A}}$ | final state probabilities of $\mathcal{A}$ |
| $n$ | number of states in an automaton ($n = \lvert Q_{\mathcal{A}} \rvert$) |
| $q_i$ | the $i^{th}$ state of a PFA or PDFA $\mathcal{A}$ |
| $D_{\mathcal{A}}$ | distribution induced by the PFA or PDFA $\mathcal{A}$ |
| $P_{\mathcal{A}}(q, s)$ | probability of the PDFA $\mathcal{A}$ generating the *prefix* $s$ starting from state $q$ |
| $P^{q}_{\mathcal{A}}(s)$ | probability of the PDFA $\mathcal{A}$ generating the *string* $s$ starting from state $q$ |
| $P_{\mathcal{A}}(q)$ | probability of reaching state $q$ |
| $R_{\mathcal{A}}(q)$ | set of strings that reach state $q$ for the first time |
| $\widehat{\mathcal{A}}$ | a hypothesis PDFA induced by an induction algorithm |
| $G$ | graph underlying a hypothesis PDFA $\widehat{\mathcal{A}}$ |
| $S$ | a multiset of strings |
| $s$ | a single string in the multiset $S$ |
| $S(s)$ | multiplicity of the string $s$ in multiset $S$ |
| $S(\sigma)$ | number of strings beginning with $\sigma$ in $S$ |
| $S_v$ | multiset of suffixes incident on node $v$ of hypothesis graph $G$ |
| $S_{u,\sigma}$ | multiset of suffixes incident on candidate node $(u, \sigma)$ of hypothesis graph $G$ |
| $W(q)$ | weight of state $q$ |
| $W(q, \widehat{q})$ | joint weight of (target) state $q$ and (hypothesis) state $\widehat{q}$ |
| $\mathcal{H}$ | class of PDFA |
| $\mathcal{O}_{\mathcal{H}}$ | an oracle matching the PDFA class $\mathcal{H}$ |