Maximum Likelihood Estimation of Factored Regular Deterministic Stochastic Languages

Chihiro Shibata
School of
Computer Science
Tokyo University of Technology
shibatachh@stf.teu.ac.jp

Jeffrey Heinz
Department of Linguistics &
Institute for Advanced Computational Science
Stony Brook University
jeffrey.heinz@stonybrook.edu

Abstract

This paper proves that for every class C of stochastic languages defined with the co-emission product of finitely many probabilistic, deterministic finite-state acceptors (PDFA) and for every data sequence D of finitely many strings drawn i.i.d. from some stochastic language, the Maximum Likelihood Estimate of D with respect to C can be found efficiently by locally optimizing the parameter values. We show that a consequence of the co-emission product is that each PDFA behaves like an independent factor in a joint distribution. Thus, the likelihood function decomposes in a natural way. We also show that the negative log likelihood function is convex. These results are motivated by the study of Strictly k-Piecewise (SP_k) Stochastic Languages, which form a class of stochastic languages which is both linguistically motivated and naturally understood in terms of the co-emission product of certain PDFAs.

1 Introduction

Stochastic languages are probability distributions over all possible strings of finite length. A class C of stochastic languages is often defined parametrically: an assignment of values to the parameters uniquely determines some stochastic language L in C and thus the probabilities that L assigns to strings. An important learning criterion for a class of stochastic languages C is whether there is an algorithm which reliably returns a Maximum-Likelihood Estimate (MLE) of an observed data sample D. The MLE is the parameter values which maximize the probability of D with respect to C.

This paper focuses on *regular deterministic* stochastic languages. These are stochastic languages that can be defined with a probabilistic, deterministic, finite-state acceptors (PDFA).

The problem of finding the MLE, however, is not only about some single stochastic language L, but also about the *class* of stochastic languages that L belong to. It is well-understood that each PDFA $\mathcal{M}$ naturally defines a class of stochastic languages $C_{\mathcal{M}}$ because the transitional probabilities in the PDFA provide a range of possible parameter values, as we explain in detail in section 2. In this case, it is well-understood how to find the MLE of a sequence of strings drawn i.i.d. from L with respect to $C_{\mathcal{M}}$ (Vidal et al., 2005a,b). This paper is concerned with finding the MLE for *different* classes of stochastic languages.

In particular, we consider the case where C is defined by the range of parametric values over *finitely many* PDFA $\mathcal{A} = \{\mathcal{M}_1 \dots \mathcal{M}_K\}$, whose *co-emission product* determines the probabilities each $L \in C$ assigns to strings. Essentially, the co-emission product of these PDFAs *factor* the probabilities each $L \in C$ assigns to strings. Each L is a complex joint distribution, and each PDFA $\mathcal{M}_j$ represents a 'more basic' regular stochastic language whose parameter values independently contribute to L. At a high level, the problem we are considering is like those addressed with Bayesian networks and Markov random fields, where complex probability distributions decompose into simpler factors (Bishop, 2006; Koller and Friedman, 2009). We refer to the classes C we study in this paper as factored, regular, probabilistic, and deterministic (FRPD).

The main result is to show how the parameters of a FRPD class C can be efficiently updated to find those parameter values which maximize the likelihood of the observed sequences (Theorem 2). We also show directly that each negative log likelihood associated with each FRPD class C is convex (Theorem 3). Together these results imply that the efficient method we present for updating the parameter values will yield the MLE.

There are several reasons for being interested in such factored classes C. Perhaps the most important from our perspective is expressed by Koller and Friedman (p. 1134) "The ability to exploit structure in the distribution is the basis for providing a compact representation of high-dimensional ... probability spaces." In our case, the size of the representation of the class given by $\mathcal{A} = \{\mathcal{M}_1 \ldots \mathcal{M}_K\}$ is simply the sum of the size of each M_j. In contrast, the representation of the class given by the co-emission product is in the worst case the product of the sizes of each M_j. One direct benefit of this is that the number of parameters is reduced, which makes it possible to more accurately estimate them with less data. Other advantages discussed by Koller and Friedman, such as modularity, we return to in the discussion in the conclusion.

There are also linguistic reasons to be interested in FRPD classes. The Strictly Piecewise (SP) class of languages encode certain types of long-distance dependencies found in natural languages. For example, SP languages can express generalizations like "at most one b per string" and "no b may follow an a" (Rogers et al., 2010). Generalizations with this formal character are known to occur in the phonologies of the world's languages (Heinz, 2010a; Rogers et al., 2013; Heinz, 2014, 2018). As Rogers et al. (2010) explain, Strictly Piecewise languages are characterized by the intersection product of finitely many deterministic finite-state acceptors (DFA). Heinz and Rogers (2010) used this characterization and the co-emission product to define the class of Strictly Piecewise stochastic languages because they were interested in the learnability of long-distance dependencies in natural languages probabilistically. They presented a learning algorithm for a class of SP stochastic languages and claimed (p. 894) that it returns the MLE.

This results in this paper can be seen as providing a more generalized, more meaningful, and more rigorous proof of their basic claim. Theorem 2 establishes how to update the parametric values which locally optimize the model of *any* FRPD class. Theorem 3 shows the negative log likelihood function of *any* FRPD class is convex, so there is in fact only one set of optimal parametric values for any sequence of data. Furthermore, we prove these results in terms of the standard definition of co-emission product, and not the variant used in Heinz and Rogers (2010). (While the results here work for both, we only prove the standard case.) These general results make it possible to explore not only the learning of SP_k stochastic languages, but also *any* finite combination of PDFAs that characterize different kinds of local and non-local dependencies which can be expressed with regular grammars. We return to this issue in the discussion.

To our knowledge, such results for FRPD classes have not been previously discussed in the literature. One reason for this is that much work on natural language processing uses probabilistic *non-deterministic* automata. These describe the same class of stochastic languages as Hidden Markov Models (HMMs) (Vidal et al., 2005a,b). Non-determinism can make a big difference when it comes to parsing and learning. In a deterministic model $\mathcal{M}$, each string w can be associated with at most one path through $\mathcal{M}$, whereas in non-deterministic $\mathcal{M}$, there can be infinitely many paths for w. This is one reason why methods used for learning HMM are not guaranteed to return a MLE. Since the states are 'hidden' one uses methods like Expectation Maximization, which may converge to a local optimum that is not a global optimum (Jurafsky and Martin, 2008; Heinz et al., 2015).

On the other hand, we are showing that, by carefully choosing the class of stochastic languages C—which the MLE which is to be found will be 'with respect to'—we can exploit the structure we assume to be present to guarantee we find a MLE. This paper takes one step in establishing the theoretical soundness of this approach.

Finally, one reviewer commented that these results may follow from fundamental theorems in the literature on probabilistic graphical models (Koller and Friedman, 2009). Regardless of whether this is true, the correctness of the proofs here stand. Also, the general results of Bayesian networks and Markov random fields say nothing about the concrete forms of the algorithm for obtaining the MLE with respect to a FRPD class C given data D, and similarly for its time complexity. Malouf (2002) makes a similar point, writing "While all parameter estimation algorithms we will consider take the same general form, the method for computing the updates ... differs substantially." Nonetheless, how probabilistic graphical models relate to this line of research ought to

be made clear.

The remainder of the paper is organized as follows. In section 2 we review languages, stochastic languages, deterministic finite-state acceptors and probabilistic versions thereof, the intersection and co-emission products, and the statement of the learning problem. Before presenting our main results, section 3 defines Strictly Piecewise (stochastic) languages, which provide a running example to illustrate the main results, which are presented in section 4. The computational complexity of the updates are analyzed in section 5 and section 6 concludes.

2 Preliminaries

2.1 Sets of Strings

Σ denotes a finite set of symbols and Σ^k, $\Sigma^{\leq k}$, and Σ^* denote all strings over this alphabet of length k, of length less than or equal to k, and of any finite length, respectively. λ denotes the empty string. The length of a string w is written $|w|$. The prefixes of a string w are $\mathsf{Pref}(w) = \{v \mid \exists u \in \Sigma^*, vu = w\}$. A string $w = \sigma_1 \dots \sigma_n$ is a *subsequence* of a string v if and only if $v \in \Sigma^*\sigma_1\Sigma^* \dots \Sigma^*\sigma_n\Sigma^*$, in which case we write $w \sqsubseteq v$.

A *language* L is a subset of Σ^*. The *complement* of a language L, denoted $\overline{L}$ is Σ^*/L. The *shuffle ideal* of w is the language of all strings containing w as a subsequence:

$$SI(w) = \{v \mid w \sqsubseteq v\}.$$

A *stochastic language* L is a probability distribution over Σ^*. The probability P of word w with respect to L is written $P_L(w) = p$. Thus, all stochastic languages L satisfy

$$\sum_{w \in \Sigma^*} P_L(w) = 1.$$

2.2 Probabilistic Deterministic Finite-state Acceptors

A *Deterministic Finite-state Acceptor* (DFA) is a tuple $\mathcal{M} = \langle Q, \Sigma, q_0, \delta, F\rangle$ where Q is the state set, Σ is the alphabet, q_0 is the start state, δ is a deterministic transition function with domain $Q \times \Sigma$ and codomain Q, and F is the set of accepting states. Let $\delta^* : Q \times \Sigma^* \to Q$ be the (partial) path function of $\mathcal{M}$. When discussing partial functions, the notation $\uparrow$ and $\downarrow$ indicates that the function is not defined, respectively, is defined, for particular arguments. Thus $\delta^*(q, w)$ is the (unique) state reachable from state q via the sequence w, if any, or $\delta^*(q, w)\uparrow$ otherwise. The language recognized by a DFA $\mathcal{M}$ is $L(\mathcal{M}) \stackrel{\text{def}}{=} \{w \in \Sigma^* \mid \delta^*(q_0, w)\downarrow \in F\}$.

A *Probabilistic Deterministic Finite-state Acceptor* (PDFA) is a tuple $\mathcal{M} = \langle Q, \Sigma, q_0, \delta, F, T\rangle$ where Q, Σ, q_0, and δ are the same as with DFA, and F and T are partial functions representing the final-state and transition probabilities. In particular, $T : Q \times \Sigma \to \mathbb{R}^+$ and $F : Q \to \mathbb{R}^+$ such that

$$\text{for all } q \in Q,\ F(q) + \sum_{\sigma \in \Sigma} T(q, \sigma) = 1. \quad (1)$$

A PDFA $\mathcal{M}$ generates a stochastic language $L(\mathcal{M})$. If it exists, the unique *path* for a word $w = \sigma_0 \dots \sigma_N$ belonging to Σ^* through a PDFA is a sequence $\langle (q_0, \sigma_0), (q_1, \sigma_1), \dots, (q_N, \sigma_N)\rangle$, where $q_{i+1} = \delta(q_i, \sigma_i)$. The probability a PDFA assigns to w is obtained by multiplying the transition probabilities along w's path if it exists with the final probability, and zero otherwise. So $P_{L(\mathcal{M})}(w) =$

$$\left(\prod_{i=0}^{N} T(q_i, \sigma_i)\right) \cdot F\big(\delta(q_N, \sigma_N)\big)$$
$$\text{if } \delta^*(q_0, w)\downarrow \text{ and 0 otherwise} \quad (2)$$

A probability distribution is *regular deterministic* iff there is a PDFA which generates it. We sometimes write $\mathcal{M}(w)$ instead of $P_{L(\mathcal{M})}(w)$.

The *structural components* of a PDFA $\mathcal{M}$ are its states Q, its alphabet Σ, its transitions δ, and its initial state q_0. By *structure* of a PDFA, we mean its structural components. The structure of each PDFA $\mathcal{M}$ defines a class of stochastic languages given by the possible instantiations of T and F satisfying Equation 1. These distributions have at most $|Q|\cdot(|\Sigma| + 1)$ independent parameters (since for each state there are $|\Sigma|$ possible transitions plus the possibility of finality.)

2.3 The co-emission product

The *intersection product* of K DFAs $\mathcal{M}_1 \dots \mathcal{M}_K$ is given by the standard construction over the state space $Q_1 \times \dots \times Q_K$ (Hopcroft et al., 2001). We write $\bigotimes_{1 \leq j \leq K} \mathcal{M}_j = \mathcal{M} = \langle Q, \Sigma, q_0, \delta, F\rangle$ where $Q = Q_1 \times \dots \times Q_K$, $q_0 = \langle q_{01}, \dots q_{0K}\rangle$. For all $\langle q_1, \dots q_K\rangle \in Q$ and $\sigma \in \Sigma$, $\delta(\langle q_1, \dots q_K\rangle, \sigma) = \langle q'_1, \dots q'_K\rangle$ if and only if $\delta_1(q_1, \sigma) = q'_1, \dots \delta_K(q_K, \sigma) = q'_K$. Finally, let $F = F_1 \times \dots \times F_K$ It is well-known that $L(\bigotimes_{1 \leq j \leq K} \mathcal{M}_j) = \bigcap_{1 \leq j \leq K} L(\mathcal{M}_j)$.

The *co-emission product* of K PDFAs $\mathcal{M}_1 \ldots \mathcal{M}_K$ is also given by a construction over the state space $Q_1 \times \ldots \times Q_K$. The probability that σ is co-emitted from $\langle q_1, \ldots, q_K \rangle$ in $Q_1 \times \ldots] \times Q_K$ is the product of the probabilities of its emission at each $q_j \in Q_j$. Let $\mathsf{CoT}(\langle \sigma, q_1 \ \ldots \ q_K \rangle) = \prod_{j=1}^{K} T_j(q_j, \sigma)$. Similarly, the probability that a word simultaneously ends at $q_1 \in Q_1, \ \ldots \ q_K \in Q_K$ is

$$\mathsf{CoF}(\langle q_1 \ \ldots \ q_K \rangle) = \prod_{j=1}^{K} F_j(q_j).$$

Finally, for $q = \langle q_1 \ \ldots \ q_K \rangle$, let

$$Z(q) = \mathsf{CoF}(q) \ + \ \sum_{\sigma \in \Sigma} \mathsf{CoT}(\langle \sigma, q \rangle)$$

be the *normalization term*. Next we define the co-emission product.

Definition 1 (Co-emission Product) *For* $\mathcal{A} = \{\mathcal{M}_1, \ldots \mathcal{M}_K\}$, *let* $\bigotimes \mathcal{A} = \langle Q, \Sigma, q_0, \delta, F, T \rangle$ *where*

1. Q, q_0, *and* δ *are defined as with DFA product; and*
2. *For all* $q \in Q$ *and* $\sigma \in \Sigma$*:*

$$F(q) = \frac{\mathsf{CoF}(q)}{Z(q)}$$

and

$$T(q, \sigma) = \frac{\mathsf{CoT}(\sigma, q)}{Z(q)}.$$

In other words, the numerators of T and F are defined to be the co-emission probabilities and division by Z ensures that co-emission product $\bigotimes \mathcal{A}$ defines a well-formed probability distribution over Σ^*.

Observe that $\mathcal{A}$ also defines a class of stochastic languages by the possible instantiations of T_j and F_j for each $M_j \in \mathcal{A}$. The structural components of $\mathcal{A}$ are the structural components of each $M_j \in \mathcal{A}$. By *structure* of $\mathcal{A}$, we mean its structural components. The structure of $\mathcal{A}$ defines a class of stochastic languages given by the possible instantiations of T_j and F_j satisfying Equation 1 for each $M_j \in \mathcal{A}$.

If $\bigotimes \mathcal{A} = \mathcal{M}$ then the class of stochastic languages induced by the structure of $\mathcal{A}$ is a subset of the class of stochastic languages obtained with the structure of the PDFA $\mathcal{M}$. This is another way of saying that a factorized model may have fewer parameters and so the class of stochastic languages it represents can become smaller.

2.4 Statement of the Learning Problem

Let D be a finite sequence of $|D|$ i.i.d. drawn examples from a stochastic language L. It follows that the $P_L(D) = \prod_{w \in D} P_L(w)$.

Let $\mathcal{A} = \{\mathcal{M}_1 \ldots \mathcal{M}_K\}$ be a set of PDFAs and let $C_\mathcal{A}$ denote the FRPD class of stochastic languages induced by the structure of $\mathcal{A}$. The *likelihood* of D w.r.t. $C_\mathcal{A}$ is determined by the parameters (the T_j and F_j functions for each $\mathcal{M}_j \in \mathcal{A}$). Let us group these parameters under the symbol Θ. Each Θ identifies some stochastic language $L_\Theta \in C_\mathcal{A}$. The *likelihood* of D w.r.t. $C_\mathcal{A}$ is defined as follows:

$$\mathsf{lhd}(D \mid \Theta) = \prod_{w \in D} P_{L_\Theta}(w).$$

The problem of finding a Maximum Likelihood Estimate (MLE) is to find those parameter values $\hat{\Theta}$ of $\mathcal{A}$ that maximize the likelihood of D w.r.t. $C_\mathcal{A}$. Formally,

$$\hat{\Theta} = \arg\max_{\Theta} \big(\mathsf{lhd}(D \mid \Theta)\big) \tag{3}$$

where Θ under the $\arg\max$ ranges over all possible parameter values of $\mathcal{A}$.

When $|\mathcal{A}| = 1$ the problem has a known solution. As mentioned, a single PDFA $\mathcal{M}$ defines a class of stochastic languages given by possible parameter values of $\mathcal{M}$. In this case, it is well-known how to find $\hat{\Theta}$. Essentially, each transition probability $T(q, \sigma)$ equals the relative frequency that symbol σ is emitted at a state q (Vidal et al., 2005a,b). In this paper, we solve this problem when $|\mathcal{A}| > 1$.

3 Strictly k-Piecewise stochastic languages

In this section, we introduce the Strictly k-Piecewise stochastic languages, which serve as a running example of a FRPD class in the remainder of the paper.

Rogers et al. (2010) define and provide multiple characterizations of Strictly Piecewise (SP) languages. We review the most relevant ones for this paper here. SP languages are exactly those formal languages that are closed under subsequence.

$$\text{SP} = \{L \subseteq \Sigma^* \mid \forall w, v \in \Sigma^* \ (v \in L, w \sqsubseteq v \Rightarrow w \in L)\}$$

Rogers et al. (2010, p. 260) prove that every SP language L can be associated with a finite set of

strings S such that L is the intersection of the complements of the shuffle ideals of S.

Theorem 1 $\forall L \in \text{SP}, \exists S \subseteq \Sigma^*, n \in \mathbb{N}$ *such that* $|S| < n$ *and* $L = \bigcap_{w \in S} \overline{SI(w)}$.

The SP languages are parameterized by a value $k \in \mathbb{N}$. This number corresponds to the length of the longest string in S. For each SP language L, if there is a set S whose longest string is equal to k, then L belongs to the SP_k class of languages.

If k is known a priori then the SP_k languages are both PAC-learnable and identifiable in the limit in polynomial time and data (Heinz, 2010b; Heinz et al., 2012).[1]

Theorem 1 allows one to construct concrete computational models for SP languages with DFA. For any nonempty string $w = \sigma_1 \ldots \sigma_n$, $SI(w) = L(\mathcal{M}_w)$ where $\mathcal{M}_w$ is defined as follows. The states are the prefixes of w, the start state is λ, and the final state is w. For all prefixes p of w and $\sigma \in \Sigma$, let $\delta(p, \sigma) = p\sigma$ whenever $p\sigma$ is a prefix of w and p otherwise. Figure 1 gives an examples of DFA for $\mathcal{M}_{abba}$.

The complement $SI(w)$ is essentially obtained from $\mathcal{M}_w$ by removing its maximal state and making every state final. In other words, if $w = va$ then the $\overline{SI(w)}$ can be recognized by an automaton where the states are the prefixes of v, the start state is λ, and each state is a final state. For all prefixes p of v and $\sigma \in \Sigma$, $\delta(p, \sigma) = p\sigma$ whenever $p\sigma$ is a prefix of v. When $p\sigma$ is not a prefix of v and $\sigma \neq a$ then $\delta(p, \sigma) = p$. Finally, $\delta(v, a)$ is not defined. We denote such a DFA as $\mathcal{M}_{\overline{w}}$. Figure 2 shows the DFA $\mathcal{M}_{\overline{abba}}$ which recognizes the complement of $SI(abba)$. Both $\mathcal{M}_w$ and the DFA recognizing its complement are minimal.

It follows that for any $L \in \text{SP}$, one can construct a DFA recognizing L by taking the *product* of the complements of the shuffle ideals of the strings in S.

Note the size of $\mathcal{M}_1 \ldots \mathcal{M}_K$ is $\sum_{1 \leq i \leq K} \mathcal{M}_j$ whereas the size of $\mathcal{M} = \bigotimes_{1 \leq j \leq K} \mathcal{M}_j$ is in the worst case $\prod_{1 \leq j \leq K} \mathcal{M}_j$. Therefore, to decide whether a string w belongs to some SP language L, it may be preferable to run w on each $\mathcal{M}_j$ instead of on $\mathcal{M}$ to avoid the potentially large increase in the state space. See Heinz and Rogers (2013) for additional discussion of this point.

Heinz and Rogers (2010) use the fact that SP languages are the intersection of the complements of shuffle ideals to define their stochastic counterpart. They define stochastic versions of $\mathcal{M}_{\overline{w}}$ (Figure 2), which they call *w-subsequence-distinguishing PDFA*.

Definition 2 (Subsequence-distinguishing PDFA) *Let* $w \in \Sigma^{k-1}$ *and* $w = \sigma_1 \cdots \sigma_{k-1}$. $\mathcal{M}_w = \langle Q, \Sigma, q_0, \delta, F, T \rangle$ *is a* w-subsequence-distinguishing PDFA *(w-SD-PDFA) iff* F *and* T *satisfy Equation 1 and* $\delta(u, \sigma) = u\sigma$ *whenever* $u\sigma \in \mathsf{Pref}(w)$ *and* u *otherwise.*

Apart from the stochastic components T and F, the w-subsequence-distinguishing PDFA differs from $\mathcal{M}_{\overline{w}}$ in one key way. Suppose. $w = va$. Then $\delta(v, a) = v$ in the w-subsequence-distinguishing PDFA is not undefined as was the case with $\mathcal{M}_{\overline{w}}$. This transition exists and may have a nonzero probability.

A set $\mathcal{A}$ of PDFAs is *a k-set of SD-PDFAs* iff, for each $w \in \Sigma^{\leq k-1}$, it contains exactly one w-SD-PDFA. For example, let $\Sigma = \{a, b\}$ and consider the 2-set of SD-PDFAs shown in Figure 3. There are three SD-PDFAs in this set corresponding to $\mathcal{M}_\lambda, \mathcal{M}_a$, and $\mathcal{M}_b$.

Heinz and Rogers (2010) define SP_k stochastic languages as a product of a k-set of SD-PDFAs. Specifically, the adapt the notion of co-emission probability (Vidal et al., 2005a). Heinz and Rogers (2010) actually use what they call the *positive co-emission product* which restricts the standard co-emission probability to particular circumstances.

In this work, we define SP stochastic languages with the standard definition of *co-emission probability* used to define products of PDFA as in Definition 1 (Vidal et al., 2005a).

Definition 3 (SP Stochastic Languages) *A probability distribution P over Σ^* is a SP stochastic language iff there exists a k-set of SD-PDFAs $\mathcal{A}$, whose co-emission product is $\mathcal{M} = \bigotimes \mathcal{A}$, such that for all $w \in \Sigma^*$, it is the case that $P(w) = \mathcal{M}(w)$.*

It follows immediately from this definition that the class of SP stochastic languages is a FRPD class. In this case, the parameters of such a distribution are the T and F values on each w-subsequence-distinguishing PDFA in the k-set. In the example in Figure 3, there are thus 15 parameters of the model, 10 of which are free. This is be-

[1] Also, SP languages suggest a different representation for strings (Rogers et al., 2013), which inform machine learning in other ways. The winning paper of the SPiCE competition (Balle et al., 2016), in which machine learning models competed to best predict the next symbol in a natural and artificial sequences was won by Shibata and Heinz (2016), who integrated SP-style representations into a neural network.

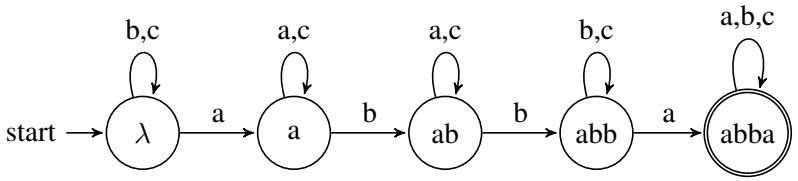

Figure 1: The DFA $\mathcal{M}_{abba}$ for $SI(abba)$ (left) with $\Sigma = \{a, b, c\}$.

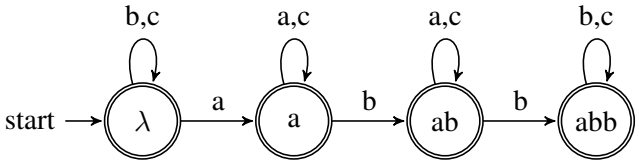

Figure 2: The DFA $\mathcal{M}_{\overline{abba}}$ for $\overline{SI(abba)}$ with $\Sigma = \{a, b, c\}$.

cause there are three actions associated with each state (a, b, and finality); there are five states; but since the probabilities must add to one only two parameters per state are free. More generally, a k-set of SD-PDFAs $\mathcal{A}$ has $|\Sigma| \cdot \sum_{j \in \mathcal{A}} |Q_j|$ free parameters.

4 Main Theorem for MLE of FRPD classes

We provide our main results here, using the 2-set of SD-PDFAs shown in Figure 3 as an illustrative example.

4.1 The Co-emission Probability Given a Prefix

It is useful to consider the co-emission probability of the symbol σ given the prefix $\sigma_1 \cdots \sigma_{i-1}$, which we denote $\mathsf{Coemit}(\sigma, i)$. It follows from Definitions 1 and 3 that this value is the normalized product of the path through $\bigotimes \mathcal{A}$ given by the prefix $\sigma_1 \cdots \sigma_{i-1}$.

Formally, let $M_1 = \langle Q_1, \Sigma, q_{01}, \delta_1, F_1, T_1 \rangle$, $\cdots$, $M_K = \langle Q_K, \Sigma, q_{0K}, \delta_K, F_K, T_K \rangle$ be exactly those PDFAs in $\mathcal{A}$. Suppose that $w = \sigma_1 \cdots \sigma_N$, where $\sigma_i \in \Sigma$ for all $1 \leq i \leq N$. Let $q(j, i)$ denote a state in Q_j that is reached after M_j reads the prefix $\sigma_1 \cdots \sigma_{i-1}$. If $i = 1$ then $q(j, i)$ represents the initial state of M_j. Then it follows from Definition 1 that the probability that a symbol σ is emitted after the product machine $\bigotimes_{1 \leq j \leq K} \mathcal{M}_j$ reads the prefix $\sigma_1 \cdots \sigma_{i-1}$ is the following: $\mathsf{Coemit}(\sigma, i) =$

$$\frac{\prod_{j=1}^{K} T_j(q(j,i), \sigma)}{\sum_{\sigma' \in \Sigma} \left(\prod_{j=1}^{K} T_j(q(j,i), \sigma') \right) + \prod_{j=1}^{K} F_j(q(j,i))} \tag{4}$$

To simplify the notation and analysis, we assume that there is a end marker $\ltimes \in \Sigma$ which uniquely occurs at the end of words. This lets us replace $F_j(q)$ with $T_j(q, \ltimes)$. Then $\mathsf{Coemit}(\sigma, i)$ is simply written as

$$\mathsf{Coemit}(\sigma, i) = \frac{\prod_{j=1}^{K} T_j(q(j,i), \sigma)}{\sum_{\sigma' \in \Sigma} \prod_{j=1}^{K} T_j(q(j,i), \sigma').} \tag{5}$$

The probability that the machine $\bigotimes_{1 \leq j \leq K} M_j$ accepts w is obtained by taking the product of the co-emission probabilities for all i:

$$P(w\ltimes) = \prod_{i=1}^{N+1} \mathsf{Coemit}(\sigma_i, i), \tag{6}$$

where $\sigma_{N+1} = \ltimes$.

Since we are concerned with the co-emission probabilities, which is a ratio, it is noteworthy that in fact it does not matter if the sum $\sum_{\sigma' \in \Sigma} T_j(q, \sigma')$ is 1. The ratio $\mathsf{Coemit}(\sigma, i)$ and thus $P(w\ltimes)$ are invariant with respect to the scale of $T_j(q, \sigma')$ and the sum $\sum_{\sigma' \in \Sigma} T_j(q, \sigma')$. Writing this last value as $z(j, q)$, it can easily be confirmed by the fact that multiplying both the denominator and the numerator by $1/z(j, q)$ does not change the value of $\mathsf{Coemit}(\sigma, i)$ while normalizing $T_j(q, \cdot)$. Thus, we can relax the condition in Equation 1 when discussing co-emission probabilities. The only condition that needs to be satisfied with respect to the transitions is that $T_j(q, \sigma') \geq 0$ for all j, q, σ'. Note that relaxing this condition does not affect the number of free parameters. This is because the numerical values associated with the transitions, once normalized, will always sum to 1. In the following, we assume this relaxed condition.

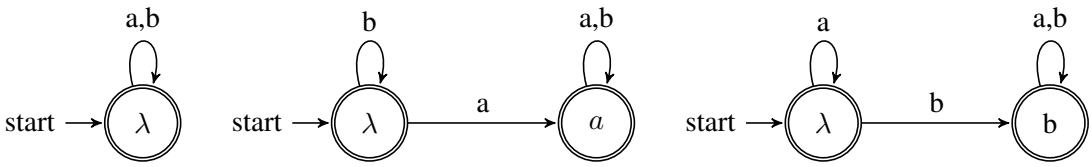

Figure 3: The 2-set of of SD-PDFAs with $\Sigma = \{a, b\}$.

4.2 Frequency and Empirical Mean of Co-emission Probability

Before describing the main theorem, we define two terms; *the frequency of an emission* and *the empirical mean of a co-emission probability*, which play important roles in estimating transition probabilities for product machines.

Definition 4 (Frequency of Emission) *For given w, we define the frequency of σ at $q \in Q_j$ as follows. Let*

- $m_w(M_j, q, \sigma) \in \mathbb{Z}^+$ *denotes how many times σ is emitted at the state q while the machine M_j emits w.*
- $n_w(M_j, q) \in \mathbb{Z}^+$ *denotes how many times the state q is visited while the machine M_j emits w.*

Then

$$\mathsf{freq}_w(\sigma|M_j, q) = \frac{m_w(M_j, q, \sigma)}{n_w(M_j, q)}, \qquad (7)$$

So $\mathsf{freq}_w(\sigma|M_j, q)$ represents the relative frequency that M_j emits σ at q during emission of w.

These concepts can be lifted to a sequence of strings D drawn i.i.d. from some stochastic language. Let

$$m_D(M_j, q, \sigma) = \sum_{w \in D} m_w(M_j, q, \sigma)$$

and

$$n_D(M_j, q) = \sum_{w \in D} n_w(M_j, q)\,.$$

It follows that

$$\mathsf{freq}_D(\sigma|M_j, q) = \frac{m_D(M_j, q, \sigma)}{n_D(M_j, q)}.$$

So $\mathsf{freq}_D(\sigma|M_j, q)$ represents the relative frequency that M_j emits σ at q during emission of D.

As an example, consider the 2-set of PDFAs in Figure 3 and consider the sample data $D = \langle abb\ltimes, aba\ltimes\rangle$. Figure 4 shows the paths of these strings through each SD-PDFA. Figure 5 shows some of the frequency computations.

If $K = 1$, i.e., the product machine consists of one PDFA then $\mathsf{freq}_w(\sigma|M_1, q)$ is the MLE of $T_1(q, \sigma)$ (Vidal et al., 2005a,b). Meanwhile, if $K \geq 2$, the probability of the emission, which equals the co-emission probability, fluctuates with states that other machines are currently at. Thus $\mathsf{freq}_w(\sigma|M_k, q)$, as a random variable, is not independent from other machines' states. This motivates the following definition.

Definition 5 (Empirical Mean) *Let*

$$\mathsf{sumCoemit}_w(\sigma, M_j, q) = \sum_{i \text{ s.t. } q(j,i)=q} \mathsf{Coemit}(\sigma, i).$$

The empirical mean of a co-emission probability *is defined as follows:*

$$\overline{\mathsf{Coemit}}_w(\sigma|M_j, q) = \frac{\mathsf{sumCoemit}_w(\sigma, M_j, q)}{n_w(M_j, q)}, \qquad (8)$$

i.e., the sample average of the co-emission probability when $q \in Q_j$ is visited.

When a state in M_j is visited more than once while emitting w, it does not imply that some other state in M_h is also visited more than once. In other words, if there are positions $i \neq \ell$ such that $q(j, i) = q(j, \ell)$ then it does not have to follow that $q(h, i) = q(h, \ell)$ for another machine M_h. Thus, even when M_j and the value of $q(j, i)$ are fixed, $\mathsf{Coemit}(\sigma, i)$ fluctuates. The empirical mean is the average taken over such fluctuating co-emission probabilities.

4.3 Main Theorem and Convexity

Theorems 2 and 3 are our main results. We simplify the proofs by assuming that D consists of a single sentence. That is, in both theorems, we consider $D = \{w\ltimes\}$. We can do this without loss of generality because any finite sequence of strings D drawn i.i.d. from a stochastic language can be converted into a single sentence

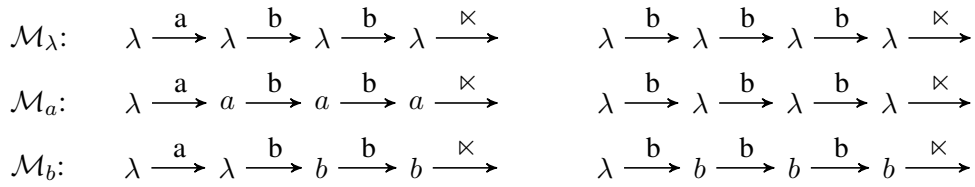

Figure 4: The paths of {abb$\ltimes$,bbb$\ltimes$} through the 2-set of of SD-PDFAs with $\Sigma = \{a, b\}$.

$$\begin{aligned}
&\mathsf{freq}_D(a|\mathcal{M}_\lambda, \lambda) = 1/8 && \mathsf{freq}_D(a|\mathcal{M}_a, \lambda) = 1/5 && \mathsf{freq}_D(a|\mathcal{M}_a, a) = 0/3,\\
&\mathsf{freq}_D(b|\mathcal{M}_\lambda, \lambda) = 5/8 && \mathsf{freq}_D(b|\mathcal{M}_a, \lambda) = 3/5 && \mathsf{freq}_D(b|\mathcal{M}_a, a) = 2/3,\\
&\mathsf{freq}_D(\ltimes|\mathcal{M}_\lambda, \lambda) = 2/8 && \mathsf{freq}_D(\ltimes|\mathcal{M}_a, \lambda) = 1/5 && \mathsf{freq}_D(\ltimes|\mathcal{M}_a, a) = 1/3,\\
& && \mathsf{freq}_D(a|\mathcal{M}_b, \lambda) = 1/3 && \mathsf{freq}_D(a|\mathcal{M}_b, b) = 3/5,\\
& && \mathsf{freq}_D(b|\mathcal{M}_b, \lambda) = 2/3 && \mathsf{freq}_D(b|\mathcal{M}_b, b) = 0/5,\\
& && \mathsf{freq}_D(\ltimes|\mathcal{M}_b, \lambda) = 0/3 && \mathsf{freq}_D(\ltimes|\mathcal{M}_b, b) = 2/5,
\end{aligned}$$

Figure 5: Frequency computations with D={abb$\ltimes$,bbb$\ltimes$} and the 2-set of of SD-PDFAs in Figure 4.

without changing the probability of its production. To see why, we can adjust the transition function of each PDFA $\mathcal{M}_j$ so that $\delta_j(q, \ltimes) = q_{0j}$ for each $q \in Q_j$. In other words, once $\ltimes$ is emitted, the machines reset to their start states. Then for any $D = \{w_1\ltimes, \cdots, w_k\ltimes\}$, we have $P(D) = P(\mathtt{concat}(D))$ where $\mathtt{concat}(D) = w_1 \ltimes w_2 \ltimes \cdots \ltimes w_k\ltimes$. Thus, $w\ltimes$ in both theorems can be understood as $\mathtt{concat}(D)$.

Theorem 2 *Suppose that $P(w\ltimes)$ is defined as Equation 6 for a product machine $\bigotimes_{1\le j\le K} M_j$ and a word w. Then, $\partial P(w\ltimes)/\partial T_j = 0$ holds for all j if and only if the following equation is satisfied for all $1 \le j \le K$:*

$$\mathsf{freq}_w(\sigma|M_j, q) = \overline{\mathsf{Coemit}}_w(\sigma|M_j, q)\ .$$

From Theorem 3, it will then follow that $T_1, \ldots T_K$ are the MLE.

Proof By taking the log of Eq. 6 , we have

$$\begin{aligned}
\log P(w\ltimes) = \sum_{i=1}^{N+1} \Bigg(&\sum_{j=1}^{K} \log T_j(q(j,i), \sigma_i)\\
&- \log \sum_{\sigma'\in\Sigma} \prod_{j=1}^{K} T_j(q(j,i), \sigma') \Bigg)\\
= \sum_{i=1}^{N+1} &\sum_{j=1}^{K} \log T_j(q(j,i), \sigma_i)\\
- \sum_{i=1}^{N+1} &\log \sum_{\sigma'\in\Sigma} \prod_{j=1}^{K} T_j(q(j,i), \sigma').
\end{aligned}$$

We differentiate this by a log emission probability $\log T_h(q, \sigma)$ for some $1 \le h \le K$. Let

$$A = \frac{\partial}{\partial \log T_h(q,\sigma)} \sum_{i=1}^{N+1} \sum_{j=1}^{K} \log T_j(q(j,i), \sigma_i)\ ,$$

and

$$B = \frac{\partial}{\partial \log T_h(q,\sigma)} \sum_{i=1}^{N+1} \log \sum_{\sigma'\in\Sigma} \prod_{j=1}^{K} T_j(q(j,i), \sigma')\ .$$

Then

$$\frac{\partial}{\partial \log T_h(q,\sigma)} \log P(w\ltimes) = A - B.$$

First, we calculate A. Since

$$\frac{\partial T_j(q(j,i),\sigma_i)}{\partial \log T_h(q,\sigma)} = \begin{cases} 1 & \text{if } \langle M_h, q, \sigma\rangle \\ & = \langle M_j, q(j,i), \sigma_i\rangle\ , \\ 0 & \text{otherwise,} \end{cases}$$

we have

$$\begin{aligned}
A &= \sum_{i=1}^{N+1} \sum_{j=1}^{K} \mathbf{I}\big[\langle M_h, q, \sigma\rangle = \langle M_j, q(j,i), \sigma_i\rangle\big]\\
&= \sum_{i=1}^{N+1} \mathbf{I}\big[\langle q, \sigma\rangle = \langle q(h,i), \sigma_i\rangle\big]\\
&= m_w(M_h, q, \sigma) \qquad (9)
\end{aligned}$$

where $\mathbf{I}[\,\cdot\,]$ denotes the indicator function and $m_w(M_h, q, \sigma)$ is defined as in Definition 4.

$$
\begin{aligned}
B =& \frac{\partial}{\partial \log T_h(q,\sigma)} \sum_{i=1}^{N+1} \log \left(\sum_{a\in\Sigma} \prod_{j=1}^{K} T_j(q(j,i),a) \right) \\
=& \sum_{i=1}^{N+1} \frac{\frac{\partial}{\partial \log T_h(q,\sigma)} \sum_{a\in\Sigma} \prod_{j=1}^{K} T_j(q(j,i),a)}{\sum_{a\in\Sigma} \prod_{j=1}^{K} T_j(q(j,i),a)} \\
=& \sum_{i=1}^{N+1} \frac{\frac{\partial}{\partial \log T_h(q,\sigma)} \sum_{a\in\Sigma} \exp\left(\sum_{j=1}^{K} \log T_j(q(j,i),a)\right)}{\sum_{a\in\Sigma} \prod_{j=1}^{K} T_j(q(j,i),a)} \\
=& \sum_{i=1}^{N+1} \frac{\sum_{a\in\Sigma} \left(\exp\left(\sum_{j=1}^{K} \log T_j(q(j,i),a)\right) \sum_{j=1}^{K} \frac{\partial \log T_h(q(j,i),a)}{\partial \log T_h(q,\sigma)} \right)}{\sum_{a\in\Sigma} \prod_{j=1}^{K} T_j(q(j,i),a)} \\
=& \sum_{i=1}^{N+1} \frac{\sum_{a\in\Sigma} \left(\prod_{j=1}^{K} T_j(q(j,i),a) \sum_{j=1}^{K} \frac{\partial \log T_j(q(j,i),a)}{\partial \log T_h(q,\sigma)} \right)}{\sum_{a\in\Sigma} \prod_{j=1}^{K} T_j(q(j,i),a)} \\
=& \sum_{i=1}^{N+1} \sum_{a\in\Sigma} \left(\frac{\prod_{j=1}^{K} T_j(q(j,i),a)}{\sum_{b\in\Sigma} \prod_{j=1}^{K} T_j(q(j,i),b)} \sum_{j=1}^{K} \frac{\partial \log T_j(q(j,i),a)}{\partial \log T_h(q,\sigma)} \right)
\end{aligned}
$$

Figure 6: Initial calculation of B in the proof of Theorem 2.

Second, we calculate B as shown in Figure 6. There are two large terms in the large parentheses in the last line of the calculation of B in Figure 6. The first one is is the co-emission probability by Equation 5. Thus B =

$$\sum_{i=1}^{N+1} \sum_{a\in\Sigma} \sum_{j=1}^{K} \mathsf{Coemit}(a,i) \frac{\partial \log T_j(q(j,i),a)}{\partial \log T_h(q,\sigma)}.$$

Recall that

$$\frac{\partial \log T_j(q(j,i),a)}{\partial \log T_h(q,\sigma)}$$

equals

$$\mathbf{I}[\langle M_h, q, \sigma\rangle = \langle M_j, q(j,i), a\rangle].$$

This indicator function equals $\mathbf{I}[\,h = j\,]\mathbf{I}[\,q = q(j,i)\,]\mathbf{I}[\,\sigma = a\,]$. Abbreviating $\mathbf{I}[\,h = j\,]$ with $\mathbf{I}_1$, $\mathbf{I}[\,q = q(j,i)\,]$ with $\mathbf{I}_2$, and $\mathbf{I}[\,\sigma = a\,]$ with $\mathbf{I}_3$, we see that

$$
\begin{aligned}
&\sum_{a\in\Sigma} \sum_{j=1}^{K} \mathsf{Coemit}(a,i)\, \mathbf{I}_1\mathbf{I}_2\mathbf{I}_3 \\
=& \sum_{j=1}^{K} \mathsf{Coemit}(\sigma,i)\, \mathbf{I}_1\mathbf{I}_2 \\
=& \mathsf{Coemit}(\sigma,i)\, \mathbf{I}[q = q(h,i)].
\end{aligned}
$$

We conclude that

$$
\begin{aligned}
B &= \sum_{i=1}^{N+1} \mathsf{Coemit}(\sigma,i)\mathbf{I}[q = q(h,i)] \\
&= \sum_{i \text{ s.t. } q(h,i)=q} \mathsf{Coemit}(\sigma,i) \\
&= \mathsf{sumCoemit}_w(\sigma, M_h, q). \qquad (10)
\end{aligned}
$$

By plugging our calculations of A (Eq. 9) and B (Eq. 10) into $A = B$ and dividing the both sides by $n_w(M_h, q)$, we obtain the result

$$\mathrm{freq}_w(\sigma|M_h,q) = \overline{\mathsf{Coemit}}_w(\sigma|M_h,q)$$

from the definitions of the relative frequency of an emission (Eq. 7) and the empirical mean of a co-emission probability (Eq. 8). This concludes the proof of Theorem 2. □

Next we prove that maximizing $P(w)$ is a *convex optimization problem* to ensure that the solution is the maximum point.

Following Boyd and Vandenberghe (2004), A set of points C in $\mathbb{R}^n$ is *convex* if the line segment between any two points in C also lies in C. Formally, C is *convex* provided for any $x_1, x_2 \in C$ and any t with $0 \le t \le 1$, we have $tx_1 + (1-t)x_2 \in C$. A function $f : \mathbb{R}^n \to \mathbb{R}$ is *convex* if

the domain of f is a convex set and if for all x, y in the domain of f, and t with $0 \leq t \leq 1$, we have $f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$. We say f is *concave* if $-f$ is convex.

Recall from section 2.4 that the likelihood of a sequence of data D to a stochastic language L belonging to a class with parameters Θ is $\mathsf{lhd}(D \mid \Theta) = \prod_{w \in D} P_L(w)$. The likelihood function is a function $f : \mathbb{R}^n \to \mathbb{R}$ where n is the number of parameters $|\Theta|$.

Let $\tau_{j,q,\sigma}$ denote $\log T_j(q, \sigma)$; i.e. the log of some parameter in Θ. There are $n = |\Sigma| \sum_{j=1}^{K} |Q_j|$ parameters in Θ since $\sigma \in \Sigma$, $1 \leq j \leq K$, and $q \in Q_j$. This τ can be thought of as a vector in $\mathbb{R}^n$.

The problem of maximizing $P(w\ltimes)$ is the same as minimizing $-\log P(w\ltimes)$ as a function of τ. We show that $\log P(w\ltimes)$ is concave with respect to $\log T_j(q, \sigma)$ (Theorem 3). If so, it is true that the solution shown in Theorem 2 is a global maximum.

Theorem 3 *$\log P(w\ltimes)$ is concave with respect to $\tau \in \mathbb{R}^n$.*

Proof By taking the log of Eq. 6 , we have $\log P(w\ltimes) =$

$$\sum_{i=1}^{N+1} \left(\sum_{j=1}^{K} \log T_j(q(j,i), \sigma_i) - \log \sum_{a \in \Sigma} \prod_{j=1}^{K} T_j(q(j,i), a) \right).$$

Substituting in τ, it follows that $\log P(w\ltimes) =$

$$\sum_{i=1}^{N+1} \left(\sum_{j=1}^{K} \tau_{j,q(j,i),\sigma_i} - \log \sum_{a \in \Sigma} \prod_{j=1}^{K} \exp(\tau_{j,q(j,i),a}) \right).$$

Since

$$\prod_{j=1}^{K} \exp(\tau_{j,q(j,i),a}) = \exp\left(\sum_{k=1}^{K} \tau_{k,q(j,i),a} \right),$$

and by letting $g_a(\tau) = \sum_j \tau_{j,q(j,i),a}$, we obtain $\log P(w\ltimes) =$

$$\sum_{i=1}^{N+1} \left(g_{\sigma_i}(\tau) - \log \sum_{a \in \Sigma} \exp\left(g_a(\tau)\right) \right).$$

Generally speaking, a composition $f(x) = h(g_1(x), \cdots, g_k(x))$ obeys the following rule: f is convex if h is convex, h is non-decreasing in each argument, and g_i is convex (see vector composition in Boyd and Vandenberghe, 2004, section 3.2.4)). Furthermore, it is known that $\log \sum \exp(\cdot)$ is convex (see section 3.1.5), and $\log \sum \exp(\cdot)$ is non-decreasing in each argument since both $\exp(\cdot)$ and $\log(\cdot)$ are non-decreasing. In addition, $g_a(\cdot)$ is both convex and concave since every linear function is so from the definition (see section 3.1.1). Thus, $\log \sum_a \exp(g_a(\cdot))$ is convex, and $-\log \sum_a \exp(g_a(\cdot))$ is concave.

Finally, from the fact that non-negative weighted sum preserves convexity and concavity (Boyd and Vandenberghe, 2004, section 3.2.1), $\log P(w\ltimes)$ is concave. □

It follows that the negative log of $P(w\ltimes)$ is convex.

It is noteworthy to point out that establishing concavity does not mean the solution is unique. In fact, the solutions can be a set of points. An example FRPD class illustrating this is one which contains two PDFA $\mathcal{M}_1$ and $\mathcal{M}_2$ with the same structure. For example suppose each had exactly one state with self-loop transitions for every symbol in Σ. The co-emission product $\mathcal{M}_1 \bigotimes \mathcal{M}_2$ does not uniquely factorize though the above theorem establishes its convexity.

Of course it is also of interest to know when the solution is unique. In this case, we have to show the negative log probability is strictly convex except for multiplying the emission probability by a constant. We leave this as an area of future research.

5 Optimization and Time Complexity

In this section, we discuss the time complexity and also how to optimize. From the proof of Theorem 2, we have the following fact immediately.

Corollary 1 *The update equation for maximization of $\log P(w\ltimes)$ is represented as:* $\log T_j(q, \sigma) :=$

$$\log T_j(q, \sigma) + \eta \left(\mathsf{freq}_w(\sigma | M_j, q) - \overline{\mathsf{Coemit}}_w(\sigma | M_j, q) \right) \quad (11)$$

if the simplest gradient method is applied, and where η is the step size. The time complexity for each update is $O(NK|\Sigma|)$.

The time complexity for $\mathsf{freq}_w(\sigma | M_j, q)$ and $\overline{\mathsf{Coemit}}_w(\sigma | M_j, q)$ are shown in Lemma 1 and Lemma 2. The time complexity for

$\overline{\mathsf{Coemit}}_w(\sigma|M_j, q)$ is a little higher than that of $\mathsf{freq}_w(\sigma|M_j, q)$.

Lemma 1 *For all M_j and $q \in Q_j$, $\mathrm{freq}_w(\sigma|M_j, q)$ are computed in the time $O(NK)$.*

Proof We trace all machines while they are emitting $\sigma_1, \cdots, \sigma_N$. Suppose that machines are at $q(1, i), \cdots, q(K, i)$ after $\sigma_1, \cdots, \sigma_{i-1}$ are emitted sequentially. For each step i, for all machines M_j, we have to update the counting for the pair of $q(k, i)$ and σ_i, in order to calculate $m_w(M_j, q, \sigma)$. So the computational cost for each step i is $O(K)$. □

Lemma 2 *For all M_j and $q \in Q_j$, $\overline{\mathsf{Coemit}}_w(\sigma|M_j, q)$ are computed in the time $O(NK|\Sigma|)$.*

Proof We trace all machines while they are emitting $\sigma_1, \cdots, \sigma_N$. Suppose that machines are at $q(1, i), \cdots, q(K, i)$ after $\sigma_1, \cdots, \sigma_{i-1}$ are emitted sequentially. The critical part is calculating $\mathsf{sumCoemit}(\sigma)_{\langle M_j, q\rangle}(w)$. For each step i, we have to update emission probabilities for all pairs of M_j and $\sigma \in \Sigma$. This update is in the time $O(K|\Sigma|)$. Thus, the time complexity for calculating $\mathsf{sumCoemit}_w(\sigma, M_j, q)$ is $O(NK|\Sigma|)$. □

6 Conclusion

The negative log likelihood function associated with a FRPD class C is convex, and it is possible to efficiently find a MLE of any sequences of data generated i.i.d. with respect to C. Essentially, the parameters of the model are found by running the corpus through each of the individual factor PDFAs and calculating the relative frequencies. While this was the approach adopted by Heinz and Rogers (2010) for SP stochastic languages, we have generalized it to sets of finitely many PDFAs.

There are several directions for future research, both theoretical and applied. On the theoretical side, one clear avenue is to better understand these results in terms of probabilistic graphical models (Koller and Friedman, 2009). As a reviewer pointed out, the application of those methods to formal language theory and grammatical inference (de la Higuera, 2010) appears fruitful.

On the applied side, there are several different opportunities. One area of interest is language modeling. The results here permit a modular approach to constructing language models, where certain primitive factors are included or excluded. For example, we expect that language models which incorporate both n-gram models (Jurafsky and Martin, 2008) (which cannot describe long-distance dependencies) and SP stochastic languages (which can describe some kinds of long-distance dependencies) will have lower perplexity, a hypothesis under current investigation. More generally, researchers can use aspects of the subregular hierarchies of languages (Thomas, 1997; Rogers et al., 2013) to identify a range of 'primitive factors' whose DFA models can form the basis of various FRPD classes.

Finally, we are also interested in extending these results to weighted deterministic automata for computing regular relations (Beros and de la Higuera, 2016) or elements of other monoids (Gerdjikov, 2018).

Acknowledgments

We would like to thank two anonymous reviewers for helpful comments, and another anonymous reviewer in particular for making clear the scope of this work, which resulted in a significant revisions to our original submission. This work was supported by NIH grant #R01HD87133-01 to JH and JSPS KAKENHI grant #JP18K11449 to CS.

References

Borja Balle, Rémi Eyraud, Franco M. Luque, Ariadna Quattoni, and Sicco Verwer. 2016. Results of the sequence prediction challenge (SPiCe): a competition on learning the next symbol in a sequence. In *Proceedings of The 13th International Conference on Grammatical Inference*, volume 57 of *JMLR: Workshop and Conference Proceedings*, pages 132–136.

Achilles Beros and Colin de la Higuera. 2016. A canonical semi-deterministic transducer. *Fundamenta Informaticae*, 146(4):431–459.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer.

S. Boyd and L. Vandenberghe. 2004. *Convex optimization*. Cambridge.

Stefan Gerdjikov. 2018. A general class of monoids supporting canonisation and minimisation

of (sub)sequential transducers. In *Language and Automata Theory and Applications - 12th International Conference, LATA 2018, Ramat Gan, Israel, April 9-11, 2018, Proceedings*, pages 143–155.

J. Heinz and J. Rogers. 2010. Estimating Strictly Piecewise Distributions. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 886–896.

Jeffrey Heinz. 2010a. Learning long-distance phonotactics. *Linguistic Inquiry*, 41(4):623–661.

Jeffrey Heinz. 2010b. String extension learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 897–906, Uppsala, Sweden. Association for Computational Linguistics.

Jeffrey Heinz. 2014. Culminativity times harmony equals unbounded stress. In Harry van der Hulst, editor, *Word Stress: Theoretical and Typological Issues*, chapter 8. Cambridge University Press, Cambridge, UK.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. In Larry Hyman and Frans Plank, editors, *Phonological Typology*, Phonetics and Phonology, chapter 5, pages 126–195. De Gruyter Mouton.

Jeffrey Heinz, Colin de la Higuera, and Menno van Zaanen. 2015. *Grammatical Inference for Computational Linguistics*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool.

Jeffrey Heinz, Anna Kasprzik, and Timo Kötzing. 2012. Learning with lattice-structured hypothesis spaces. *Theoretical Computer Science*, 457:111–127.

Jeffrey Heinz and James Rogers. 2013. Learning subregular classes of languages with factored deterministic automata. In *Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13)*, pages 64–71, Sofia, Bulgaria. Association for Computational Linguistics.

Colin de la Higuera. 2010. *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press.

John Hopcroft, Rajeev Motwani, and Jeffrey Ullman. 2001. *Introduction to Automata Theory, Languages, and Computation*. Boston, MA: Addison-Wesley.

Daniel Jurafsky and James Martin. 2008. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*, 2nd edition. Prentice-Hall, Upper Saddle River, NJ.

Daphne Koller and Nir Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Robert Malouf. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, pages 1–7. Association for Computational Linguistics.

James Rogers, Jeffrey Heinz, Gil Bailey, Matt Edlefsen, Molly Visscher, David Wellcome, and Sean Wibel. 2010. On languages piecewise testable in the strict sense. In *The Mathematics of Language*, volume 6149 of *Lecture Notes in Artifical Intelligence*, pages 255–265. Springer.

James Rogers, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert, and Sean Wibel. 2013. Cognitive and sub-regular complexity. In *Formal Grammar*, volume 8036 of *Lecture Notes in Computer Science*, pages 90–108. Springer.

Chihiro Shibata and Jeffrey Heinz. 2016. Predicting sequential data with lstms augmented with strictly 2-piecewise input vectors. In *Proceedings of The 13th International Conference on Grammatical Inference*, volume 57 of *JMLR: Workshop and Conference Proceedings*, pages 137–142.

Wolfgang Thomas. 1997. Languages, automata, and logic. In *Handbook of Formal Languages*, volume 3, chapter 7. Springer.

Enrique Vidal, Franck Thollard, Colin de la Higuera, Francisco Casacuberta, and Rafael C. Carrasco. 2005a. Probabilistic finite-state machines-part I. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1013–1025.

Enrique Vidal, Frank Thollard, Colin de la Higuera, Francisco Casacuberta, and Rafael C. Carrasco. 2005b. Probabilistic finite-state machines-part II. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1026–1039.