Learning Probabilistic Residual Finite State Automata

Yann Esposito[1], Aurélien Lemay[2], François Denis[1], and Pierre Dupont[3]

[1] LIF, UMR 6166, Université de Provence, Marseille, France.
esposito@cmi.univ-mrs.fr, fdenis@cmi.univ-mrs.fr
[2] GRAPPA-LIFL, Université de Lille, Lille, France.
lemay@lifl.fr
[3] INGI, University of Louvain, Louvain-la-Neuve, Belgium.
pdupont@info.ucl.ac.be

Abstract. We introduce a new class of probabilistic automata: Probabilistic Residual Finite State Automata. We show that this class can be characterized by a simple intrinsic property of the stochastic languages they generate (the set of residual languages is finitely generated by residuals) and that it admits canonical minimal forms. We prove that there are more languages generated by PRFA than by Probabilistic Deterministic Finite Automata (PDFA). We present a first inference algorithm using this representation and we show that stochastic languages represented by PRFA can be identified from a characteristic sample if words are provided with their probabilities of appearance in the target language.

Introduction

In the field of machine learning, most realistic situations deal with data provided by a stochastic source and probabilistic models, such as Hidden Markov Models (HMMs) or probabilistic automata (PA), become increasingly important. For example, speech recognition, computational biology and more generally, every field where statistical sequence analysis is needed, may use this kind of models. In this paper, we focus on Probabilistic Automata.

A probabilistic automata can be described by its structure (a Finite State Automata) and by a set of continuous parameters (probability to emit a given letter from a given state or to end the generation process). There exist several fairly good methods to adjust the continuous parameters of a given structure to a training set of examples. However the efficient building of the structure from given data is still an open problem. Hence most applications of HMMs or PA assume a fixed model structure, which is either chosen as general as possible (i.e. a complete graph) or *a priori* selected using domain knowledge.

Several learning algorithms, based on previous works in the field of grammatical inference, have been designed to output a deterministic structure (Probabilistic Deterministic Finite State Automata: PDFA) from training data ([CO94], [CO99], [TDdlH00]; see also [Ang88], [SO94] for early works) and several interesting theoretical and experimental results have been obtained. However, unlike

P. Adriaans, H. Fernau, and M. van Zaanen (Eds.): ICGI 2002, LNAI 2484, pp. 77–91, 2002.

to the case of non stochastic languages, DFA structures are not able to represent as many stochastic languages as non deterministic ones. Therefore using these algorithms to infer probabilistic automata structures introduces a strong, and possibly wrong, learning bias.

A new class of non deterministic automata, the Residual Finite State Automata (RFSA), has been introduced in [DLT01b]. RFSA have interesting properties from a language theory point of view, including the existence of a canonical minimal form which can offer a much smaller representation than an equivalent DFA ([DLT01a]). Several learning algorithms that output RFSA have been designed in [DLT00] and [DLT01a]. The present paper describes an extension to these works to deal with stochastic regular languages.

We introduce in Section 1 classical notions of probabilistic automata and stochastic languages. In Section 2 we explain how the definition of residual languages can be extended to stochastic languages, and we define a new class of probabilistic automata: Probabilistic Residual Finite State Automata (PRFA). We prove that this class has canonical minimal representations. In Section 3 we introduce an intrinsic characterization of stochastic languages represented by PDFA: a stochastic language can be represented by a PDFA if and only if the number of its residual languages is finite. We extend this characterization to languages represented by PRFA: a stochastic language can be represented by a PRFA if and only if the set of its residual languages is finitely generated. We prove in Section 4 that the class of languages represented by PRFA is more expressive than the one represented by PDFA. This results is promising as it means that algorithms that would identify stochastic languages represented by PRFA would be able to identify a larger class of languages than PDFA inference algorithms. Section 5 presents a preliminary result along this line: stochastic languages represented by PRFA can be identified from a characteristic sample if words are provided with their actual probabilities of appearance in the target language.

1 Probabilistic Automata and Stochastic Languages

Let Σ be a finite *alphabet* and let Σ^* be the set of finite words built on Σ. A language L is a subset of Σ^*. Let u be a word of Σ^*, the length of u is denoted by $|u|$, the empty word is denoted by ε. Σ^* is ordered in the usual way, i.e. $u \leq v$ if and only if $|u| < |v|$ or $|u| = |v|$ and u is before v in lexicographical order. Let u be a word of Σ^*, v is a *prefix* of u if there exists a word w such that $u = vw$. A language L is *prefixial* if for every word u of L, the set of prefixes of u is a subset of L. Let E be a set, $\mathcal{D}(E) = \left\{(\alpha_e)_{e\in E} \in [0,1]^{\mathrm{Card}(E)} \mid \sum_{e\in E} \alpha_e = 1\right\}$ denotes the set of distributions over E and $\mathcal{D}(\{1,\ldots,n\})$ is denoted by $\mathcal{D}(n)$.

A *stochastic language* L on Σ is a function from Σ^* to $[0,1]$ such that $\sum_{u\in\Sigma^*} L(u) = 1$. We note $p(w \mid L) = L(w)$, or simply $p(w)$ when there is no ambiguity. If W is a set of words, $p(W) = \sum_{w\in W} p(w)$. Let $SL(\Sigma)$ be the set of stochastic languages on Σ.

A *probabilistic finite state automaton (PFA)* is a quintuple $\langle \Sigma, Q, \varphi, \iota, \tau \rangle$ where Q is a finite set of states, $\varphi : Q \times \Sigma \times Q \to [0,1]$ is the transition function, $\iota : Q \to [0,1]$ is the probability for each state to be initial and $\tau : Q \to [0,1]$ is the probability for each state to be terminal. A PFA need satisfy $\sum_{q \in Q} \iota(q) = 1$ and for each state q,

$$\tau(q) + \sum_{a \in \Sigma} \sum_{q' \in Q} \varphi(q, a, q') = 1. \tag{1}$$

Let φ also denote the extension of the transition function, defined on $Q \times \Sigma^* \times Q$ by $\varphi(q, wa, q') = \sum_{q'' \in Q} \varphi(q, w, q'')\varphi(q'', a, q')$ and $\varphi(q, \varepsilon, q') = 1$ if $q = q'$ and 0 otherwise.

We extend φ again on $Q \times 2^{\Sigma^*} \times Q$ by $\varphi(q, U, q') = \sum_{w \in U} \varphi(q, w, q')$.

The set of initial states is defined by $Q_I = \{q \in Q \mid \iota(q) > 0\}$, the set of reachable states is defined by $Q_{reach} = \{q \in Q \mid \exists q_I \in Q_I, \varphi(q_I, \Sigma^*, q) > 0\}$ and the set of terminal states is defined by $Q_T = \{q \in Q \mid \tau(q) > 0\}$. We only consider here PFA such that: $\forall q \in Q_{reach}, \exists q_T \in Q_T, \varphi(q, \Sigma^*, q_T) > 0$.

Let $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ be a PFA. Let p_A be the function defined on Σ^* by

$$p_A(u) = \sum_{q, q' \in Q \times Q} \iota(q)\varphi(q, u, q')\tau(q'). \tag{2}$$

It can be proved that p_A is a stochastic language on Σ which is called the stochastic language generated by A.

For every state q, we denote by A_q the PFA $A_q =< \Sigma, Q, \varphi, \iota_q, \tau >$ where $\iota_q(q) = 1$. We denote $p_{A_q}(w)$ by $p_A(w|q)$.

A *probabilistic deterministic finite state automaton (PDFA)* is a PFA $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ with a single initial state and such that for any state q and for every letter a, there is at much one state q' such that $\varphi(q, a, q') > 0$.

The class of *stochastic regular languages* on Σ is denoted by $L_{PFA}(\Sigma)$. It consists of all stochastic languages generated by probabilistic finite automata. Also, the class of *stochastic deterministic regular languages* on Σ is denoted by $L_{PDFA}(\Sigma)$. It consists of all stochastic languages generated by probabilistic deterministic finite automata.

2 Probabilistic Residual Finite State Automata (PRFA)

We introduce in this section the class of probabilistic residual finite state automata (PRFA). This class extends the notion of RFSA defined in [DLT01b]. We extend the notion of residual language for stochastic languages and we define a class of probabilistic automata based on this new notion. We study its properties and prove that the class of PRFA also defines a new class of stochastic languages strictly including the class of stochastic deterministic regular languages. PRFA also have a canonical form, a property in common with RFSA and PDFA.

Let L be a language, and let u be a word. The *residual language* of L with respect to u is $u^{-1}L = \{v \mid uv \in L\}$. We extend this notion to the stochastic

case as follows. Let L be a *stochastic* language, the *residual language* of L with respect to u, also denoted by $u^{-1}L$, associates to every word w the probability $p(w|u^{-1}L) = p(uw|L)/p(u\Sigma^*|L)$ if $p(u\Sigma^*|L) \neq 0$. If $p(u\Sigma^*|L) = 0$, $u^{-1}L$ is not defined. Let $L_{fr}(\Sigma)$ be the class of stochastic languages on Σ having a finite number of residual languages.

A RFSA recognizing a regular language L is an automaton whose states are associated with residual languages of L. We propose here a similar definition in the stochastic case.

Definition 1. *A PRFA is a PFA $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ such that every state defines a residual language. More formally*

$$\forall q \in Q, \exists u \in \Sigma^*, L_q = u^{-1}p_A. \tag{3}$$

The class of stochastic residual regular languages *on the alphabet Σ is denoted by $L_{PRFA}(\Sigma)$. It consists of all stochastic languages generated by probabilistic residual finite automata on Σ. Figures 2 and 3.4 show two examples of PRFA.*

Let L be a stochastic language on Σ, let U be a finite subset of Σ^* and let $w \in \Sigma^*$. We define the set of *linearly generated residual languages* of L associated with U:

$$LG_L(U) = \left\{ l \in SL(\Sigma) \mid \exists (\alpha_u)_{u\in U} \in \mathcal{D}(U), l = \sum_{u\in U} \alpha_u.u^{-1}L \right\} \tag{4}$$

and we define *the set of linear decompositions* of w associated with U in L:

$$Decomp_L(w, U) = \left\{ (\alpha_u)_{u\in U} \in \mathcal{D}(U) \mid w^{-1}L = \sum_{u\in U} \alpha_u.u^{-1}L \right\}. \tag{5}$$

Let U be a finite set of words. We say that U is a *finite residual generator of* L if every residual language of L belongs to $LG_L(U)$. U is short if and only if for all word u of U there is no smaller word v such that $v^{-1}L = u^{-1}L$. Let $L_{frg}(\Sigma)$ be the class of stochastic languages on Σ having a finite residual generator. Note that $L_{fr} \subseteq L_{frg}$.

We prove now that we can associate with every language L generated by a PRFA a unique minimal short residual generator, called the *base* $\mathcal{B}_L$ of L.

Remark 1. Let $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ be a PRFA generating a language L. We can observe that $\{u \in \Sigma^* \mid \exists q \in Q, L_q = u^{-1}L \wedge \nexists v < u, u^{-1}L = v^{-1}L\}$ is a finite short residual generator of L with the same cardinality as Q. Therefore finding a minimal residual generator of L gives us the possibility to construct a minimal PRFA generating L.

Theorem 1. *Base unicity*

Every language L of L_{frg} has a unique minimal short residual generator denoted by $\mathcal{B}_L$.

Proof. Let $U = \{u_1, \ldots, u_l\}$ and $V = \{v_1, \ldots, v_m\}$ be two minimal short residual generators of L and suppose that $l \geq m$. We prove that $U = V$. From the definition of a residual generator, we can deduce that for all i in $\{1, \ldots, l\}$ there exists $\alpha_{i,j} \in \mathcal{D}(l)$ and for all j in $\{1, \ldots, m\}$ there exists $\beta_{j,k} \in \mathcal{D}(m)$ such that $u_i^{-1}L = \sum_{j=1}^{m} \alpha_{i,j} v_j^{-1}L$ and $v_j^{-1}L = \sum_{k=1}^{l} \beta_{j,k} u_k^{-1}L$. Therefore

$$u_i^{-1}L = \sum_{j=1}^{m} \alpha_{i,j} \left(\sum_{k=1}^{l} \beta_{j,k} u_k^{-1}L \right) = \sum_{k=1}^{l} \left(\sum_{j=1}^{m} \alpha_{i,j}\beta_{j,k} \right) u_k^{-1}L.$$

This implies that $\sum_{j=1}^{m} \alpha_{i,j}\beta_{j,k} = 1$ if $i = k$ and 0 otherwise. Indeed, if there exist $(\gamma_k)_{1 \leq k \leq l} \in \mathcal{D}(l)$ such that $u_i^{-1}L = \sum_{k=1}^{l} \gamma_k u_k^{-1}L$ with $\gamma_i \neq 1$ then

$$u_i^{-1}L = \gamma_i u_i^{-1}L + \sum_{k=1, k\neq i}^{l} \gamma_k u_k^{-1}L = \sum_{k=1, k\neq i}^{n} \frac{\gamma_k}{1-\gamma_i} u_k^{-1}L.$$

Hence $U \setminus \{u_i\}$ would be a residual generator which contradicts the fact that U is a minimal residual generator.

Let j_0 be such that $\alpha_{i,j_0} \neq 0$. Thus for all $k \neq i$, $\alpha_{i,j_0}\beta_{j_0,k} = 0$. Hence $\beta_{j_0,k} = 0$ which implies that $\beta_{j_0,i} = 1$. As a consequence, $v_{j_0}^{-1}L = \sum_{k=1}^{l} \beta_{j_0,k} u_k^{-1}L = u_i^{-1}L$. Finally for all i there exists j such that $u_i^{-1}L = v_j^{-1}L$, that is $U = V$. □

As we can associate with every language L of L_{frg} a base $\mathcal{B}_L$, we can build a minimal PRFA from $\mathcal{B}_L$ using the following definition. We call this automaton a minimal PRFA of L and we prove that it is a PRFA generating L and having the minimal number of states.

Definition 2. *Let L be a language of L_{frg}. For all word u, let $(\alpha_{u,v})_{v \in \mathcal{B}_L}$ be an element of $Decomp_L(u, \mathcal{B}_L)$ such that $u^{-1}L = \sum_{v \in \mathcal{B}_L} \alpha_{u,v} v^{-1}L$. A* minimal PRFA *of L is a PFA $A = \langle \Sigma, \mathcal{B}_L, \varphi, \iota, \tau \rangle$ such that*

$$\begin{gathered} \forall u \in \mathcal{B}_L,\ \iota(u) = \alpha_{\varepsilon,u}, \\ \forall (u, u') \in \mathcal{B}_L, \forall a \in \Sigma,\ \varphi(u, a, u') = \alpha_{ua,u'} \cdot p(a\Sigma^* | u^{-1}L), \\ \forall u \in \mathcal{B}_L,\ \tau(u) = p(\varepsilon | u^{-1}L). \end{gathered} \tag{6}$$

Theorem 2. *Let L be a language of L_{frg}, a minimal PRFA of L generates L and has the minimal number of states.*

Proof.
A is minimal:

It is clear that a PRFA generating L must have at least as many states as words in the base of L, i.e. at least as many states as A.
A is a PRFA generating L:

By construction we have for all u in $\mathcal{B}_L$, $p_A(\varepsilon | u) = p(\varepsilon | u^{-1}L)$.

Now suppose that for any word w such that $|w| \leq k$ and for all u in $\mathcal{B}_L$, we have $p_A(w|u) = p(w|u^{-1}L)$. Then considering the letter a

$$\begin{aligned}
p_A(aw|u) &= \textstyle\sum_{u' \in Q} \varphi(u, a, u') p_A(w|u') \\
&= \textstyle\sum_{u' \in Q} \alpha_{ua,u'} p(a\Sigma^*|u^{-1}L) p_A(w|u') \\
&= \textstyle\sum_{u' \in Q} \alpha_{ua,u'} p(a\Sigma^*|u^{-1}L) p(w|u'^{-1}L) \text{ (by the induction hypothesis)} \\
&= p(a\Sigma^*|u^{-1}L) p(w|ua^{-1}L) \qquad (\text{by } (ua)^{-1}L = \textstyle\sum_{u' \in \mathcal{B}_L} \alpha_{ua,u'} u'^{-1}L) \\
&= p(aw|u^{-1}L).
\end{aligned}$$

We proved that $\forall u \in \mathcal{B}_L, u^{-1}L = p_{A_u}$. Given that

$$L = \varepsilon^{-1}L = \sum_{u \in \mathcal{B}_L} \alpha_{\varepsilon,u} u^{-1}L = \sum_{u \in \mathcal{B}_L} \alpha_{\varepsilon,u} p_{A_u} = \sum_{u \in \mathcal{B}_L} \iota(u) p_{A_u} = p_A$$

A generates L. □

One can observe that the above definition does not define a unique minimal PRFA, but a family of minimal PRFA. Every minimal PRFA is built on the base of the language, but probabilities on the transition depend on the choice of the values $\alpha_{u,v}$.

3 Characterization of Stochastic Languages

We propose here a characterization of stochastic languages represented by PDFA based on residual languages. We then prove that PRFA also have a similar characterization.

3.1 PDFA Generated Languages

The class of stochastic languages having a finite number of residual languages is equal to the class of stochastic languages generated by PDFA.

Theorem 3. $L_{fr} = L_{PDFA}$

Proof.
(1) Let $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ be a PDFA. For any state $q \in Q_{reach}$, we note u_q the smallest word such that $\varphi(q_0, u_q, q) > 0$ where $Q_I = \{q_0\}$. For any $w \in \Sigma^*$, if $w^{-1}L$ is defined, there exists a unique q such that $\varphi(q_0, w, q) > 0$ and therefore $w^{-1}L = u_q^{-1}L$.

(2) Let $L \in L_{fr}(\Sigma)$, let us construct a PDFA generating L. Let U be a minimal set of words such that for any word w such that $w^{-1}L$ is defined, there is a word u in U such that $w^{-1}L = u^{-1}L$.

For such words w, the word u of U such that $w^{-1}L = u^{-1}L$ is denoted by u_w. We construct $A = \langle \Sigma, U, \varphi, \iota, \tau \rangle$ with $\iota(u_\varepsilon) = 1$, $\tau(u) = p(\varepsilon|u^{-1}L)$ for all $u \in U$ and $\varphi(u, a, u') = p(a\Sigma^*|u^{-1}L)$ if $u' = u_{ua}$ and 0 otherwise, for all words u and u' of U and for every letter a.

In order to prove that $p_A = L$, it is sufficient to prove that $\forall w \in \Sigma^*$, $p_A(w\Sigma^*) = p(w\Sigma^*|L)$. By construction, for all word u in U:

$$p_A(\varepsilon\Sigma^*|u) = p(\varepsilon\Sigma^*|u^{-1}L) = 1 \text{ and } \forall a \in \Sigma, p_A(a\Sigma^*|u) = p(a\Sigma^*|u^{-1}L)$$

Let us assume that for every w in $\Sigma^{\leq k}$ and for every u in U, $p_A(w\Sigma^*|u) = p(w\Sigma^*|u^{-1}L)$. Let $w \in \Sigma^k$, $a \in \Sigma$ and $u \in U$, If $p(ua\Sigma^*|L) = 0$, we have $p_A(aw\Sigma^*|u) = p(aw\Sigma^*|u^{-1}L) = 0$ and otherwise let u' the unique word of U such that $\varphi(u, a, u') > 0$

$$\begin{aligned} p_A(aw\Sigma^*|u) &= \varphi(u, a, u')p_A(w\Sigma^*|u') \\ &= p_A(a\Sigma^*|u)p_A(w\Sigma^*|u') \\ &= p(a\Sigma^*|u^{-1}L)p(w\Sigma^*|u'^{-1}L) \text{ (by the induction hypothesis)} \\ &= p(aw\Sigma^*|u^{-1}L) \text{ since } (ua)^{-1}L = u'^{-1}. \end{aligned}$$

Then $\forall u \in U, p_{A_u} = u^{-1}L$. In particular, as $\iota(u_\varepsilon) = 1$, $p_A = p_{A_{u_\varepsilon}} = \varepsilon^{-1}L = L$. Therefore, A is a PDFA generating the language L. □

We propose here a similar characterization for L_{PRFA}, also based on intrinsic properties of the associated languages.

3.2 PRFA Generated Languages

We prove that L_{PRFA} is the class of languages having finite residual generators; it includes languages which may have an infinite number of residual languages.

Theorem 4. $L_{frg} = L_{PRFA}$.

Proof. Let $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$, we prove that $p_A \in L_{frg}(\Sigma)$. For all words w and u such that $p_A(u\Sigma^*) \neq 0$,

$$\begin{aligned} p_A(w|u^{-1}p_A) &= \left(\sum_{q\in Q} \sum_{q'\in Q_I} \iota(q')\varphi(q', u, q)p_A(w|q)\right) / p_A(u\Sigma^*) \\ &= \sum_{q\in Q} \alpha_q \cdot p_A(w|u_q^{-1}p_A) \end{aligned}$$

where u_q is the smallest word such that $(u_q)^{-1}p_A$ is the stochastic residual language generated by the state q, and $\alpha_q = \sum_{q'\in Q_I} \iota(q')\varphi(q', u, q)/p_A(u\Sigma^*)$. Verify that $\sum_{q\in Q} \alpha_q = 1$.

The converse is clear from Theorem 2. □

4 Expressiveness of L_{PRFA}

In this section, we prove that the class of stochastic languages defined by PRFA is more expressive than the one defined by PDFA, although not as expressive as the one generated by general PFA.

Theorem 5.

$$L_{PDFA} \subsetneq L_{PRFA} \subsetneq L_{PFA}.$$

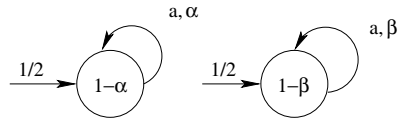

Fig. 1. A PFA generating a language not in L_{PRFA}

Proof. Inclusions are clear, we only have to show the strict inclusions.

(1) $L_{PRFA} \subsetneq L_{PFA}$

Let L be the language generated by the PFA described on Figure 1. As $\Sigma = \{a\}$, all residuals are $(a^n)^{-1}L$. We consider $\alpha = \beta^2$.

$$p(\varepsilon|(a^n)^{-1}L) = \frac{p(a^n|L)}{p(a^n\Sigma^*|L)} = \frac{\alpha^n(1-\alpha)+\beta^n(1-\beta)}{\alpha^n+\beta^n} = 1 - \frac{\alpha^{n+1}+\beta^{n+1}}{\alpha^n+\beta^n}$$
$$= 1 - \frac{\beta^{2(n+1)}+\beta^{n+1}}{\beta^{2n}+\beta^n} = 1 - \beta^2 - \frac{\beta-\beta^2}{\beta^n+1}$$

Hence $p(\varepsilon|(a^n)^{-1}L)$ is a strictly decreasing function (as $0 < \beta < 1$). Suppose that p_A has a finite residual generator U. Let $u_0 = a^{n_0} \in U$ such that for all u in U, $p(\varepsilon|u_0^{-1}L) \leq p(\varepsilon|u^{-1}L)$ and let $n > n_0$. Then there exists $(\alpha_u^n)_{u\in U} \in \mathcal{D}(\mathrm{Card}(U))$ such that

$$p(\varepsilon|(a^n)^{-1}L) = \sum_{u\in U} \alpha_u^n p(\varepsilon|u^{-1}L) \geq \sum_{u\in U} \alpha_u^n p(\varepsilon|u_0^{-1}L) = p(\varepsilon|u_0^{-1}L)$$

which is impossible since $p(\varepsilon|(a^n)^{-1}L)$ is strictly decreasing.

(2) $L_{PDFA} \subsetneq L_{PRFA}$

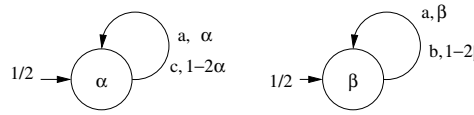

Fig. 2. A PRFA generating a language not in L_{PDFA}.

Let L be the language generated by the PRFA described on Figure 2. Let us consider the case when $\alpha = \beta^2$, then

$$p(a^m|L) = \frac{\alpha^{n+1} + \beta^{n+1}}{2} = \frac{\beta^{2n+2} + \beta^{n+1}}{2}$$

and

$$p(\varepsilon|(a^n)^{-1}L) = \frac{p(a^n|L)}{p(a^n\Sigma^*|L)} = \frac{\beta^{2(n+1)} + \beta^{n+1}}{\beta^{2n} + \beta^n} = \beta^2 + \frac{\beta - \beta^2}{\beta^n + 1}$$

as $p(\varepsilon|(a^n)^{-1}L)$ is a strictly increasing function ($0 < \beta < 1$), it is clear that the number of residual languages cannot be finite. Therefore it can not be generated by a PDFA. □

5 PRFA Learning

We present in this section an algorithm that identifies stochastic languages. Unlike other learning algorithms, this algorithm takes as input words associated with their true probability to be a prefix in the target stochastic language. In this context, we prove that any target stochastic language L generated by a PRFA can be identified. We prove that the sample required for this identification task has a polynomial size as function of the size of the minimal prefix PRFA of L. As $L_{PDFA} \subsetneq L_{PRFA}$, the class of stochastic languages identified in this way strictly includes the class of stochastic languages identified by algorithms based on identification of PDFA. The use of an exact information on the probability of appearance of words is unrealistic. We will further extend our work to cases where probabilities are replaced by sample estimates.

5.1 Preliminary Definitions

Definition 3. *The* minimal prefix set *of a stochastic language L is the prefixial set composed of the words whose associated residual languages cannot be decomposed by residual generated by smaller words. More formally,*

$$Pm(L) = \left\{u \in L \mid p(u\Sigma^*|L) > 0 \wedge u^{-1}L \notin LG_L(\{v \in \Sigma^* \mid v < u\})\right\}. \quad (7)$$

Remark 2. For all word u, $LG_L(\{v \in \Sigma^* \mid v<u\}) = LG_L(\{v \in Pm(L) \mid v<u\})$.

When L is a stochastic language generated by a PRFA, $Pm(L)$ is finite. This set will be the set of states of the PRFA output by our algorithm.

Definition 4. *The* kernel of L *contains $Pm(L)$ and some successors of elements of $Pm(L)$. More formally,*

$$K(L) = \{\varepsilon\} \cup \{wa \in \Sigma^* \mid p(wa\Sigma^*|L) > 0 \wedge w \in Pm(L) \wedge a \in \Sigma\}. \quad (8)$$

$K(L)$ contains the words which will be tested by the algorithm in order to know whether they are states of the output PRFA.

Remark 3. It can be easily be shown that $Pm(L)$ and $K(L)$ are prefixial sets, $\mathcal{B}_L \subseteq Pm(L) \subseteq K(L)$, and therefore $Pm(L)$ and $K(L)$ are finite residual generators of L.

Definition 5. *Prefix PRFA are PRFA based on a prefixial set of words and whose non deterministic transitions only occur on maximal words.*

Let $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ be a PRFA. A is a prefix PRFA *if*

- *Q is a finite prefixial set of Σ^*,*
- *$\varphi(w, a, w') \neq 0 \Rightarrow w' = wa \vee (wa \notin Q \wedge w' < wa)$.*

Example 1. Automaton 4 in Figure 3 is an example of a minimal prefix PRFA. Its set of states is $Pm(L) = \{\varepsilon, a, b\}$, and $K(L) = \{\varepsilon, a, b, aa, ba\}$, where L is the generated language.

Proposition 1. *Every stochastic language L generated by a PRFA can be generated by a prefix PRFA whose set of states is $Pm(L)$. We call them minimal prefix PRFA of L.*

Proof. The proof is similar to the proof of Theorem 2. □

Definition 6. *Let L be a stochastic language, a* rich sample *of L is a set S of couples $(u, p(u\Sigma^*|L)) \in \Sigma^* \times [0,1]$. Let $\pi_1(S)$ denote the set $\{u \in \Sigma^* \mid \exists (u,p) \in S\}$.*

Definition 7. *Let L be a stochastic language, v a word and U and W two finite sets of words such that: $\forall u \in U \cup \{v\}, u^{-1}L$ is defined. $E_L(v, U, W)$ is defined as the linear system composed of:*

1. $0 \leq \alpha_u \leq 1$ *for all* $u \in U$,
2. $\sum_{u \in U} \alpha_u = 1$,
3. $p(w\Sigma^*|v^{-1}L) = \sum_{u \in U} \alpha_u p(w\Sigma^*|u^{-1}L)$ *for every word w in W.*

Definition 8. *Linear systems associated with a rich sample.*

Let L be a stochastic language, v a word and U a finite set of words. Let S be a rich sample of L such that: $\forall u \in U \cup \{v\}, u \in \pi_1(S)$ and $u^{-1}L$ is defined. Let $E_S(v, U) = E_L(v, U, W)$ where $W = \{w \in \Sigma^* | \forall u \in U \cup \{v\}, uw \in \pi_1(S)\}$. Note that $E_S(v, U)$ can be computed from S.

The set of solutions of $E_S(v, U)$ is denoted by $sol(E_S(v, U))$.

We shall use this linear system to test whether a given stochastic residual language $v^{-1}L$ is in $LG_L(U)$.

Definition 9. *A* characteristic sample *of a stochastic language $L \in L_{frg}$ is a rich sample S such that*

$$K(L) \subseteq \pi_1(S) \text{ and } \forall v \in K(L), \text{ let } U_v = \{u \in Pm(L) \mid u < v\}$$
$$sol(E_S(v, U_v)) = Decomp_L(v, U_v).$$

Remark 4. Every rich sample containing a characteristic sample is characteristic.

If S is a characteristic sample of L then for every v in $K(L)$, $sol(E_S(v, U_v)) \neq \emptyset$ is equivalent to $Decomp_L(v, U_v) \neq \emptyset$ which is equivalent to $v^{-1}L \in LG_L(U_v)$.

Lemma 1. *Every language L of L_{frg} has a finite characteristic sample containing $O(Card(Pm(L))^2 Card(\Sigma))$ elements.*

Proof. We note S_∞ the rich sample such that $\pi_1(S_\infty) = \Sigma^*$. It is clear that S_∞ is a characteristic sample. For every v in $K(L)$, solutions of $E_S(v, U_v)$ can be described as the intersection of an affine subspace of $\mathbb{R}^{\mathrm{Card}(U_v)}$ with $[0,1]^{\mathrm{Card}(U_v)}$. Hence there exists a finite set of equations (at most $\mathrm{Card}(U_v)+1$), and thus a finite set of words of Σ^*, providing the same solutions. The rich sample generated by these equations for every v in $K(L)$ is characteristic. Such a minimal rich sample contains at most $\mathrm{Card}(K(L)) \times (\mathrm{Card}(Pm(L)) + 1) = O\left(\mathrm{Card}(Pm(L))^2 \times \mathrm{Card}(\Sigma)\right)$ elements. □

The learning algorithm described below outputs a minimal prefix PRFA of the target language when the input is a characteristic sample.

5.2 `lmpPRFA` Algorithm

```
lmpPRFA
input : a rich sample S
output : a prefix PRFA A = ⟨Σ, Q, φ, ι, τ⟩

begin
   Q ← {ε} ; ι(ε) ← 1 ; ∀a ∈ Σ, φ(ε, a, ε) ← 0
   W ← {a ∈ Σ | a ∈ π1(S) and p(aΣ*) > 0};
   do while W ≠ ∅
      v ← min W ; W ← W − {v}; Let w ∈ Σ*, a ∈ Σ s.t. v = wa ;
      if sol(E_S(v, Q)) = ∅ then
         Q ← Q ∪ {v} ; ∀u ∈ Q, ∀x ∈ Σ, φ(v, x, u) ← 0 ;
         W ← W ∪ {vx ∈ π1(S) | x ∈ Σ and p(vxΣ*) > 0};
         φ(w, a, v) ← p(waΣ*)/p(wΣ*);
      else
         let (α_u)_{u∈Q} ∈ sol(E_S(v, Q))
         for all u ∈ Q do φ(w, a, u) ← α_u × (p(waΣ*)/p(wΣ*))
      end if
   end do
   for all q ∈ Q do τ(q) ← 1 − φ(q, Σ, Q);
end
```

Theorem 6. *Let $L \in L_{PRFA}$ and let S be a characteristic sample of L, then given input S, algorithm* `lmpPRFA` *outputs the minimal prefix PRFA in polynomial time as function of the size of S.*

Proof. **When the algorithm terminates, the set of states Q is $Pm(L)$**

Let $Q^{[i]}$ (resp. $v^{[i]}$) denote the set Q (the word v) obtained at iteration i just before the `if`. Considering $W \neq \emptyset$ at the beginning (else $p_A(\varepsilon) = 1$).

From the definition of $Pm(L)$, $Q^{[1]} = \{\varepsilon\} = \left\{u \in Pm(L) \mid u < v^{[1]}\right\}$.

Let us assume that $Q^{[k]} = \left\{u \in Pm(L) \mid u < v^{[k]}\right\}$. At step $k+1$ there are two possibilities:

1. If $sol(E_S(v, Q^{[k]})) \neq \emptyset$ then as $v^{-1}L \in LG_L(Q^{[k]})$, $v \notin Pm(L)$ and
$$Q^{[k+1]} = Q^{[k]} = \left\{u \in Pm(L) \mid u < v^{[k]}\right\} = \left\{u \in Pm(L) \mid u < v^{[k+1]}\right\}.$$
2. If $sol(E_S(v^{[k]}, Q^{[k]})) = \emptyset$ then as $(v^{[k]})^{-1}L \notin LG_L(Q^{[k]})$, $v \in Pm(L)$. It follows that $Q^{[k+1]} = Q^{[k]} \cup \{v^{[k]}\}$ and as the word v is increasing to each iteration
$$Q^{[k+1]} = \left\{u \in Pm(L) \mid u < v^{[k+1]}\right\}.$$

As a consequence $Q \subseteq Pm(L)$. We also have $Pm(L) \subseteq Q$. Indeed, if we assume that there exists w in $Pm(L)$ and not in Q then there exists a prefix x of w such that $Decomp_L(x, \{u \in Pm(L) \mid u < x\}) \neq \emptyset \Rightarrow x \notin Pm(L)$ and as $Pm(L)$ is a prefixial set this is contradictory. Consequently the output state set Q is $Pm(L)$.

The algorithm terminates

Let $W^{[i]}$ denote the set W obtained at iteration i. From the definition of the kernel of L
$$W^{[0]} = \{a \in \Sigma \mid a \in \pi_1(S) \text{ and } p(a\Sigma^*) > 0\} \subseteq K(L).$$
Assume that $W^{[k]} \subseteq K(L)$ then at step $k+1$ there are two possible cases:

1. If $sol(E_S(v, Q)) \neq \emptyset$ then $W^{[k+1]} = W^{[k]} - \{v\} \subseteq K(L)$
2. If $sol(E_S(v, Q)) = \emptyset$ then as $Decomp_L(v, Q) = \emptyset$, and $Q = \{u \in Pm(L) \mid u < v\}$ (see the first part of the proof), $v \in Pm(L)$ and for every letter a if $va \in \pi_1(S)$ and $p(va\Sigma^*) > 0$ then $va \in K(L)$. It follows
$$W^{[k+1]} = \left(W^{[k]} - \{v\}\right) \cup \{va \in \pi_1(S) \mid a \in \Sigma \text{ and } p(va\Sigma^*) > 0\} \subseteq K(L)$$

As at every step one element of W is removed and $K(L)$ is finite, the algorithm terminates.

The output automaton is a minimal prefix PRFA of L

By construction the automaton is a minimal prefix PRFA of L. Thus A generates L (see the proof of Proposition 1).

Complexity of the algorithm

There is $\mathrm{Card}(K(L))$ iterations of the main loop, and we operate a resolution of a linear system of maximal size $\mathrm{Card}(S) + \mathrm{Card}(Pm(L)) + 1$ (solvable in polynomial time), and $Pm(L) \subseteq K(L) \subseteq \pi_1(S)$. Hence the algorithm complexity is polynomial in the size of $\mathrm{Card}(S)$. □

Example 2. We consider the target being automaton 4 at Figure 3. We construct a characteristic sample
$$S = \left\{(\varepsilon, 1), (a, \frac{1}{2}), (b, \frac{1}{2}), (aa, \frac{1}{2}), (ba, \frac{1}{6}), (aaa, \frac{1}{3}), (baa, \frac{1}{18}), (aaaa, \frac{7}{18}), (baaa, \frac{1}{54}).\right\}$$

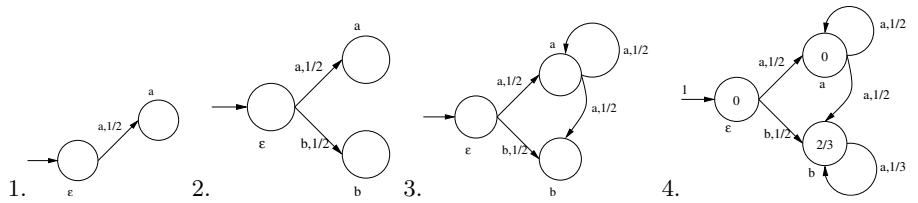

Fig. 3. An execution of algorithm `lmpPRFA`

First Step: The algorithm starts with $Q = \{\varepsilon\}$ and $W = \{a, b\}$. One considers adding the state a,

$$E_S(a, \{\varepsilon\}) = \begin{cases} \alpha_\varepsilon = 1 \\ p(a\Sigma^*|\varepsilon^{-1}L)\alpha_\varepsilon = p(a\Sigma^*|a^{-1}L) \\ \vdots \end{cases}$$

then $\alpha_\varepsilon = \frac{p(a\Sigma^*|a^{-1}L)}{p(a\Sigma^*|\varepsilon^{-1}L)} = \frac{p(aa\Sigma^*)/p(a\Sigma^*)}{p(a\Sigma^*)} = 2$. As this system has no solution, the state a is added (see Figure 3.1).

Step 2: $Q = \{\varepsilon, a\}$, $W = \{b, aa\}$
One considers adding the state b,

$$E_S(b, \{\varepsilon, a\}) = \begin{cases} \alpha_\varepsilon + \alpha_a = 1 \\ \frac{1}{2}\alpha_\varepsilon + 1\alpha_a = \frac{1}{3}(\text{obtained using } a) \\ \vdots \end{cases}$$

As this system has no solution in $[0, 1]^2$, the state b is added (see Figure 3.2).

Step 3: $Q = \{\varepsilon, a, b\}$, $W = \{aa, ba\}$
One considers adding the state aa,

$$E_S(aa, \{\varepsilon, a, b\}) = \begin{cases} \alpha_\varepsilon + \alpha_a + \alpha_b = 1 \\ \frac{1}{2}\alpha_\varepsilon + 1\alpha_a + \frac{1}{3}\alpha_b = \frac{2}{3}(\text{obtained using } a) \\ \frac{1}{2}\alpha_\varepsilon + \frac{2}{3}\alpha_a + \frac{1}{9}\alpha_b = \frac{7}{18}(\text{obtained using } aa) \end{cases}$$

wich is equivalent to $\alpha_\varepsilon = 0, \alpha_a = \frac{1}{2}, \alpha_b = \frac{1}{2}$. The state aa is not added and two transitions are added (see Figure 3.3).

Step 4: $Q = \{\varepsilon, a, b\}$, $W = \{ba\}$
One considers adding the state ba. The system $E_S(ba, \{\varepsilon, a, b\})$ is equivalent to $\alpha_\varepsilon = 0, \alpha_a = 0, \alpha_b = 1$. The state ba is not added and the target automaton is returned (see Figure 3.4).

Conclusion

Several grammatical inference algorithms can be described as looking for natural components of target languages, namely their residual languages. For example, in the deterministic framework, RPNI-like algorithm ([OG92], [LPP98]) try to identify the residual languages of the target language, while DELETE algorithms ([DLT00] and [DLT01a]) try to find inclusion relations between these languages. In the probabilistic framework, algorithms such as ALERGIA [CO94] or MDI [TDdlH00] also try to identify the residual languages of the target stochastic language. However these algorithms are restricted to the class L_{fr} of stochastic languages which have a finite number of residual languages.

We have defined the class L_{frg} of stochastic languages whose (possibly infinitely many) residual languages can be described by means of a linear expression of a finite subset of them. This class strictly includes the class L_{fr}.

A first learning algorithm for this class was proposed. It assumes the availability of a characteristic sample in which words are provided with their actual probabilities in the target language. Using similar techniques to those described in [CO99] and [TDdlH00], we believe that this algorithm can be adapted to infer correct structures from sample estimates. Work in progress aims at developing this adapted version and at evaluating this technique on real data.

References

[Ang88] D. Angluin. Identifying languages from stochastic examples. Technical Report YALEU/DCS/RR-614, Yale University, New Haven, CT, 1988.

[CO94] R.C. Carrasco and J. Oncina. Learning stochastic regular grammars by means of a state merging method. In *International Conference on Grammatical Inference*, pages 139–152, Heidelberg, September 1994. Springer-Verlag.

[CO99] 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.

[DLT00] F. Denis, A. Lemay, and A. Terlutte. Learning regular languages using non deterministic finite automata. In *ICGI'2000, 5th International Colloquium on Grammatical Inference*, volume 1891 of *Lecture Notes in Artificial Intelligence*, pages 39–50. Springer Verlag, 2000.

[DLT01a] F. Denis, A. Lemay, and A. Terlutte. Learning regular languages using rfsa. In *ALT 2001*. Springer Verlag, 2001.

[DLT01b] F. Denis, A. Lemay, and A. Terlutte. Residual finite state automata. In *18th Annual Symposium on Theoretical Aspects of Computer Science*, volume 2010 of *Lecture Notes in Computer Science*, pages 144–157, 2001.

[LPP98] K. J. Lang, B. A. Pearlmutter, and R. A. Price. Results of the Abbadingo one DFA learning competition and a new evidence-driven state merging algorithm. In *Proc. 4th International Colloquium on Grammatical Inference - ICGI 98*, volume 1433 of *Lecture Notes in Artificial Intelligence*, pages 1–12. Springer-Verlag, 1998.

[OG92] J. Oncina and P. Garcia. Inferring regular languages in polynomial update time. In *Pattern Recognition and Image Analysis*, pages 49–61, 1992.

[SO94] A. Stolcke and S. Omohundro. Inducing probabilistic grammars by Bayesian model merging. *Lecture Notes in Computer Science*, 862:106–118, 1994.

[TDdlH00] Franck Thollard, Pierre Dupont, and Colin de la Higuera. Probabilistic DFA inference using Kullback-Leibler divergence and minimality. In *Proc. 17th International Conf. on Machine Learning*, pages 975–982. Morgan Kaufmann, 2000.