Learning Classes of Probabilistic Automata

François Denis and Yann Esposito

LIF-CMI, 39, rue F. Joliot Curie
13453 Marseille Cedex 13 FRANCE,
fdenis,esposito@cmi.univ-mrs.fr

Abstract. Probabilistic finite automata (PFA) model stochastic languages, i.e. probability distributions over strings. Inferring PFA from stochastic data is an open field of research. We show that PFA are identifiable in the limit with probability one. Multiplicity automata (MA) is another device to represent stochastic languages. We show that a MA may generate a stochastic language that cannot be generated by a PFA, but we show also that it is undecidable whether a MA generates a stochastic language. Finally, we propose a learning algorithm for a subclass of PFA, called PRFA.

1 Introduction

Probabilistic automata (PFA) are formal objects which model *stochastic languages*, i.e. probability distributions over words [1]. They are composed of a *structure* which is a finite automaton (NFA) and of *parameters* associated with states and transitions which represent the probability for a state to be initial, terminal or the probability for a transition to be chosen. Given the structure of a probabilistic automaton A and a sequence of words $u_1, \ldots, u_n$ independently distributed according to a probability distribution P, computing parameters for A which maximize the likelihood of the observation is NP-hard [2]. However in practical cases, algorithms based on the EM (*Expectation-Maximization*) method [3] can be used to compute approximate values. On the other hand, inferring a probabilistic automaton (structure and parameters) from a sequence of words is a widely open field of research. In some applications, prior knowledge may help to choose a structure (for example, the standard model for biological sequence analysis [4]). Without prior knowledge, a complete graph structure can be chosen. But it is likely that in general, inferring both the appropriate structure and parameters from data would provide better results (see for example [5]).

Several learning frameworks can be considered to study inference of PFA. They often consist in adaptations to the stochastic case of classical learning models. We consider a variant of the identification in the limit model of Gold [6], adapted to the stochastic case in [7]. Given a PFA A and a sequence of words $u_1, \ldots, u_n, \ldots$ independently drawn according to the associated distribution P_A, an inference algorithm must compute a PFA A_n from each subsequence $u_1, \ldots, u_n$ such that with probability one, the support of A_n is stationary from

some index n and P_{A_n} converges to P_A; moreover, when parameters of the target are rational numbers, it can be requested that A_n itself is stationary from some index. The set of probabilistic automata whose structure is deterministic (PDFA) is identifiable in the limit with probability one [8,9,10], the identification being exact when the parameters of the target are rational numbers. However, PDFA are far less expressive than PFA, i.e. the set of probability distributions associated with PDFA is stricly included in the set of distributions generated from general PFA. We show that PFA are identifiable in the limit, with exact identification when the parameters of the target are rational numbers (Section 3).

Multiplicity automata (MA) are devices which model functions from Σ^* to $\mathbb{R}$. It has been shown that functions that can be computed by MA are very efficiently learnable in a variant of the exact learning model of Angluin, where the learner can ask *equivalence* and *extended membership queries*[11,12,13]. As PFA are particular MA, they are learnable in this model. However, the learning is improper in the sense that the output function is not a PFA but a multiplicity automaton. We show that a MA is maybe not a very convenient representation scheme to represent a PFA if the goal is to learn it from stochastic data. This representation is not robust, i.e. there are MA which do not compute a stochastic language and which are arbitrarily close to a given PFA. Moreover, we show that it is undecidable whether a MA generates a stochastic language. That is, given a MA computed from stochastic data: it is possible that it does not compute a stochastic language and there may be no way to detect it! We also show that MA can compute stochastic languages that cannot be computable by PFA. These two results are proved in Section 4: they solve problems that were left open in [1].

Our identification in the limit algorithm of PFA is far from being efficient while algorithms that identifies PDFA in the limit can also be used in practical learning situations (ALERGIA [8], RLIPS [9], MDI [14]). Note also that we do not have a model that describes algorithms "that can be used in practical cases": identification in the limit model is clearly too weak, exact learning via queries is irrealistic, PAC-model is maybe too strong (PDFA are not PAC-learnable [15]). So, it is important to define subclasses of PFA, as rich as possible, while keeping good empirical learnability properties. We have introduced in [16,17] a new class of PFA based on the notion of *residual languages*: a *residual language* of a stochastic language P is the language $u^{-1}P$ defined by $u^{-1}P(v) = P(uv)/P(u\Sigma^*)$. It can be shown that a stochastic language can be generated by a PDFA iff it has a finite number of residual languages. We consider the class of Probabilistic Residual Finite Automata (PRFA): a PFA A is a PRFA iff each of its states generates a residual language of P_A. It can be shown that a stochastic language can be generated by a PRFA iff P_A has a finite number of *prime* residual languages $u_1^{-1}P, \ldots, u_n^{-1}P$ sufficient to express all the residual languages as a convex linear combination of $u_1^{-1}P, \ldots, u_n^{-1}P$, i.e. for every word v, there exist non negative real numbers α_i such that $v^{-1}P = \sum \alpha_i u_i^{-1}P$ ([17,16]). Clearly, the class of PRFA is much more expressive than PDFA. We introduce a first learning algorithm for PRFA, which identifies this class in the limit with probability one, and can be used in practical cases (Section 5).

2 Preliminaries

2.1 Automata and Languages

Let Σ be a finite *alphabet*, and Σ^* be the set of words on Σ. The empty word is denoted by ε and the length of a word u is denoted by $|u|$. Let $<$ denote the length-lexicographic order on Σ^*. A *language* is a subset of Σ^*. For any language L, let $\mathrm{pref}(L) = \{u \in \Sigma^* | \ \exists v \in \Sigma^*, uv \in L\}$. L is *prefixial* iff $L = \mathrm{pref}(L)$.

A *non deterministic finite automaton (NFA)* is a 5-tuple $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ where Q is a finite set of states, $Q_0 \subseteq Q$ is the set of initial states, $F \subseteq Q$ is the set of terminal states, δ is the *transition* function defined from $Q \times \Sigma$ to 2^Q. Let δ also denote the extension of the transition function defined from $2^Q \times \Sigma^*$ to 2^Q. An NFA is *deterministic (DFA)* if $|Q_0| = 1$ and if $\forall q \in Q$, $\forall x \in \Sigma$, $|\delta(q, x)| \leq 1$. An NFA is *trimmed* if for any state q, $q \in \delta(Q_0, \Sigma^*)$ and $\delta(q, \Sigma^*) \cap F \neq \emptyset$. Let $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ be an NFA. A word $u \in \Sigma^*$ is *recognized* by A if $\delta(Q_0, u) \cap F \neq \emptyset$. The language recognized by A is $L_A = \{u \in \Sigma^* | \ \delta(Q_0, u) \cap F \neq \emptyset\}$. Let $q \in Q$, we denote $L_{A,q}$ the language $\{v \in \Sigma^* | \ \delta(q, v) \cap F \neq \emptyset\}$.

2.2 Multiplicity and Probabilistic Automata, Stochastic Languages

A *multiplicity automaton (MA)* is a 5-tuple $\langle \Sigma, Q, \varphi, \iota, \tau \rangle$ where Q is a finite set of states, $\varphi : Q \times \Sigma \times Q \to \mathbb{R}$ is the transition function, $\iota : Q \to \mathbb{R}$ is the initialization function and $\tau : Q \to \mathbb{R}$ is the termination function. We extend the transition function φ to $Q \times \Sigma^* \times Q$ by $\varphi(q, wx, r) = \sum_{s \in Q} \varphi(q, w, s)\varphi(s, x, r)$ where $x \in \Sigma$ and $\varphi(q, \varepsilon, r) = 1$ if $q = r$ and 0 otherwise. We extend again φ to $Q \times 2^{\Sigma^*} \times 2^Q$ by $\varphi(q, U, R) = \sum_{w \in U} \sum_{r \in R} \varphi(q, w, r)$. Let $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ be a MA. Let P_A be the function defined by: $P_A(u) = \sum_{q \in Q} \sum_{r \in Q} \iota(q)\varphi(q, u, r)\tau(r)$. The *support* of A is the NFA $\langle \Sigma, Q, Q_I, Q_T, \delta \rangle$ where $Q_I = \{q \in Q \mid \iota(q) \neq 0\}$, $Q_T = \{q \in Q \mid \tau(q) \neq 0\}$ and $\delta(q, x) = \{r \in Q | \ \varphi(q, x, r) \neq 0\}$ for any state q and letter x. An MA is said to be *trimmed* if its support is a trimmed NFA.

A *semi-PFA* is a MA such that ι, φ and τ take their values in $[0, 1]$, $\sum_{q \in Q} \iota(q) \leq 1$ and for any state q, $\tau(q) + \varphi(q, \Sigma, Q) \leq 1$. A *Probabilistic Finite Automaton (PFA)* is a trimmed semi-PFA such that $\sum_{q \in Q} \iota(q) = 1$ and for any state q, $\tau(q) + \varphi(q, \Sigma, Q) = 1$. A *Probabilistic Deterministic Finite Automaton (PDFA)* is a PFA whose support is deterministic.

A *stochastic language* on Σ is a probability distribution over Σ^*, i.e. a function P defined from Σ^* to $[0, 1]$ such that $\sum_{u \in \Sigma^*} P(u) = 1$. The function P_A associated with a PFA A is a stochastic language. Let us denote by $\mathcal{S}$ the set of stochastic languages on Σ. Let $P \in \mathcal{S}$ and let $\mathrm{res}(P) = \{u \in \Sigma^* | P(u\Sigma^*) \neq 0\}$. Let $u \in \mathrm{res}(P)$, the *residual language* of P associated with u is the stochastic language $u^{-1}P$ defined by $u^{-1}P(w) = P(uw)/P(u\Sigma^*)$. Let $\mathrm{Res}(P) = \left\{u^{-1}P \middle| \ u \in \mathrm{res}(P)\right\}$. It can easily be shown that $\mathrm{Res}(P)$ spans a finite dimension vector space iff P can be generated by a MA. Let $\mathrm{MA}_{\mathcal{S}}$ be the set composed of MA which generate stochastic languages. Let us denote by $\mathcal{S}_{\mathrm{MA}}$ (resp. $\mathcal{S}_{\mathrm{PFA}}, \mathcal{S}_{\mathrm{PDFA}}$) the set of stochastic languages generated by MA (resp. PFA, PDFA) on Σ. Let $R \subseteq MA$. Let us denote by $R[\mathbb{Q}]$ the set of elements of R, the parameters of which are all in $\mathbb{Q}$.

2.3 Learning Stochastic languages

We are interested in learnable subsets of MA which generate stochastic languages. Several learning model can be used, we consider two of them.

Identification in the limit with probability 1. The identification in the limit learning model of Gold [6] can be adapted to the stochastic case ([7]).

Let $P \in \mathcal{S}$ and let S be a finite sample drawn according to P. For any $X \subseteq \Sigma^*$, let $P_S(X) = \frac{1}{\text{Card}(S)} \sum_{x \in S} \mathbb{1}_{x \in X}$ be the empirical distribution associated with S. A *complete presentation* of P is an infinite sequence S of words generated according to P. Let S_n be the sequence composed of the n first words (not necessarily different) of S. We shall write $P_n(A)$ instead of $P_{S_n}(A)$.

Definition 1. *Let $\mathcal{R} \subseteq \text{MA}_{\mathcal{S}}$. $\mathcal{R}$ is said to be* identifiable in the limit with probability one *if there exists a learning algorithm $\mathcal{L}$ such that for any $R \in \mathcal{R}$, with probability 1, for any complete presentation S of P_R, $\mathcal{L}$ computes for each S_n given as input, a hypothesis R_n such that the support of R_n is stationary from some index n^* and such that $P_{R_n} \to P_R$ as $n \to \infty$. Moreover, $\mathcal{R}$ is* strongly identifiable in the limit with probability one *if P_{R_n} is also stationary from some index.*

It has been shown that PDFA is identifiable in the limit with probability one [8,9] and that PDFA[$\mathbb{Q}$] is strongly identifiable in the limit [10].

We show below that PFA is identifiable in the limit with probability one and that PFA[$\mathbb{Q}$] is strongly identifiable in the limit.

Learning using queries The MAT model of Angluin [18], which allows to use *membership queries* (MQ) and *equivalence queries* (EQ) has been extended to functions computed by MA. Let P be the target function, let u be a word and let A be a MA. The answer to the query MQ(u) is the value $P(u)$; the answer to the query EQ(A) is YES if $P_A = P$ and NO otherwise. Functions computed by MA can be learned exactly within polynomial time provided that the learning algorithm can make extended membership queries and equivalence queries. Therefore, any stochastic language in $\mathcal{S}_{\text{MA}}$ can be learned by this algorithm.

However, using MA to represent stochastic languages has some drawbacks: first, this representation is not robust, i.e. a MA may compute a stochastic language for a given set of parameters θ_0 and computes a function which is not a stochastic language for any $\theta \neq \theta_0$; moreover, it is undecidable whether a MA computes a stochastic language. That is, by using MA to represent stochastic languages, a learning algorithm using approximate data might infer a MA which does not compute a stochastic language and with no means to detect it.

3 Identifying $\mathcal{S}_{\text{PFA}}$ in the limit.

We show in this Section that $\mathcal{S}_{\text{PFA}}$ is identifiable in the limit with probability one. Moreover, the identification is strong when the target can be generated by a PFA whose parameters are rational numbers.

3.1 Weak identification

Let P be a stochastic language over Σ, let $\mathcal{A} = (A_i)_{i \in I}$ be a family of subsets of Σ^*, let S be a finite sample drawn according to P, and let P_S be the empirical distribution associated with S. It can be shown [19,20] that for any confidence parameter δ, with a probability greater than $1 - \delta$, for any $i \in I$,

$$|P_S(A_i) - P(A_i)| \leq c\sqrt{\frac{\mathrm{VC}(\mathcal{A}) - \log \frac{\delta}{4}}{Card(S)}} \tag{1}$$

where $\mathrm{VC}(\mathcal{A})$ is the dimension of Vapnik-Chervonenkis of $\mathcal{A}$ and c is an universal constant. When $\mathcal{A} = (\{w\})_{w \in \Sigma^*}$, $\mathrm{VC}(\mathcal{A}) = 1$. Let $\Psi(\epsilon, \delta) = \frac{c^2}{\epsilon^2}(1 - \log \frac{\delta}{4})$.

Lemma 1. *Let $P \in \mathcal{S}$ and let S be a complete presentation of P. For any precision parameter ϵ, any confidence parameter δ, any $n \geq \Psi(\epsilon, \delta)$, with a probability greater than $1 - \delta$, $|P_n(w) - P(w)| \leq \epsilon$ for all $w \in \Sigma^*$.*

Proof. Use Inequality (1). □

For any integer k, let $Q_k = \{1, \ldots, k\}$ and let $\Theta_k = \{\iota_i, \tau_i, \varphi^x_{i,j} | i, j \in Q_k, x \in \Sigma\}$ be a set of variables. We consider the following set of constraints C_k on Θ_k:

$$C_k = \begin{cases} 0 \leq \iota_i, \tau_i, \varphi^x_{i,j} \leq 1 \text{ for any } i, j \in Q_k, x \in \Sigma, \\ \sum_{i \in Q_k} \iota_i \leq 1, \\ \tau_i + \sum_{j \in Q_k, x \in \Sigma} \varphi^x_{i,j} \leq 1 \text{ for any } i \in Q_k. \end{cases}$$

Any assignment θ of these variables satisfying C_k is said to be *valid*; any valid assignement θ defines a semi-PFA A^θ_k by letting $\iota(i) = \iota_i$, $\tau(i) = \tau_i$ and $\varphi(i, x, j) = \varphi^x_{i,j}$ for any states i and j and any letter x. We simply denote by P_θ the function $P_{A^\theta_k}$ associated with A^θ_k. Let V_k be the sets of valid assignments. For any $\theta \in V_k$, let θ^t be the associated trimmed assignment which set to 0 every parameter which is never effectively used to compute the probability $P_\theta(w)$ of some word w. Clearly, θ^t is valid and $P_\theta = P_{\theta^t}$.

For any w, $P_\theta(w)$ is a polynomial and is therefore a continuous function of θ. On the other hand, the series $\sum_w P_\theta(w)$ are convergent but not uniformly convergent and $P_\theta(w\Sigma^*)$ is not a continous function of θ (see Fig. 1). However, we show below that the function $(\theta, w) \to P_\theta(w)$ is uniformly continuous.

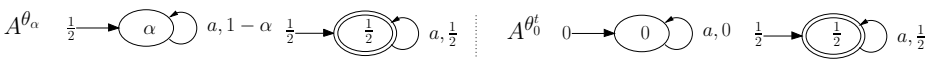

Fig. 1. $P_{\theta_\alpha}(\varepsilon) = 1/4 + \alpha/2$; $P_{\theta_0}(\Sigma^*) = 1/2$ and $P_{\theta_\alpha}(\Sigma^*) = 1$ when $\alpha > 0$.

Proposition 1. *For any $k \in \mathbb{N}$, the function $(\theta, w) \to P_\theta(w)$ is uniformly continuous: $\forall \epsilon, \exists \alpha, \forall w \in \Sigma^*, \forall \theta, \theta' \in V_k, \|\theta - \theta'\| < \alpha \Rightarrow |P_\theta(w) - P_{\theta'}(w)| < \epsilon$.*

Proof. We prove the proposition in several steps.

1. Let $\theta_0 \in V_k$, let $A_k^{\theta_0^t} = \langle \Sigma, Q_k, \varphi_0, \iota_0, \tau_0 \rangle$ and let $\beta_0 = \max\left\{\varphi_0(q, \Sigma^k, Q_k) \middle| \ q \in Q_k\right\}$. For any state q s.t. $\varphi_0(q, \Sigma^k, Q_k) > 0$, there must exist a word w of length $< k$ and a state q' s.t. $\varphi_0(q, w, q') \neq 0$ and $\tau_0(q') \neq 0$. Hence, $\beta_0 < 1$.
2. For any integer n and any state q, $\varphi_0(q, \Sigma^{nk}, Q_k) \leq \beta_0^n$. Proof by induction on n: clearly true when $n = 0$ and
   $\varphi_0(q, \Sigma^{nk}, Q_k) \leq \sum_{q' \in Q_k, w \in \Sigma^k} \varphi_0(q, w, q')\varphi_0(q', \Sigma^{(n-1)k}, Q_k)$
   $\leq \beta_0^{n-1} \sum_{q' \in Q_k, w \in \Sigma^k} \varphi_0(q, w, q') \leq \beta_0^n$.
3. For any integer n, $P_{\theta_0^t}(\Sigma^{nk}\Sigma^*) = \sum_{q \in Q_k} \iota_0(q)\varphi_0(q, \Sigma^{nk}, Q_k) \leq \beta_0^n$.
4. For any state q, $\varphi_0(q, \Sigma^*, Q_k) = \sum_{n \in \mathbb{N}} \sum_{m=0}^{k} \varphi_0(q, \Sigma^{nk+m}, Q_k)$
   $\leq \sum_{n \in \mathbb{N}} \sum_{m=0}^{k} \sum_{q' \in Q_k} \varphi_0(q, \Sigma^m, q')\varphi_0(q', \Sigma^{nk}, Q_k)$
   $\leq \sum_{n \in \mathbb{N}} \sum_{m=0}^{k} \sum_{q' \in Q_k} \beta_0^n \varphi_0(q, \Sigma^m, q') \leq k/(1 - \beta_0)$.
5. Let α_0 be the minimal non null parameter in θ_0^t, let $\alpha < \alpha_0/2$, let θ be a valid assignement such that $||\theta - \theta_0|| < \alpha$ and let $A_k^{\theta^t} = \langle \Sigma, Q_k, \varphi, \iota, \tau \rangle$. Note that any non null parameter in θ_0^t corresponds to a non null parameter in θ^t but that the converse is false (see Fig. 1). Let θ' be the assignment obtained from θ^t by setting to 0 every parameter which is null in θ_0^t, let $A_k^{\theta'} = \langle \Sigma, Q_k, \varphi', \iota', \tau' \rangle$ and let $\beta' = \max\left\{\varphi'(q, \Sigma^k, Q_k) \middle| \ q \in Q_k\right\}$. As θ' and θ_0^t have the same set of non null parameters, there exists $\alpha_1 < \alpha_0/2$ such that $||\theta - \theta_0|| < \alpha_1$ implies $\beta' < (1 + \beta_0)/2$. Let $\beta_1 = (1 + \beta_0)/2$.
6. Let w be a word of length $\geq nk$. There are two categories of derivations of w in $A_k^{\theta^t}$:
   - those which exist in $A_k^{\theta'}$. Their contribution to $P_{\theta^t}(w)$ is not greater than β_1^n.
   - those which do not entirely exist in $A_k^{\theta'}$ and one parameter of which is $\leq \alpha_1$. Let $q_0, \ldots, q_{|w|}$ be such a derivation. Either $\iota(q) \leq \alpha_1$, either $\tau(q_{|w|}) \leq \alpha_1$, or there exists a first state q_i such that $q_0, \ldots, q_i$ is a derivation in $A_k^{\theta'}$ and $\varphi(q_i, w_i, q_{i+1}) \leq \alpha_1$, where w_i is the ith letter of w. The contribution of these derivations to $P_{\theta^t}(w)$ is bounded by

$$\sum_{q, \iota(q) \leq \alpha_1} \alpha_1 \varphi(q, w, Q) + \sum_{q, q', \iota(q') \leq \alpha_1} \iota(q)\varphi(q, w, q')\alpha_1 + \sum_{q_0, q_i \in Q_k} \iota'(q_0)\varphi'(q_0, \Sigma^*, q_i)\alpha_1 \leq \alpha_1(k + 1 + k/(1 - \beta_1)) \ .$$

   Therefore, $P_{\theta^t}(w) \leq \beta_1^n + \alpha_1(k + 1 + k/(1 - \beta_1))$.
7. Let $\epsilon > 0$. Let $\alpha_2 = \min(\alpha_1, \epsilon/[4(k + 1 + k/(1 - \beta_1))])$ and let N be such that $\beta_1^N < \epsilon/4$. As for any fixed w, $P_\theta(w)$ is continuous, there exists $\alpha \leq \alpha_2$ such that $||\theta - \theta_0|| < \alpha$ implies that for any $w \in \Sigma^{\leq N}$, $|P_{\theta_0}(w) - P_\theta(w)| < \epsilon$. As $P_{\theta_0}(w) \leq \epsilon/2$ and $P_\theta(w) \leq \epsilon/2$ when $|w| \geq N$, we conclude that for all words w, $|P_{\theta_0}(w) - P_\theta(w)| < \epsilon$.

8. We have shown that: $\forall \epsilon, \forall \theta_0 \in V_k, \exists \alpha, \forall w \in \Sigma^*, \forall \theta \in V_k, ||\theta - \theta_0|| < \alpha \Rightarrow |P_\theta(w) - P_{\theta_0}(w)| < \epsilon$. Now, suppose that:

$$\exists \epsilon, \forall n \in \mathbb{N}, \exists w_n \in \Sigma^*, \exists \theta_n,\ \theta'_n \in V_k \text{ s.t.}$$
$$\|\theta_n - \theta'_n\| < 1/n \text{ and } \left|P_{\theta_n}(w_n) - P_{\theta'_n}(w_n)\right| \geq \epsilon$$

As valid assignments are elements of a compact set, there would exist a valid assignement θ_0 such that $\theta_{\sigma(n)} \to \theta_0$ and $\theta'_{\sigma(n)} \to \theta_0$ (for some subsequence $\sigma(n)$). We know that there exists $\alpha > 0$ such that $||\theta - \theta_0|| < \alpha$ implies that for all w, $|P_{\theta_0}(w) - P_\theta(w)| < \epsilon/2$. When $1/n < \alpha$, the hypothesis leads to a contradiction. □

Let $P \in \mathcal{S}$ and let S be a complete presentation of P. For any integers n and k and for any $\epsilon > 0$, let $I_{\Theta_k}(S_n, \epsilon)$ be the following system

$$I_{\Theta_k}(S_n, \epsilon) = C_k \cup \{|P_\theta(w) - P_n(w)| \leq \epsilon \text{ for } w \in S_n\}.$$

Lemma 2. *Let $P \in \mathcal{S}$ be a stochastic language and let S be a complete presentation of P. Suppose that there exists an integer k and a PFA $A_k^{\theta_0}$ such that $P = P_{\theta_0}$. Then, for any precision parameter ϵ, any confidence parameter δ and any $n \geq \Psi(\epsilon/2, \delta)$, with a probability greater than $1-\delta$, $I_{\Theta_k}(S_n, \epsilon)$ has a solution that can be computed.*

Proof. From Lemma 1, with a probability greater than $1 - \delta$, we have $|P_{\theta_0}(w) - P_n(w)| \leq \epsilon/2$ for all $w \in S_n$. For any $w \in S_n$, $P_\theta(w)$ is a polynomial in θ whose coefficients are all equal to 1. A bound M_w of $||\frac{dP_\theta(w)}{d\theta}||$ can easily be computed. We have

$$|P_\theta(w) - P_{\theta'}(w)| \leq M_w||\theta - \theta'||.$$

Let $\alpha = \inf\{\frac{\epsilon}{2M_w}|w \in S_n\}$. If $||\theta - \theta'|| < \alpha$, $|P_\theta(w) - P_{\theta'}(w)| \leq \epsilon/2$ for all $w \in S_n$. So, we can compute a finite number of assignments: $\theta_1^\alpha, \ldots \theta_{N_\alpha}^\alpha$ such that for all valid assignment θ, there exists $1 \leq i \leq N_\alpha$ such that $||\theta - \theta_i^\alpha|| \leq \alpha$. Let i be such that $||\theta_0 - \theta_i^\alpha|| \leq \alpha$: θ_i^α is a solution of $I_{\Theta_k}(S_n, \epsilon)$. □

The Borel-Cantelli Lemma is often used to show that a given property holds with probability 1: let $(A_n)_{n \in \mathbb{N}}$ be a sequence of events such that $\sum_{n \in \mathbb{N}} P(A_n) < \infty$; then, the probability that a finite number of A_n occur is 1.

For any integer n, let $\epsilon_n = n^{-\frac{1}{3}}$ and $\delta_n = n^{-2}$. Clearly, $\epsilon_n \to 0$ and $\sum_{n \in \mathbb{N}} \delta_n < \infty$. Moreover, there exists an integer N s.t. $\forall n > N, n \geq \psi_1(\epsilon_n/2, \delta_n)$.

Proposition 2. *Let P be a stochastic language and let S be a complete presentation of P. Suppose that there exists an integer k and a PFA $A_k^{\theta_0}$ such that $P = P_{\theta_0}$. With probability 1 there exists an integer N such that for any $n > N$, $I_{\Theta_k}(S_n, \epsilon_n)$ has a solution θ_n and $\lim_{n \to \infty} P_{\theta_n}(w) \to P(w)$ uniformly in w.*

Proof. The Borel-Cantelli Lemma proves that with probability 1 there exists an integer N s.t. for any $n > N$, $I_{\Theta_k}(S_n, \epsilon_n)$ has a solution θ_n. Now suppose that

$$\exists \epsilon, \forall N, \exists n \geq N, \exists w_n \in \Sigma^*, |P_{\theta_n}(w_n) - P(w_n)| \geq \epsilon.$$

Let $(\theta_{\sigma(n)})$ be a subsequence of (θ_n) such that for every integer n, $\sigma(n) \geq n$, $|P_{\theta_{\sigma(n)}}(w_{\sigma(n)}) - P(w_{\sigma(n)})| \geq \epsilon$ and $\theta_{\sigma(n)} \to \theta$. As each $\theta_{\sigma(n)}$ is a solution of $I_{\Theta_k}(S_{\sigma(n)}, \epsilon_{\sigma(n)})$, θ is a valid assignement such that for all w such that $P(w) \neq 0$, $P(w) = P_\theta(w)$. As P is a stochastic language, we must have $P(w) = P_\theta(w)$ for every word w, i.e. $P = P_\theta$. From Proposition 1, $P_{\theta_{\sigma(n)}}$ converges uniformy to P, which contradicts the hypothesis. □

It remains to show that when the target cannot be expressed by a PFA on k states, the system $I_{\Theta_k}(S_n, \epsilon_n)$ has no solution from some index.

Proposition 3. *Let P be a stochastic language and let S be a complete presentation of P. Let k be an integer such that there exists no θ satisfying $P = P_\theta$. Then, with probability 1, there exists an integer N such that for any $n > N$, $I_{\Theta_k}(S_n, \epsilon_n)$ has no solution.*

Proof. Suppose that $\forall N \in \mathbb{N}$, $\exists n \geq N$ such that $I_{\Theta_k}(S_n, \epsilon_n)$ has a solution. Let $(n_i)_{i \in \mathbb{N}}$ be an increasing sequence such that $I_{\Theta_k}(S_{n_i}, \epsilon_{n_i})$ has a solution θ_i and let (θ_{k_i}) be a subsequence of (θ_i) that converges to a limit value $\overline{\theta}$.

Let $w \in \Sigma^*$ be such that $P(w) \neq 0$. We have $|P_{\overline{\theta}}(w) - P(w)| \leq |P_{\overline{\theta}}(w) - P_{\theta_i}(w)| + |P_{\theta_i}(w) - P_{n_i}(w)| + |P_{n_i}(w) - P(w)|$ for any integer i.

With probability 1, the last term converges to 0 as i tends to infinity (Lemma 1). With probability 1, there exists an index i such that $w \in S_{n_i}$. From this index, the second term is less than ϵ_{n_i} which tends to 0 as i tends to infinity. Now, as $P_\theta(w)$ is a continuous function of θ, the first term tends to 0 as i tends to infinity. Therefore, $P_{\overline{\theta}}(w) = P(w)$ and $P_{\overline{\theta}} = P$, which contradicts the hypothesis. □

Theorem 1. *$\mathcal{S}_{\text{PFA}}$ is identifiable in the limit with probability one.*

Proof. Consider the following algorithm $\mathcal{A}$:

`Input: A stochastic sample` S_n `of length` n`.`
`for` $k = 1$ `to` n `do {`
  `compute` α `and` $\theta_1^\alpha, \ldots \theta_{N_\alpha}^\alpha$ `as in Lemma 2`
  `if` $\exists 1 \leq i \leq N_\alpha$ `s.t.` θ_i^α `is a solution of` $I_{\Theta_k}(S_n, \epsilon_n)$ `then`
    `{return the smallest solution (in some order)` $A_k^{\theta_i^\alpha}$ `}}`
`return a default hypothesis if no solution has been found`

Let P be the target and let $A_k^{\theta_0}$ be a minimal state PFA which computes P. Previous propositions prove that with probability one, from some index N, the algorithm shall output a PFA $A_k^{\theta_n}$ such that P_{θ_n} converges uniformely to P. □

3.2 Strong identification

When the target can be computed by a PFA whose parameters are in $\mathbb{Q}$, an equivalent PFA can be identified in the limit with probability 1. In order to show a similar property for PDFA, a method based on Stern-Brocot trees was used in [10]. Here we use the representation of real numbers by continuous fractions [21].

Let $x \geq 0$. Define $x_0 = x$, $a_0 = \lfloor x_0 \rfloor$ and while $x_n \neq a_n$, $x_{n+1} = 1/(x_n - a_n)$ and $a_{n+1} = \lfloor x_n \rfloor$. The sequences (x_n) and (a_n) are finite iff $x \in \mathbb{Q}$. Suppose

from now on that $x \in \mathbb{Q}$, let N be the greatest index such that $x_N \neq a_N$, and for any $n \leq N$, let the nth *convergent* of x be the fraction

$$p_n/q_n = a_0 + 1/\left(a_1 + 1/\left(\cdots(1/a_n)\cdots\right)\right)$$

where $gcd(p_n, q_n) = 1$.

Lemma 3 ([21]). *We have $x = \frac{p_N}{q_N}$ and $\forall n < N$, $\left|x - \frac{p_n}{q_n}\right| \leq \frac{1}{q_n q_{n+1}} < \frac{1}{q_n^2}$. If a and b are two integers such that $\left|\frac{a}{b} - x\right| < \frac{1}{2b^2}$, then there is an integer $n \leq N$ such that $\frac{a}{b} = \frac{p_n}{q_n}$. For any integer A, there exists only a finite number of rational numbers $\frac{p}{q}$ such that $\left|x - \frac{p}{q}\right| \leq \frac{A}{q^2}$.*

Let $x = 5/14$. We have $p_0/q_0 = 0$, $p_1/q_1 = 1/2$, $p_2/q_2 = 1/3$ and $p_3/q_3 = x$.

Lemma 4. *Let (ϵ_n) be a sequence of non negative real numbers which converges to 0, let $x \in \mathbb{Q}$, let (y_n) be a sequence of elements of $\mathbb{Q}$ such that $|x - y_n| \leq \epsilon_n$ for all but finitely many n. Let $\frac{p_m^n}{q_m^n}$ the convergents associated with y_n. Then, there exists an integer N such that, for any $n \geq N$, there is an integer m such that $x = \frac{p_m^n}{q_m^n}$. Moreover, $\frac{p_m^n}{q_m^n}$ is the unique rational number such that $\left|y_n - \frac{p_m^n}{q_m^n}\right| \leq \epsilon_n \leq \frac{1}{(q_m^n)^2}$.*

Proof. Omitted. All proofs omitted here can be found in a complete version of the paper available `http://www.cmi.univ-mrs.fr/~fdenis` .

Example 1. Let $y_n = 1/2 - 1/n$ and $\epsilon_n = 1/n$. Then $y_3 = 1/6$, $y_4 = 1/4$, $y_5 = 3/10$, $y_6 = 1/3$, $y_7 = 5/14$. The first n s.t. $\left|y_n - \frac{p_m^n}{q_m^n}\right| \leq \frac{1}{n} \leq \frac{1}{(q_m^n)^2}$ has a solution is $n = 4$. Let z_n be the first solution. We have $z_4 = 1/4$, $z_5 = 1/3$, $z_6 = 1/3$ and $z_n = 1/2$ for $n \geq 7$.

Theorem 2. *Let $\mathcal{S}_{\mathrm{PFA}}[\mathbb{Q}]$ be the set of stochastic languages that can be generated from a PFA whose parameters are in $\mathbb{Q}$. $\mathcal{S}_{\mathrm{PFA}}[\mathbb{Q}]$ is strongly identifiable in the limit with probability one.*

Proof. Omitted.

4 $\mathcal{S}_{\mathbf{MA}}$ and $\mathcal{S}_{\mathbf{PFA}}$

The representation of stochastic languages by MA is not robust. Fig. 2 shows two MA which depend on parameter x. They define a stochastic language when $x = 0$ but not when $x > 0$. When $x > 0$, the first one generates negative values, and the second one generates unbounded values.

Let $P \in \mathcal{S}_{\mathrm{MA}}$ and let A be the MA which generates P output by the exact learning algorithm defined in [12]. A sample S drawn according to P defines an empiric distribution P_S that could be used by some variant of this learning algorithm. In the best case, this variant is expected to output a hypothesis

$\hat{A}$ having the same support as A and with approximated parameters close to those of A. But there is no guaranty that $\hat{A}$ defines a stochastic language. More seriously, we show below that it is impossible to decide whether a given MA generates a stochastic language. The conclusion is that MA representation of stochastic languages is maybe not appropriate to learn stochastic languages.

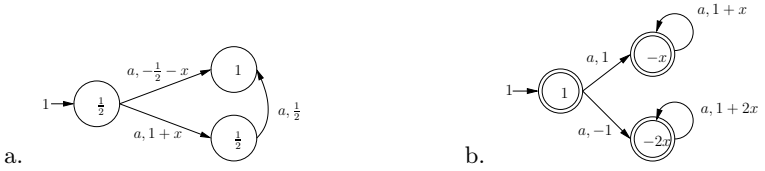

Fig. 2. Two MA generating stochastic language if $x = 0$. If $x > 0$, the first generates negative values and the second unbounded values.

4.1 Membership to $\mathcal{S}_{\mathbf{MA}}$ is undecidable

We reduce the decision problem to a problem about *acceptor PFA*. An MA $\langle \Sigma, Q, \varphi, \iota, \tau \rangle$ is an *acceptor PFA* if φ, ι and τ are non negative functions, $\sum_{q \in Q} \iota(q) = 1$, $\forall q \in Q, \forall x \in \Sigma, \sum_{r \in Q} \varphi(q, x, r) = 1$ and if there exists a unique terminal state t such that $\tau(t) = 1$.

Theorem 3 ([22]). *Given an acceptor PFA A whose parameters are in $\mathbb{Q}$ and $\lambda \in \mathbb{Q}$, it is undecidable whether there exists a word w such that $P_A(w) < \lambda$.*

The following lemma shows some constructions on MA.

Lemma 5. *Let A and B be two MA and let $\lambda \in \mathbb{Q}$. We can construct:*

1. *a MA I_λ such that $\forall w \in \Sigma^*, P_{I_\lambda}(w) = \lambda$,*
2. *a MA $A + B$ such that $P_{A+B} = P_A + P_B$*
3. *a MA $\lambda \cdot A$ such that $P_{\lambda \cdot A} = \lambda P_A$,*
4. *a MA $\mathrm{tr}(A)$ such that for any word w, $P_{\mathrm{tr}(A)}(w) = \frac{P_A(w)}{(|\Sigma|+1)^{|w|+1}}$*

Proof. Proofs are omitted. See Fig. 3.

Lemma 6. *Let $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ be a semi-PFA, let $Q^t = \{q \in Q | \varphi(Q_I, \Sigma^*, q) > 0$ and $\varphi(q, \Sigma^*, Q_T) > 0\}$ and let $A^t = \langle \Sigma, Q^t, \varphi_{|Q_t}, \iota_{|Q_t}, \tau_{|Q_t} \rangle$. Then, A^t is a trimmed semi-PFA such that $P_A = P_{A^t}$ and which can be constructed from A.*

Proof. Straightforward.

Lemma 7. *Let A be a trimmed semi-PFA, we can compute $P_A(\Sigma^*)$.*

Proof. Omitted.

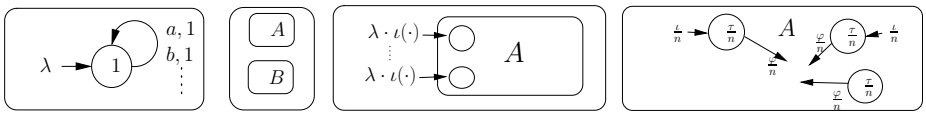

Fig. 3. How to construct I_λ, $A + B$, $\lambda \cdot A$ and $\mathrm{tr}(A)$, where $n = |\Sigma| + 1$. Note that when A is an acceptor PFA, $\mathrm{tr}(A)$ is a semi-PFA.

Proposition 4. *It is undecidable whether a MA generates a stochastic language.*

Proof. Let A be an acceptor PFA on Σ and $\lambda \in \mathbb{Q}$. For every word w, we have $P_{\mathrm{tr}(A-I_\lambda)}(w) = (|\Sigma|+1)^{-(|w|+1)}(P_A(w) - \lambda) = P_{\mathrm{tr}(A)}(w) - \lambda(|\Sigma|+1)^{-(|w|+1)}$ and therefore $P_{\mathrm{tr}(A-I_\lambda)}(\Sigma^*) = P_{\mathrm{tr}(A)}(\Sigma^*) - \lambda$.

- If $P_{\mathrm{tr}(A)}(\Sigma^*) = \lambda$ then either $\exists w$ s.t. $P_A(w) < \lambda$ or $\forall w, P_A(w) = \lambda$. Let B be the PFA such that $P_B(w) = 1$ if $w = \varepsilon$ and 0 otherwise. We have, $P_{B+\mathrm{tr}(A-I_\lambda)}(\Sigma^*) = 1$. Therefore, $\forall w, P_A(w) \geq \lambda$ iff $P_A(\varepsilon) \geq \lambda$ and $B + \mathrm{tr}(A - I_\lambda)$ generates a stochastic language.
- If $P_{\mathrm{tr}(A)}(\Sigma^*) \neq \lambda$, let $B = \left|P_{\mathrm{tr}(A)}(\Sigma^*) - \lambda\right|^{-1} \cdot \mathrm{tr}(A - I_\lambda)$. Check that B is computable from A, that $P_B(\Sigma^*) = 1$ and that $P_B(w) = \left|P_{\mathrm{tr}(A)}(\Sigma^*) - \lambda\right|^{-1} \left(\mathrm{Card}(\Sigma+1)^{|w|+1}\right)^{-1} (P_A(w) - \lambda)$. So, $\exists w \in \Sigma^*, P_A(w) < \lambda$ iff B does note generate a stochastic language.

In both cases, we see that deciding whether a MA generates a stochastic language would solve the decision problem on PFA acceptors. □

Remark that in fact, we have proved a stronger result: it is undecidable whether a MA A such that $\sum_{w\in\Sigma^*} P_A(w) = 1$ generates a stochastic language. As a consequence, it can be proved that there exist stochastic languages that can be computed by MA but not by PFA.

Theorem 4. $\mathcal{S}_{\mathrm{PFA}} \subsetneq \mathcal{S}_{\mathrm{MA}}$.

Proof. Omitted.

5 Learning PRFA

The inference algorithm given in Section 3 is highly inefficient and cannot be used for real applications. It is unknown whether PFA can be efficiently learned. Here, we study a subclass of PFA, for which there exists a learning algorithm which can be efficiently implemented.

5.1 Probabilistic Residual Finite Automata

Definition 2 (Probabilistic Residual Finite Automaton). *A* PRFA *is a PFA $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ whose states define residual languages of P_A, i.e.such that $\forall q \in Q$, $\exists u \in \Sigma^*$, $P_{A,q} = u^{-1}P_A$, where $P_{A,q}$ denotes the stochastic language generated by $< \Sigma, Q, \varphi, \iota_q, \tau >$ where $\iota_q(q) = 1$ [16].*

Remark that PDFA are PRFA but that the converse is false. Fig. 4 represents a PRFA $\langle \Sigma, Q, \varphi, \iota, \tau \rangle$ where $\Sigma = \{a, b\}$, $Q = \{\varepsilon, a, b\}$, $\iota(\varepsilon) = 1$, $\tau(b) = \frac{2}{3}$, $\varphi(\varepsilon, a, a) = \frac{1}{2}$, $\varphi(\varepsilon, b, b) = \frac{1}{2}$, $\varphi(a, a, a) = \frac{1}{2}$, $\varphi(a, a, b) = \frac{1}{2}$ and $\varphi(b, a, b) = \frac{1}{3}$.

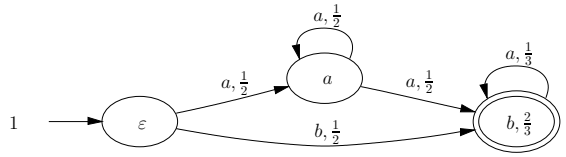

Fig. 4. A prefix PRFA.

Let $\mathcal{P}$ be a finite subset of $\mathcal{S}$. The convex closure of $\mathcal{P}$ is denoted by $\text{conv}(\mathcal{P}) = \{P \in \mathcal{S} | \exists P_1, \ldots, P_n \in \mathcal{P}, \exists \lambda_1, \ldots \lambda_n \geq 0, P = \sum_{i=1}^{n} \lambda_i P_i\}$. We say that $\mathcal{P}$ is a *residual net* if for every $Q \in \mathcal{P}$ and every $u \in \text{res}(Q)$, $u^{-1}Q \in \text{conv}(\mathcal{P})$. A residual net $\mathcal{P}$ is a *convex generator* for $P \in \mathcal{S}$ if $P \in \text{conv}(\mathcal{P})$.

It can be shown that $\mathcal{S}_{\text{PDFA}} \subsetneq \mathcal{S}_{\text{PRFA}} \subsetneq \mathcal{S}_{\text{PFA}} \subsetneq \mathcal{S}_{\text{MA}} \subsetneq \mathcal{S}$ [16]. More precisely, let $P \in \mathcal{S}$:

- $P \in \mathcal{S}_{\text{PDFA}}$ iff P has a finite number of residual languages.
- $P \in \mathcal{S}_{\text{PRFA}}$ iff there exists a convex generator for P composed of residual languages of P.
- $P \in \mathcal{S}_{\text{PFA}}$ iff there exists a convex generator for P.
- $P \in \mathcal{S}_{\text{MA}}$ iff $\text{res}(P)$ spans a finite dimensional vector space.

Any $P \in \mathcal{S}_{\text{PDFA}}$ can be generated by a minimal (in number of states) PDFA whose states correspond to the residual languages of P. In a similar way, it can be shown that any $P \in \mathcal{S}_{\text{PRFA}}$ has a unique minimal convex generator, composed of *prime* residual languages of P which correspond to the states of a minimal (in number of states) PRFA generating P (see [17] for a complete study). Such a canonical form does not exist for PFA or MA.

A PRFA $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$ is *prefix* if Q is a prefixial subset of Σ^*, $\iota(\varepsilon) = 1$, and $\forall (u, x, v) \in Q \times \Sigma \times Q$, $\varphi(u, x, v) \neq 0$ implies $ux = v$ or $ux \notin Q$. Transitions of the form (u, x, ux) are called *internal transitions*; the others are called *return transitions*. For example, automaton on Fig. 4, which can be built on the set $\{\varepsilon, a, b\}$, is a prefix PRFA, the transitions (ε, a, a) and (ε, b, b) are internal while (a, a, a), (a, a, b) and (b, a, b) are return transitions. Prefix PRFA are sufficient to generate all languages in $\mathcal{S}_{\text{PRFA}}$.

Let $P \in \mathcal{S}_{\text{PRFA}}$, $\text{Pm}(P)$ is the smallest prefixial subset of Σ^* such that $\forall u \in \text{Pm}(P)$, $\forall x \in \Sigma \cap \text{res}\left(u^{-1}P\right)$, $(ux)^{-1}P \in \text{conv}(\{v \in \text{Pm}(P) \mid v < ux\}) \Rightarrow ux \notin Pm(P)$. Let $U_{ux} = \{v \in \text{Pm}(P) \mid v < ux\}$ and for any word $u, v \in \text{Pm}(P)$, any $x \in \Sigma$ let $(\alpha_v^{ux})_{v \in U_{ux}}$ be positive parameters such that $(ux)^{-1}P = \sum_{v \in U_{ux}} \alpha_v^{ux} v^{-1}P$. Consider now the following PFA $A_P = \langle \Sigma, \text{Pm}(P), \varphi, \iota, \tau \rangle$ where $\iota(\epsilon) = 1$, $\varphi(u, x, v) = P(ux\Sigma^*)/P(u\Sigma^*)$ if $v = ux$ and $\varphi(u, x, v) = \alpha_v^{ux} P(ux\Sigma^*)/P(u\Sigma^*)$

if $(ux)^{-1}P = \sum_{v \in U_{ux}} \alpha_v^{ux} v^{-1}P$. It can be proved that A_P is a prefix PRFA which generates P [16]. See Fig. 4 for an example, where $\mathrm{Pm}(P) = \{\epsilon, a, b\}$.

5.2 The inference algorithm

For any finite prefixial set Q, let $\Theta_Q = \left\{\iota_u, \tau_u, \varphi_{u,v}^x \middle| \, u, v \in Q, x \in \Sigma\right\}$ be a set of variables. We consider the following set of constraints C_Q on Θ_Q:

$$C_Q = \begin{cases} 0 \leq \iota_u, \tau_u, \varphi_{u,v}^x \leq 1 & \text{for any } u, v \in Q, x \in \Sigma, \\ \iota_\varepsilon = 1 & \\ \iota_u = 0 & \text{for any } u \in Q \setminus \{\varepsilon\}, \\ \tau_u + \sum_{v \in Q, x \in \Sigma} \varphi_{u,v}^x = 1 & \text{for any } u \in Q, \\ \varphi_{u,v}^x = 0 & \text{for any } u, v, x \text{ s.t. } ux \neq v \text{ and } ux \in Q \ . \end{cases} \tag{2}$$

Any assignment θ of these variables satisfying C_Q defines a prefix PRFA A^θ.

Let $P \in \mathcal{S}$, let S be a complete presentation of P, for any finite prefixial set Q, any $\epsilon > 0$, any integer n and any $v \in \mathrm{res}(P)$ such that $\forall u \in Q$, $v > u$, let $I_{\Theta_Q}(v, S_n, \epsilon)$ be the following system: $I_{\Theta_Q}(v, S_n, \epsilon) = C_Q \cup C_{\text{internal}} \cup C_{\text{return}}(v)$ where $C_{\text{internal}} = \{P_{A^\theta}(w) = P_n(w) | \, w \in Q\}$ and $C_{\text{return}}(vx)$ $(x \in \Sigma)$ is the set of constraints $\left|(vx)^{-1}P_n(w\Sigma^*) - \sum_{u \in Q} \frac{\varphi_{v,u}^x}{P_n(vx\Sigma^*)} u^{-1}P_n(w\Sigma^*)\right| \leq \epsilon$ for all $w \in \mathrm{pref}(S_n)$ successors of vx. Let $I_{\Theta_Q}(S_n, \epsilon) = C_Q \cup C_{\text{internal}} \bigcup_{vx \in \mathrm{fr}(Q, P_n)} C_{\text{return}}(vx)$.

The constraint set C_{internal} can be solved immediatly and give parameters of the internal part of the automaton. It can be solved with $\iota(\varepsilon) = 1$, $\forall (u, x, ux) \in Q \times \Sigma \times Q$, $\varphi(u, x, ux) = P_n(ux\Sigma^*)/P_n(u\Sigma^*)$ and for all $u \in Q$, $\tau(u) = P_n(u)/P_n(u\Sigma^*)$. C_{return} is used to get parameters of return transitions. Remark $I_{\Theta_Q}(S_n, \epsilon)$ is a system composed of linear inequations.

Let DEES be the following algorithm:

`Input: a stochastic sample` S_n
`Output: a prefix PRFA` $A = \langle \Sigma, Q, \varphi, \iota, \tau \rangle$
$Q \leftarrow \{\varepsilon\}$, $F \leftarrow \Sigma \cap \mathrm{res}(P_n)$
`while` $F \neq \emptyset$ `do {`
  $v = \min F$, $F \leftarrow F \setminus \{v\}$
  `if` $I_{\Theta_Q}(v, S_n, \epsilon_n)$ has no solution `then{`
    $Q \leftarrow Q \cup \{v\}$, $F \leftarrow F \cup \{vx \in \mathrm{res}(P_n) \mid x \in \Sigma\}$`}}`
`if` $I_{\Theta_Q}(S_n, \epsilon_n)$ has some solution A^θ `then return` A^θ.
`else return` the prefix tree automaton of S_n.

DEES identifies $\mathcal{S}_{\text{PRFA}}$ in the limit with probability 1.

Theorem 5. *Let $P \in \mathcal{S}_{\text{PRFA}}$ and let S be a complete presentation of P, then with probability one, there exists $N \in \mathbb{N}$, such that for any $n > N$, the set of states of* $\mathrm{DEES}(S_n)$ *is* $\mathrm{Pm}(P)$ *and* $P_{\mathrm{DEES}(S_n)}$ *converges to P.*

Proof. It can be proved that, with probability one, after some rank, $I_{\Theta_Q}(S_n, \epsilon_n)$ has solutions if and only if there exists a prefix PRFA A^θ such that $P_{A^\theta} = P$. More precisely, it can be shown that $\mathrm{Pm}(P)$ is identified as the set of states from some index. Proofs are similar as the proofs of Prop. 2 and Prop. 3. □

Example. The the target be the prefix PRFA of Fig. 4. Let S_{20} be the sample such that $\text{pref}(S_{20}) = \{(\varepsilon : 20), (a : 12), (b : 8), (aa : 12), (ba : 2), (aaa : 11), (baa : 1), (aaaa : 4), (aaaaa : 3), (aaaaaa : 2)\}$ where $(u : n)$ means that n occurrences of u are counted.

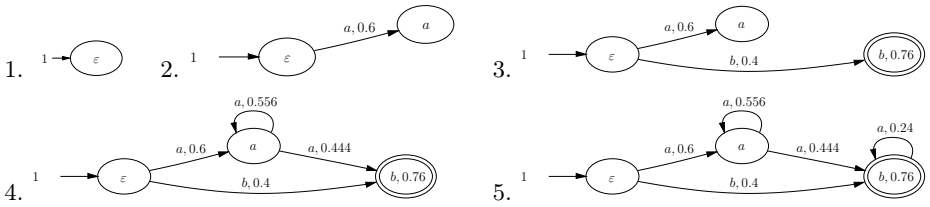

Fig. 5. DEES on S_{20}.

In the first step of the algorithm, $Q = \{\varepsilon\}$ (see Fig. 5.1).
$I_{\Theta_Q}(a, S_n, \epsilon)$ is the system:

$$\begin{cases} \left|a^{-1}P_n(\Sigma^*) - \frac{\varphi^a_{\varepsilon,\varepsilon}}{P_n(a\Sigma^*)}\varepsilon^{-1}P_n(\Sigma^*)\right| & \leq \epsilon \\ \left|a^{-1}P_n(a\Sigma^*) - \frac{\varphi^a_{\varepsilon,\varepsilon}}{P_n(a\Sigma^*)}\varepsilon^{-1}P_n(a\Sigma^*)\right| & \leq \epsilon \\ \vdots & \end{cases} \Leftrightarrow \begin{cases} \left|1 - \varphi^a_{\varepsilon,\varepsilon}\frac{20}{12}\cdot 1\right| & \leq \epsilon \\ \left|\frac{12}{12} - \varphi^a_{\varepsilon,\varepsilon}\frac{20}{12}\cdot\frac{12}{20}\right| & \leq \epsilon \\ \vdots & \end{cases}$$

which has no solution. Then we add the state a to Q (see Fig. 5.2). In the second step, $Q = \{\varepsilon, a\}$ and $I_{\Theta_Q}(b, S_{20}, \epsilon)$ has no solution. Then b is added to Q (see Fig. 5.3). In the third step, $Q = \{\varepsilon, a, b\}$ and as $\varphi^a_{a,\epsilon} = 0$, $\varphi^a_{a,a} = 0,556$ and $\varphi^a_{a,b} = 0,444$ is a solution of $I_{\Theta_Q}(aa, S_n, \epsilon)$, we construct the automaton with these values (see Fig. 5.4). In the last step, $Q = \{\epsilon, a, b\}$, and $\varphi^a_{b,\varepsilon} = \varphi^a_{b,a} = 0$, $\varphi^a_{b,b} = 0,24$ is a valid solution of $I_{\Theta_Q}(ba, S_n, \epsilon)$. The returned automaton is a prefix PRFA close to the target represented on Fig. 4.

6 Conclusion

We have shown that PFA are identifiable in the limit with probability one, that representing stochastic languages using Multiplicity Automata presents some serious drawbacks and we have proposed a subclass of PFA, the class of PRFA, and a learning algorithm which identifies this class and which can be implemented efficiently. In the absence of models which could precisely measure the performances of learning algorithms of PFA, we plan to compare experimentally our algorithm to other learning algorithms used in this field. We predict that we shall have better performances than algorithms that infer PDFA, since PRFA is a much more expressive class, but this has to be experimentally established. The questions remains whether richer subclasses of PFA can be efficiently inferred, and what is the level of expressivity needed in practical learning situations.

References

1. Paz, A.: Introduction to probabilistic automata. Academic Press, London (1971)
2. Abe, N., Warmuth, M.: On the computational complexity of approximating distributions by probabilistic automata. Machine Learning **9** (1992) 205–260
3. Dempster, A., Laird, N.M., Rubin, D.B.: Maximum likelyhood from incomplete data via the em algorithm. Journal of the Royal Statistical Society **39** (1977) 1–38
4. Baldi, P., Brunak, S.: Bioinformatics: The Machine Learning Approach. MIT Press (1998)
5. Freitag, D., McCallum, A.: Information extraction with HMM structures learned by stochastic optimization. In: AAAI/IAAI. (2000) 584–589
6. Gold, E.: Language identification in the limit. Inform. Control **10** (1967) 447–474
7. Angluin, D.: Identifying languages from stochastic examples. Technical Report YALEU/DCS/RR-614, Yale University, New Haven, CT (1988)
8. Carrasco, R., Oncina, J.: Learning stochastic regular grammars by means of a state merging method. In: International Conference on Grammatical Inference, Heidelberg, Springer-Verlag (1994) 139–152
9. Carrasco, R.C., Oncina, J.: Learning deterministic regular grammars from stochastic samples in polynomial time. RAIRO (Theoretical Informatics and Applications) **33** (1999) 1–20
10. de la Higuera, C., Thollard, F.: Identification in the limit with probability one of stochastic deterministic finite automata. Volume 1891 of Lecture Notes in Artificial Intelligence., Springer (2000) 141–156
11. Bergadano, F., Varricchio, S.: Learning behaviors of automata from multiplicity and equivalence queries. In: Italian Conference on Algorithms and Complexity. (1994)
12. Beimel, A., Bergadano, F., Bshouty, N.H., Kushilevitz, E., Varricchio, S.: On the applications of multiplicity automata in learning. In: IEEE Symposium on Foundations of Computer Science. (1996) 349–358
13. Beimel, A., Bergadano, F., Bshouty, N.H., Kushilevitz, E., Varricchio, S.: Learning functions represented as multiplicity automata. Journal of the ACM **47** (2000) 506–530
14. Thollard, F., Dupont, P., de la Higuera, C.: Probabilistic DFA inference using Kullback-Leibler divergence and minimality. In: Proc. 17th International Conf. on Machine Learning, (KAUFM) 975–982
15. Kearns, M., Mansour, Y., Ron, D., Rubinfeld, R., Schapire, R.E., Sellie, L.: On the learnability of discrete distributions. (1994) 273–282
16. Esposito, Y., Lemay, A., Denis, F., Dupont, P.: Learning probabilistic residual finite state automata. In: ICGI'2002, 6th International Colloquium on Grammatical Inference. LNAI, Springer Verlag (2002)
17. Denis, F., Esposito, Y.: Residual languages and probabilistic automata. In: 30th International Colloquium, ICALP 2003. Number 2719 in LNCS, SV (2003) 452–463
18. Angluin, D.: Queries and concept learning. Machine Learning **2** (1988) 319–342
19. Vapnik, V.N.: Statistical Learning Theory. John Wiley (1998)
20. Lugosi, G.: Pattern classification and learning theory. In: Principles of Nonparametric Learning. Springer (2002) 1–56
21. Hardy, G.H., Wright, E.M.: An introduction to the theory of numbers. Oxford University Press (1979)
22. Blondel, V.D., Canterini, V.: Undecidable problems for probabilistic automata of fixed dimension. Theory of Computing Systems **36** (2003) 231–245