Grammatical Inference as a Principal Component Analysis Problem

Raphaël Bailly RAPHAEL.BAILLY@LIF.UNIV-MRS.FR
François Denis FRANCOIS.DENIS@LIF.UNIV-MRS.FR
Liva Ralaivola LIVA.RALAIVOLA@LIF.UNIV-MRS.FR
Laboratoire d'Informatique Fondamentale de Marseille, Aix-Marseille Université, F-13013 Marseille, FRANCE

Abstract

One of the main problems in probabilistic grammatical inference consists in inferring a stochastic language, i.e. a probability distribution, in some class of probabilistic models, from a sample of strings independently drawn according to a fixed unknown target distribution p. Here, we consider the class of *rational stochastic languages* composed of stochastic languages that can be computed by *multiplicity automata*, which can be viewed as a generalization of probabilistic automata. Rational stochastic languages p have a useful algebraic characterization: all the mappings $\dot{u}p : v \to p(uv)$ lie in a finite dimensional vector subspace V_p^* of the vector space $\mathbb{R}\langle\langle\Sigma\rangle\rangle$ composed of all real-valued functions defined over Σ^*. Hence, a first step in the grammatical inference process can consist in identifying the subspace V_p^*. In this paper, we study the possibility of using Principal Component Analysis to achieve this task. We provide an inference algorithm which computes an estimate of this space and then build a multiplicity automaton which computes an estimate of the target distribution. We prove some theoretical properties of this algorithm and we provide results from numerical simulations that confirm the relevance of our approach.

1. Introduction

A *stochastic language* over the finite alphabet Σ is a probability distribution defined on the set of strings Σ^*, i.e. a mapping $p : \Sigma^* \to \mathbb{R}$ which satisfies (i) $0 \leq p(u) \leq 1$ for any string u and (ii) $\sum_{u\in\Sigma^*} p(u) = 1$. Given a set of strings independently drawn according to a fixed unknown stochastic language p, a usual goal in grammatical inference is to infer an estimate of p in some class of probabilistic models (Carrasco & Oncina, 1994; Thollard et al., 2000). Here, we consider the class of *rational stochastic languages* composed of stochastic languages that can be computed by a *multiplicity automaton* (Beimel et al., 2000; Denis & Esposito, 2008). A multiplicity automaton (MA) is a tuple $\langle\Sigma, Q, \iota, \tau, \varphi\rangle$ where Q is a set of states, $\iota : Q \to \mathbb{R}$ is an initialization function, $\tau : Q \to \mathbb{R}$ is a termination function and $\varphi : Q \times \Sigma \times Q \to \mathbb{R}$ is a transition function. The class of rational stochastic languages ($\mathcal{S}^{rat}(\Sigma)$) strictly encompasses the set of stochastic languages that can be defined by *probabilistic automata*, or equivalently, *Hidden Markov Models*. The main interest in considering rational stochastic languages relies in the fact that they have an algebraic characterization which proves to be useful for inference purpose (Denis et al., 2006). For any string u and any stochastic language p, let us denote by $\dot{u}p$ the function defined by $\dot{u}p(v) = p(uv)$. Then, a stochastic language p is rational if and only if the set $\{\dot{u}p|u \in \Sigma^*\}$ spans a finite dimensional vector subspace V_p^* of the vector space $\mathbb{R}\langle\langle\Sigma\rangle\rangle$ composed of all real-valued functions defined over Σ^*. The dimension of V_p^* is called the rank of p. It coincides with the minimal number of states needed by a multiplicity automaton to compute p. Hence, a first step in the grammatical inference process can consists in identifying the subspace $\{\dot{u}p|u \in \Sigma^*\}$. In this paper, we study the possibility to use Principal Component Analysis (PCA) techniques to achieve this task (see (Clark et al., 2006) for another use of PCA in the field of Grammatical Inference).

Next section (Section 2) gives a high level overview of the results presented in this work. Preliminaries on rational stochastic languages are given in Section 3. The main inference algorithm is described in Section 4. Consistency properties are proved in Section 5, which also contains the method to infer the rank of the target. Some experiments are provided in Section 6.

2. Overview of Results

The class of rational stochastic languages is included in the Hilbert space $l_2(\Sigma^*)$ composed of all real valued func-

Appearing in *Proceedings of the 26^{th} International Conference on Machine Learning*, Montreal, Canada, 2009.

tions r such that the sum $\sum_u r(u)^2$ is convergent and equipped with the corresponding dot product. Given a sample of strings S drawn according to an unknown stochastic language p and a dimension parameter k, we build the k-dimensional subspace $V^*_{S,k}$ of $\mathbb{R}\langle\langle\Sigma\rangle\rangle$ that minimizes $\sum_u ||\dot{u}p_S - \Pi_{V^*_{S,k}}\dot{u}p_S||^2$ where p_S denotes the empirical distribution induced by S and where Π_V denotes the orthogonal projection of the subspace V. Then, we use the linear dependencies between the projections on $V^*_{S,k}$ of the elements $\dot{u}p_S$ to build a multiplicity automaton A_S. The method is consistent: if $p_S = p$, the space $V^*_{S,k}$ is equal to the space spanned by the set $\{\dot{u}p|u \in \Sigma^*\}$, and the automaton A_S computes the target p. And we prove that if k is equal to the rank of the target, the linear dependencies computed in the inferred space $V^*_{S,k}$ converge to the correct ones with a rate of convergence equal to $O(|S|^{-1/2})$. Lastly, we show that the dimension d of the target p can be inferred from the sample S: in order to build the subspace $V^*_{S,k}$, using PCA techniques, we compute the eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m \geq 0$ of a positive semi-definite matrix N and we show that the sum $\sum_{i>d}\lambda_i$ of the last $m-d$ eigen values tends to 0 as the size of the sample S increases. This property provides an algorithm to infer the dimension d.

3. Preliminaries

Let Σ^* be the set of strings on the finite alphabet Σ. The empty string is denoted by ε, and the length of a string u is denoted by $|u|$. For any integer k, we denote by Σ^k the set $\{u \in \Sigma^* \mid |u| = k\}$ and by $\Sigma^{\geq k}$ the set $\{u \in \Sigma^* \mid |u| \geq k\}$. Let $L \subseteq \Sigma^*$. We denote by $pref(L)$ (resp. $suf(L)$) the set $\{u \in \Sigma^* | \exists v \in \Sigma^* uv \in L\}$ (resp. $\{u \in \Sigma^* | \exists v \in \Sigma^* vu \in L\}$).

A *formal power series* is a mapping r from Σ^* to $\mathbb{R}$. The set of all formal power series is denoted by $\mathbb{R}\langle\langle\Sigma\rangle\rangle$. It is an $\mathbb{R}$-vector space. For any series r and any string $u \in \Sigma^*$, we denote by $\dot{u}r$ the series defined by $\dot{u}r(w) = r(uw)$.

A *multiplicity automaton (MA)* is a 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 $\varphi(q, wx, q') = \sum_{q'' \in Q} \varphi(q, w, q'')\varphi(q'', x, q')$ and $\varphi(q, \varepsilon, q') = 1$ if $q = q'$ and 0 otherwise, for any $q, q' \in Q$, $x \in \Sigma$ and $w \in \Sigma^*$. For any MA $A = \langle\Sigma, Q, \varphi, \iota, \tau\rangle$, we define the series r_A by $r_A(w) = \sum_{q,q' \in Q} \iota(q)\varphi(q, w, q')\tau(q')$. A series r is *rational* if it can be computed by a multiplicity automaton. It can be proved that a series r is rational iff the vector subspace of $\mathbb{R}\langle\langle\Sigma\rangle\rangle$ spanned by the set $\{\dot{u}r|u \in \Sigma^*\}$ has a finite dimension, which is called the *rank* of r. This dimension coincides with the minimal number of states needed for a MA to compute r. The family of all rational series is denoted by $\mathbb{R}^{rat}\langle\langle\Sigma\rangle\rangle$.

A *stochastic language* is a formal series p which only takes non negative values and such that $\sum_{u\in\Sigma^*} p(u) = 1$. A stochastic language defines a probability distribution over Σ^*. A *rational stochastic language* (Denis & Esposito, 2008) is a stochastic language which can be computed by a multiplicity automaton. The set of all stochastic languages (resp. rational stochastic languages) defined on Σ^* is denoted by $\mathcal{S}(\Sigma)$ (resp. $\mathcal{S}^{rat}(\Sigma)$). It can be shown that all probability distributions computed by Hidden Markov Models (or equivalently, probabilistic automata) are rational stochastic languages but the converse is false.

The values of a rational stochastic language decrease exponentially fast. It can be shown that for any $p \in \mathcal{S}^{rat}(\Sigma)$, there exists $0 < \rho < 1$ such that for any integer $n \geq 0$, $p(\Sigma^{\geq n}) = O(\rho^n)$. Let $A = \langle\Sigma, Q, \varphi_A, \iota_A, \tau_A\rangle$ be an MA that computes the rational stochastic language p and suppose that A has a minimal number of states d (equal to the dimension of the space spanned by $\{\dot{u}p|u \in \Sigma^*\}$). Let $B = \langle\Sigma, Q, \varphi_B, \iota_B, \tau_B\rangle$ be another MA defined on the same set of states as A. It can be shown (Denis et al., 2006) that if the parameters of B converge to the parameters of A, then, the series computed by B converges to the series computed by A for the norm $||\cdot||_1$. More precisely, for any $\epsilon > 0$, there exists $\alpha > 0$ such that if $|\iota_A(q) - \iota_B(q)| < \alpha$, $|\tau_A(q) - \tau_B(q)| < \alpha$, $|\varphi_A(q, x, q') - \varphi_B(q, x, q')| < \alpha$ for any states q, q' and any letter x then $\sum_{u\in\Sigma^*} |p(u) - r_B(u)| < \epsilon$. As a first consequence, if the parameters of B are sufficiently close to the parameters of A, then the series computed by B is absolutely convergent and $\sum_{u\in\Sigma^*} r_B(u)$ is arbitrarily close to 1. A second consequence is that the absolutely convergent series r_B can be used to compute a stochastic language p_{r_B} which approximates p (see algorithm 1). It can be shown that if $\sum_{u\in\Sigma^*} |p(u) - r_B(u)| < \epsilon$ then, $\sum_{u\in\Sigma^*} |p_{r_B}(u) - r_B(u)| < \frac{\varepsilon(3+\varepsilon)}{1-\varepsilon}$ and therefore, $\sum_{u\in\Sigma^*} |p(u) - p_{r_B}(u)| = O(\varepsilon)$.

We consider the Hilbert space $l_2(\Sigma^*)$ composed of the rational series $r \in \mathbb{R}\langle\langle\Sigma\rangle\rangle$ such that $\sum_{w\in\Sigma^*} r(w)^2 < \infty$ and where the inner product $\langle\cdot,\cdot\rangle$ is defined by $\langle r, s\rangle = \sum_{w\in\Sigma^*} r(w)s(w)$. Hence, $||r|| = \left(\sum_{w\in\Sigma^*} r(w)^2\right)^{1/2}$. Clearly, $\mathcal{S}^{rat}(\Sigma^*) \subseteq l_2(\Sigma^*)$ and if $r \in l_2(\Sigma^*)$ then $\dot{u}r \in l_2(\Sigma^*)$. Given a vector subspace V of $l_2(\Sigma^*)$, we denote by Π_V the orthogonal projection on V.

4. Principle of the algorithm

Let $p \in \mathcal{S}^{rat}(\Sigma)$ be a rational stochastic language, let V^*_p be the vector space spanned by $\{\dot{u}p|u \in \Sigma^*\}$ and let $d = dim(V^*_p)$. Let S be a sample independently drawn according to p and let p_S be the empirical distribution defined from S. We first build an estimate V_S of V^*_p from S. Then, we show that V_S can be used to build a MA A_V whose as-

Algorithm 1 RandomDraw. Defines a stochastic language p_r such that for any string u, $p_r(u) = 0$ if $r(u) \leq 0$ and $p_r(u) \leq r(u)/r(\Sigma^*)$ otherwise.

Data : An absolutely convergent rational series r s.t. $\sum_{u\in\Sigma^*} r(u) > 0$.
Result : A string u drawn randomly
$X = \{\varepsilon\} \cup \Sigma$
For $x \in \Sigma$, let $p_x = max(0, \dot{x}r(\Sigma^*))$
Let $p_\varepsilon = max(0, r(\varepsilon)$.
Let $s = \sum_{x\in X} p_x$
Draw an element z of X according to the multinomial model $(p_x/s)_{x\in X}$
if $z = \varepsilon$ **then**
  return ε
else
  return z+RandomDraw($\dot{z}r$)
end if

sociated rational series approximates the target p. In this section, we shall implictly suppose that the dimension d of V_p^* is known. We will show in the next section how it may be estimated from the data.

4.1. Estimating the target space

Let $k \geq 0$ be an integer. The first step consists in finding the k-dimensional vector subspace $V_{S,k}^*$ of $l_2(\Sigma^*)$ which minimizes the distance to $\{\dot{u}p_S | u \in \Sigma^*\}$:

$$V_{S,k}^* = Argmin_{dim(V)=k} \sum_{u\in\Sigma^*} ||\dot{u}p_S - \Pi_V \dot{u}p_S||^2.$$

Since the support of p_S is finite, $V_{S,k}^*$ can be computed by using PCA. Let $v_1, \ldots, v_m$ be an enumeration of $suf(S)$. For any $u \in pref(S)$, let x_u be the vector of $\mathbb{R}^m$ defined by $x_u[i] = \dot{u}p_S(v_i) - \frac{1}{Card(suf(S))} \sum_{v\in suf(S)} \dot{u}p_S(v)$. Let X_S be the matrix containing the vectors x_u as rows. The matrix $M_S = X_S' X_S$ is positive semi-definite and the eigenvectors $(q_1, \ldots, q_k)$ corresponding to the k largest (positive) eigenvalues form an orthogonal basis of $V_{S,k}^*$.

4.2. Building the automaton

Now, let V be a k-dimensional vector subspace of $l_2(\Sigma^*)$ and let $B = \{q_1, \ldots, q_k\}$ be a basis of V. We define the multiplicity automaton $A_{S,B} = \langle \Sigma, Q_{S,B}, \varphi_{S,B}, \iota_{S,B}, \tau_{S,B} \rangle$ by

- $Q_{S,B} = \{q_1, \ldots, q_k\}$,
- $\varphi_{S,B}(q_i, x, q_j) = \alpha_{i,x}^j$ where $\alpha_{i,x}^j$ is the j-th component of $\Pi_V(\dot{x}q_i)$ in the basis $(q_1, \ldots, q_k)$
- $\iota_{S,B}(q_i) = \iota_i$ where ι_i is the i-th component of $\Pi_V(p_S)$ in the basis $(q_1, \ldots, q_k)$,
- $\tau_{S,B} = q_i(\varepsilon)$.

The automaton $A_{S,B}$ computes a rational series which only depends on S and V. Indeed, let us define the linear operator $\psi : \Sigma^* \to \mathcal{L}(\mathbb{R}\langle\langle\Sigma\rangle\rangle)$ by

$$\psi(u) = \begin{cases} \Pi_V & \text{if } u = \varepsilon \\ \Pi_V \circ \dot{x} \circ \psi(v) & \text{if } u = xv \end{cases}$$

Let r be the series computed by $A_{S,B}$. The following proposition shows that $r(u) = \psi(u)(p_S)(\varepsilon)$, which does not depend on B.

Proposition 1. *For any string u, $r(u) = \psi(u)(p_S)(\varepsilon)$.*

Proof. It can easily be shown, by induction on the length of u, that $\sum_{q'\in B} \varphi(q, u, q')q' = \psi(u)(q)$. Then, we have:

$$\begin{aligned} r(u) &= \sum_{q,q'} \iota(q)\varphi(q, u, q')\tau(q') \\ &= \left(\sum_q \iota(q) \sum_{q'} \varphi(q, u, q')q' \right)(\varepsilon) \\ &= \left(\sum_q \iota(q)\psi(u)(q) \right)(\varepsilon) \\ &= \psi(u)(\sum_q \iota(q)q(\varepsilon) \\ &= \psi(u)(p_S)(\varepsilon). \end{aligned}$$

□

Note that if we take $V = V_p^*$ and if we set $\iota(q)$ to the i-th component of $\Pi_{V_p^*}(p)$ in the basis $(q_1, \ldots, q_k)$, then $\psi(u) = \dot{u}\Pi_{V_p^*}$ and $r = p$: the automaton computes the target p, which can be seen as a weak consistency property of the algorithm.

4.3. The algorithm

Algorithm 4.3 takes a sample S and an estimate of the rank d of the target space as input and outputs a multiplicity automaton. It first computes the space $V_{S,k}^*$, then it computes an automaton $A_{S,B}$ from S and a basis of $V_{S,k}^*$, which defines a rational series that only depends on S and d.

4.4. Example

Let us consider the stochastic language p defined by the probabilistic automaton described on Figure 1 and let S be a sample composed of 1000 examples independently drawn according to p. Table 1 shows the beginning of the array X_S (before centering).

Algorithm 2 Building an automaton corresponding to a sample S and a dimension d

Data : A sample $S = \{s_i \in \Sigma^*, 1 \leq i \leq |S|\}$ i.i.d. according to a distribution p, a dimension d
Result : A Multiplicity Automaton A

$W = \{w_i, 1 \leq i \leq |W|\} \leftarrow$ prefixes of S
$U = \{u_i, 1 \leq i \leq |U|\} \leftarrow$ suffixes of S
$X \leftarrow$ a $|W| \times |U|$ matrix
$X_{i,j} \leftarrow \dot{w}_i p_S(u_j) - \frac{1}{|U|} \sum_{v \in U} \dot{u} p_S(v)$
$N = X'X$
$(\lambda_i, q_i) \leftarrow$ eigenvalues of N in decreasing order, and corresponding eigenvectors
$B \leftarrow (q_1, \ldots, q_d)$
/*$\Pi_B =$ orth. proj. on the vector subspace spanned by B*/
for i from 1 to d **do**
  $\iota(q_i) \leftarrow$ the i-th coordinate of $\Pi_B(p_S)$ in B
  $\tau(q_i) \leftarrow q_i(\varepsilon)$
  for each $x \in \Sigma$ **do**
    for j from 1 to d **do**
      $\varphi(q_i, x, q_j) \leftarrow$ the j-th coordinate of $\Pi_B(\dot{x} q_i)$ in B
    end for
  end for
end for
return $A = \langle \Sigma, \{q_1, \ldots, q_n\}, \varphi, \iota, \tau \rangle$

Table 1. First rows and columns of X_S (before centering).

X_S	ε	a	b	aa	ab	ba	bb	...
ε	0.0	0.091	0.08	0.027	0.026	0.074	0.058	
a	0.091	0.027	0.026	0.009	0.01	0.008	0.007	
b	0.08	0.074	0.058	0.016	0.026	0.055	0.052	
...								

The 3 largest eigenvalues of the matrix $M_S = X'_S X_S$ are (in decreasing order) $7.71 \cdot 10^{-1}, 2.14 \cdot 10^{-1}, 5.98 \cdot 10^{-3}$.

Figure 1 shows the automaton output by the learning algorithm for $d = 1$ and $d = 2$. The quadratic distance between the target and the learned automaton is $1.758 \cdot 10^{-2}$ for the first automaton ($d = 1$), $1.487 \cdot 10^{-4}$ for the second ($d = 2$) and $2.325 \cdot 10^{-4}$ for $d = 3$ (the automaton is not represented). We can remark that the distance is minimal when the number of states is correct. Table 2 shows the first values computed by the target and the learned automaton.

5. Consistency

5.1. Consistency when the rank of the target is known

We show that the solution computed by the algorithm converges to the target as the size of the sample S increases.

Table 2. The first values computed with the target automaton A (Fig. 1 (a)), the first learned MA A_1 (Fig. 1 (b)), the second learned MA A_2 (Fig. 1 (c)) and the stochastic language $p_{r_{A_2}}$ derived from A_2 by using algorithm *RandomDraw*.

	ε	a	b	aa	ab	ba	bb
p_A	0.0	0.083	0.083	0.028	0.028	0.069	0.069
r_{A_1}	0.057	0.021	0.041	0.007	0.015	0.015	0.030
r_{A_2}	0.000	0.092	0.080	0.025	0.028	0.071	0.066
$p_{r_{A_2}}$	0.000	0.10	0.086	0.028	0.030	0.077	0.072

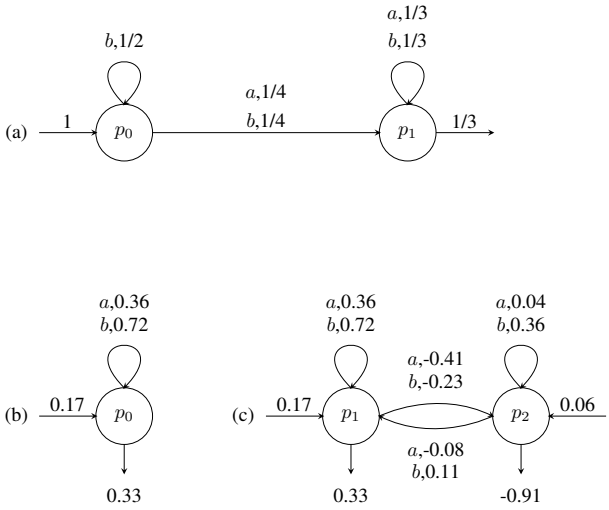

Figure 1. A target stochastic language defined by the probabilistic automaton (a). The learned automaton with $d = 1$ (b) and $d = 2$ (c). The parameters have been trunked after the second decimal. The quadratic distance between the target (a) and the learned automaton is $1.76 \cdot 10^{-2}$ for the first automaton (b) and $1.49 \cdot 10^{-4}$ for the second (c).

Proposition 2. *Let S be a sample i.i.d. according to a rational stochastic language p with rank d. Then $\mathbb{E}[\|\Pi_{V^*_{S,d}}(\dot{w}p_S) - \dot{w}p\|] \to 0$ uniformly wrt w as the size of S increases.*

Proof. Let $w \in \Sigma^*$. As $p_S(w)$ is a binomial distribution with parameters $|S|$ and $p(w)$, $\mathbb{V}[p_S(w)] = \frac{p(w)(1-p(w))}{|S|} \leq \frac{p(w)}{|S|}$. Thus, $\mathbb{E}[\|\dot{w}p_S - \dot{w}p\|^2] \leq \sum_u \mathbb{E}[p_S(wu) - p(wu)]^2 \leq \sum_u \frac{p(wu)}{|S|} = \frac{p(w\Sigma^*)}{|S|}$. Therefore, $\mathbb{E}[\|\dot{w}p_S - \dot{w}p\|^2]$ tends to 0 as the size of S increases.

We have $\|\dot{w}p_S - \Pi_{V^*_p}(\dot{w}p_S)\| \leq \|\dot{w}p_S - \dot{w}p\|$ since $\dot{w}p \in V^*_p$. Furthermore, $\sum_w \mathbb{E}[\|\dot{w}p_S - \dot{w}p\|^2] \leq \sum_w \frac{p(w\Sigma^*)}{|S|} = \sum_{n \in \mathbb{N}} \frac{p(\Sigma^{\geq n})}{|S|} \leq \sum_{n \in \mathbb{N}} \frac{O(\rho^n)}{|S|} = O(1/|S|)$ for some $0 < \rho < 1$.

Therefore, $(\mathbb{E}[\|\dot{w}p_S - \Pi_{V^*_p}(\dot{w}p_S)\|])^2 \leq (\mathbb{E}[\|\dot{w}p_S -$

$\dot{w}p\|])^2 \leq \mathbb{E}[\|\dot{w}p_S - \dot{w}p\|^2] \leq \sum_w \mathbb{E}[\|\dot{w}p_S - \dot{w}p\|^2] = O(1/|S|)$ which proves that $\mathbb{E}[\|\Pi_{V^*_{S,d}}(\dot{w}p_S) - \dot{w}p\|] \leq \mathbb{E}[\|\dot{w}p_S - \Pi_{V^*_p}(\dot{w}p_S)\| + \|\dot{w}p_S - \dot{w}p\|] \leq O(|S|^{-1/2})$.

That is, $\mathbb{E}[\|\Pi_{V^*_{S,d}}(\dot{w}p_S) - \dot{w}p\|] \to 0$ uniformly wrt w as the size of S increases. □

It can easily be deduced from Proposition 2 that if $\dot{u}_1 p, \ldots, \dot{u}_p p$ form a basis of V^*_p, if $\dot{v}p = \sum_{i=1}^d \alpha_i \dot{u}_i p$ and if $\Pi_{V^*_{S,d}}(\dot{v}p_S) = \sum_{i=1}^d \hat{\alpha_i} \Pi_{V^*_{S,d}}(\dot{u}_i p_S)$, then $\mathbb{E}[Max(\|\alpha_i - \hat{\alpha_i}\|)] = O(|S|^{-1/2})$.

In particular, linear relations between residuals are exactly identified in the limit.

5.2. Convergence of eigenvalues

We will use two results about eigenvalues:

Lemma 1 (Shawe-Taylor et al. (2001))**.** *Let M be the variance-covariance matrix of the set $\{\dot{w}p, w \in W\}$, with $|W| = m$. Let $\{\lambda_i, 1 \leq i \leq m\}$ be the set of eigenvalues of M. Let V^*_d be the d-dimension vector subspace wich minimizes $\sum_{w\in W} \|\dot{w}p - \Pi_{V^*_d}(\dot{w}p)\|^2$. Then, $\forall d$ s.t. $1 \leq d \leq m$,*

- $\lambda_d = \max_{dim(V)=d} \min_{v\in V\setminus\{0\}} \sum_{w\in W} \|\Pi_v(\dot{w}p)\|^2$;
- $\sum_{d+1\leq i\leq m} \lambda_i = \sum_{w\in W} \|\dot{w}p - \Pi_{V^*_d}(\dot{w}p)\|^2$.

Proposition 3. *Let S be a sample i.i.d. according to a rational stochastic language p with rank d, and p_S the empirical distribution deduced from S. Let Π_V be the orthogonal projection on V. Let V^*_S be the d-dimension vector subspace wich minimizes $\sum_{w\in pref(S)} \|\dot{w}p_S - \Pi_{V^*_S}(\dot{w}p_S)\|^2$, $V^* = Res(p)$. Let $\{\lambda_i, 1 \leq i \leq m\}$ be the set of eigenvalues of the variance-covariance matrix of $\{\dot{w}p_S, w \in pref(S)\}$, with $m = |pref(S)|$.*

Then $\mathbb{E}[\sum_{d+1\leq i\leq m} \lambda_i]$ tends to 0 as the size of S increases.

Proof. $\mathbb{E}[\sum_{d+1\leq i\leq |pref(S)|} \lambda_i] = \sum_{w\in pref(S)} \|\dot{w}p_S - \Pi_{V^*_S}(\dot{w}p_S)\|^2 \leq \frac{M}{|S|}$, with $M = max\{|w|, w \in pref(S)\}$. This tends to 0 as $|S|$ tends to infinity. □

Proposition 4. *Let S be an infinite sample i.i.d. according to a rational stochastic language p with rank d, and S_n the n first elements. Let p_S the empirical distribution deduced from S, and p_S the one deduced from S_n. Let*

$$\lambda_k = \max_{dim(V)=k} \min_{v\in V\setminus\{0\}} \sum_{w\in\Sigma^*} \|\Pi_V(\dot{w}p)\|^2$$

$$\lambda_{k,n} = \max_{dim(V)=k} \min_{v\in V\setminus\{0\}} \sum_{w\in\Sigma^*} \|\Pi_V(\dot{w}p_{S_n})\|^2$$

Then $\liminf_{n\to\infty} \lambda_{k,n} \geq \lambda_k$.

Proof. Let $\epsilon > 0$. There exists $W_{k,\epsilon} \subset \Sigma^*$, $|W_{k,\epsilon}| < \infty$ such that

$$\lambda_k - \epsilon/2 \leq \max_{dim(V)=k} \min_{v\in V\setminus\{0\}} \sum_{w\in W_{k,\epsilon}} \|\Pi_V(\dot{w}p)\|^2 = \lambda'_{k,\epsilon}$$

We have also $\lambda'_{k,\epsilon} \leq \lambda_k$ ($\sum_{w\in W_{k,\epsilon}} \|\Pi_V(\dot{w}p)\|^2 \leq \sum_{w\in\Sigma^*} \|\Pi_V(\dot{w}p)\|^2$ since $W_{k,\epsilon} \subset \Sigma^*$).

There exists N_ϵ such that $\forall w \in W_{k,\epsilon}, \forall v \in \mathbb{R}^{\Sigma^*}, \forall n > N_\epsilon$, $\|\Pi_v(\dot{w}p)\|^2 - \|\Pi_v(\dot{w}p_{S_n})\|^2 \leq \frac{\epsilon}{2|W_{k,\epsilon}|}$ because Π_v is 1-lipschitz. Let

$$\lambda''_{k,\epsilon} = \max_{dim(V)=k} \min_{v\in V\setminus\{0\}} \sum_{w\in\Sigma^*} \|\Pi_V(\dot{w}p_{S_{N_\epsilon}})\|^2 \leq \lambda_{k,N_\epsilon}$$

We have:

$$\lambda_{k,N_\epsilon} \geq \lambda''_{k,\epsilon} \geq \lambda'_{k,\epsilon} + \epsilon/2 \geq \lambda_k + \epsilon$$

This is true for every ϵ, and we have the conclusion. □

Algorithm 3 first finds a lower bound of the rank with the method we just described, then it detects the point (greater than the bound previously found) on the curve of the logarithm of the eigenvalues that corresponds to the highest second order slope (see the last loop of the algorithm).

6. Numerical Simulations

We carry out two types of simulations: first we focus on the ability of our algorithm to retrieve the structure and coefficients of a specific automaton, and then we perform experiments on randomly drawn automata.

We intend to study the possibility of determining the rank of the target p from a finite sample S i.i.d. according to p.

Comparison with other methods to find the structure (or the number of states) of a statistical model from a learning sample are not carried out here and are left for future work. However, we would like to stress that existing inference algorithms either provide models of far lower expressiveness (such as PDFA), or are designed to work in an asymptotic way, and still need be tuned to obtain an efficient implementation (such as DEES). We therefore anticipate our method to provide more conclusive empirical results.

6.1. Rank Estimation Procedures

We propose four criteria to estimate the target rank.

Eigenvalue-based Test We use the algorithm described before, based on eigenvalues, but we don't take the value $\frac{L_{max}}{|S|}$ as bound, we use instead annother value. Let $p'(w) = p_S(w) + \frac{1}{|S|}$ be a smoothing of p_S. Variance of $p(w)$ is estimated with $\mathbb{V}'(w) = \frac{p'(w)(1-p'(w))}{|S|}$. Finally:

Algorithm 3 Finding the rank d

Data : A sample $S = \{s_i, 1 \leq i \leq |S|\}$ i.i.d. according to a distribution p
Result : A dimension d

$W = \{w_i, 1 \leq i \leq |W|\} \leftarrow$ prefixes of S
$U = \{u_i, 1 \leq i \leq |U|\} \leftarrow$ suffixes of S
$L_{max} \leftarrow max\{|s_i|, s_i \in S\}$
$M \leftarrow$ a $(|W|, |U|)$ matrix
$M_{i,j} \leftarrow \dot{w}_i p_S(u_j) = \frac{|\{s_k = w_i u_j, s_k \in S\}|}{|S|}$
$N = MM^T$
$(\lambda_i)_{1 \leq i \leq |U|} \leftarrow$ eigenvalues of N in decreasing order
$d \leftarrow |U|$
$sum \leftarrow 0$
while $sum < \frac{L_{max}}{|S|}$ and $d > 0$ **do**
  $sum \leftarrow sum + \lambda_d$
  $d \leftarrow d - 1$
end while
$d \leftarrow d + 1$
$e \leftarrow d$
$\delta'' \leftarrow \frac{\lambda_d \lambda_{d+2}}{\lambda_{d+1}^2}$
while $e < |U| - 2$ **do**
  $e \leftarrow e + 1$
  if $\delta'' \leq \frac{\lambda_e \lambda_{e+2}}{\lambda_{e+1}^2}$ **then**
    $d \leftarrow e$
  end if
end while
return d

$$L'_{max} = \sum_{u \in W, v \in W} \left(p_S(uv) + \sqrt{\mathbb{V}'(uv)} \right)$$

We then estimate the "elbow" position by maximizing the second order slope of the eigenvalues logarithm curve, that is, if $\{\lambda_1, \ldots, \lambda_n\}$ are the eigenvalues, computing $\arg\max_i \frac{\lambda_i \lambda_{i+2}}{\lambda_{i+1}^2}$ for i greater than the dimension previously found.

Distance to Test Sample For each target automaton, we build automata for rank d from 1 to 11. We want to check here if it is possible to infer the rank by minimizing the distance beetween each of the eleven automata and a test sample.

The inferred automata do not, in general, compute a probability distribution. We simulate one from each inferred automaton with Algorithm 1. We then compute here an approximation of the distance beetween the target probability p and each inferred probability p_n for $1 \leq n \leq 11$, only for strings of length lower than 10. The distances considered are l_1 and l_2 distances, and KL-divergence.

$$D = \sum_{w \in \Sigma^{\leq 10}} d(p(w), p_n(w))$$

6.2. Single case study

We start with the case of a single 5-state automaton on the alphabet $\Sigma = \{0, 1\}$ described on Fig. 2. Fig. 3 represents the eigenvalues curves, in log scale, for a sample size of 1000, 5000, 20000 and 100000 strings.

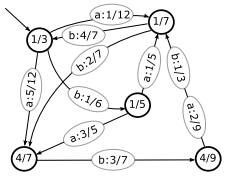

Figure 2. The 5-states automaton

One can see the eigenvalues decrease exponentially, so the sum from the i-th eigenvalue to the last one is close to the i-th eigenvalue. Considering this, the eigenvalues-based algorithm identifies the highest second order slope point under the horizontal line, representing the value $L'_{max}/|S|$:

With a sample size of 1000, the retained dimension is 3 while for larger sizes the correct dimension is identified.

In the case of the l_1 distance, the correct rank is found for all samples.

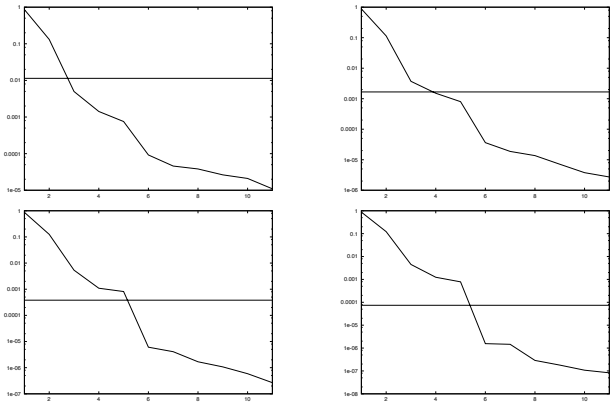

Figure 3. Curves of Eigenvalues (in logarithm scale) for sample size of 1000, 5000, 20000 and 100000 strings, from left to right and top to down. The horizontal line represents the value $L'_{max}/|S|$.

6.3. General case study

We implement the following protocol. We first randomly generate 500 probabilistic 4-state automata on the alphabet

$\Sigma = \{0, 1\}$ and we generate for each of them one sample of 1000, 5000, 20000 and 100000 strings. From each sample, we generate a data matrix on which we perform a PCA. The test for the dimension is a first estimation of the target rank. We generate weighted automata for dimensions from 1 to 11, and we compute the l_1, l_2-distances and the KL-divergence between each target automaton and its aproximations of ranks from 1 to 11. This gives three other estimations of the target rank.

Randomly generated automata Each coefficient of the initial vector and final vector is uniformly generated between 0 and 1. Each coefficient of the transition matrices is uniformly generated with probability $1/3$, and is equal to 0 with probability $2/3$.

We first divide the initial vector's coordinates by their sum. Then, for every state, we do the sum of final coefficient and all the outgoing transitions. This must sum up to 1, so we divide each term by the sum to normalize.

We obtain this way a weighted automaton wich computes a probability distribution over Σ^*.

Data Matrix We do not consider the whole data matrix but the one formed with the prefixes and suffixes of length lower or equal than 4, in lexicographic order. Let us call this set $W = \{\epsilon, 0, 1, 00, 01, 10, 11, \ldots, 1110, 1111\}$. $|W| = 31$. The matrix X_S is defined by: $X_{Si,j} = p_S(w_i w_j)$.

After performing the PCA, we obtain 31 eigenvalues and eigenvectors.

The Fig.4 represents the curve of the average eigenvalues computed by the algorithm for each sample size. One can see clearly the "elbow" at position 5 with a sample size greater than 20000. The horizontal line represents the value $L'_{max}/|S|$.

The Fig. 5, 6, 7 and 8 represents the compared dimensions obtained by the eigenvalues-based algorithm, l_1 and l_2-distance minimization, and KL-divergence minimization.

Results From our experiments of rank prediction, one can observe that:

- The KL-divergence criterion performances decrease with the sample size.
- l_1 and l_2 minimization criteria slowly increase with the sample size.

We chose to compute distances with the real target distribution, in order to have a convenient way to compare results. In reality, one only knows the empirical distribution deduced from learning sample, and this should decrease performances of those three criteria.

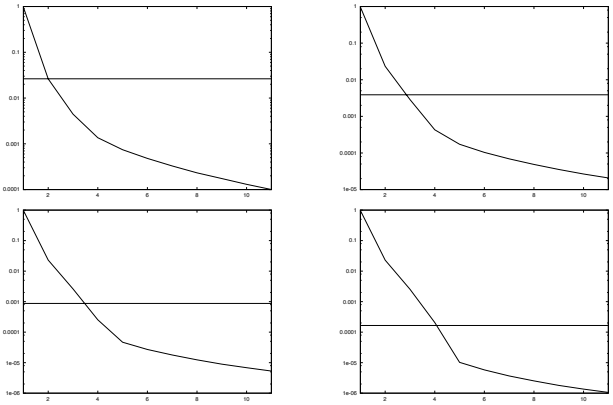

Figure 4. Curves of the average eigenvalues (in logarithm scale) for sample size of 1000, 5000, 20000 and 100000 strings, from left to right and top to down. The horizontal line represents the average value $L'_{max}/|S|$.

- The eigenvalues criterion performances seem to increase faster than the others when the size of the sample increases.

7. Conclusion

We introduce a new approach in probabilistic grammatical inference which differs from previous one on several aspects. Most classical inference algorithms used in the field of probabilistic grammatical build an automaton or a grammar iteratively from a sample S; starting from an automaton composed of only one state, they have to decide whether a new state must be added to the structure. This iterative decision relies on a statistical test with a known drawback: as the structure grows, the test relies on less and less examples. Instead of this iterative approach, we tackle the problem globally and our algorithm computes in one step the space needed to build the output automaton. That is, we have reduced the problem set in the classical probabilistic grammatical inference framework into a classical optimization problem. We now need to experimentally compare our approach to existing ones on real data: this is a work in progress. A further consequence of our approach is that it will be possible to introduce non linearity via the kernel PCA technique developed in (Shawe-Taylor et al., 2001) and by the Hilbert space embedding of distributions proposed in (Smola et al., 2007).

Acknowledgments

This work is partially supported by the IST Program of the EC, under the FP7 Pascal 2 Network of Excellence, ICT-216886-NOE and by the ANR project SEQUOIA.

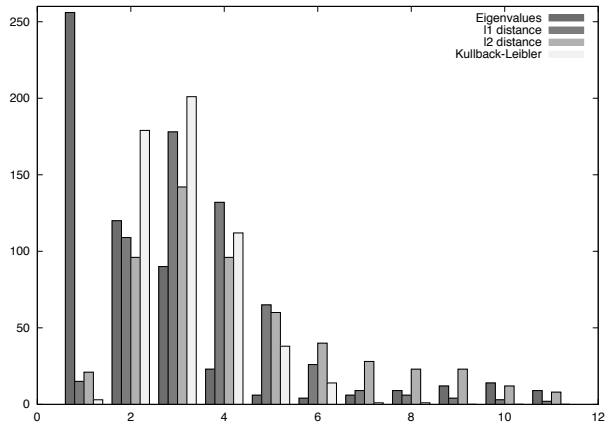

Figure 5. Estimated rank with the four methods $|S| = 1000$.

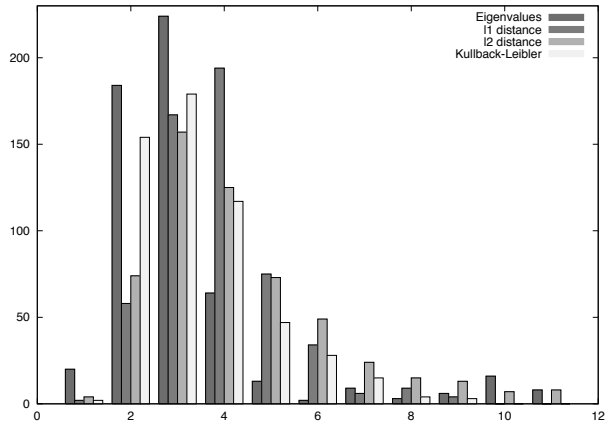

Figure 6. Estimated rank with the four methods $|S| = 5000$.

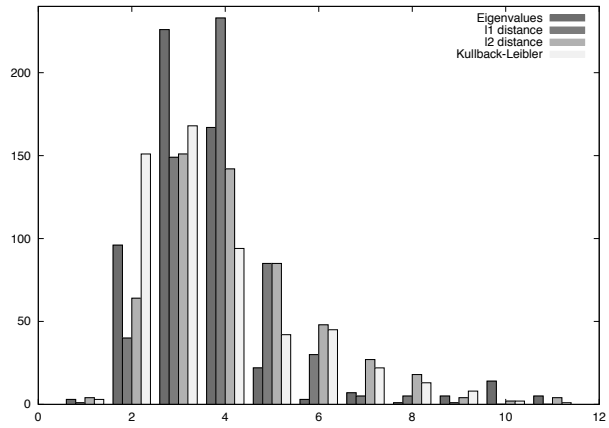

Figure 7. Estimated rank with the four methods $|S| = 20000$.

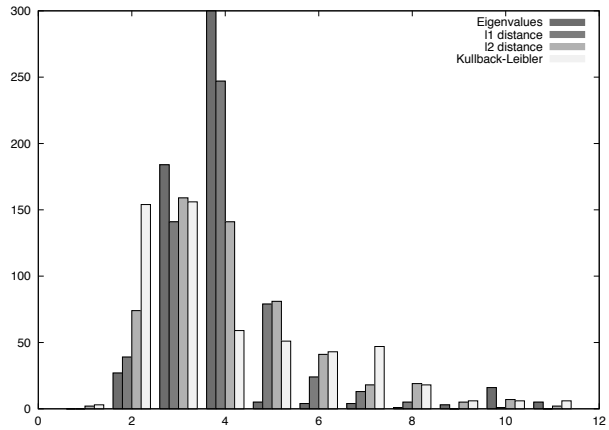

Figure 8. Estimated rank with the four methods $|S| = 100000$.

References

Beimel, A., Bergadano, F., Bshouty, N. H., Kushilevitz, E., & Varricchio, S. (2000). Learning functions represented as multiplicity automata. *Journal of the Association for Computing Machinery*, *47*, 506–530.

Carrasco, R. C., & Oncina, J. (1994). Learning stochastic regular grammars by means of a state merging method. *Second International Colloquium on Grammatical Inference* (pp. 139–152). Springer.

Clark, A., Florêncio, C. C., & Watkins, C. (2006). Languages as hyperplanes: Grammatical inference with string kernels. *17th European Conference on Machine Learning* (pp. 90–101). Springer.

Denis, F., & Esposito, Y. (2008). On rational stochastic languages. *Fundamenta Informaticae*, *86*, 41–77.

Denis, F., Esposito, Y., & Habrard, A. (2006). Learning rational stochastic languages. *19th Conference on Learning Theory* (pp. 274–288). Springer.

Shawe-Taylor, J., Cristianini, N., & Kandola, J. S. (2001). On the concentration of spectral properties. *Advances in Neural Information Processing Systems, 14* (pp. 511–517). MIT Press.

Smola, A. J., Gretton, A., Song, L., & Schölkopf, B. (2007). A Hilbert space embedding for distributions. *18th International Conference on Algorithmic Learning Theory* (pp. 13–31). Springer.

Thollard, F., Dupont, P., & de la Higuera, C. (2000). Probabilistic DFA Inference using Kullback-Leibler Divergence and Minimality. *17th International Conference on Machine Learning* (pp. 975–982). Morgan Kaufmann.