Computation of moments for probabilistic finite-state automata

Joan Andreu Sánchez, Verónica Romero

Pattern Recognition and Human Language Technologies Center
Universitat Politècnica de València
Camino de Vera s/n, 46022 , València, Spain

Abstract

The computation of moments of probabilistic finite-state automata (PFA) is researched in this article. First, the computation of moments of the length of the paths is introduced for general PFA, and then, the computation of moments of the number of times that a symbol appears in the strings generated by the PFA is described. These computations require a matrix inversion. Acyclic PFA, such as word graphs, are quite common in many practical applications. Algorithms for the efficient computation of the moments for acyclic PFA are also presented in this paper.

Keywords: moments, probabilistic finite-state automata

1. Introduction

Probabilistic graphs have been used in many applications of information science. Probabilistic graphs are related to formal grammars and automata. Thus, Probabilistic Context-Free Grammars (PCFG) are a core concept that has been used for probabilistic parsing in Natural Language Processing [1], RNA mod-

[*] Tel: +34 96 387 7253, Fax: +34 96 387 7239. e-mail: jandreu@prhlt.upv.es

elling [2], Mathematical Expression Recognition [3], and Machine Translation [4]. Probabilistic regular models such as N-gram models and Hidden Markov Models are used for Automatic Speech Recognition (ASR) [5] and Handwritten Text Recognition (HTR) [6]. Probabilistic Finite-State Automata (PFA) are regular models that are used in many applications such as ASR [7], Machine Translation (MT) [8], and the computation of Confidence Measure for different purposes [9, 10, 11]. PFA are very relevant for practical reasons since many algorithms that deal with these models have polynomial time complexity in many real applications.

The probabilistic properties of these syntactical models are very relevant since these properties reflect their representation capabilities. For example, entropy has been studied for PCFG [12, 13, 14], HMM [15], and regular models [16, 17, 18, 19], and conditions about the consistency of PCFG have been researched in [20, 21, 22, 23, 24].

Moments are an important concept related to statistics. The first moment is a relevant concept since it represents the average value of a stochastic variable. The second moment and higher order moments are of interest for knowing the aspect of a distribution and other convergence properties.

The computation of moments for PCFG has been researched in [25, 26], cross-moments computation has been researched for factor graphs in [27], and the computation of moments in other types of graphs has been researched in [28, 29]. Generating functions are used in [25, 26] as a core mathematical tool for computing moments that are related to the derivation and string lengths produced by PCFG. The algorithms described in [25, 26] are based on matrix computations that require a matrix inversion, and, therefore, the time complexity is at least the time complexity of this matrix inversion.

This paper studies the computation of moments for PFA [30]. The main contribution of this paper is that this computation is not based on generating functions but rather on matrix computations, which allow for a very intuitive interpretation. The formulation of the problem introduced in this paper allows us also to obtain efficient algorithms for acyclic PFA. We study moments of path length and also moments of the number of times that a symbol appears in the strings generated by a PFA[1]. Acyclic PFA are a relevant formalism since they are related to trellises that are used in ASR [10] and HTR [11]. From a practical point of view, the first and second moments of the path lengths are relevant for computing the variance of the string represented in a word graph that, in turn, represents a speech signal or a handwritten text line.

Section 2 introduces the notation used in this article. The computation of moments for general PFA is presented in Section 3. Section 4 introduces the computation of moments for acyclic PFA, and presents efficient algorithms for carrying out these computations.

2. Notation and definitions

We introduce the notation related to probabilistic finite-state automata (PFA) that will be used in this article. We mainly follow the notation of [30] and [31] and some notation from [23].

Definition 2.1. A *PFA* is a tuple $\mathcal{A} = \langle Q, \Sigma, \delta, I, F, P \rangle$, where: Q is a finite set of states; Σ is the alphabet; $\delta \subseteq Q \times \Sigma \times Q$ is a set of transitions; $I : Q \rightarrow \mathbb{R}^{\geq 0}$ is the probability function of a state being an initial state; $P : \delta \rightarrow \mathbb{R}^{\geq 0}$ is a probability

[1] In formal language theory, automata are considered to be string acceptors, but PFA may be considered to be generative processes (see Section 2.2 in [30]).

function of transition between states; $F : Q \to \mathbb{R}^{\geq 0}$ is the probability function of a state being a final state. I, P, and F are functions such that:

$$\sum_{i \in Q} I(i) = 1 , \tag{1}$$

$$\forall i \in Q, \;\; F(i) + \sum_{k \in \Sigma, j \in Q} P(i,k,j) = 1 . \tag{2}$$

For the sake of notation simplification, P is assumed to be extended with $P(i,k,j) = 0$ for all $(i,k,j) \notin \delta$. An automaton is said to be *proper* if it satisfies equation (2). Without loss of generality, throughout this paper, we assume that there are no useless states in the PFA [30].

In this article, we assume that all states are nominated by integers from 0 to $|Q|-1$. For simplicity without loss of generality, we assume that the PFA have only one initial state, named 0, and, therefore, the sum in the left part of (1) has only one term. We assume without loss of generality that the PFA have only one final state, named $|Q|-1$, which moreover does not have loops. We also assume that all symbols are nominated by integers from 0 to $|\Sigma|-1$. The difference between integers that refer to states and integers that refer to symbols will be clear from the context. These assumptions greatly simplify the notation. For practical reasons, we assume that the final state differs from the initial state and therefore the empty string is not in $L(\mathcal{A})$.

Given a PFA $\mathcal{A}$, a *path* s in $\mathcal{A}$ is a sequence of transitions:

$$0 = i_0 \xrightarrow[P(0,k_1,i_1)]{k_1} i_1 \cdots i_{l_s-1} \xrightarrow[P(i_{l_s-1},k_{l_s},i_{l_s})]{k_{l_s}} i_{l_s} = |Q|-1 .$$

such that $P(i_{j-1},k_j,i_j) \in \delta$, $0 < j \leq l_s$, and l_s is the length of the path s, that is, the number of transitions. Its probability is defined as $p_{\mathcal{A}}(s) = \prod_{j=1}^{l_s} P(i_{j-1},k_j,i_j)$. We denote as $\mathrm{c}_{k,s}$ the number of times that the terminal $k, k \in \Sigma$, appears in the path s.

In this work, a *cycle* is a path $i_0 \xrightarrow[P(i_0,k_1,i_1)]{k_1} i_1 \cdots i_{n-1} \xrightarrow[P(i_{n-1},k_n,i_n)]{k_n} i_n$ such that $i_r \neq i_s$, $0 \leq r, s < n$ and $i_0 = i_n$, and a *loop* is a cycle in which $n = 1$ [30].

Definition 2.2. An *acyclic PFA* is defined as a PFA that cannot have cycles.

If the graph is acyclic, then a topological order can be defined over the nodes[2]. If the graph is acyclic, then we assume that the nodes are numbered according to this topological order from 0 (the initial state) to $|Q| - 1$ (the final state).

Definition 2.3. Given a PFA $\mathcal{A}$, we define the *characteristic matrix* $E = (e_{i,j})$ [21] of dimensions $|Q| \times |Q|$ as[3]:

$$e_{i,j} = \sum_{k \in \Sigma} P(i, k, j) \ . \tag{3}$$

where $0 \leq i, j \leq |Q| - 1$.

Definition 2.4. The *terminal expectation matrix* [22] $Z = (z_{i,k})$, $0 \leq i \leq |Q| - 1$ and $0 \leq k \leq |\Sigma| - 1$, of the PFA $\mathcal{A}$ is defined as:

$$z_{i,k} = \sum_{j} P(i, k, j) \ . \tag{4}$$

3. Moments for general PFA

This section describes the computation of moments of path lengths for general PFA, i.e., PFA that can have cycles and loops. This problem has been researched in the past by using formal power series [25, 26]. We present similar results with a different formalism. The computations described in this section will be useful for a more efficient computation for acyclic PFA, which is described in the following section. This section also describes the computation of moments for the number of times that a symbol appears in the language generated by a PFA.

[2] Note that different topological orders can exist, but this is not relevant in this article. We consider just one of them.

[3] Throughout the article, we assume that the rows and columns of any matrix with dimensions $n \times n$ are indexed from 0 to $n - 1$.

3.1. The first moment of the path lengths

For PFA as described in Section 2, the infinite sum $\mathcal{Q} = \sum_{i=0}^{\infty} E^i$ converges to $(I - E)^{-1}$, where I is the identity matrix. The convergence of $\mathcal{Q}$ is guaranteed since we have assumed that the PFA does not have useless states [30].

Definition 3.1. The *expected path length* of all possible paths in $\mathcal{A}$ is defined as:

$$\bar{l}_{\mathcal{A}} \;=\; \sum_{s \in \mathcal{A}} p_{\mathcal{A}}(s)\, l_s \;. \tag{5}$$

Note that this expression is the first moment of the path lengths.

The *expected path length* of expression (5) $\bar{l}_{\mathcal{A}}$ is [22]:

$$\bar{l}_{\mathcal{A}} \;=\; \sum_{i=0}^{|Q|-2} (\mathcal{Q})_{0,i} \;. \tag{6}$$

The range of the right sum of expression (6) ends with $|Q| - 2$ because each $(\mathcal{Q})_{0,i}$ can also be understood to be the expected number of times that the state i is used in paths from the initial state 0 to the final state $|Q| - 1$. Note that each of these paths has a length equal to the number of states in the path minus 1, which is:

$$\bar{l}_{\mathcal{A}} = \sum_{i=0}^{|Q|-1} (\mathcal{Q})_{0,i} - 1 = \sum_{i=0}^{|Q|-2} (\mathcal{Q})_{0,i} + \underbrace{(\mathcal{Q})_{0,|Q|-1}}_{=1} - 1.$$

The time complexity of this computation is at least the time required to compute the inverse of a matrix.

In order to illustrate this computation, consider the PFA in Figure 1. This PFA includes cycles, loops, and transitions from one state to a different state using two different labels. The possible paths that this PFA can generate and the corresponding probability for each path can be computed from paths that are shown in Figure 2.

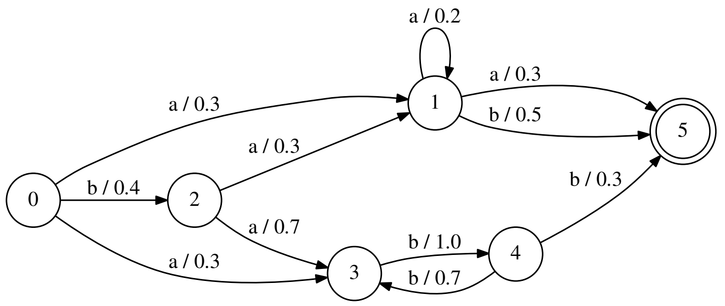

Figure 1: Example of a PFA.

$$0 \xrightarrow[0.3]{a} 1 \quad \underbrace{\xrightarrow[0.2]{a} 1}_{i \text{ times}, i \geq 0} \quad \xrightarrow[0.3]{a} 5$$

$$0 \xrightarrow[0.3]{a} 1 \quad \underbrace{\xrightarrow[0.2]{a} 1}_{i \text{ times}, i \geq 0} \quad \xrightarrow[0.5]{b} 5$$

$$0 \xrightarrow[0.4]{b} 2 \xrightarrow[0.3]{a} 1 \quad \underbrace{\xrightarrow[0.2]{a} 1}_{i \text{ times}, i \geq 0} \quad \xrightarrow[0.3]{a} 5$$

$$0 \xrightarrow[0.4]{b} 2 \xrightarrow[0.3]{a} 1 \quad \underbrace{\xrightarrow[0.2]{a} 1}_{i \text{ times}, i \geq 0} \quad \xrightarrow[0.5]{b} 5$$

$$0 \xrightarrow[0.4]{b} 2 \xrightarrow[0.7]{a} 3 \xrightarrow[1.0]{b} 4 \underbrace{\xrightarrow[0.7]{b} 3 \xrightarrow[1.0]{b} 4}_{i \text{ times}, i \geq 0} \xrightarrow[0.3]{b} 5$$

$$0 \xrightarrow[0.3]{a} 3 \xrightarrow[1.0]{b} 4 \underbrace{\xrightarrow[0.7]{b} 3 \xrightarrow[1.0]{b} 4}_{i \text{ times}, i \geq 0} \xrightarrow[0.3]{b} 5$$

Figure 2: Possible paths for the PFA in Figure 1.

Therefore, the expected path length is:

$$\begin{aligned} \bar{l}_{\mathcal{A}} &= 0.09 \sum_{i=0}^{\infty} 0.2^i (i+2) + 0.15 \sum_{i=0}^{\infty} 0.2^i (i+2) + 0.036 \sum_{i=0}^{\infty} 0.2^i (i+3) + \\ & \quad 0.06 \sum_{i=0}^{\infty} 0.2^i (i+3) + 0.084 \sum_{i=0}^{\infty} 0.7^i (2i+4) + 0.09 \sum_{i=0}^{\infty} 0.7^i (2i+3) \\ &= 0.253 + 0.422 + 0.146 + 0.244 + 2.427 + 2.3 = 5.792 \ . \end{aligned}$$

Note that, in this computation, we have used the following results for a real value

r such that $0 < r < 1.0$:

$$\sum_{i=0}^{\infty} r^i = \frac{1}{1-r} , \tag{7}$$

$$\sum_{i=0}^{\infty} i r^i = \frac{r}{(1-r)^2} . \tag{8}$$

The second equation follows from the first, by differentiating both sides in (7), and then both sides are multiplied by $-1/r$. It is easy to show that the infinite series ir^i is convergent as $i \to \infty$ for $0 < r < 1.0$. We include these well-known results because they will help us to explain further demonstrations.

Continuing with the example, we now compute expression (5) by using the matrix form introduced in Definition 2.3. Matrix E, which is associated to the PFA in Figure 1, is:

$$E = \begin{pmatrix} 0 & 0.3 & 0.4 & 0.3 & 0 & 0 \\ 0 & 0.2 & 0 & 0 & 0 & 0.8 \\ 0 & 0.3 & 0 & 0.7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.7 & 0 & 0.3 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

and the matrix $\mathcal{Q}$ for matrix E is:

$$\mathcal{Q} = \begin{pmatrix} 1.0 & 0.525 & 0.4 & 1.933 & 1.933 & 1.0 \\ 0 & 1.25 & 0 & 0 & 0 & 1.0 \\ 0 & 0.375 & 1.0 & 2.333 & 2.333 & 1.0 \\ 0 & 0 & 0 & 3.333 & 3.333 & 1.0 \\ 0 & 0 & 0 & 2.333 & 3.333 & 1.0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \end{pmatrix}$$

Therefore, the following result is obtained using expression (6):

$$\bar{l}_{\mathcal{A}} = \sum_{i=0}^{4} (\mathcal{Q})_{0,i} = 5.792 \; .$$

3.2. The second moment of the path lengths

The variance of the expected path length of the paths in a PFA $\mathcal{A}$ is defined as:

$$\sigma^2_{l_{\mathcal{A}}} = \sum_{s \in \mathcal{A}} p_{\mathcal{A}}(s) \; (l_s - \bar{l}_{\mathcal{A}})^2 = \sum_{s \in \mathcal{A}} p_{\mathcal{A}}(s) \; l_s^2 - \bar{l}^2_{\mathcal{A}} \; . \qquad (9)$$

The sum in expression (9) is the second moment of the path lengths. We now describe how to compute this second moment. Note that $(E^k)_{0,|Q|-1}$ represents the probability accumulated in all paths whose length is k that start in state 0 and reach the final state $|Q| - 1$. Consequently, we have the following result:

Lemma 3.1.

$$\sum_{s \in \mathcal{A} \,|\, l_s = k} p_{\mathcal{A}}(s) \; l_s^2 = k^2 \; (E^k)_{0,|Q|-1} \; . \qquad (10)$$

Proof. As we stated above, $(E^k)_{0,|Q|-1}$ is the addition of the probability of all paths that start in state 0 and reach the final state $|Q| - 1$ whose length is k. This is exactly the addition represented on the left side of (10). □

For instance, in the example in Figure 1, expression (10) for the paths with length 3 and their probabilities are the following:

$0 \xrightarrow[0.3]{a} 1 \xrightarrow[0.2]{a} 1 \xrightarrow[0.3]{a} 5$	0.018
$0 \xrightarrow[0.3]{a} 1 \xrightarrow[0.2]{a} 1 \xrightarrow[0.5]{b} 5$	0.03
$0 \xrightarrow[0.4]{b} 2 \xrightarrow[0.3]{a} 1 \xrightarrow[0.3]{a} 5$	0.036
$0 \xrightarrow[0.4]{b} 2 \xrightarrow[0.3]{a} 1 \xrightarrow[0.5]{b} 5$	0.06
$0 \xrightarrow[0.3]{a} 3 \xrightarrow[1.0]{b} 4 \xrightarrow[0.3]{b} 5$	0.09

and the left part of equation (10) is:

$$3^2 \sum_{s\in\mathcal{A}|l_s=3} p_{\mathcal{A}}(s) = 2.106$$

while the right part is

$$3^2 (E^3)_{0,5} = 2.106 \ .$$

We now have the following result.

Theorem 3.1.

$$\sum_{s\in\mathcal{A}} p_{\mathcal{A}}(s)\ l_s^2 = \left(\frac{E^2+E}{(I-E)^3}\right)_{0,|Q|-1} \ . \tag{11}$$

Proof. It holds that:

$$\begin{aligned}\sum_{s\in\mathcal{A}} p_{\mathcal{A}}(s)\ l_s^2 &= \sum_{k=1}^{\infty} \sum_{s\in\mathcal{A}|l_s=k} p_{\mathcal{A}}(s)\ l_s^2 \qquad (12)\\ &= \sum_{k=1}^{\infty} k^2\ (E^k)_{0,|Q|-1} \ .\end{aligned}$$

To solve the last sum, we apply the same ideas from expressions (7) and (8). First, note that the following equation is immediate:

$$E + E^2 + \ldots = (I-E)^{-1}E \ . \tag{13}$$

Each term in the sum in the left part of the equation is a matrix with dimensions $|Q| \times |Q|$, and the right part is also a matrix with the same dimensions. Let E be a square matrix of real values with dimensions $|Q| \times |Q|$; we define the following differentiating operation for this matrix:

$$\sum_{i=0}^{|Q|-1} \frac{\partial E}{\partial e_{i,i}} \ .$$

Appendix A shows the application of this differentiating operation to the matrices that are shown above. Then, with some operations, expression (13) becomes:

$$E + E^2 + \ldots = (I - E)^{-1} E$$

{apply differentiation}

$$I + 2E + 3E^2 + \ldots = (I - E)^{-2}$$

{multiply both sides by E}

$$E + 2E^2 + 3E^3 + \ldots = E(I - E)^{-2}$$

{apply differentiation}

$$I + 2^2 E + \ldots = (I - E)^{-2} + 2E(I - E)^{-3}$$

{multiply both sides by E}

$$E + 2^2 E^2 + \ldots = E(I - E)^{-2} + 2E^2(I - E)^{-3}$$

$$\sum_{k=1}^{\infty} k^2 \; E^k = \frac{E^2 + E}{(I - E)^3}$$

Finally,

$$\sum_{s \in \mathcal{A}} p_{\mathcal{A}}(s) \; l_s^2 = \left(\frac{E^2 + E}{(I - E)^3} \right)_{0,|Q|-1} .$$

□

The time complexity of the right part of this expression is at least the time complexity of computing the inverse of a matrix, i.e., it is at least cubic with the dimension of the matrix. In addition, note that several matrix products are involved. Also note that other moments of higher order can be computed following the above demonstration.

Continuing with the example, let us now compute the left part of expression (11). We will make use of the following result which is obtained from equation (8) by first differentiating both sides and then multiplying both sides by r:

$$\sum_{i=0}^{\infty} i^2 r^i \quad = \quad \frac{r(1+r)}{(1-r)^3} \; . \qquad (14)$$

It is also easy to show that the infinite series ir^i is convergent as $i \to \infty$ for $0 < r < 1.0$. Then we have:

$$\begin{aligned}
&\sum_{s \in \mathcal{A}} p_{\mathcal{A}}(s)\ l_s^2 \\
&= 0.09 \sum_{i=0}^{\infty} 0.2^i (i+2)^2 + 0.15 \sum_{i=0}^{\infty} 0.2^i (i+2)^2 + 0.036 \sum_{i=0}^{\infty} 0.2^i (i+3)^2 \\
&\quad + 0.06 \sum_{i=0}^{\infty} 0.2^i (i+3)^2 + 0.084 \sum_{i=0}^{\infty} 0.7^i (2i+4)^2 + 0.09 \sum_{i=0}^{\infty} 0.7^i (2i+3)^2 \\
&= 0.605 + 1.008 + 0.489 + 0.816 + \ 29.742 + 26.967 = 59.627\ .
\end{aligned}$$

In matrix form according to expression (11), $((E^2 + E)(I - E)^{-3})_{0,|Q|-1} = 59.627$.

Therefore, the variance of the expected path length of the paths in the PFA in Figure 1 according to expression (9) is:

$$\sigma^2_{l_{\mathcal{A}}} = 59.627 - 5.792^2 = 26.08\ .$$

3.3. The first moment of the number of occurrences of a symbol

In addition to the moments of the path lengths, other interesting moments can be computed. For example, the moments of the number of times that a symbol k appears in the language generated by a PFA $\mathcal{A}$ allow us to compute the variance of this expected value. This means that:

$$\sigma^2_{k_{\mathcal{A}}} = \sum_{s \in \mathcal{A}} p_{\mathcal{A}}(s) c^2_{k,s} - \bar{c}^2_{k_{\mathcal{A}}}\ . \tag{15}$$

where $c_{k,s}$ is the number of times that the symbol k appears in the path s, and $\bar{c}_{k_{\mathcal{A}}}$ is the expected value of the number of times that the symbol k appears in the language generated by $\mathcal{A}$.

This section describes the computation of the first moment of the number of occurrences of a symbol in the language generated by PFA $\mathcal{A}$, which is defined as:

$$\bar{c}_{k_{\mathcal{A}}} = \sum_{s \in \mathcal{A}} p_{\mathcal{A}}(s)\ c_{k,s}\ . \tag{16}$$

This problem is researched in [20] for probabilistic grammars, and the solution provided, which is adapted to the PFA defined in this paper, is:

$$\bar{c}_{k_{\mathcal{A}}} = \sum_{s \in \mathcal{A}} p_{\mathcal{A}}(s)\ c_{k,s} = (\mathcal{Q}Z)_{0,k}\ . \tag{17}$$

The time complexity of this solution is the same time complexity of computing the inverse of a matrix.

For the example in Figure 1 and following similar computations as in Figure 2, for the symbol "a":

$$\begin{aligned}
\bar{c}_{a_{\mathcal{A}}} &= \sum_{s \in \mathcal{A}} p_{\mathcal{A}}(s)\ c_{a,s} \\
&= 0.09 \sum_{i=0}^{\infty} 0.2^i (i+2) + 0.15 \sum_{i=0}^{\infty} 0.2^i (i+1) + 0.036 \sum_{i=0}^{\infty} 0.2^i (i+2) \\
&\quad + 0.06 \sum_{i=0}^{\infty} 0.2^i (i+1) + 0.084 \sum_{i=0}^{\infty} 0.7^i + 0.09 \sum_{i=0}^{\infty} 0.7^i \\
&= 0.253 + 0.234 + 0.101 + 0.094 + 0.28 + 0.3 = 1.262\ .
\end{aligned}$$

In matrix form according to expression (17) and assuming that the first column of Z is associated to symbol "a", $(\mathcal{Q}Z)_{0,0} = 1.262$. If we assume that the second column of Z is associated to symbol "b", then $(\mathcal{Q}Z)_{0,1} = 4.529$. As expected, the addition of both values is exactly $\bar{l}_{\mathcal{A}}$.

3.4. The second moment of the number of occurrences of a symbol

The definition of the second moment of the number of times that a symbol k appears in a path s is:

$$\sum_{s\in\mathcal{A}} p_{\mathcal{A}}(s)\ c_{k,s}^2\ . \tag{18}$$

and the computation in the matrix notation introduced in the above sections is the following.

From Definition 2.3, let E_k be the characteristic matrix for symbol k, which is

$$(E_k)_{i,j} = P(i,k,j)\ . \tag{19}$$

and let $E_{\tilde{k}}$ be the characteristic matrix for the other symbols that are different from k, which is

$$(E_{\tilde{k}})_{i,j} = \sum_{m\in\Sigma | m\neq k} P(i,m,j)\ . \tag{20}$$

Note that for any $k \in \Sigma$, it holds that $E = E_k + E_{\tilde{k}}$.

Theorem 3.2.

$$\sum_{s\in\mathcal{A}} p_{\mathcal{A}}(s)\ c_{k,s}^2 = \left((I - E_{\tilde{k}})^{-1}(R^2 + R)(I - R)^{-3}\right)_{0,|Q|-1}\ . \tag{21}$$

where $R = E_k(I - E_{\tilde{k}})^{-1}$

Proof. Let us add all of the paths from the left side of equation (21) in the PFA that have the symbol k only once:

$$\left((I+E_{\tilde{k}}+E_{\tilde{k}}^2+\ldots)E_k(I+E_{\tilde{k}}+E_{\tilde{k}}^2+\ldots)\right)_{0,|Q|-1} = \left((I - E_{\tilde{k}})^{-1}E_k(I - E_{\tilde{k}})^{-1}\right)_{0,|Q|-1}\ .$$

Note that the infinite sum $I + E_{\tilde{k}} + E_{\tilde{k}}^2 + \ldots$ is convergent [22]. Analogously, let us add all of the paths from the left side of equation (21) in the PFA that have the symbol k twice:

$$2^2\left((I - E_{\tilde{k}})^{-1}E_k(I - E_{\tilde{k}})^{-1}E_k(I - E_{\tilde{k}})^{-1}\right)_{0,|Q|-1}\ .$$

In general, the addition of all of the paths with i times the symbol k is:

$$i^2 \left((I - E_{\tilde{k}})^{-1}(E_k(I - E_{\tilde{k}})^{-1})^i\right)_{0,|Q|-1} \ .$$

Consequently, if $R = E_k(I - E_{\tilde{k}})^{-1}$, then:

$$\sum_{s \in \mathcal{A}} p_{\mathcal{A}}(s)\ c_{k,s}^2 = \left((I - E_{\tilde{k}})^{-1} \sum_{i=1}^{\infty} i^2 R^i\right)_{0,|Q|-1} = \left((I - E_{\tilde{k}})^{-1}(R^2 + R)(I - R)^{-3}\right)_{0,|Q|-1} \ .$$

□

Note that the infinite sum in the last expression is analogous to the infinite sum that appeared in Theorem 3.1.

The time complexity of this computation is again at least the time complexity of computing the inverse of a matrix, i.e., it is at least cubic with the dimension of the matrix. In addition, many matrix products are involved in the computation.

For the example in Figure 1, for the symbol "a" it holds that:

$$\begin{aligned}
&\sum_{s \in \mathcal{A}} p_{\mathcal{A}}(s)\ c_{a,s}^2 \\
&= \ 0.09 \sum_{i=0}^{\infty} 0.2^i (i+2)^2 + 0.15 \sum_{i=0}^{\infty} 0.2^i (i+1)^2 + 0.036 \sum_{i=0}^{\infty} 0.2^i (i+2)^2 \\
&\quad + 0.06 \sum_{i=0}^{\infty} 0.2^i (i+1)^2 + 0.084 \sum_{i=0}^{\infty} 0.7^i + 0.09 \sum_{i=0}^{\infty} 0.7^i \\
&= \ 0.605 + 0.352 + 0.242 + 0.14 + 0.28 + 0.3 = 1.919 \ .
\end{aligned}$$

The matrix obtained according to expression (21) for the PFA in Figure 1 is:

$$\begin{pmatrix} 0 & 0.984 & 0 & 1.933 & 1.933 & 1.919 \\ 0 & 0.469 & 0 & 0 & 0 & 0.938 \\ 0 & 0.703 & 0 & 2.333 & 2.333 & 1.656 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

With these computations, the variance of the expected number of times that the symbol "a" appears in the language generated by the PFA in Figure 1 that is computed using expression (15) is:

$$\sigma^2_{a_{\mathcal{A}}} = 1.919 - 1.262^2 = 0.326 \ .$$

4. Moments for acyclic PFA

In the case of acyclic PFA, the computations introduced in the above sections can be largely simplified and the time complexity can be decreased for all of them. This section describes new algorithms for computing moments for acyclic PFA. The subsections in this section are in parallel correspondence with the subsections in Section 3.

4.1. The first moment of the path lengths

If the PFA is acyclic, then each $(\mathcal{Q})_{0,i}$, $0 \leq i \leq |Q|-1$ can also be understood to be the posterior probability of the i-th state given all of the paths that start in the initial state 0 and reach the final state $|Q|-1$ and go through the i-th state. In this case, the first row of $\mathcal{Q}$ can be computed using this simple *forward* algorithm for an acyclic PFA $\mathcal{A}$:

$$\begin{aligned}
\alpha_{\mathcal{A}}(0) &= 1.0 \,, && (22)\\
\alpha_{\mathcal{A}}(i) &= \sum_{j=0}^{i-1}\sum_{k\in\Sigma}\alpha_{\mathcal{A}}(j)P(j,k,i) \quad 0<i\leq|Q|-1 \,. && (23)
\end{aligned}$$

In this algorithm, value $\alpha_{\mathcal{A}}(i)$ also represents the probability accumulated in all prefixes starting in the initial state 0 and reaching state i. This algorithm coincides with algorithm 3.1 in [30]. It is immediate to see that the probability accumulated in all suffixes that can be obtained from the subgraph that is induced from all states that are reachable from i adds up to 1.0. That is the reason we stated above that $\alpha_{\mathcal{A}}(i)$ can also be understood to be a posterior probability. Note that this algorithm can be implemented in linear time with the number of edges in the PFA, i.e., $O(|\delta|)$, which is a better time complexity than the time complexity of a matrix inversion. In real applications of ASR, HTR, or MT where trellises are used, it is usual that $|\delta| \ll |Q|^2$.

Lemma 4.1.

$$\alpha_{\mathcal{A}}(i) = \sum_{j=0}^{i}(E^j)_{0,i} \;\; \forall i \text{ such that } 0 \leq i \,.$$

Intuition. Each new term in the addition in the right part of this equation includes the accumulated probability of all of the paths from the initial state 0 to the i^{th} state that requires j transitions. The elements in the upper row of matrix $\mathcal{Q}$ are being fixed from left to right as a new power of matrix E is computed. The computation of this row coincides with the iterative computation of $\alpha_{\mathcal{A}}(\cdot)$.

Proof. The demonstration is by induction on i. Note that E^j is null for $j \geq |Q|$ since E is an upper triangular matrix. For $i = 0$, it holds that

$$\alpha(0) = (E^0)_{0,0} = I_{0,0} = 1.0 \,.$$

Note that in $(E^k)_{i,j}$, $1 \leq k \leq |Q|-1$, $0 \leq i \leq |Q|-k-1$, $k+i \leq j \leq |Q|-1$ may be non-null values. Then, for $0 < i \leq |Q|-1$:

$$
\begin{aligned}
&\sum_{j=0}^{i}(E^j)_{0,i} \\
&= \underbrace{I_{0,i}}_{=0} + \sum_{j=1}^{i}(E^{j-1}E)_{0,i} \\
&= (E)_{0,i} + \\
&\quad \underbrace{(E)_{0,0}}_{=0}(E)_{0,i} + (E)_{0,1}(E)_{1,i} + (E)_{0,2}(E)_{2,i} + \ldots + (E)_{0,i}\underbrace{(E)_{i,i}}_{=0} + \\
&\quad \underbrace{(E^2)_{0,0}}_{=0}(E)_{0,i} + \underbrace{(E^2)_{0,1}}_{=0}(E)_{1,i} + (E^2)_{0,2}(E)_{2,i} + \ldots + (E^2)_{0,i}\underbrace{(E)_{i,i}}_{=0} + \\
&\quad \ldots \\
&\quad \underbrace{(E^{i-1})_{0,0}}_{=0}(E)_{0,i} + \underbrace{(E^{i-1})_{0,1}}_{=0}(E)_{1,i} + \ldots + (E^{i-1})_{0,i-1}(E)_{i-1,i} + (E^{i-1})_{0,i}\underbrace{(E)_{i,i}}_{=0} + \\
&= \alpha_{\mathcal{A}}(0)\,(E)_{0,i} + \alpha_{\mathcal{A}}(1)\,(E)_{1,i} + \ldots + \underbrace{\sum_{l=0}^{i-1}(E^l)_{0,i-1}}_{=\alpha_{\mathcal{A}}(i-1)}(E)_{i-1,i} \\
&= \sum_{j=0}^{i-1}\sum_{k\in\Sigma}\alpha_{\mathcal{A}}(j)P(j,k,i) = \alpha_{\mathcal{A}}(i)\ .
\end{aligned}
$$

□

For example, consider the PFA in Figure 3. Since this PFA does not include cycles or loops, it is an acyclic PFA.

The algorithm in expressions (22) and (23) produces the following result for the PFA in Figure 3:

0	1	2	3	4
1.0	0.4	0.42	0.916	1.0

In matrix form, these computations are as follows. Matrix E associated to the

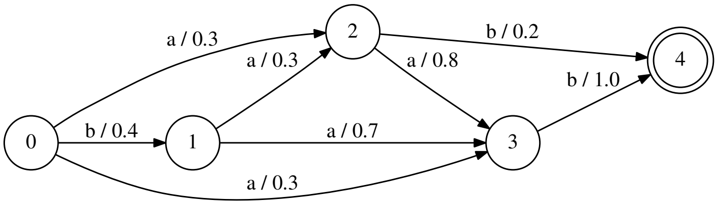

Figure 3: Example of an acyclic PFA.

PFA in Figure 3 is:

$$E = \begin{pmatrix} 0 & 0.4 & 0.3 & 0.3 & 0 \\ 0 & 0 & 0.3 & 0.7 & 0 \\ 0 & 0 & 0 & 0.8 & 0.2 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

and the matrix $\mathcal{Q}$ for this matrix E is:

$$\mathcal{Q} = \begin{pmatrix} 1.0 & 0.4 & 0.42 & 0.916 & 1.0 \\ 0 & 1.0 & 0.30 & 0.94 & 1.0 \\ 0 & 0 & 1.0 & 0.8 & 1.0 \\ 0 & 0 & 0 & 1.0 & 1.0 \\ 0 & 0 & 0 & 0 & 1.0 \end{pmatrix}$$

Note that the first row of this matrix is equal to the computation performed with the *forward* algorithm in (22) and (23). The first moment of the path lengths for this example is $\bar{l}_{\mathcal{A}} = 2.736$.

4.2. The second moment of the path lengths

This section describes how to compute the left part of expression (11) for an acyclic PFA in an efficient way that does not require a matrix inversion. We take advantage of an algorithm that is described in [31], however, it is used for a different purpose in this article. We include that algorithm here for completeness.

We use the right part of equation (12) for this purpose. Note that the inner sum of this expression is grouped for paths with the same length. Based on this idea, we define $\widehat{\alpha}_{\mathcal{A}}(i,l)$, $0 \leq i \leq |Q|-1$ and $1 \leq l \leq |Q|$ as the probability accumulated in all paths starting in the initial state 0, each of which has l different states and reaches state i:

$$\widehat{\alpha}_{\mathcal{A}}(i,l) = \sum_{\substack{s=(s_0=0,k_0,s_1\ldots,s_{l-1}=i)\ | \\ P(s_j,k_j,s_{j+1})\in\delta, 0\leq j<l-1}} p_{\mathcal{A}}(s)\ .$$

Thus, $\widehat{\alpha}_{\mathcal{A}}(|Q|-1,l)$ represents the probability accumulated in all paths starting in the initial state 0 and reaching the final state $|Q|-1$ whose length is $l-1$. Consequently, expression (12) becomes:

$$\sum_{s\in\mathcal{A}} p_{\mathcal{A}}(s)\ l_s^2 \ = \ \sum_{k=1}^{\infty} \sum_{s\in\mathcal{A}|l_s=k} p_{\mathcal{A}}(s)\ l_s^2 = \sum_{l=1}^{|Q|} \widehat{\alpha}_{\mathcal{A}}(|Q|-1,l)\ (l-1)^2\ . \qquad (24)$$

The computation of $\widehat{\alpha}_{\mathcal{A}}(\cdot,\cdot)$ can be performed with this new *forward* algorithm:

$$\widehat{\alpha}_{\mathcal{A}}(0,1) = 1 \qquad (25)$$

$$\widehat{\alpha}_{\mathcal{A}}(i,l) = \sum_{\substack{0\leq j<i \\ k\in\Sigma}} \widehat{\alpha}_{\mathcal{A}}(j,l-1)P(j,k,i) \quad 1 < l \leq |Q|, l-1 \leq i \leq |Q|-1\ . \qquad (26)$$

This algorithm can also be implemented in linear time with the number of edges in the PFA and the number of states, i.e., $O(|Q|\ |\delta|)$. In real applications where $|\delta| \ll |Q|^2$, this algorithm is more efficient than computing expression (11), which requires a matrix inversion and several matrix products.

If this algorithm is applied to the PFA in Figure 3, the following result is obtained:

	1	2	3	4	5
0	1.0				
1		0.4			
2		0.3	0.12		
3		0.3	0.52	0.096	
4			0.36	0.544	0.096

Note that the addition of all values in each row in matrix $\widehat{\alpha}_{\mathcal{A}}(\cdot,\cdot)$ is equal to each value in $\alpha_{\mathcal{A}}(\cdot)$, that is,

$$\sum_{j=1}^{|Q|} \widehat{\alpha}_{\mathcal{A}}(i,j) = \alpha_{\mathcal{A}}(i) \quad 0 \leq i \leq |Q| - 1\ .$$

This is because values in $\alpha_{\mathcal{A}}(\cdot)$ are distributed according to path lengths in $\widehat{\alpha}_{\mathcal{A}}(\cdot,\cdot)$.

Now, using equation (24) we obtain:

$$\sum_{s\in\mathcal{A}} p_{\mathcal{A}}(s)\ l_s^2 = 0.36\ \ 2^2 + 0.544\ \ 3^2 + 0.096\ \ 4^2 = 7.872\ .$$

In matrix form according to (11), we obtain $((E^2 + E)(I - E)^{-3})_{0,|Q|-1} = 7.872$.

Therefore, the variance is obtained with expression (9):

$$\sigma^2_{l_{\mathcal{A}}} = 7.872 - 2.736^2 = 0.386\ .$$

4.3. The first moment of the number of occurrences of a symbol

The computation of the first moment of the number of occurrences of a symbol for an acyclic PFA is immediate from expression (17) and taking into account Lemma 4.1. Only the first row of expression (17) is needed, and this row is computed with expressions (22) and (23). Consequently, the computation that is needed is:

$$\bar{c}_k = \sum_{i=0}^{|Q|-1} \alpha_{\mathcal{A}}(i) z_{i,k} \ . \tag{27}$$

The time complexity for the computation of this expression is clearly better than the time complexity for the computation of (17).

For the example in Figure 3, $\bar{c}_0 = 1.336$ if we assume that the first column of Z is associated to symbol "a", and $\bar{c}_1 = 1.4$ if we assume that the second column of Z is associated to symbol "b". As expected, $\bar{c}_0 + \bar{c}_1 = \bar{l}_{\mathcal{A}}$.

4.4. The second moment of the number of occurrences of a symbol

The computation of the second moment of the number of occurrences of a symbol can be carried out using an algorithm that is similar to the algorithm introduced in Section 4.2. The main idea is similar to the definition of $\widehat{\alpha}_{\mathcal{A}}(\cdot,\cdot)$ in Section 4.2. The definition is as follows:

$$\widehat{\alpha}'_{\mathcal{A}}(i,r) = \sum_{\substack{s=(s_0=0,k_o,s_1\ldots,s_{l_s}=i) \ | \\ P(s_j,k_j,s_{j+1})\in\delta,0\leq j<l_s \ \wedge \\ |\{P(s_j,k_j,s_{j+1})\in\delta,0\leq j<l_s \ \wedge k_j=m\}|=r}} p_{\mathcal{A}}(s) \ .$$

where $\widehat{\alpha}'_{\mathcal{A}}(i,r)$ represents the probability of all paths that reach state i that includes r times the m^{th} symbol. Note that r is finite since the PFA is acyclic. The computation of $\widehat{\alpha}'_{\mathcal{A}}(\cdot,\cdot)$ is as follows:

$$\begin{aligned}
\widehat{\alpha}'_{\mathcal{A}}(0,0) &= 1 && \text{for completeness,} && (28)\\
\widehat{\alpha}'_{\mathcal{A}}(i,0) &= \sum_{\substack{0\leq j<i\\ k\in\Sigma:k\neq m}} \widehat{\alpha}'_{\mathcal{A}}(j,0)P(j,k,i) && 1\leq i\leq |Q|-1\ . && (29)\\
\widehat{\alpha}'_{\mathcal{A}}(i,r) &= \sum_{0\leq j<i} \widehat{\alpha}'_{\mathcal{A}}(j,r-1)P(j,m,i)+\sum_{\substack{0\leq j<i\\ k\in\Sigma:k\neq m}}\widehat{\alpha}'_{\mathcal{A}}(j,r)P(j,k,i) && && (30)\\
& && 1\leq i\leq |Q|-1, 1\leq r<i\ . &&
\end{aligned}$$

This algorithm can be implemented with time complexity $O(|Q|\ |\delta|)$. Note that this computation is performed independently for every symbol in the alphabet.

Finally, expression (18) is computed as:

$$\sum_{1\leq r\leq |Q|-1} \widehat{\alpha}'_{\mathcal{A}}(|Q|-1,r)\ r^2\ . \qquad (31)$$

For example, the set of paths for the graph in Figure 3 are the following:

$$\begin{array}{rl}
0 \xrightarrow[0.4]{b} 1 \xrightarrow[0.3]{a} 2 \xrightarrow[0.8]{a} 3 \xrightarrow[1.0]{b} 4 & 0.096\\
0 \xrightarrow[0.4]{b} 1 \xrightarrow[0.3]{a} 2 \xrightarrow[0.2]{b} 4 & 0.024\\
0 \xrightarrow[0.4]{b} 1 \xrightarrow[0.7]{a} 3 \xrightarrow[1.0]{b} 4 & 0.28\\
0 \xrightarrow[0.3]{a} 2 \xrightarrow[0.8]{a} 3 \xrightarrow[1.0]{b} 4 & 0.24\\
0 \xrightarrow[0.3]{a} 2 \xrightarrow[0.2]{b} 4 & 0.06\\
0 \xrightarrow[0.3]{a} 3 \xrightarrow[1.0]{b} 4 & 0.3
\end{array}$$

and expression (18) for symbol “a” is:

$$\begin{aligned}
&0.096\ 2^2+0.024+0.28+0.24\ 2^2+0.06+0.3\\
&=2.008\ .
\end{aligned}$$

Using the algorithm in expressions (28), (29), and (30) for symbol “a” produces the following results:

	0	1	2	3	4
0	1.0				
1	0.4	0.0			
2	0.0	0.42	0.0		
3	0.0	0.58	0.336	0.0	
4	0.0	0.664	0.336	0.0	0.0

Finally, expression (18) can be computed with the final row of this using (31) as follows:

$$0.664\ 1^2 + 0.336\ 2^2 = 2.008\ .$$

In matrix form using expression (21):

$$\left((I - E_{\tilde{k}})^{-1}(R^2 + R)(I - R)^{-3}\right)_{0,|Q|-1} = 2.008\ .$$

5. Conclusions

In this paper, we have studied the computation of moments for probabilistic finite-state automata, specifically, the computation of moments of path lengths for general PFA. We also studied the computation of moments of the number of times that a symbol appears in the language generated by a PFA. The algorithms introduced can be easily generalized for computing moments of higher order and for other kinds of models like Probabilistic Context-Free Grammars. The generalization for computing moments in matrix form of higher order would require lemmas that are similar to the lemmas in Appendix A. These moments would provide additional information about the aspect of the distributions. We also studied computations for acyclic PFA, which are quite common in ASR, MT, and HTR. The algorithms introduced for acyclic PFA have lower time complexity than the algorithms for general PFA.

Acknowledgments

This work has been partially supported by the Ministerio de Ciencia y Tecnología under the grant TIN2017-91452-EXP (IBEM), by the Generalitat Valenciana under the grant PROMETEO/2019/121 (DeepPattern), and by the grant "Ayudas Fundación BBVA a equipos de investigación científica 2018" (PR[18]_HUM_C2_0087).

References

[1] S. Petrov, D. Klein, Improved inference for unlexicalized parsing, in: Proc. NAACL-HLT, 2007, pp. 404–411.

[2] Y. Sakakibara, M. Brown, R. Hughey, I. Mian, K. Sjölander, R. Underwood, D. Haussler, Stochastic context-free grammers for tRNA modeling, Nucleic Acids Research 22 (23) (1994) 5112–5120.

[3] F. Álvaro, J. Sánchez, J. Benedí, An integrated grammar-based approach for mathematical expression recognition, Pattern Recognition 51 (2016) 135–147.

[4] D. Wu, Stochastic inversion transduction grammars and bilingual parsing of parallel corpora, Computational Linguistics 23 (3) (1997) 377–404.

[5] F. Jelinek, Statistical Methods for Speech Recognition, MIT Press, 1998.

[6] A. Toselli, E. Vidal, F. Casacuberta (Eds.), Multimodal Interactive Pattern Recognition and Applications, 1st Edition, Springer, 2011.

[7] M. Mohri, F. Pereira, M. Riley, Weighted finite-state transducers in speech recognition, Computer Speech & Language 16 (1) (2002) 69–88.

[8] F. Casacuberta, E. Vidal, Machine translation with inferred stochastic finite-state transducers, Computational Linguistics 30 (2) (2004) 205–225.

[9] S. Ortmanns, H. Ney, A word graph algorithm for large vocabulary continuous speech recognition, Computer Speech and Language 11 (1997) 43–72.

[10] N. Ueffing, F. Och, H. Ney, Generation of word graphs in statitistical machine translation, in: Proceedings on Empirical Method for Natural Language Processing, 2002, pp. 156–163.

[11] J. Puigcerver, A. Toselli, E. Vidal, Word-graph and character-lattice combination for KWS in handwritten documents, in: International Conference on Frontiers in Handwriting Recognition (ICFHR), 2014, pp. 181–186.

[12] U. Grenander, Syntax controlled probabilities, Tech. rep., Brown University, Div. of Applied Mathematics (December 1967).

[13] S. Soule, Entropies of probabilistic grammars, Information and Control 25 (1974) 57–74.

[14] J. Justesen, K. Larsen, On the probabilistic context-free grammars that achieve capacity, Information and Controls 29 (1975) 268–285.

[15] D. Hernando, V. Crespi, G. Cybenko, Efficient computation of the hidden Markov model entropy for a given observation sequence, IEEE Transactions of Information Theory 51 (7) (2005) 2681–2685.

[16] G. Mann, A. McCallum, Efficient computation of entropy gradient for semi-supervised conditional random fields, in: Proceedings of HLT-NAACL, Companion Volume, Short Papers, Association for Computational Linguistics, Stroudsburg, PA, USA, 2007, pp. 109–112.

[17] M. Nederhof, G. Satta, Computation of distances for regular and context-free probabilistic languages, Theoretical Computer Science 395 (2008) 235–254.

[18] C. Cortes, M. Mohri, A. Rastogi, M. Riley, On the computation of the relative entropy of probabilistic automata, Int. J. Found. Comput. Sci. 19 (1) (2008) 219–242.

[19] V. Ilic, M. Stankovic, B. Todorivic, Entropy message passing, IEEE Trans. on Information Theory 57 (1) (2011) 91–99.

[20] T. Booth, R. Thompson, Applying probability measures to abstract languages, IEEE Transactions on Computers C-22 (5) (1973) 442–450.

[21] R. Thompson, Determination of probabilistic grammars for funtionally specified probability-measure languages, IEEE Transactions of Computers c-23 (6) (1974) 603–614.

[22] C. Wetherell, Probabilistic languages: A review and some open questions, Computing Surveys 12 (4) (1980) 361–379.

[23] J. Sánchez, J. Benedí, Consistency of stochastic context-free grammmars from probabilistic estimation based on growth transformation, IEEE Trans. Pattern Analysis and Machine Intelligence 19 (9) (1997) 1052–1055.

[24] Z. Chi, Statistical properties of probabilistic context-free grammar, Computational Linguistics 25 (1) (1999) 131–160.

[25] S. Hutchins, Moments of string and derivation lengths of stochastic context-free grammars, Information Sciences 4 (1972) 179–191.

[26] V. Ilic, M. Ciric, M. Stankovic, Cross-moments computation for stochastic context-free grammars, Tech. rep., https://arxiv.org/abs/1108.0353v2 (2013).

[27] V. Ilic, M. Stankovic, B. Todorovic, Computation of cross-moments, Advances in Mathematics of Communications 6 (3) (2012) 363–384.

[28] A. Heim, V. Sidorenko, U. Sorger, Computation of distributions and their moments in the trellis, Adv. Math. Commun. 2 (4) (2008) 373–391.

[29] Z. Li, J. Eisner, First- and second-order expectation semirings with applications to minimum-risk training on translation forests, in: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1, 2009, pp. 40–51.

[30] E. Vidal, F. Thollard, C. de la Higuera, F. Casacuberta, R. Carrasco, Probabilistic finite-state machines - Part I, IEEE Transactions on Pattern Analysis Machine Intelligence 27 (7) (2005) 1013–1039.

[31] J. Sánchez, M. Rocha, V. Romero, M. Villegas, On the derivational entropy of left-to-right probabilistic finite-state automata and hidden Markov models, Computational Linguistics 44 (1) (2017) 17–37.

Appendix A

Lemma 5.1. Consider the following square matrix E of real values with dimension $n \times n$:

$$E = \begin{pmatrix} e_{0,0} & e_{0,1} & \dots \\ e_{1,0} & \ddots & \\ \vdots & & e_{n-1,n-1} \end{pmatrix}$$

Then

$$\sum_{i=0}^{|Q|-1} \frac{\partial E^n}{\partial e_{i,i}} = nE^{n-1} \ .$$

Proof. First note that:

$$\sum_{i=0}^{|Q|-1} \frac{\partial E}{\partial e_{i,i}} = \begin{pmatrix} 1 & 0 & \dots \\ 0 & \ddots & \\ \vdots & & 0 \end{pmatrix} + \dots + \begin{pmatrix} 0 & 0 & \dots \\ 0 & \ddots & \\ \vdots & & 1 \end{pmatrix} = I \ .$$

Now:

$$\begin{aligned}
\sum_{i=0}^{|Q|-1} \frac{\partial E^n}{\partial e_{i,i}} &= \sum_{i=0}^{|Q|-1} \frac{\partial (E^{n-1}E)}{\partial e_{i,i}} = \sum_{i=0}^{|Q|-1} E\frac{\partial E^{n-1}}{\partial e_{i,i}} + E^{n-1}\frac{\partial E}{\partial e_{i,i}} \\
&= \sum_{i=0}^{|Q|-1} E\frac{\partial (E^{n-2}E)}{\partial e_{i,i}} + E^{n-1} = \sum_{i=0}^{|Q|-1} E(E\frac{\partial E^{n-2}}{\partial e_{i,i}} + E^{n-2}) + E^{n-1} \\
&= \sum_{i=0}^{|Q|-1} E^2\frac{\partial E^{n-2}}{\partial e_{i,i}} + 2E^{n-1} = \dots = \sum_{i=0}^{|Q|-1} E^{n-1} + (n-1)E^{n-1} \\
&= \sum_{i=0}^{|Q|-1} nE^{n-1} \ .
\end{aligned}$$

□

Lemma 5.2. Consider the following square matrix E of real values with dimension $n \times n$:

$$E = \begin{pmatrix} e_{0,0} & e_{0,1} & \dots \\ e_{1,0} & \ddots & \\ \vdots & & e_{n-1,n-1} \end{pmatrix}$$

Then

$$\sum_{i=0}^{|Q|-1} \frac{\partial (I-E)^{-1}E}{\partial e_{i,i}} = (I-E)^{-2} \ .$$

Proof.

$$\begin{aligned}
\sum_{i=0}^{|Q|-1} \frac{\partial (I-E)^{-1}E}{\partial e_{i,i}} &= \frac{(I-E)\begin{pmatrix}1 & 0 & \dots \\ 0 & \ddots & \\ \vdots & & 0\end{pmatrix} - E\begin{pmatrix}-1 & 0 & \dots \\ 0 & \ddots & \\ \vdots & & 0\end{pmatrix}}{(I-E)^2} \\
&\quad + \dots + \frac{(I-E)\begin{pmatrix}0 & 0 & \dots \\ 0 & \ddots & \\ \vdots & & 1\end{pmatrix} - E\begin{pmatrix}0 & 0 & \dots \\ 0 & \ddots & \\ \vdots & & -1\end{pmatrix}}{(I-E)^2} \\
&= \frac{(I-E)+E}{(I-E)^2} = (I-E)^{-2} \ .
\end{aligned}$$

□