Learning Partially Observable Markov Models from First Passage Times

Jérôme Callut[1,2] and Pierre Dupont[1,2]

[1] Department of Computing Science and Engineering, INGI
Université catholique de Louvain,
Place Sainte-Barbe 2,
B-1348 Louvain-la-Neuve, Belgium
{Jerome.Callut,Pierre.Dupont}@uclouvain.be
[2] UCL Machine Learning Group
http://www.ucl.ac.be/mlg/

Abstract. We propose a novel approach to learn the structure of Partially Observable Markov Models (POMMs) and to estimate jointly their parameters. POMMs are graphical models equivalent to Hidden Markov Models (HMMs). The model structure is built to support the First Passage Times (FPT) dynamics observed in the training sample. We argue that the FPT in POMMs are closely related to the model structure. Starting from a standard Markov chain, states are iteratively added to the model. A novel algorithm POMMPHit is proposed to estimate the POMM transition probabilities to fit the sample FPT dynamics. The transitions with the lowest expected passage times are trimmed off from the model. Practical evaluations on artificially generated data and on DNA sequence modeling show the benefits over Bayesian model induction or EM estimation of ergodic models with transition trimming.

1 Introduction

This paper is concerned with the induction of Hidden Markov Models (HMMs). These models are widely used in many pattern recognition areas, including speech recognition [9], biological sequence modeling [2], and information extraction [3], to name a few. The estimation of such models is twofolds: (i) the model structure, *i.e.* the number of states and the presence of transitions between these states, has to be defined and (ii) the probabilistic parameters of the model have to be estimated. The structural design is a discrete optimization problem while the parameter estimation is continuous by nature. In most cases, the model structure, also referred to as topology, is defined according to some prior knowledge of the application domain. However, automated techniques for designing the HMM topology are interesting as the structures are sometimes hard to define *a priori* or need to be tuned after some task adaptation. The work described here presents a new approach towards this objective.

Classical approaches to structural induction includes the Bayesian merging technique due to Stolcke [10] and the maximum likelihood state-splitting method of Ostendorf and Singer [8]. The former approach however has not been shown

J.N. Kok et al. (Eds.): ECML 2007, LNAI 4701, pp. 91–103, 2007.

to clearly outperform alternative approaches while the latter is specific to the subclass of left-to-right HMMs modeling speech signals. A more recent work [6] proposes a maximum *a priori* (MAP) technique using entropic model priors. This technique mainly focus on learning the correct number of states of the model but not its underlying transition graph. Another approach [11] attempts to design the model structure in order to fit the length distribution of the sequences. This problem can be considered as a particular case of the problem considered here since length distributions are the First Passage Times (FPT) between start and end sequence markers. Furthermore, in [11], sequence lengths are modeled with a mixture of negative binomial distributions which form a particular subclass of the general phase-type (PH) distributions considered here.

This paper presents a novel approach to the structural induction of Partially Observable Markov Models (POMMs). These models are equivalent to HMMs in the sense that they can generate the same class of distributions [1]. The model structure is built to support the First Passage Times (FPT) dynamics observed in the training sample. The FPT relative to a pair of symbols $(\mathtt{a}, \mathtt{b})$ is the number of steps taken to observe the next occurrence of `b` after having observed `a`. The distribution of the FPT in POMMs are shown to be of phase type (PH). POMMStruct aims at fitting these PH distributions from the FPT observed in the sample. We motivate the use of the FPT in POMMStruct by showing that they are informative about the model structure to be learned. Starting from a standard Markov chain (MC), POMMStruct iteratively adds states to the model. The probabilistic parameters are estimated using a novel method based on the EM algorithm, called POMMPHit. The latter computes the POMM parameters that maximize the likelihood of the observed FPT. POMMPHit differs from the standard Baum-Welch procedure since the likelihood function to be maximized is concerned with times between events (*i.e.* emission of symbols) rather than with the complete generative process. Additionally, a procedure based on the FPT is proposed to trim unnecessary transitions in the model. In contrast with a previous work [1], POMMStruct does not only focus on the mean of the FPT but on the complete distribution of these dynamical features. Consequently, a new parameter estimation technique is proposed here. In addition, a transition trimming procedure as well as a feature selection method to select the most relevant pairs $(\mathtt{a}, \mathtt{b})$ are also proposed.

Section 2 reviews the FPT in sequences, POMMs, PH distributions and the Jensen-Shannon divergence used for feature selection. Section 3 focus on the FPT dynamics in POMMs. Section 4 presents the induction algorithm POMMStruct. Finally, section 5 shows experimental results obtained with the proposed technique applied on artificial data and DNA sequences.

2 Background

The induction algorithm POMMStruct presented in section 4 relies on the First Passage Times (FPT) between symbols in sequences. These features are reviewed in section 2.1. Section 2.2 presents Partially Observable Markov Models

(POMMs) which are the models considered in POMMStruct. The use of POMMs is convenient in this work as the definition of the FPT distributions in these models readily matches the standard parametrization of phase-type (PH) distributions (see section 3). Discrete PH distributions are reviewed in section 2.3. Finally, the Jensen-Shannon (JS) divergence used to select the most relevant pairs of symbols is reviewed in subsection 2.4.

2.1 First Passage Times in Sequences

Definition 1. *Given a sequence s defined on an alphabet Σ and two symbols $\mathtt{a}, \mathtt{b} \in \Sigma$. For each occurrence of $\mathtt{a}$ in s, the first passage time to $\mathtt{b}$ is the finite number of steps taken before observing the next occurrence of $\mathtt{b}$.* $\mathrm{FPT}_s(\mathtt{a}, \mathtt{b})$ *denotes the first passage times to $\mathtt{b}$ for all occurrences of $\mathtt{a}$ in s. It is represented by a set of pairs $\{(z_1, w_1), \ldots, (z_l, w_l)\}$ where z_i denotes a passage time and w_i is the frequency of z_i in s.*

For instance, let us consider the sequence $s = aababba$ defined over the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. The FPT from $\mathtt{a}$ to $\mathtt{b}$ in s are $\mathrm{FPT}_s(\mathtt{a}, \mathtt{b}) = \{(2, 1), (1, 2)\}$. The *empirical FPT distribution* relative to a pair $(\mathtt{a}, \mathtt{b})$ is obtained by computing the relative frequency of each distinct passage time from $\mathtt{a}$ to $\mathtt{b}$. In contrast with N-gram features (*i.e.* contiguous substring of length N), the FPT does not only focus on the local dynamics in sequences as there is no *a priori* fixed maximum time (*i.e.* number of steps) between two events. For this reason, such features are well-suited to model long-term dependencies [1]. In section 3, we motivate the use of the FPT in the induction algorithm by showing that they are informative about the model topology to be learned.

2.2 Partially Observable Markov Models (POMMs)

Definition 2 (POMM). *A* Partially Observable Markov Model (POMM) *is a HMM $H = \langle \Sigma, Q, A, B, \iota \rangle$ where Σ is an alphabet, Q is a set of states, $A : Q \times Q \to [0, 1]$ is a mapping defining the probability of each transition, $B : Q \times \Sigma \to [0, 1]$ is a mapping defining the emission probability of each symbol on each state, and $\iota : Q \to [0, 1]$ is a mapping defining the initial probability of each state. Moreover, the emission probabilities satisfy: $\forall q \in Q, \exists a \in \Sigma$* such that $B(q, a) = 1$.

In other words, each state of a POMM only emits a single symbol. This model is called *partially* observable since, in general, several distinct states can emit the same symbol. As for a HMM, the observation of a sequence emitted by a POMM does not identify uniquely the states from which each symbol was emitted. However, the observations define *state subsets* or *blocks* from which each symbol may have been emitted. Consequently one can define a partition $\kappa = \{\kappa_{\mathtt{a}}, \kappa_{\mathtt{b}}, \ldots, \kappa_{\mathtt{z}}\}$ of the state set Q such that $\kappa_{\mathtt{a}} = \{q \in Q \,|\, B(q, \mathtt{a}) = 1\}$. Each block of the partition κ gathers the states emitting the same symbol. Whenever each block contains only a single state, the POMM is fully observable and equivalent to an order 1 MC. A POMM is depicted in the left part of Figure 1. The state label $\mathtt{1a}$ indicates that it

is the first state of the block $\kappa_{\mathtt{a}}$ and the emission distributions are defined according to state labels. There is a probability one to start in state 1d. Any probability distribution over Σ^* generated by a HMM with $|Q|$ states over an alphabet Σ can be represented by a POMM with $\mathcal{O}(|Q|.|\Sigma|)$ states [1].

2.3 Phase-Type Distributions

A discrete finite Markov chain (MC) is a stochastic process $\{X_t \mid t \in \mathbb{N}\}$ where the random variable X takes its value at any discrete time t in a finite set Q and such that: $P[X_t = q \mid X_{t-1}, X_{t-2}, \ldots, X_0] = P[X_t = q \mid X_{t-1}, \ldots, X_{t-p}]$. This condition states that the probability of the next outcome only depends on the last p values of the process (Markov property). A MC can be represented by a 3-tuple $T = \langle Q, A, \boldsymbol{\iota} \rangle$ where Q is a finite set of states, A is a $|Q| \times |Q|$ transition probability matrix and $\boldsymbol{\iota}$ is a $|Q|-$dimensional vector representing the initial probability distribution. A MC is *absorbing* if the process has a probability one to get trapped into a state q. Such a state is called *absorbing*. The state set can be partitioned into the *absorbing set* $Q_A = \{q \in Q \mid A_{qq} = 1\}$ and its complementary set, the *transient set* Q_T. The *time to absorption* is the number of steps the process takes to reach an absorbing state.

Definition 3 (Discrete Phase-type (PH) Distribution). *A probability distribution $\varphi(.)$ on $\mathbb{N}^0$ is a* distribution of phase-type *(PH) if and only if it is the distribution of the time to absorption in an absorbing MC.*

The probability distribution of $\varphi(.)$ is classically computed using matrix operations [5]. However, this computation is performed here via *forward* and *backward* variables, similar to those used in the Baum-Welch algorithm [9], which are useful in the POMMPhit algorithm (see section 4.2). Strictly speaking, computing $\varphi(.)$ only requires one of these two kinds of variables but both of them are needed in POMMPhit. Given a set $\mathcal{S} \subseteq Q_T$ of starting states, a state $q \in Q$ and a time $t \in \mathbb{N}$, the forward variable $\alpha^{\mathcal{S}}(q,t)$ computes the probability that the process started in $\mathcal{S}$ reaches state q after having moved over transient states during t steps: $\alpha^{\mathcal{S}}(q,t) = P[X_t = q, \{X_k\}_{k=1}^{t-1} \in Q_T \mid X_0 \in \mathcal{S}]$. Given a set $\mathcal{E} \subseteq Q_A$ of absorbing states, a state $q \in Q$ and a time $t \in \mathbb{N}$, the backward variable $\beta^{\mathcal{E}}(q,t)$ computes the probability that state q is reached by the process t steps before getting absorbed in $\mathcal{E}$: $\beta^{\mathcal{E}}(q,t) = P[X_0 = q, \{X_k\}_{k=1}^{t-1} \in Q_T \mid X_t \in \mathcal{E}]$. The forward variables can be computed using the following recurrence for $q \in Q$ and $t \in \mathbb{N}$:

$$\alpha^{\mathcal{S}}(q,0) = \begin{cases} \iota_q^{\mathcal{S}} & \text{if } q \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases} \qquad \alpha^{\mathcal{S}}(q,t) = \sum_{q' \in Q_T} \alpha^{\mathcal{S}}(q', t-1) A_{q'q} \tag{1}$$

where $\boldsymbol{\iota}^{\mathcal{S}}$ denotes an initial distribution over $\mathcal{S}$. The following recurrence computes the backward variables for $q \in Q$ and $t \in \mathbb{N}$:

$$\beta^{\mathcal{E}}(q,0) = \begin{cases} 1 & \text{if } q \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \qquad \beta^{\mathcal{E}}(q,t) = \begin{cases} 0 & \text{if } q \in \mathcal{E} \\ \sum_{q' \in Q} \beta^{\mathcal{E}}(q', t-1) A_{qq'} & \text{otherwise} \end{cases} \tag{2}$$

Using these variables, the probability distribution of φ is computed as follows for all $t \in \mathbb{N}^0$:

$$\varphi(t) = \sum_{q \in Q_A} \alpha^{Q_T}(q,t) = \sum_{q \in Q_T} \iota_q^{Q_T} \beta^{Q_A}(q,t) \tag{3}$$

where $\boldsymbol{\iota}^{Q_T}$ is the initial distribution of the MC for transient states. Each transient state of the absorbing MC is called a *phase*. This technique is powerful since it decomposes complex distributions such as the hyper-geometric or the Coxian distribution as a combination of phases. These distributions can be defined using specific absorbing MC structures. A distribution with an initial vector and a transition matrix with no structural constraints is called here a *general PH distribution*.

2.4 Jensen-Shannon Divergence

The Jensen-Shannon divergence is a function which measures the distance between two distributions [7]. Let $\mathcal{P}$ denote the space of all probability distributions defined over a discrete set of events Ω. The JS divergence is a function $\mathcal{P} \times \mathcal{P} \to \mathbb{R}$ defined by $D_{JS}(P_1, P_2) = H(M) - \frac{1}{2}H(P_1) - \frac{1}{2}H(P_2)$ where $P_1, P_2 \in \mathcal{P}$ are two distributions, $M = \frac{1}{2}(P_1 + P_2)$ and $H(P) = -\sum_{e \in \Omega} P[e] \log P[e]$ is the Shannon entropy. The JS divergence is non-negative and is bounded by 1 [7]. It can be thought of as a symmetrized and smoothed variant of the KL divergence as it is relative to the mean of the distributions.

3 First Passage Times in POMMs

In this section, the distributions of the FPT in POMMs are studied. We show that the FPT distributions between blocks are of phase-type by constructing their representing absorbing MC. POMMStruct aims at fitting these PH distributions from the FPT observed in a training sample. We motivate the use of these distributions by showing that they are informative about the model structure to be learned.

First, let us formally define the FPT for a pair of symbols $(\mathtt{a}, \mathtt{b})$ in a POMM.

Definition 4 (First Passage Times in POMMs). *Given a POMM* $H = \langle \Sigma, Q, A, B, \iota \rangle$*, the* first passage time *(FPT) is a function* $\mathrm{fpt} : \Sigma \times \Sigma \to \mathbb{N}^0$ *such that* $\mathrm{fpt}(\mathtt{a}, \mathtt{b})$ *is the number of steps before reaching the block* $\kappa_\mathtt{b}$ *for the first time, leaving initially from the block* $\kappa_\mathtt{a}$*:* $\mathrm{fpt}(\mathtt{a}, \mathtt{b}) = \inf_t \{t \in \mathbb{N}^0 | X_t \in \kappa_\mathtt{b} \text{ and } X_0 \in \kappa_\mathtt{a}\}$.

The FPT from block $\kappa_\mathtt{a}$ to block $\kappa_\mathtt{b}$ are drawn from a phase-type distribution obtained by (i) defining an initial distribution[1] $\boldsymbol{\iota}^{\kappa_\mathtt{a}}$ over $\kappa_\mathtt{a}$ such that $\iota_q^{\kappa_\mathtt{a}}$ is the expected[2] proportion of time the process reaches state q relatively to the states in κ_a and (ii) transforming the states in $\kappa_\mathtt{b}$ to be absorbing. It is assumed here that

[1] $\boldsymbol{\iota}^{\kappa_\mathtt{a}}$ is not the initial distribution of the POMM but it is the initial distribution for the FPT starting in $\kappa_\mathtt{a}$.

[2] This expectation can be computed using standard MC techniques (see [4]).

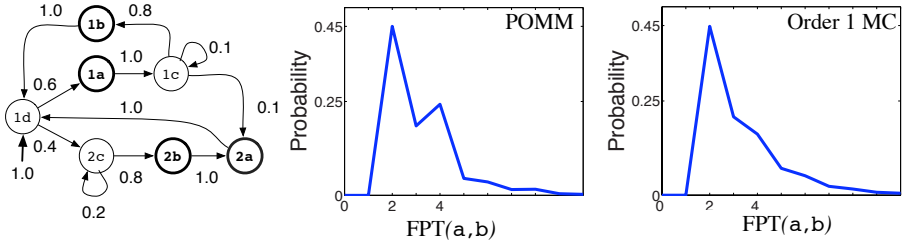

Fig. 1. Left: an irreducible POMM H. Center: the distribution of the FPT from block $\kappa_{\mathtt{a}}$ to block $\kappa_{\mathtt{b}}$ in H. Right: the FPT distribution from $\mathtt{a}$ to $\mathtt{b}$ in an order 1 MC estimated from 1000 sequences of length 100 generated from H.

$\mathtt{a} \neq \mathtt{b}$. Otherwise, a similar absorbing MC can be constructed but the states in $\kappa_{\mathtt{a}}$ have to be duplicated such that the original states are used as starting states and the duplicated ones are transformed to be absorbing. The probability distribution of $\text{fpt}(\mathtt{a}, \mathtt{b})$ is computed as follows for all $t \in \mathbb{N}^0$:

$$P[\text{fpt}(\mathtt{a}, \mathtt{b}) = t] \propto \sum_{q \in \kappa_{\mathtt{b}}} \alpha^{\kappa_{\mathtt{a}}}(q, t) = \sum_{q \in \kappa_{\mathtt{a}}} \iota_q^{\kappa_{\mathtt{a}}} \beta^{\kappa_{\mathtt{b}}}(q, t) \tag{4}$$

An irreducible POMM H and its associated PH distribution from block $\kappa_{\mathtt{a}}$ to block $\kappa_{\mathtt{b}}$ are depicted respectively in the left and center parts of Figure 1. The obtained PH distribution has several modes (*i.e.* maxima), the most noticeable being at times 2 and 4. These modes reveal the presence of paths of length[3] 2 and 4 from $\kappa_{\mathtt{a}}$ to $\kappa_{\mathtt{b}}$ having a large probability. For instance, the paths `1a,1c,1b` and `2a,1d,1a,1c,1b` have a respective probability equal to 0.45 and 0.21 (other paths of length 4 yield a total probability equal to 0.25 for this length). Other informations related to the model structure such as long-term dependencies can also be deduced from the FPT distributions [1]. These structural informations, available in the learning sequences, are exploited in the induction algorithm POMMStruct presented in section 4. It starts by estimating a standard MC from the training sequences. The right part of Figure 1 shows the FPT distribution from $\mathtt{a}$ to $\mathtt{b}$ in an order 1 MC estimated from sequences drawn from H. The FPT dynamics from $\mathtt{a}$ to $\mathtt{b}$ in the MC poorly approximates the FPT dynamics from $\kappa_{\mathtt{a}}$ to $\kappa_{\mathtt{b}}$ in H as there is only a single mode. POMMStruct iteratively adds states to the estimated model and reestimate its probabilistic parameters in order to best match the observed FPT dynamics.

4 The Induction Algorithm: POMMStruct

This section presents the POMMStruct algorithm which learns the structure and the parameters of a POMM from a set of training sequences S_{train}. The objective is to induce a model that best reproduces the FPT dynamics extracted

[3] The length of a path is defined here in terms of number of steps.

from S_{train}. Section 4.1 presents the general structure of the induction algorithm. Reestimation formulas for fitting FPT distributions are detailed in section 4.2.

4.1 POMM Induction

The pseudo-code of POMMSTRUCT is presented in Algorithm 1.

Algorithm POMMSTRUCT
Input: • A training sample S_{train}
• The order r of the initial model
• The number p of pairs
• A precision parameter ϵ
Output: A collection of POMMs

$EP_0 \leftarrow \texttt{initialize}(S_{train}, r)$;
$FPT_{train} \leftarrow \texttt{extractFPT}(S_{train})$;
$\mathcal{F} \leftarrow \texttt{selectDivPairs}(EP_0, FPT_{train}, p)$;
$EP_0 \leftarrow \text{POMMPHIT}(EP_0, FPT_{train}, \mathcal{F})$;
$Lik_{train} \leftarrow \texttt{FPTLikelihood}(EP_0, FPT_{train})$;
$i \leftarrow 0$
repeat
  $Lik_{last} \leftarrow Lik_{train}$;
  $\kappa_j \leftarrow \texttt{probeBlocks}(EP_i, FPT_{train})$;
  $EP_{i+1} \leftarrow \texttt{addStateInBlock}(EP_i, \kappa_j)$;
  $EP_{i+1} \leftarrow \text{POMMPHIT}(EP_{i+1}, FPT_{train}, \mathcal{F})$
  $Lik_{train} \leftarrow \texttt{FPTLikelihood}(EP_{i+1}, FPT_{train})$;
  $i \leftarrow i + 1$
until $\frac{|Lik_{train} - Lik_{last}|}{|Lik_{last}|} < \epsilon$;
return $\{EP_0, \ldots, EP_i\}$

Algorithm 1. POMM Induction by fitting FPT dynamics

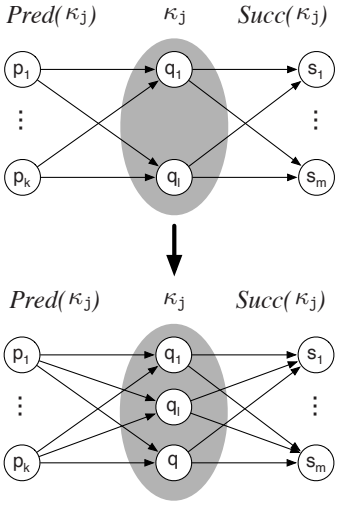

Fig. 2. Adding a new state q in the block κ_j

An initial order r MC is estimated first from S_{train} by the function `initialize`. Next, the function `extractFPT` extracts the FPT in the sample for each pair of symbols according to definition 1. Using the Jensen-Shannon (JS) divergence, `selectDivPairs` compares the FPT distributions of the initial MC with the empirical FPT distributions of the sample. The p most diverging pairs $\mathcal{F}$ are selected to be fit during induction process, where p is an input parameter. In addition, the selected pairs can be weighted according to their JS divergence in order to give more importance to the poorly fitted pairs. This is achieved by multiplying the parameters w_i in $FPT(\mathtt{a}, \mathtt{b})$ (see definition 1) by the JS divergence obtained for this pair. The JS divergence is particularly well-suited for this feature weighting as it is positive and upper bounded by one. The parameters of the initial model are reestimated using the POMMPHIT algorithm presented in section 4.2. This EM-based method computes the model parameters that maximize the likelihood of the selected FPT pairs.

States are iteratively added to the model in order to improve the fit to the observed dynamics. At the beginning of each iteration, the procedure `probeBlocks` determines the block κ_j of the model in which a new state is added. This block is selected as the one leading to the larger FPT likelihood improvement. To do so, `probeBlocks` tries successively to add a state in each block using the `addStateInBlock` procedure detailed hereafter. For each candidate block, a few iterations of POMMPHit is applied to reestimate the model parameters. The block κ_j offering the largest improvement is returned. The `addStateInBlock` function (illustrated in Figure 2) inserts a new state q in κ_j such that q is connected to all the predecessors (*i.e.* states having at least one outgoing transition to a state in κ_j) and successors (*i.e.* states having at least one incoming transition from a state in κ_j) of κ_j. These two sets need not to be disjoint and may include states in κ_j (if they are connected to some state(s) in κ_j).

The probabilistic parameters of the augmented model are estimated using POMMPHit until convergence. An interesting byproduct of POMMPHit are the expected transition passage times (see section 4.2). It provides the average number of times the transitions are triggered when observing the FPT in the sample. According to this criterion, the less frequently used transitions are successively trimmed off from the model. Whenever a transition is removed, the parameters of the model are reestimated using POMMPHit. In general, the convergence is attained after a few iterations as the parameters not affected by the trimming are already well estimated. Transitions are trimmed until the likelihood function no longer increases. This procedure has several benefits: (i) it can move POMMPHit away from a local minimum of the FPT likelihood function (ii) it makes the model sparser and therefore reduces the computational resources needed in the forward-backward computations (see section 4.2) and (iii) the obtained model is more interpretable. POMMStruct is iterated until convergence of the FPT likelihood up to a precision parameter ϵ. A validation procedure is used to select the best model from the collection of models $\{EP_0, \ldots, EP_i\}$ returned by POMMStruct. Each model is evaluated on an independent validation set of sequences and the model offering the highest FPT likelihood is chosen. At each iteration, the computational complexity is dominated by the complexity of POMMPHit (see section 4.2).

POMMStruct does not maximize the likelihood of the training sequences in the model but the likelihood of the FPT extracted from these sequences. We argued in section 3 that maximizing this criterion is relevant to learn an adequate model topology. If one wants to perform sequence prediction, *i.e.* predicting the next outcomes of a process given its past history, the parameters of the model may be adjusted towards this objective. This can be achieved by applying the standard Baum-Welch procedure initialized with the model resulting from POMMStruct.

4.2 Fitting the FPT: POMMPHit

In this section, we introduce the POMMPHit algorithm for fitting the FPT distributions between blocks in POMMs from the FPT observed in the sequences.

POMMPHIT is based on the Expectation-Maximization (EM) algorithm and extends the PHIT algorithm presented in [1] for fitting a single PH distribution. For each pair of symbol $(\mathtt{a},\mathtt{b})$, the observations consist of the FPT $\{(z_1,w_1),\ldots,(z_l,w_l)\}$ extracted from the sequences according to definition 1. The observations for a given pair $(\mathtt{a},\mathtt{b})$ are assumed to be independent from the observations for the other pairs. While this assumption is generally not satisfied, it drastically simplifies the reestimation formula and consequently offers an important computational speed-up. Moreover, good results are obtained in practice. A passage time z_i is considered here as an incomplete observation of the pair (z_i,h_i) where h_i is the sequence of states reached by the process to go from block $\kappa_\mathtt{a}$ to block $\kappa_\mathtt{b}$ in z_i steps. In the sequel, $\mathcal{H}^{\mathtt{a},\mathtt{b}}$ denotes the set of hidden paths from block $\kappa_\mathtt{a}$ to block $\kappa_\mathtt{b}$. Before presenting the expectation and maximization steps in POMMPHIT, let us introduce auxiliary hidden variables which provide sufficient statistics to compute the complete FPT likelihood function $P[Z,\mathcal{H}\mid\lambda]$ conditioned to the model parameters λ:

- $S^{\mathtt{a},\mathtt{b}}(q)$: the number of observations in $\mathcal{H}^{\mathtt{a},\mathtt{b}}$ starting in state $q\in\kappa_\mathtt{a}$,
- $N^{\mathtt{a},\mathtt{b}}(q,q')$: the number of times state q' immediately follows state q in $\mathcal{H}^{\mathtt{a},\mathtt{b}}$.

The complete FPT likelihood function is defined as follows:

$$P[Z,\mathcal{H}\mid\lambda]=\prod_{\mathtt{a},\mathtt{b}\in\mathcal{F}}\prod_{q\in\kappa_\mathtt{a}}(\iota_q^{\kappa_\mathtt{a}})^{S^{\mathtt{a},\mathtt{b}}(q)}\prod_{q,q'\in Q}A_{qq'}^{N^{\mathtt{a},\mathtt{b}}(q,q')} \tag{5}$$

where $\boldsymbol{\iota}^{\kappa_\mathtt{a}}$ is the initial distribution over $\kappa_\mathtt{a}$ for the FPT starting in $\kappa_\mathtt{a}$.

Expectation step

The expectation of the variables $S^{\mathtt{a},\mathtt{b}}(q)$ and $N^{\mathtt{a},\mathtt{b}}(q,q')$ are conveniently computed using the *forward* and *backward* variables respectively introduced in equations (1) and (2). These reccurences are efficiently computed using a $|Q|\times L^{\mathtt{a},\mathtt{b}}$ lattice structure where $L^{\mathtt{a},\mathtt{b}}$ is the longest observed FPT from $\mathtt{a}$ to $\mathtt{b}$. The conditional expectation of the auxiliary variables given the observations $\overline{S^{\mathtt{a},\mathtt{b}}}(q)=E[S^{\mathtt{a},\mathtt{b}}(q)\mid FPT(\mathtt{a},\mathtt{b})]$ and $\overline{N^{\mathtt{a},\mathtt{b}}}(q,q')=E[N^{\mathtt{a},\mathtt{b}}(q,q')\mid FPT(\mathtt{a},\mathtt{b})]$ are:

$$\overline{S^{\mathtt{a},\mathtt{b}}}(q)=\sum_{(z,w)\in FPT(\mathtt{a},\mathtt{b})}w\;\frac{\iota_q^{\kappa_\mathtt{a}}\beta^{\kappa_\mathtt{b}}(q,z)}{\sum_{q\in\kappa_\mathtt{a}}\iota_q^{\kappa_\mathtt{a}}\beta^{\kappa_\mathtt{b}}(q,z)} \tag{6}$$

$$\overline{N^{\mathtt{a},\mathtt{b}}}(q,q')=\sum_{(z,w)\in FPT(\mathtt{a},\mathtt{b})}w\sum_{t=0}^{z-1}\frac{\alpha^{\kappa_\mathtt{a}}(q,t)A_{qq'}\beta^{\kappa_\mathtt{b}}(q',z-t-1)}{\sum_{q\in\kappa_\mathtt{a}}\iota_q^{\kappa_\mathtt{a}}\beta^{\kappa_\mathtt{b}}(q,z)} \tag{7}$$

The previous computations assume that $\mathtt{a}\neq\mathtt{b}$. In the other case, the states in $\kappa_\mathtt{a}$ have to be preliminary duplicated as described in section 3. The obtained conditional expectations are used in the maximization step of POMMPHIT but also in the trimming procedure of POMMSTRUCT. In particular, $\sum_{(\mathtt{a},\mathtt{b})\in\mathcal{F}}\overline{N^{\mathtt{a},\mathtt{b}}}(q,q')$ provides the average number of times the transition $q\rightarrow q'$ is triggered while observing the sample FPT.

Maximization step
Given the conditional expectations, $\overline{S^{\mathtt{a},\mathtt{b}}}(q)$ and $\overline{N^{\mathtt{a},\mathtt{b}}}(q,q')$, the maximum likelihood estimates of the POMM parameters are the following for all $q, q' \in Q$:

$$\iota_q^{\kappa_\mathtt{a}} = \frac{\sum_{\mathtt{b}\in\{\mathtt{b}|(\mathtt{a},\mathtt{b})\in\mathcal{F}\}} \overline{S^{\mathtt{a},\mathtt{b}}}(q)}{\sum_{q\in\kappa_\mathtt{a}} \sum_{\mathtt{b}\in\{\mathtt{b}|(\mathtt{a},\mathtt{b})\in\mathcal{F}\}} \overline{S^{\mathtt{a},\mathtt{b}}}(q)} \text{ where } q \in \kappa_\mathtt{a}, \quad A_{qq'} = \frac{\sum_{\mathtt{a},\mathtt{b}\in\mathcal{F}} \overline{N^{\mathtt{a},\mathtt{b}}}(q,q')}{\sum_{q'\in Q} \sum_{\mathtt{a},\mathtt{b}\in\mathcal{F}} \overline{N^{\mathtt{a},\mathtt{b}}}(q,q')} \tag{8}$$

The computational complexity per iteration is $\Theta(pL^2m)$ where p is the number of selected pairs, L is the longest observed FPT and m is the number of transitions in the current model. An equivalent bound for this computation is $\mathcal{O}(pL^2|Q|^2)$, but this upper bound is tight only if the transition matrix A is dense.

5 Experiments

This section presents experiments conducted with POMMStruct on artificially generated data and on DNA sequences. In order to report comparative results, experiments were also performed with the Baum-Welch algorithm and the Bayesian state merging algorithm due to Stolcke [10]. The Baum-Welch algorithm is applied on fully connected graphs of increasing sizes. For each considered model size, three different random seeds are used and the model having the largest likelihood is kept. Additionally, a transition trimming procedure, based on the transition probabilities, has been used. The optimal model size is selected on a validation set obtained by holding out 25% of the training data. The Bayesian state merging technique of Stolcke has been reimplemented according to the setting described in the section 3.6.1.6 of [10]. The *effective sample size* parameter, defining the weight of the prior versus the likelihood, has been tuned[4] in the set $\{1, 2, 5, 10, 20\}$. The POMMStruct algorithm is initialized with an order $r \in \{1, 2\}$ MC. All observed FPT pairs are considered (*i.e.* $p = |\Sigma|^2$) without feature weighting. Whenever applied, the POMMPHit algorithm is initialized with three different random seeds and the parameters leading to the largest FPT likelihood are kept. The optimal model size is selected similarly as for the Baum-Welch algorithm.

Artificially generated sequences were drawn from target POMMs having a complex FPT dynamics and with a tendency to include long-term dependencies [1]. From each target model, 500 training sequences and 250 test sequences of length 100 were generated. The evaluation criterion considered here is the Jensen-Shannon (JS) divergence between the FPT distributions of the model and the empirical FPT distributions extracted from the test sequences. This is a good measure to assess whether the model structure represents well the dynamics in the test sample. The JS divergence is averaged over all pairs of symbols. The left part of Figure 3 shows learning curves for the 3 considered techniques on test sequences drawn from an artificial target model with 32 states and an

[4] The fixed value of 50 recommended in [10] performed poorly in our experiments.

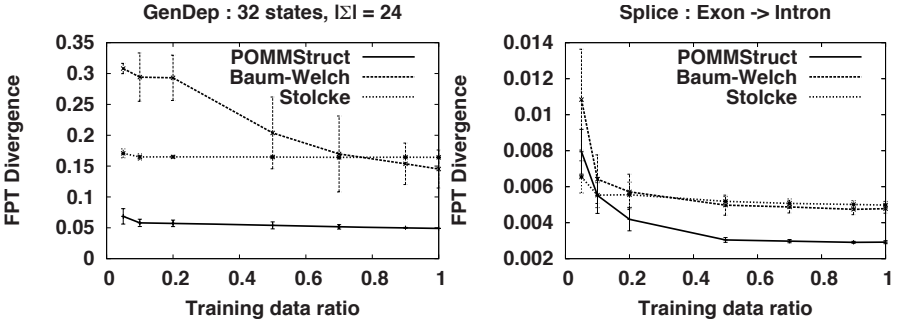

Fig. 3. Left: Results obtained on test sequences generated by an artificial target model with 32 states. Right: Results obtained on the `Splice` test sequences.

alphabet size equal to 24. For each training size, results are averaged over 10 samples of sequences[5]. POMMStruct outperforms its competitors for all training set sizes. Knowledge of the target machine size is not provided to our induction algorithm. However, if one would stop the iterative state adding using this target state number, the resulting number of transitions very often matches the target. The algorithm of Stolcke performed well for small amounts of data but the performance does not improve much when more training data are available. The Baum-Welch technique poorly fits the FPT dynamics when a small amount data is used. However, when more data are available ($\geq 70\%$), it provides slightly better results than the Stolcke's approach. Performances in sequence prediction (which is not the main objective of the proposed approach) can be assessed with test perplexity. The relative perplexity increases with respect to the target model, used to generate the sequences, for POMMStruct[6], the approach of Stolcke and the Baum-Welch algorithm are respectively 2%, 18% and 21%. When all the training data are used, the computational run-times are the following: about 3.45 hours for POMMStruct, 2 hours for Baum-Welch and 35 minutes for Stolcke's approach . Experiments were also conducted on DNA sequences containing exon-intron boundaries from the `Splice`[7] dataset. The training and the test sets contain respectively 500 and 235 sequences of length 60. The FPT dynamics in these sequences is less complex than in the generated sequences, leading to smaller absolute JS divergences for all techniques. The right part of Figure 3 shows learning curves for the 3 induction techniques. Again, POMMStruct, initialized here with an order 2 MC, exhibits the best overall performance. When more than 50% of the training data are used, the Baum-Welch algorithm performs slightly better than the technique of Stolcke. The perplexity obtained with POMMStruct and Baum-Welch are comparable while the approach of

[5] The errorbars in the plot represent standard deviations.

[6] Emissions and transitions probabilities of the model learned by POMMStruct have been reestimated here with the Baum-Welch algorithm without adapting the model structure.

[7] `Splice` is available from the UCI repository.

Stolcke performs slightly worse (4% of relative perplexity increase). When all the training data are used, the computational run-times are the following: 25 minutes for Baum-Welch and 17 minutes for Stolcke's approach and 6 minutes for POMMStruct.

6 Conclusion

We propose in this paper a novel approach to the induction of the structure of Partially Observable Markov models (POMMs) which are graphical models equivalent to Hidden Markov Models. A POMM is constructed to best fit the First Passage Times (FPT) dynamics between symbols observed in the learning sample. Unlike N-grams, these features are not local as there is no fixed maximum time (*i.e.* number of steps) between two events. Furthermore, the FPT distributions contain relevant informations, such as the presence of dominant path lengths or long-term dependencies, about the structure of the model to be learned. The proposed algorithm, POMMStruct, induces the structure and the parameters of a POMM that best fit the FPT observed in the training sample. Additionally, the less frequently used transitions in the FPT are trimmed off from the model. POMMStruct is iterated until the convergence of the FPT likelihood function. Experimental results illustrate that the proposed technique is better suited to fit a process with a complex FPT dynamics than the Baum-Welch algorithm applied with a fully connected graph with transition trimming or the Bayesian state merging approach of Stolcke.

Our future work includes extension of the proposed approach to model FPT between substrings rather than between individual symbols. An efficient way to take into account the dependencies between the FPT in the reestimation procedure of POMMPHit will also be investigated. Applications of the proposed approach to other datasets will also be considered, typically in the context of novelty detection where the FPT might be very relevant features.

References

1. Callut, J., Dupont, P.: Inducing hidden markov models to model long-term dependencies. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) ECML 2005. LNCS (LNAI), vol. 3720, pp. 513–521. Springer, Heidelberg (2005)
2. Durbin, R., Eddy, S., Krogh, A., Mitchison, G.: Biological sequence analysis. Cambridge University Press, Cambridge (1998)
3. Freitag, D., McCallum, A.: Information extraction with HMM structures learned by stochastic optimization. In: Proc. of the Seventeenth National Conference on Artificial Intelligence, AAAI, pp. 584–589 (2000)
4. Kemeny, J.G., Snell, J.L.: Finite Markov Chains. Springer, Heidelberg (1983)
5. Latouche, G., Ramaswami, V.: Introduction to Matrix Analytic Methods in Stochastic Modeling. Society for Industrial & Applied Mathematics, U.S. (1999)
6. Li, J., Wang, J., Zhao, Y., Yang, Z.: Self-adaptive design of hidden markov models. Pattern Recogn. Lett. 25(2), 197–210 (2004)

7. Lin, J.: Divergence measures based on the shannon entropy. IEEE Trans. Information Theory 37, 145–151 (1991)
8. Ostendorf, M., Singer, H.: HMM topology design using maximum likelihood successive state splitting. Computer Speech and Language 11, 17–41 (1997)
9. Rabiner, L., Juang, B.-H.: Fundamentals of Speech Recognition. Prentice-Hall, Englewood Cliffs (1993)
10. Stolcke, A.: Bayesian Learning of Probabilistic Language Models. Ph. D. dissertation, University of California (1994)
11. Zhu, H., Wang, J., Yang, Z., Song, Y.: A method to design standard hmms with desired length distribution for biological sequence analysis. In: Bücher, P., Moret, B.M.E. (eds.) WABI 2006. LNCS (LNBI), vol. 4175, pp. 24–31. Springer, Heidelberg (2006)