Inducing Hidden Markov Models to Model Long-Term Dependencies

Jérôme Callut and Pierre Dupont

Department of Computing Science and Engineering,
INGI, Université catholique de Louvain,
Place Sainte-Barbe 2,
B-1348 Louvain-la-Neuve, Belgium
{jcal, pdupont}@info.ucl.ac.be

Abstract. We propose in this paper a novel approach to the induction of the structure of Hidden Markov Models. The induced model is seen as a lumped process of a Markov chain. It is constructed to fit the dynamics of the target machine, that is to best approximate the stationary distribution and the mean first passage times observed in the sample. The induction relies on non-linear optimization and iterative state splitting from an initial order one Markov chain.

Keywords: HMM topology induction, Partially observable Markov models, Mean first passage times, Lumped Markov process, State splitting algorithm.

1 Introduction

Hidden Markov Models (HMMs) are widely used in many pattern recognition areas, including applications to speech recognition [10], biological sequence modeling [4], information extraction [5,6] and optical character recognition [8], to name a few. In most cases, the model structure, also referred to as topology, is defined according to some prior knowledge of the application domain. Automatic techniques for inducing 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.

Previous works with HMMs mainly concentrated either on hand-built models (*e.g.* [5]) or heuristics to refine predefined structures combined with EM estimation [6]. More principled approaches are the Bayesian merging technique due to Stolcke [12] and the maximum likelihood state-splitting method of Ostendorf and Singer [9]. The former approach however has not been shown to clearly outperform alternative approaches while the latter is specific to the subclass of left-to-right HMMs modeling speech signals.

The present contribution describes a novel approach to the structural induction of HMMs. The general objective is to induce the structure and to estimate the parameters of a HMM from a sample assumed to have been drawn from an unknown target HMM. The goal however is not the identification of the target

J. Gama et al. (Eds.): ECML 2005, LNAI 3720, pp. 513–521, 2005.

model but the induction of a model sharing with the target the main features of the distribution it generates. We restrict here our attention to features that can be deduced from the sample. These features are closely related to fundamental quantities of a Markov process, namely the *stationary distribution* and *mean first passage times* (MFPT). In other words, the induced model is built to fit the dynamics of the target machine observed in the sample, not necessarily to match its structure.

Section 2 reviews some useful definitions coming from the theory of discrete Hidden Markov Models and Markov Chains. We use here a specific representation class for distributions generated by HMMs, called *Partially Observable Markov Models* (POMMs). This class is general enough since any discrete HMM can equivalently be represented by a POMM [2].

HMMs are able to model a class of distributions broader than finite order Markov chains. In particular, section 3 describes why HMMs, with an appropriate topology, are well suited to represent long term probabilistic dependencies in a compact way. We also argue why accurate modeling of these dependencies cannot be achieved through the classical approach of Baum-Welch estimation of a fully connected model. These observations motivate the use of MFPT to guide the search of an appropriate model. The resulting induction algorithm is presented in section 4. Comparative results given in [3] illustrate the superiority of POMM induction over variable order Markov chains (equivalent to back-off smoothed Ngrams) and EM estimation of a fully connected HMM.

2 Partially Observable Markov Models, Markov Chains and Lumped Processes

We introduce here Partially Observable Markov Models and we review some fundamental notions of the Markov chains theory.

Definition 1 (POMM). *A* Partially Observable Markov Model (POMM) *is a HMM* $M = \langle \Sigma, Q, A, B, \iota \rangle$ *where* Σ *is an alphabet,* Q *is a set of states,* $A : Q \times Q \rightarrow [0,1]$ *is a mapping defining the probability of each transition,* $B : Q \times \Sigma \rightarrow [0,1]$ *is a mapping defining the emission probability of each letter on each state, and* $\iota : Q \rightarrow [0,1]$ *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 letter. This model is called *partially* observable since, in general, several distinct states can emit the same letter. As for a HMM, the observation of a string emitted during a random walk does not allow one to identify the states from which each letter was emitted. However, the observations define *state subsets* from which each letter may have been emitted. Any distribution generated by a HMM with $|Q|$ states over an alphabet Σ can be represented by a POMM with $\mathcal{O}(|Q|.|\Sigma|)$ states [2].

The notion of POMM is closely related to a standard Markov Chain (MC). Indeed, in the particular case where all states emit a different letter, the process of a POMM is fully observable. Moreover the Markov property is satisfied as, by definition, the probability of any transition only depends on the current state. Some fundamental properties of a Markov chain are recalled hereafter and the links between a POMM and a MC are further detailed. A MC can be represented by a 3-tuple $T = \langle Q, A, \iota \rangle$ where Q is a finite set of states, A is a $|Q| \times |Q|$ transition probability matrix and ι is a $|Q|-$dimensional vector representing the initial probability distribution. The *stationary distribution* and *mean first passage times* are two fundamental quantities characterizing the dynamics of a Markov chain[1]. The stationary distribution is a $|Q|-$dimensional stochastic vector $\boldsymbol{\pi}$ such that $\boldsymbol{\pi}^T A = \boldsymbol{\pi}^T$. The q-th entry of $\boldsymbol{\pi}$ can be interpreted as the expected proportion of the time the Markov process in steady-state reaches state q. Given two states q and q', the Mean First Passage Time (MFPT) $M_{qq'}$ is the expected number of steps before reaching state q' for the first time while leaving initially from state q.

Given a MC, a partition can be defined on its state set and the resulting process is said to be *lumped*.

Definition 2 (Lumped process). *Given a regular MC, $T = \langle Q, A, \iota \rangle$, let $q^{(t)}$ be the state reached at time t during a random walk in T. The set $\kappa = \{\kappa_1, \kappa_2, \ldots, \kappa_r\}$ denotes a partition of the set of states Q. The function $K_\kappa = Q \to \kappa$ maps the state q to the block of κ that contains q. The* lumped process *$T/\!/\kappa$ outcomes $K_\kappa(q^{(t)})$ at time t.*

While the states are fully observable during a random walk in a MC, a lumped process is associated with random walks where only state *subsets* are observed. In this sense, the lumped process makes the MC only partially observable as in the case of a POMM. Conversely, a random walk in a POMM can be considered as a lumped process of its underlying MC with respect to an *observable partition* of its state set. Each block of the observable partition corresponds to the state(s) emitting a specific letter. In this case, both models define the same string distribution. The induction algorithm presented in section 4 is based on the MFPT extended to lumped processes.

Definition 3 (MFPT for a lumped process). *Given a regular MC $T = \langle Q, A, \iota \rangle$, κ a partition of Q and κ_i, κ_j two blocks of κ, an absorbing MC T^{κ_j} is created from T by transforming every state of κ_j to be absorbing. Furthermore, let $\boldsymbol{w}^{\kappa_j}$ be the MTA vector of T^{κ_j}. The mean first passage time $M_{ij}/\!/\kappa$ from κ_i to κ_j in the lumped process $T/\!/\kappa$ is defined as follows: $M_{ij}/\!/\kappa = \sum_{q \in \kappa_i} \frac{\pi_q}{\pi_{\kappa_i}} \boldsymbol{w}_q^{\kappa_j}$ if $\kappa_i \neq \kappa_j$ and $M_{ii}/\!/\kappa = \frac{1}{\pi_{\kappa_i}}$, where π_q is the stationary distribution of state q in T, $\pi_{\kappa_i} = \sum_{q \in \kappa_i} \pi_q$ is the stationary distribution of the block κ_i in the lumped process $T/\!/\kappa$ and $\boldsymbol{w}^{\kappa_j}$ is the mean time to absorption vector related to κ_j [3,7].*

[1] We focus here on *regular* MCs, which are MCs with strongly connected transition graphs and no periodic states [7].

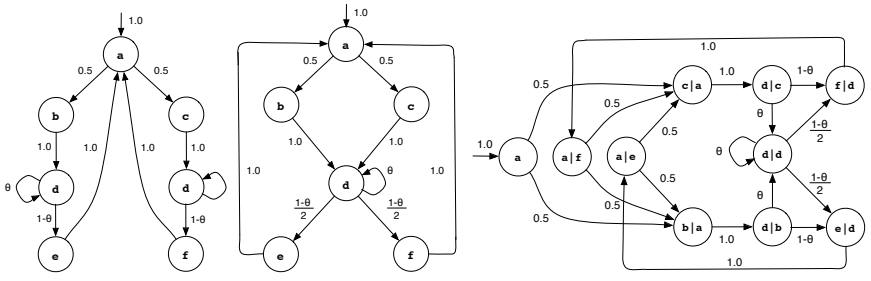

Fig. 1. A parametric POMM T_θ (left) modeled by an order 1 MC (center) or an order 2 MC (right)

3 Modeling Long-Term Probabilistic Dependencies

A stochastic process $\{X_t \,|\, t \in \mathbb{N}\}$ contains long-term dependencies if an outcome at time t significantly depends on an outcome that occurred at a much earlier time t': $P(X_t \mid X_{t-1}, \ldots, X_{t'}) \neq P(X_t \mid H)$ when $H = \{X_{t-1}, \ldots, X_{t-p}\}$ and $p < t - t'$. Hence, the *relevant history size* for such a process is defined as the minimal size of H such that $P(X_t \,|\, X_{t-1}, \ldots, X_{t'}) = P(X_t \,|\, H),\ \forall t, t' \in \mathbb{N}, t' < t$. When the size of the relevant history is bounded, Markov chains of a sufficient order can model the long-term dependencies. On the other hand, if a conditioning event $X_{t'}$ can be arbitrarily far in the past, more powerful models such as HMMs or POMMs are required.

3.1 Modeling Long-Term Dependencies with Finite Order MC

Let us consider the parametric POMM T_θ displayed on the left of Figure 1. Emission of `e` or `f` in this model depends on whether `b` or `c` was emitted right before the last consecutive `d`'s. Depending on the number of consecutive `d`'s, the `b` or `c` outcomes can be arbitrarily far in the past. In other words, the size of the relevant history (*i.e.* the number of consecutive `d`'s + 1) is unbounded. The expected number of consecutive `d`'s is however finite and given by $\sum_{i=0}^{\infty} \theta^i = \frac{1}{1-\theta}$. Consequently, the expected size of the relevant history is $\frac{1}{1-\theta} + 1$. It should be noted that when $\theta = 0$, T_θ can be modeled accurately by an order 2 MC[2] since the relevant history size equals 2.

A model would badly fit the distribution defined by T_θ if it would first emit `f` rather than `e` after having emitted `b`. The probability of such an event is $P_{error} = P(t_{\mathtt{f}} < t_{\mathtt{e}} \mid X_t = \mathtt{b})$ where $t_{\mathtt{f}}$ and $t_{\mathtt{e}}$ denote the respective times of the first `f` or `e` after the outcome `b`. In the target model T_θ, $P_{error} = 0$. If the same process is modeled by an order 1 MC (center of Figure 1), $P_{error} = 0.5$. Indeed, when the process reaches state `d`, there is an equal probability to reach states `e` or `f`. In particular, these probabilities do not depend on previous emissions of `b` or `c`.

[2] A state label `b|a` in an order 2 MC means that the process emits `b` after having emitted `a`. The probability of the transition from state `b|a` to state `d|b` encodes the second order dependence $P(X_t = d | X_{t-1} = b, X_{t-2} = a)$.

An order 2 MC, as depicted on the right of Figure 1, would have $P_{error} = 0.475$ when $\theta = 0.95$. In general, the error of an order p MC is given by $P_{error} = \frac{\theta^{p-1}}{2}$. For instance, when $\theta = 0.95$, the expected size of the relevant history is 21 and P_{error} for such a model is still 0.17. Bounding the error probability to 0.1 would require to estimate a MC of order $p = \lceil \log_{0.95}(0.2) + 1 \rceil = 33$. An accurate estimate of such a model requires a huge amount of training data, very unlikely to be available in practice. Hence, POMMs and HMMs can better model long-term dependencies when the relevant history size is unbounded.

3.2 Topology Matters to Fit Long-Term Dependencies with HMMs

Bengio has shown that the use of a good HMM topology is crucial in order to model long term dependencies [1]. Indeed, the classical Baum-Welch algorithm applied to a fully connected graph is hindered by a phenomenon of diffusion of credit: the probability of being in a state at time t becomes gradually independent of the states reached at a previous time $t' \ll t$. In other words, the dependencies on the past outcomes of the process ends up vanishing. This phenomenon is related to the powers of the transition matrix A used in the forward and backward recursions of the Baum-Welch algorithm. Let $\boldsymbol{\iota}_t$ be a row vector representing the distribution of being in each state at time t. This distribution d steps further is given by $\boldsymbol{\iota}_{t+d} = \boldsymbol{\iota}_t A^d$. If the successive powers of A converge quickly to a rank 1 matrix[3] then $\boldsymbol{\iota}_{t+d}$ becomes independent of $\boldsymbol{\iota}_t$. In such a case, the estimation algorithm is likely to be stuck in an inappropriate local minimum of the likelihood function.

For a primitive matrix[4] A, the rate of convergence to rank 1 can be characterized using the Perron-Frobenius theorem [11]. It implies that a primitive stochastic matrix has a unique eigenvalue equal to 1 and that all other eigenvalues are strictly smaller than 1 (in absolute value). If the rank of A is r, then the spectral decomposition of A is given by $A = \lambda_1 \boldsymbol{U}_1 \boldsymbol{V}_1^T + \ldots + \lambda_r \boldsymbol{U}_r \boldsymbol{V}_r^T$, where λ_i is the i-th largest eigenvalue in absolute value and $\boldsymbol{U}_i$, $\boldsymbol{V}_i$ are respectively the right-hand and left-hand eigenvectors associated with λ_i. Furthermore, the spectral decomposition of A^d is given by $A^d = \lambda_1^d \boldsymbol{U}_1 \boldsymbol{V}_1^T + \ldots + \lambda_r^d \boldsymbol{U}_r \boldsymbol{V}_r^T$ that is, taking A to the power d amounts to take its eigenvalues to the power d. Consequently, while taking the successive powers of A, $\lambda_1 = 1$ remains unchanged and all other eigenvalues are decreasing until cancellation. The rate of convergence to rank 1 follows a geometric progression with a ratio that can be approximated by the second[5] largest eigenvalue λ_2.

Classically, the Baum-Welch algorithm is initialized with a uniform random matrix[6]. Such a matrix typically has a very low λ_2. The Baum-Welch algorithm is thus badly conditioned to learn long-term dependencies when initialized in this way. On the other hand, initializing this algorithm with a matrix having λ_2 close to 1 requires prior knowledge of the model topology.

[3] All rows of a rank 1 stochastic matrix are equal.
[4] The transition matrix of a regular MC is primitive.
[5] In the case of the POMM T_θ of Figure 1, $\lambda_2 = \theta$.
[6] Each entry is uniformly drawn in $[0, 1]$ and rows are normalized to sum up to 1.

Table 1. MFPT in $T_{0.95}$ (left), modeled by an order 1 MC (center) or an order 2 MC (right)

$T/\!\!/\kappa$	e	f
b	21.0	67.0
c	67.0	21.0

MC_1	e	f
b	44.0	44.0
c	44.0	44.0

MC_2	e	f
b	42.85	45.15
c	45.15	42.85

3.3 Long-Term Dependencies and MFPT

The MFPT in a lumped process $T/\!\!/\kappa$ contains information about the long-term dynamics of the process. Indeed, the MFPT from the block κ_b to the block κ_e is an expectation of the length of random walks starting with b before emitting e for the first time. Let us assume that the emission of e is conditioned by the fact that the process has first emitted b. The MFPT from b to e is equal to the expected length of the relevant history to predict e from b. Table 1 shows some interesting MFPT in the example T_θ of Figure 1 with $\theta = 0.95$. In the target T_θ, $M_{be} = M_{cf}$ is equal to the expected size of the relevant history (21, see section 3.1). Furthermore, there is a rather long expected time between the outcomes b and f (equivalently between c and e). When T_θ is approximated by an order 1 MC, $M_{\mathtt{be}} = M_{\mathtt{bf}} = M_{\mathtt{ce}} = M_{\mathtt{cf}} = 44$. This means that independently of whether (b or c) were emitted, the outcomes e and f are expected to occur 44 steps later. An order 2 MC only slightly improves the fit to the correct MFPT with respect to an order 1 model.

4 POMM Induction to Model Long-Term Dependencies

A random walk in a POMM can be seen as its underlying MC lumped with respect to the observable partition, as detailed in section 2. We present here an induction algorithm making use of this relation. Given a data sample, assumed to have been drawn from a target POMM TP, our induction algorithm estimates a model EP fitting the dynamics of the MC related to TP. The estimation relies on the stationary distribution and the mean first passage times which can be derived from the sample.

In the present work, we focus on distributions that can be represented by POMMs without final (or termination) probabilities and with regular underlying MC. Since the target process TP never stops, the sample is assumed to have been observed in steady-state. Furthermore, as the transition graph of TP is strongly connected, it is not restrictive to assume that the data is a unique finite string s resulting from a random walk through TP observed during a finite time[7]. Under these assumptions, all transitions of the target POMM and all letters of its alphabet will tend to be observed in the sample. Such a sample can be called *structurally complete*.

[7] The sample statistics could equivalently be computed from repeated finite samples observed in steady-state.

Algorithm POMMStateSplit
Input: A string s from a target POMM
A precision parameter ϵ
Output: A POMM EP_{cur}

$EP \leftarrow$ `initialize`(s);
$\hat{M} \leftarrow$ `sampleMFPT`(s);
$Lik \leftarrow$ `logLikelihood`(EP, s);
repeat
  $Lik_{cur} \leftarrow Lik$;
  $EP_{cur} \leftarrow EP$;
  foreach *state* q *in* EP_{cur} **do**
    $EP_{new} \leftarrow$ `optimizeMFPT`$(EP_{cur}, q, \hat{M})$;
    $Lik_{new} \leftarrow$ `logLikelihood`(EP_{new}, s);
    if $Lik_{new} > Lik$ **then**
      $EP \leftarrow EP_{new}$;
      $Lik \leftarrow Lik_{new}$;
until $\frac{Lik - Lik_{cur}}{Lik_{cur}} < \epsilon$;
return EP_{cur}

Algorithm 1: POMM Induction by state splitting

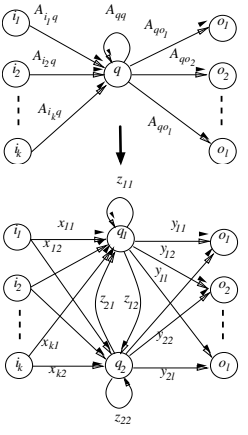

Fig. 2. Splitting of state q

As the target process TP can be considered as a lumped process, each letter of the sample s is associated with a unique state subset of the observable partition κ. All estimates introduced here are related to the state subsets of the target lumped process. The starting point of the induction algorithm is an order 1 MC estimated from the sample. For any pair of letters a, b the transition probability $\hat{A}_{ab}$ is estimated by maximum likelihood by counting how many times a letter a is immediately followed by b in the sample. The stationary distribution of this order 1 MC fits the letter distribution observed in the sample. However, this is not sufficient to reproduce the target dynamics. Hence, the induced model is further required to comply with the MFPT between the blocks of $TP/\!\!/\kappa$, that is between the letters observed in the sample. Given a string s defined on an alphabet Σ, let $\hat{M}$ denote a $|\Sigma| \times |\Sigma|$ matrix where $\hat{M}_{ab}$ is the average number of symbols after an occurrence of a in s to observe the first occurrence of b.

Algorithm 1 describes the induction algorithm. Iterative state splitting in the current model allows one to increase the fit to the MFPT as well as the likelihood of the model with respect to s, while preserving the stationary distribution. After the construction of the initial order 1 MC, $\hat{M}$ is estimated from s and the log-likelihood of the initial model is computed. At each iteration step, every state q of the current model is considered as a candidate for splitting. During the call to `optimizeMFPT`, the considered state q is split into two new states q_1 and q_2 as depicted in Fig. 2. The *input states* $i_1, \ldots, i_k$ and *output states* $o_1, \ldots, o_l$ are those directly connected to q in the current model[8], in which all transition probabilities A are known. The topology after splitting provides additional degrees of freedom in the transition probabilities. The new transition probabilities x, y, z form the variables of an optimization problem, which can be represented by the matrices X $(k \times 2), Y$ $(2 \times l)$ and Z (2×2).

[8] Input and output states are not necessarily distinct.

The objective function to be minimized measures a least squares error with respect to the target MFPT: $W(X,Y,Z) = \sum_{i,j=1,\, i\neq j}^{|\Sigma|} (\hat{M}_{ij} - M_{ij}/\!/\kappa)^2$, where $M_{ij}/\!/\kappa$ is computed according to definition 3. The best model according to the log-likelihood value is selected and the process is iterated till convergence of the log-likelihood function. The optimization problem is non-linear both in the objective function and the constraints. It can be solved using a Sequential Quadratic Programming (SQP) method [3].

5 Conclusion

We propose in this paper a novel approach to the induction of the structure of Hidden Markov Models. The induced model is constructed to fit the dynamics of the target machine, that is to best approximate the stationary distribution and the mean first passage times (MFPT) observed in the sample. HMMs are able to model a class of distributions broader than finite order Markov chains. They are well suited to represent in a compact way long term probabilistic dependencies. Accurate modeling of these dependencies cannot be achieved however through the classical approach of Baum-Welch estimation of a fully connected model. These observations motivate the use of MFPT to guide the search of an appropriate model topology. The proposed induction algorithm relies on non-linear optimization and iterative state splitting from an initial order one Markov chain. Experimental results illustrate the advantages of the proposed approach as compared to Baum-Welch HMM estimation or back-off smoothed Ngrams.

Our future work will include extension of the proposed approach to other classes of models, such as lumped processes of periodic or absorbing Markov chains. The current implementation of our induction algorithm considers all states of the current model as candidates for splitting. More efficient ways of selecting the best state to split at any given step are under study. Applications of the proposed approach to larger datasets will also be considered, typically in the context of language or biological sequence modeling.

Acknowledgment[9]

The authors wish to thank Philippe Delsarte for many fruitful discussions about this work.

References

1. Y. Bengio and P. Frasconi. Diffusion of context and credit information in markovian models. *Journal of Artificial Intelligence Research*, 3:223–244, 1995.
2. J. Callut and P. Dupont. A Markovian approach to the induction of regular string distributions. In *Grammatical Inference: Algorithms and Applications*, number 3264 in Lecture Notes in Artificial Intelligence, pages 77–90, Athens, Greece, 2004. Springer Verlag.

[9] This work is partially supported by the *Fonds pour la formation à la Recherche dans l'Industrie et dans l'Agriculture (F.R.I.A.)* under grant reference F3/5/5-MCF/FC-19271.

3. J. Callut and P. Dupont. Learning hidden markov models to fit long-term dependencies. Technical Report 2005-9, Université catholique de Louvain, July 2005.
4. R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis*. Cambridge University Press, 1998.
5. D. Freitag and A. McCallum. Information extraction with HMMs and shrinkage. In *Proc. of the AAAI-99 Workshop on Machine Learning for Information Extraction*, 1999.
6. D. Freitag and A. McCallum. Information extraction with HMM structures learned by stochastic optimization. In *Proc. of the Seventeenth National Conference on Artificial Intelligence, AAAI*, pages 584–589, 2000.
7. J.G. Kemeny and J.L. Snell. *Finite Markov Chains*. Springer-Verlag, 1983.
8. E. Levin and R. Pieraccini. Planar Hidden Markov modeling: from speech to optical character recognition. In C.L. Giles, S.J. Hanton, and J.D. Cowan, editors, *Advances in Neural Information Processing Systems*, volume 5, pages 731–738. Morgan Kauffman, 1993.
9. M. Ostendorf and H. Singer. HMM topology design using maximum likelihood successive state splitting. *Computer Speech and Language*, 11:17–41, 1997.
10. L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.
11. E. Senata. *Non-negative Matrices and Markov Chains*. Springer-Verlag, 1981.
12. A. Stolcke. *Bayesian Learning of Probabilistic Language Models*. Ph. D. dissertation, University of California, 1994.