A Nonstationary Hidden Markov Model with Approximately Infinitely-Long Time-Dependencies

Sotirios P. Chatzis[1], Dimitrios I. Kosmopoulos[2], and George M. Papadourakis[3]

[1] Cyprus University of Technology, Cyprus
[2] University of Patras, Greece
[3] Technological Educational Institute of Crete, Greece

Abstract. Hidden Markov models (HMMs) are a popular approach for modeling sequential data, typically based on the assumption of a first-order Markov chain. In other words, only one-step back dependencies are modeled which is a rather unrealistic assumption in most applications. In this paper, we propose a method for postulating HMMs with approximately infinitely-long time-dependencies. Our approach considers the whole history of model states in the postulated dependencies, by making use of a recently proposed nonparametric Bayesian method for modeling label sequences with infinitely-long time dependencies, namely the *sequence memoizer.* We manage to derive training and inference algorithms for our model with computational costs identical to simple first-order HMMs, despite its entailed infinitely-long time-dependencies, by employing a mean-field-like approximation. The efficacy of our proposed model is experimentally demonstrated.

1 Introduction

The hidden Markov model (HMM) is increasingly being adopted in applications since it provides a convenient way of modeling observations appearing in a sequential manner and tending to cluster or to alternate between different possible components (subpopulations) [1]. HMMs typically used in sequential data modeling applications postulate first-order Markov chains; i.e., they are based on the assumption that the distribution of a state transition depends only on the current state. Even though this assumption allows for a good trade-off between data modeling effectiveness and computational complexity of the resulting model training and inference algorithms, it is quite clear that the first-order state transition dependencies provide a rather poor model of the actual data dynamics in several real-world applications.

To overcome the shortcomings of first-order HMMs, several refinements have been proposed, introducing HMMs with higher-order dependencies, see, e.g., [2], [3], [4]. A major drawback of the existing approaches is their considerably increased computational costs, which become rather prohibitive as model order increases, especially when it exceeds ten. An effort to ameliorate these issues of higher-order HMMs is presented in [5]. In that work, instead of directly training R-th order HMMs on the data, a method of fast incremental training is used that progressively trains HMMs from first to R-th order. Although this approach is much faster, it is still faced with rapidly increasing computational costs with the model order R.

G. Bebis et al. (Eds.): ISVC 2014, Part II, LNCS 8888, pp. 51–62, 2014.

In this paper we attempt to obtain a non-stationary HMM approximately taking into account the whole history of state transitions. In other words, we derive an R-th order HMM on the limit $R \to \infty$, where the state transition probabilities of the Markov chain vary over time. Formulation of our model is based on introduction of a new type of state transition probabilities for HMMs which are obtained as the predictive densities of a *sequence memoizer* (SM) [6], a nonparametric Bayesian method recently proposed for modeling sequential data with discrete values and dependencies over infinitely-long time-windows. As we show, training and inference for our model can be efficiently reduced to the forward-backward and Viterbi algorithms, respectively, used in the case of first-order HMMs, by utilizing an approximation technique, based on the *mean-field principle* from statistical mechanics [7,8].

The remainder of this paper is organized as follows: In Section 2, we provide the theoretical background of our approach. In Section 3, the proposed nonstationary infinite-order HMM (HMM$^\infty$) model is introduced, and its training and inference algorithms are derived. In Section 4, we consider a number of applications of the HMM$^\infty$ model, with the aim to investigate whether coming up with a computationally tractable way of approximately introducing an infinite-order HMM is of any significance for the sequential data classification algorithm when considering real-life datasets. Finally, in the concluding section, we summarize our work and discuss our future research directives.

2 Theoretical Background

2.1 The Sequence Memoizer

The Hierarchical Pitman-Yor Process. Let us consider a vocabulary $\mathcal{Y}$ comprising K words. For each word $y \in \mathcal{Y}$, let $G(y)$ be the (to be estimated) probability of y; let also $G = [G(y)]_{y \in \mathcal{Y}}$ be the vector of word probabilities. The Pitman-Yor process [9] is a prior that can be imposed over the vector of word probabilities G. We can write

$$G|d, \theta, G_0 \sim \mathrm{PY}(d, \theta, G_0) \tag{1}$$

where $d \in [0, 1)$ is the discount parameter of the process, $\theta > -d$ is its strength parameter, and $G_0 = [G_0(y)]_{y \in \mathcal{Y}}$ is its base distribution, expressing the a priori probability of a word y before any observation, usually set to $G_0(y) = \frac{1}{K}\ \forall y \in \mathcal{Y}$. Now, consider a sequence of words $\{y_t\}_{t=1}^T$ drawn independently and identically (i.i.d.) from G

$$y_t|G \sim G, \quad t = 1, \ldots T \tag{2}$$

Integrating out G, the joint distribution of the variables $\{y_t\}_{t=1}^T$ can be shown to exhibit a clustering effect. Specifically, given the first $T-1$ samples drawn i.i.d. from G, $\{y_t\}_{t=1}^{T-1}$, it can be shown that the new sample y_T is either (a) drawn from the base distribution G_0 with probability $\frac{\theta + dK}{\theta + T - 1}$, or (b) is selected from the existing draws, according to a multinomial allocation, with probabilities proportional to the number of the previous draws with the same allocation.

The above generative procedure produces a sequence of words drawn i.i.d. from G, with G marginalized out. Notice the rich-gets-richer clustering property of the process:

the more words have been assigned to a draw from G_0, the more likely subsequent words will be assigned to the draw. Further, the more we draw from G_0, the more likely a new word will again be assigned to a new draw from G_0. These two effects produce a *power-law distribution* where unique words are observed, most of them rarely [9].

Under the Pitman-Yor process, the drawn words are always considered to be independent of each other. However, in practical applications, it is usually the case that a set of sequential observations are always closely interdependent, thus the i.i.d. assumption is clearly invalid. An nth order hierarchical Pitman-Yor process (HPYP) [10] resolves these issues by postulating a hierarchical model of the form

$$G_{\boldsymbol{u}} \sim \mathrm{PY}(d_{|\boldsymbol{u}|}, \theta_{|\boldsymbol{u}|}, G_{\pi(\boldsymbol{u})}) \tag{3}$$

where $\boldsymbol{u}$ is the context variable, denoting the set of the previously drawn (up to) n words, $G_{\boldsymbol{u}}(y)$ is the probability of the current word taking on the value y given its context $\boldsymbol{u}$, $G_{\boldsymbol{u}} = [G_{\boldsymbol{u}}(y)]_{y \in \mathcal{Y}}$ is the vector of probabilities of all the possible words $y \in \mathcal{Y}$ when the context is $\boldsymbol{u}$, and $\pi(\boldsymbol{u})$ is the prefix of $\boldsymbol{u}$ consisting of all but the latest word in $\boldsymbol{u}$.

Note that the base distribution $G_{\pi(\boldsymbol{u})}$ in (3) is also unknown; for this reason, we recursively place a prior $G_{\pi(\boldsymbol{u})}$ over it using again the general expression (3), but now with parameters $\theta_{\pi(\boldsymbol{u})}$, $d_{\pi(\boldsymbol{u})}$, and $G_{\pi(\pi(\boldsymbol{u}))}$ instead. This recursion is repeated until we get to $G_{\emptyset}$, that is we reach an empty context, on which we place a simple Pitman-Yor process prior of the form

$$G_{\emptyset} \sim \mathrm{PY}(d_0, \theta_0, G_0) \tag{4}$$

where G_0 is a simple base distribution with $G_0(y) = \frac{1}{K}\ \forall y \in \mathcal{Y}$. Inference for the HPYP model is performed using a simple Gibbs sampling scheme, described in [10].

The Sequence Memoizer as an Unbounded-Depth HPYP Model. The sequence memoizer is basically an unbounded-depth HPYP model. Specifically, the sequence memoizer is based on the postulation of an HPYP model of the form (3), with the maximum length n of its context variables $\boldsymbol{u}$ taken as tending to infinity, i.e., $n \to \infty$.

As is obvious, inference in such an unbounded-depth HPYP model might entail a large number of recursions of the form (3), a fact that could possibly give rise to prohibitive computational costs for the model inference algorithms when the length of the drawn sequences increases considerably. To constrain the learning of these latent variables, a special hierarchical Bayesian prior based on Pitman-Yor processes is employed in this work, which promotes sharing of statistical strength between subsequent symbol predictive distributions for equivalent contexts of different lengths [10]. Specifically, in this work, as a way of mitigating these issues, we exploit the following result [11]:

Theorem 1. Consider a single path in a graphical model $G_1 \to G_2 \to G_3$ with G_2 having no children other than G_3. Then, if $G_2|G_1 \sim \mathrm{PY}(d_1, 0, G_1)$ and $G_3|G_2 \sim \mathrm{PY}(d_2, 0, G_2)$, it holds $G_3|G_1 \sim \mathrm{PY}(d_1 d_2, 0, G_1)$ with G_2 marginalized out.

Based on Theorem 1, the computational complexity of the SM model inference algorithms, *which are otherwise identical to those of the HPYP model*, are considerably reduced in cases of long drawn sequences, without compromises in the model's efficacy. In this paper, we perform inference using the Gibbs sampler proposed in [10], as described in the previous section.

At test time t, inference consists in using the sequence memoizer to compute the probability $q(y_t|y_{<t})$ of the modeled variable being equal to the symbol y_t, given a context $\boldsymbol{u} = \{y_\tau\}_{\tau=1}^{t-1}$. Similar to the discussions of the previous section, the predictive probability of the sequence memoizer is taken as the posterior expectation of the distribution $G_{\boldsymbol{u}}(y_t)$ of the current word taking on the value y_t, given its context $\boldsymbol{u}$, i.e.

$$q(y_t|y_{<t}) \triangleq \mathbb{E}\left[G_{\boldsymbol{u}}(y_t)\right] \tag{5}$$

where the distribution of $G_{\boldsymbol{u}}(y_t)$ is given by (3), and $\boldsymbol{u} \triangleq \{y_\tau\}_{\tau=1}^{t-1}$.

2.2 The Mean-Field Principle

The mean-field principle is originally a method of approximation for the computation of the mean of a Markov random field. It comes from statistical mechanics (e.g. [12]), where it has been used as an analysis tool to study phase transition phenomena. More recently, it has been used in computer vision applications (e.g. [13,14]), graphical models (e.g. [15], and references therein) and other areas (e.g. [16]). The basic idea of the mean-field principle consists in neglecting the fluctuations of the variables interacting with a considered variable. As a result of this assumption, the resulting system behaves as one composed of independent variables for which computation gets tractable.

More specifically, let us consider a set of interdependent variables $\{y_t\}_{t=1}^{T}$ that define a Markov random field with a specified neighborhood system. For example, a neighborhood system of first-order sequential nature may be considered, in which case the postulated Markov random field reduces to a first-order Markov chain. Under the mean-field principle, the joint distribution $p(\{y_t\}_{t=1}^{T})$ is approximated by the product

$$p(\{y_t\}_{t=1}^{T}) \approx \prod_{t=1}^{T} \hat{p}_t(y_t) \tag{6}$$

Here, the $\hat{p}_t(y_t)$ is an approximation of the marginal distribution $p(y_t)$ of the field at the site (e.g., time point) t. This latter quantity is expressed under the mean-field principle in the following conditional form $\hat{p}_t(y_t) \approx p(y_t|\{\hat{y}_\tau\}_{\tau\in\mathcal{N}(t)})$ where $\hat{y}_\tau$ is the expected value of y_τ, i.e., $\hat{y}_\tau = \mathbb{E}[y_\tau]$ and $\mathcal{N}(t)$ is the set of neighbors of site t. E.g., in the case of an infinite-order sequential nature neighborhood system, we have $\hat{p}_t(y_t) \approx p(y_t|\hat{y}_{<t})$.

More generally, we talk about *mean field-like approximations* when the value of a variable observed at a site t is considered independent of the fluctuations of the values at other sites in its neighborhood, which are all set to constants ***(not necessarily their means),*** independently of the value at site t. This family of approximations allows for a considerable increase in computational efficiency. It has the drawback though that, while the mean-field approach has been theoretically formulated using a variational framework, mean-field-like approximations have not yet been shown to have such a theoretically founded motivation [7].

3 Proposed Approach

Let us first introduce some notation. Let us suppose an N-state HMM where the hidden emission density of each state is modeled by a K-component finite mixture model.

The postulated HMM comprises the set of parameters $\Theta = \{\boldsymbol{W}, \boldsymbol{\Psi}\}$, where $\boldsymbol{W} = (w_{ij})_{i,j=1}^{N,K}$ are the mixing weights of the component distributions of the mixture models used as the emission probabilities of the HMM states, and $\boldsymbol{\Psi}$ is the set of parameters of the mixture components of the state emission probabilities.

Let $X = \{\boldsymbol{x}_t\}_{t=1}^{T}$ be an observed data sequence, with $\boldsymbol{x}_t \in \mathcal{X} \subseteq \mathbb{R}^d$, modeled by the postulated HMM. The latent (unobserved) data associated with this sequence comprise the corresponding state sequence $S = \{s_t\}_{t=1}^{T}$, where $s_t = 1, \ldots, N$ is the indicator of the state the tth observation is emitted from, and the sequence of the corresponding mixture component indicators $L = \{l_t\}_{t=1}^{T}$, where $l_t = 1, \ldots, K$ indicates the mixture component density that generated the tth observation.

As already discussed, in this paper we aim to introduce an (approximately) infinite-order non-stationary HMM. That is, we seek a model with likelihood function

$$p(X|\Theta) = \sum_{S,L} \pi_{s_1} \left[\prod_{t=2}^{T} p(s_t|s_{<t}) \right] \left[\prod_{t=1}^{T} w_{s_t l_t} p(\boldsymbol{x}_t|\boldsymbol{\Psi}_{s_t l_t}) \right] \tag{7}$$

where we denote $s_{<t} = \{s_\tau\}_{\tau=1}^{t-1}$, and π_i are the initial state probabilities.

As we observe, key components of the sought model comprise the considered state transition probabilities $p(s_t|s_{<t})$ which take into account the whole history of past states at any time point, and vary with time, thus giving rise to a non-stationary nature for our model. Based on the discussions of Section 2.2, the desired form of the state transition probabilities of our model can be obtained by modeling them as the predictive densities of a pre-trained sequence memoizer model. Apparently, such a postulated sequence memoizer would also yield the initial state probabilities of the model; thus, we eventually obtain

$$p(X|\Theta) = \sum_{S,L} \left[\prod_{t=1}^{T} q(s_t|s_{<t}) \right] \left[\prod_{t=1}^{T} w_{s_t l_t} p(\boldsymbol{x}_t|\boldsymbol{\Psi}_{s_t l_t}) \right] \tag{8}$$

for the likelihood function of the postulated model, where $q(s_t|s_{<t})$ is given by (5).

Definition 1. We define as the infinite-order HMM (HMM$^\infty$) model an HMM the Markov chain (initial state and state transition) probabilities of which take under consideration the whole history of model states, by being modeled as the predictive probabilities of a postulated sequence memoizer model.

3.1 Model Training

Consider a training sequence of length T, $X = \{\boldsymbol{x}_t\}_{t=1}^{T}$. Training for the proposed HMM$^\infty$ model using the given sequence is performed in two phases.

First Phase. In the first phase, we need to train the sequence memoizer used to obtain the Markov chain probabilities of our model. To effect this procedure, we must obtain training sequences of the latent state variables s_t of our model. For this purpose, we resort to the following solution: We first train a simple first-order HMM with similar finite mixture state emissions, $\mathcal{M}_0$, on the training dataset X, using the EM algorithm [17]. Subsequently, we apply the Viterbi algorithm for the model $\mathcal{M}_0$ to obtain a segmentation of the training sequence X, i.e. compute the sequence of optimal states $\hat{S} = \{\hat{s}_t\}_{t=1}^{T}$. Using the estimated state sequence $\hat{S}$, we can perform ”training” of the postulated sequence memoizer (see Section 2.2), using the Gibbs sampler of [10].

Second Phase. In the second phase of our training algorithm, we proceed to estimation of the rest of the model parameters, that is the parameters of the state emission densities of the model. We employ maximization of the likelihood of the model, given by (11), considering that the sets X, S and L comprise our complete data. Nevertheless, from (11), it becomes apparent that the complete-data log-likelihood of our model is not computationally tractable, as it entails summation over all possible $s_{<t}$ configurations at each time point t. Therefore, an approximation is needed.

We resort to a mean-field-like approximation, eventually yielding

$$\begin{aligned}\mathcal{L} \approx & \sum_{t=1}^{T}\sum_{j=1}^{N}\sum_{i=1}^{N} p(s_t = j, s_{t-1} = i|X) \times \log q(s_t = j|s_{t-1} = i; \hat{s}_{<t-1}) \\ & + \sum_{j=1}^{N}\sum_{k=1}^{K} p(l_t = k, s_t = j|X) \log\big[w_{jk} p(\boldsymbol{x}_t|\boldsymbol{\Psi}_{jk})\big]\end{aligned} \tag{9}$$

In this expression, we essentially assume that in each term $q(s_t = j|s_{<t})$, the variables $\{s_\tau\}_{\tau=1}^{t-2}$ do not fluctuate with s_t and s_{t-1}, but, rather, they are constants equal to the known estimates $\{\hat{s}_\tau\}_{\tau=1}^{t-2}$, obtained at the first phase of the training algorithm of our model. This is in essence a mean-field-like approximation of the complete-data log-likelihood of our model. Note that, under such an approximation, we don't obtain a bound to the log-marginal, unlike a standard mean-field approximation. Therefore, our approximation does not allow for the training algorithm to optimize the true model objective function. However, the major advantage of our approach is that it allows for approximating the partition function of the infinite-order HMM$^\infty$ model using an efficient algorithm, similar to the method used in the case of first-order HMMs.

Indeed, based on the above approximation, the forward probabilities in the case of the HMM$^\infty$ model yield

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\, q(s_{t+1} = j|s_t = i; \hat{s}_{<t})\right] \times \sum_{k=1}^{K} w_{jk} p(\boldsymbol{x}_{t+1}|\boldsymbol{\Psi}_{jk}) \tag{10}$$

with initialization similar to first-order HMMs [18]. Similar, the backward probabilities of our model yield

$$\begin{aligned}\beta_t(i) = & \sum_{j=1}^{N} q(s_{t+1} = j|s_t = i; \hat{s}_{<t}) \\ & \times \sum_{k=1}^{K} w_{jk} p(\boldsymbol{x}_{t+1}|\boldsymbol{\Psi}_{jk})\, \beta_{t+1}(j)\end{aligned} \tag{11}$$

with initialization similar to first-order HMMs [18].

Note that the employed mean-field-like approximation does not constitute an assumption of the HMM$^\infty$ model itself, but is only applied to obtain a computationally tractable expression for its complete-data log-likelihood. In other words, the Markov chain probabilities of the HMM$^\infty$ model are still computed using the *whole history of state labels* $\{s_\tau\}_{\tau=1}^{t-1}$, a computation made possible by exploiting the sequence memoizer. Hence, the model itself continues to postulate infinite-order state transitions, taking into account the *whole history of state labels* $\{s_\tau\}_{\tau=1}^{t-1}$, with no truncations imposed in that respect. The truncation consists in only truncating **the fluctuation** of the values $\{s_\tau\}_{\tau=1}^{t-2}$, by setting them to a constant appropriately obtained in the first phase of the model training algorithm.

Based on these results, the state posterior probabilities $p(s_t|X)$ of our model yield

$$p(s_t = j|X) = \frac{\alpha_t\left(j\right)\beta_t\left(j\right)}{\sum_{i=1}^N \alpha_t\left(i\right)\beta_t\left(i\right)} \tag{12}$$

Finally, derivation of our model training algorithm completes with the computation of the posterior probabilities of the mixture components, $p(l_t = k, s_t = j|X)$, and of the estimates of the mixture model parameters $\boldsymbol{\Psi}$ and $\boldsymbol{W}$. These expressions are identical to the ones that hold for first-order HMMs with similar emission distributions, thus we omit them for brevity (see, e.g., [19]).

3.2 Inference Algorithm

Inference in the context of the HMM$^\infty$ model comprises sequence labeling, and sequence probability estimation.

Sequence Labeling. Similar to first-order HMMs, sequence labeling using the HMM$^\infty$ model can be performed by application of the Viterbi algorithm [17], which here uses the cost function

$$\xi_t(j) \triangleq \max_{s_{<t}} \left\{ q(s_t = j|s_{<t}) \sum_{k=1}^K w_{jk} p\left(\boldsymbol{x}_t|\boldsymbol{\Psi}_{jk}\right) \right\} \xi_{t-1}(s_{t-1}) \tag{13}$$

and employs the recursion

$$\hat{s}_t = \underset{1\leq i\leq K}{\operatorname{argmax}} \left\{ \xi_t(i) \right\} \tag{14}$$

The cost function (16) results in a dynamic programming problem which entails a large (theoretically infinite) number of variables over which $\xi_t(i)$ gets optimized. As such, the incurred computational costs might become prohibitive in most real-world scenarios. For this reason, we resort again to a mean-field-like approximation.

Specifically, we propose the following approximation: Let us begin with the second time step, $t = 2$. The cost function $\xi_t(i)$ reads

$$\xi_2(j) \triangleq \max_{s_1} \left\{ q(s_2 = j|s_1) \sum_{k=1}^K w_{jk} p\left(\boldsymbol{x}_2|\boldsymbol{\Psi}_{jk}\right) \right\} \xi_1(s_1) \tag{15}$$

Let us now continue to the next time-step, $t = 3$. The cost function $\xi_t(i)$ now reads

$$\xi_3(j) \triangleq \max_{s_2,s_1}\left\{ q(s_3 = j|s_2, s_1) \sum_{k=1}^{K} w_{jk} p\left(\boldsymbol{x}_3|\boldsymbol{\Psi}_{jk}\right) \right\} \xi_2(s_2) \tag{16}$$

At this point, we make the following key-hypothesis: We assume that, in $\xi_3(i)$, the variable s_1 does not fluctuate with s_2 and s_3, but, instead, it takes on a constant ("optimal") value $\hat{s}_1$. This way, we eventually yield

$$\xi_3(j) \triangleq \max_{s_2}\left\{ q(s_3 = j|s_2; \hat{s}_1) \sum_{k=1}^{K} w_{jk} p\left(\boldsymbol{x}_3|\boldsymbol{\Psi}_{jk}\right) \right\} \xi_2(s_2) \tag{17}$$

This assumption is in essence a mean-field-like approximation of $\xi_3(i)$.

Generalizing this procedure, we reduce our dynamic programming problem to a simpler one with bounded worst-case computational costs, where the cost function reads

$$\xi_t(j) \approx \max_{1\leq i\leq N}\left\{ q(s_t = j|s_{t-1} = i; \hat{s}_{<t-1}) \sum_{k=1}^{K} w_{jk} p\left(\boldsymbol{x}_t|\boldsymbol{\Psi}_{jk}\right) \right\} \\ \times \xi_{t-1}(i) \tag{18}$$

with the same backwards recursion. This construction gives, in turn, rise to another issue: what is the appropriate selection of the values $\hat{s}_{<t-1}$? Following the literature (e.g., [20,21,15,14]), the values $\{\hat{s}_\tau\}_{\tau=1}^{t-2}$ may be selected as the values of $\{s_\tau\}_{\tau=1}^{t-2}$ that optimize some criterion. Here, the values of $\{\hat{s}_\tau\}_{\tau=1}^{t-2}$ are obtained as follows:

1. First, we postulate a first-order HMM for the same problem, trained on the same data as the considered HMM$^\infty$ model. We use this model to obtain an initial optimal set $\hat{S} = \{\hat{s}_t\}_{t=1}^{T}$ using Viterbi algorithm.
2. Using this initial optimizer $\hat{S}$, we run the dynamic programing recursions (21) of the HMM$^\infty$ inference (Viterbi-like) algorithm. This way, a new sequence segmentation is derived. This procedure may be repeated for a number of iterations.

At this point, we would like to emphasize that, again, the mean-field-like approximation is not applied to the core assumptions of the HMM$^\infty$ itself. The proposed dynamic programming algorithm for HMM$^\infty$ model inference entails "full" state transition probabilities, that is state transition probabilities taking into account the *whole history of state labels* $\{s_\tau\}_{\tau=1}^{t-1}$, with no truncations imposed in that respect. Therefore, the application of the mean-field-like approximation does not affect the infinite-order nature of the model; it only consists in truncating the fluctuations of $\{s_\tau\}_{\tau=1}^{t-2}$ with the s_t and s_{t-1} when computing the cost functions $\xi_t(j)$.

Sequence Classification. Finally, computation of the probability of a sequence with respect to a trained HMM$^\infty$ model can be performed by summation of the forward probabilities at some time-point, similar to first-order HMMs.

4 Experiments

To evaluate the efficacy of our approach, we considered three datasets from different domains dealing with sequence classification. The workflow recognition [22], the RGBD-HUDAACT [23] and the speaker identification [24].

We compared the standard HMM to our method, which uses the whole history in order to quantify the merits of our approach. Both methods build models from which we can draw samples. We tested the representation capabilities of those models in time-series classification, which is an important domain for the computer vision and multimedia communities. We initialized the two methods using exactly the same initial values for $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \mathbf{A}, \mathbf{W}, \boldsymbol{\Psi}\}$, so that the comparison was fair. The best number of states and state components was experimentally decided based on classification accuracy.

Our source code was developed in MATLAB. The sequence memoizers used in our model are obtained by Gibbs sampling, as suggested in [6]. We used 100000 samples, as suggested in [10]. The implementation of this Gibbs sampler was taken from the sequence memoizer software available at: `http://www.sequencememoizer.com/`

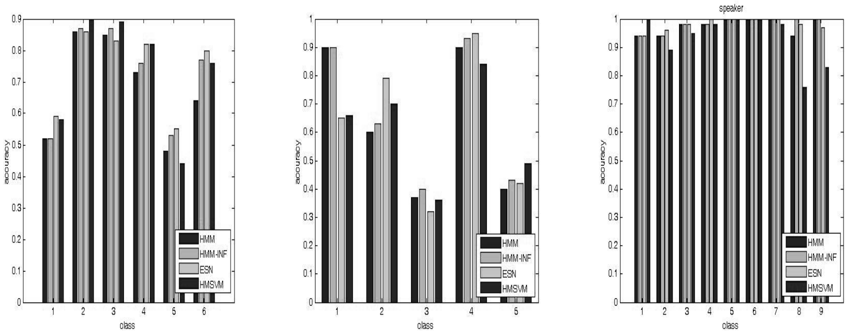
Furthermore, we have compared our method to two state-of-the-art methodologies for time series classification, which do not make any Markovian assumption as well, but are of different rationale: the Echo State Network (ESN) [25] and the Hidden Markov Support Vector Machine (HMSVM) [26]. For the ESN we have used the Matlab toolbox provided by the authors [25]. More specifically, we used a linear reservoir. We omitted a number of frames at the beginning of each sequence to account for the initialization effect of the ESN and we performed median filtering to reduce jitter at the output. For the HMSVM we used the library provided by the authors [26]. We employed for our experiments the linear kernel. Both methods assign labels to each frame, so the ”winner” class is the one to which the most frames were assigned. Below we detail our experimental methods for each experiment. For both the ESN and the HMSVM we normalized the input based on the mean and standard deviation to assist the optimization.

Workflow Recognition [22]. This dataset includes color image sequences acquired in an industrial environment and the goal is to recognize tasks that consist a *visual workflow*. Each frame was modeled by the Zernike moments, of the pixel change history images generated by the foreground objects (humans), up to order six. That yielded feature vectors of dimension 31. We used the camera 1, depicting the first workflow, which involved six tasks. We applied 50-fold cross validation. In each fold we selected randomly five scenarios for training and the rest 15 for testing. We used a diagonal covariance matrix for the HMMs, to avoid overfitting. For the ESN we used 500 plain nodes, which was efficient, small enough to avoid overfitting and effective. The number of the output nodes, was equal to the number of classes and we used spectral radius 0.60, input scaling 0.3 and smoothing of noise level 0.0003 for optimal results. Regarding the HMSVM the C parameter was set to 0.5 and for the rest we used the default values.

RGBD-HUDAACT [23]. The second application used depth images to classify *human actions* in an assistive living environment , and namely the following: ”get up from bed”, ”go to bed”, ”sit down”, ”eat meal”, and ”drink water”. Each frame was represented as a moment-based vector of 31 elements, similarly to the previous experiment, which

Table 1. Experimental results for the datasets Workflow Recognition [22], RGBD-HUDAACT [23] and Speaker Identification [24]

Dataset Method	WR	RGBD-HUDAACT	Speaker Ident
HMM	68.49	63.33	97.56
HMM-inf	72.30	66.00	98.10
ESN	75.57	62.70	99.45
HMSVM	73.73	61.12	94.23



(a) The Workflow Recognition dataset

(b) The RGBD-HUDAACT dataset

(c) The Speaker Identification dataset

Fig. 1. Experimental per class for the three datasets. The vertical axis is the accuracy (%) and the horizontal the classes

in this case encoded the backward motion history image (decrease of depth). Again we applied 50-fold cross validation using 35 samples of each action. In each fold we selected randomly five scenarios for training and the rest 30 for testing. For the ESN we used the same architecture and parameters with five output nodes this time. For the HMSVM the C parameter was set to 20.

Speaker Identification [24]. To verify the applicability in other domains (which however can be combined via a fusion framework with vision applications), we finally considered a *text-dependent speaker identification* task, using the Japanese Vowels Data Set. The pass-phrase used for speaker identification purposes comprised two Japanese vowels, /ae/, successively uttered by nine male speakers. For each utterance, a 12-degree linear prediction analysis was applied to obtain a discrete-time series with 12 LPC cepstrum coefficients. The dataset involved nine speakers, who had to be recognized. We used for training and testing the designated sets provided by the authors. For the ESN we used a reservoir of 200 nodes with nine output nodes and same parameters. For the HMSVM we used $C = 0.5$.

The accuracies of all methods of in all our experiments are summarized in Table 1. More detailed results concerning each class separately are given in Figure 1. The results are given based on the average of all folds. What is remarkable here, is that in all the experiments there is a statistically consistent benefit in using the proposed method as opposed to using the standard HMM. As presented in Figure 1 the per class accuracy is always higher on average. When we compare our method to the baseline methods of ESN and HMSVM we notice that their performance is comparable. Here we present the best results after experimentation with the parameters. Given the rather low differences we cannot claim that our method is generally better, or not, compared to the linear versions of the ESN and the HMSVM. The result depends on a variety of parameters, which are not common to the different methods. Therefore we cannot guarantee a really fair comparison, unlike in the case of comparing the standard HMM to our method.

The fact that our performance is comparable to some of the state of the art methods is also remarkable. We proposed a method for estimating a generative model, which is generally more appropriate for simulation applications (via model sampling), than for classification tasks. However, the ESN and the HMSVM are discriminative methods and thus usable in classification applications, but their models cannot be used for sampling. The proposed method can impact all generative HMM-based methods that are employed for time-series modeling. Furthermore, the discriminative methods that utilize the HMMs, such as those that optimize entropy criteria, e.g., [27] or methods that optimize the margin between classes, e.g., [28] can benefit from our approach.

5 Conclusions

We presented the hidden Markov model with infinitely long time dependencies thus effectively by-passing the Markovian assumption. We provided algorithms for model training and inference, and evaluated their efficacy in real-world applications. We tackled a non-tractable problem by employing a mean-field-like approximation. As we showed, our method outperforms the standard HMM in classification, thus it enhances its representation capability. Furthermore, it has comparable performance to state-of-the-art methods which have been trained for discriminative tasks. Currently we fit the model to the data without optimizing the discrimination capability of the model. We plan to extend our method to models trainable explicitly for discriminative tasks.

References

1. Cappé, O., Moulines, E., Rydén, T.: Inference in Hidden Markov Models. Springer, New York (2005)
2. Mari, J., Fohr, D., Junqua, J.: A second-order HMM for high-performance word and phoneme-based continuous speech recognition. In: Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 435–438 (1996)
3. Mari, J.F., Haton, J.P., Kriouile, A.: Automatic word recognition based on second-order hidden Markov models. IEEE Trans. Speech Audio Process. 5, 22–25 (1997)
4. Aycard, O., Mari, J.F., Washington, R.: Learning to automatically detect features for mobile robots using second-order HMMs. Int. J. Adv. Robotic Syst. 1, 231–245 (2004)

5. Engelbrecht, H., du Preez, J.: Efficient backward decoding of high-order hidden markov models. Pattern Recognition 43, 99–112 (2010)
6. Wood, F., Gasthaus, J., Archambeau, C., James, L., Teh, Y.W.: The sequence memoizer. Communications of the ACM 54, 91–98 (2011)
7. Celeux, G., Forbes, F., Peyrard, N.: EM procedures using mean field-like approximations for Markov model-based image segmentation. Patt. Recogn. 36, 131–144 (2003)
8. Zhang, J.: The mean field theory in EM procedures for Markov random fields. IEEE Transactions on Image Processing 2, 27–40 (1993)
9. Pitman, J., Yor, M.: The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. Annals of Probability 25, 855–900 (1997)
10. Teh, Y.W.: A hierarchical Bayesian language model based on Pitman-Yor processes. In: Proc. Assoc. for Comp. Linguistics, pp. 985–992 (2006)
11. Wood, F., Archambeau, C., Gasthaus, J., James, L.F., Teh, Y.: A stochastic memoizer for sequence data. In: Proc. Int. Conference on Machine Learning (ICML) (2009)
12. Chandler, D.: Introduction to Modern Statistical Mechanics. Oxford Univ. Press (1987)
13. Geiger, D., Girosi, F.: Parallel and deterministic algorithms from MRFs: surface reconstruction. IEEE Trans. Pattern Anal. Mach. Intell. 13, 401–412 (1991)
14. Zerubia, J., Chellappa, R.: Mean field approximation using compound Gauss-Markov random field for edge detection and image restoration. In: Proc. ICASSP, pp. 2193–2196 (1990)
15. Jaakkola, T., Jordan, M.: Improving the mean field approximation via the use of mixture distributions. In: Jordan, M. (ed.) Learning in Graphical Models, pp. 163–173. Kluwer (1998)
16. Hofmann, T., Buhmann, J.: Pairwise data clustering by deterministic annealing. IEEE Trans. Pattern Anal. Mach. Intell. 19, 1–14 (1997)
17. Rabiner, L.: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77, 245–255 (1989)
18. Rabiner, L.: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77, 245–255 (1989)
19. McLachlan, G., Peel, D.: Finite Mixture Models. Wiley Ser. Probability and Statistics (2000)
20. Chatzis, S.P., Varvarigou, T.A.: A fuzzy clustering approach toward hidden Markov random field models for enhanced spatially constrained image segmentation. IEEE Trans. on Fuzzy Systems 16, 1351–1361 (2008)
21. Chatzis, S.P., Tsechpenakis, G.: The infinite hidden Markov random field model. IEEE Transactions on Neural Networks 21, 1004–1014 (2010)
22. Voulodimos, A., et al.: A threefold dataset for activity and workflow recognition in complex industrial environments. IEEE Multimedia 19, 42–52 (2012)
23. Ni, B., Wang, G., Moulin, P.: RGBD-HuDaAct: A color-depth video database for human daily activity recognition. In: ICCV Workshops, pp. 1147–1153 (2011)
24. Kudo, M., Toyama, J., Shimbo, M.: Multidimensional curve classification using passing through regions. Pattern Recogn. Lett. 20, 1103–1111 (1999)
25. Jaeger, H., Maass, W., Principe, J.: Special issue on echo state networks and liquid state machines. Neural Networks 20, 287 (2007)
26. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. J. Mach. Learn. Res. 6, 1453–1484 (2005)
27. McCallum, A., Freitag, D., Pereira, F.C.N.: Maximum entropy markov models for information extraction and segmentation. In: Proc. of the Int. Conf. on Mach. Learning, ICML 2000, pp. 591–598 (2000)
28. Sha, F., Saul, L.K.: Large margin hidden markov models for automatic speech recognition. In: Advances in Neural Information Processing Systems 19, pp. 1249–1256. MIT Press (2007)