Preprint of paper published in the Pattern Recognition Journal, vol 45, no 11, 3985-3996, Nov 2012.

A Reservoir-Driven Non-Stationary Hidden Markov Model

Sotirios P. Chatzis and Yiannis Demiris

Department of Electrical and Electronic Engineering
Imperial College London
Exhibition Road, South Kensington Campus, SW7 2BT

Abstract

In this work, we propose a novel approach towards sequential data modeling that leverages the strengths of hidden Markov models and echo-state networks (ESNs) in the context of nonparametric Bayesian inference approaches. We introduce a *non-stationary* hidden Markov model, the time-dependent state transition probabilities of which are driven by a high-dimensional signal that encodes the whole history of the modeled observations, namely the state vector of a postulated observations-driven ESN reservoir. We derive an efficient inference algorithm for our model under the variational Bayesian paradigm, and we examine the efficacy of our approach considering a number of sequential data modeling applications.

Keywords: hidden Markov model; Dirichlet process; reservoir

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]. Specifically, HMMs with continuous observation densities have been used in a wide spectrum of applications in ecology, encryption, image understanding, speech recognition, and machine vision applications [2].

Hidden Markov models are based on the assumption that each data point in a sequence of observations is generated by a latent (hidden) model state. Usually, a first-order hidden Markov chain is postulated, thus limiting the considered state dependencies only to successive observations. Longer dependencies between data over time may be also considered, by postulating higher-order hidden Markov chains; however, such a selection may also give rise to an overwhelming increase in the computational complexity of the model, rendering it unattractive in most practical applications [2].

Echo-state networks are a groundbreaking and surprisingly efficient network structure for recurrent neural network (RNN) training [3, 4, 5, 6]. ESNs avoid

the shortcomings of typical, gradient-descent-based RNN training by randomly creating a recurrent neural network which remains unchanged during training. This RNN is called the *reservoir*. It is passively excited by the input signal and maintains in its state a nonlinear transformation of the input history. Indeed, the function of the reservoir in ESNs can be compared to that of the kernel function in kernel machine approaches (e.g., support vector machines [7], relevance vector machines [8], and their variants) [9]: input signals drive the nonlinear reservoir and produce a high-dimensional dynamical "echo response," which is used as a non-orthogonal basis to reconstruct the desired outputs. The obtained reservoir state values of the ESN networks capture long-term dependencies between the modeled data, by encoding the history of the observed values of their driving signals.

Motivated by these advances, in this paper we exploit the merits of ESN reservoirs in order to provide a novel *non-stationary HMM* formulation for sequential data modeling. The proposed model is based on the fundamental assumption that the probabilities of HMM state transition are not stationary, but instead they depend on time, and specifically on the whole history of observed data, as encoded in the state vectors of an echo-state network reservoir. That is, an HMM with reservoir-driven non-stationary state transition probabilities is essentially introduced. The main advantage of the proposed approach is that it allows to model longer temporal dependencies compared to conventional HMMs, by introducing the dynamic information captured from the postulated ESN reservoirs into the state transition mechanics of the latent Markov chain. Derivation of our model is conducted under a nonparametric Bayesian approach to allow for automatic data-driven determination of the appropriate model size.

Nonparametric Bayesian modeling techniques, especially Dirichlet process (DP) prior-based models, have become very popular in statistics over the last few years, for performing nonparametric density estimation [10, 11, 12]. Briefly, a realization of a DP prior-based model can be seen as an infinite mixture of distributions with given parametric shape (e.g., Gaussian, HMM, etc.). This theory is based on the observation that an infinite number of component distributions in an ordinary finite mixture model tends on the limit to a Dirichlet process prior [11, 13]. Exploitation of the merits of nonparametric Bayesian statistics has allowed for coming up with computationally efficient formulations of HMMs that allow for doing inference over the number of model states, thus obviating the need of model order selection. For example, in [14], an infinite HMM was proposed, based on the introduction of a hierarchical Dirichlet process (HDP) prior over the model state transition probabilities. In [15], hierarchical stick-breaking priors were imposed over the model state transition probabilities instead of the HDP, to allow for more efficient model inference by means of a truncated variational Bayesian inference technique.

As we shall discuss in the following sections, the formulation of our model consists in introduction of a joint stick-breaking and ESN reservoir-driven prior over the model state transition probabilities, which gives rise to an elaborate reservoir-driven HMM in the context of a nonparametric Bayesian inference setting. We derive an efficient truncated algorithm for model inference based

on the variational Bayesian paradigm, and we experimentally demonstrate the efficacy of our approach. We dub the resulting model the echo-state stick-breaking HMM (ES-SB-HMM).

Indeed, our approach towards non-stationary HMMs with observation-driven state transitions is related to the approach taken by conditional random fields (CRFs). A CRF is simply a log-linear model representing the conditional distribution of the model states given the observed data with an associated graphical structure. In other words, they explicitly model data-driven transitions. Because the model is conditional, dependencies among the observed variables do not need to be explicitly represented, affording the use of rich, global features of the input [16]. A drawback of CRFs is that they cannot be used for classification of whole sequences into a number of learned classes. The hidden CRF (HCRF) [17] is a discriminative model that caters to these needs, by modeling the class labels of whole sequences of observations conditional on the observed sequential data, considering that each observation is also assigned a latent label variable which is optimized as model parameter.

The remainder of this work is organized as follows: In Section 2, we provide a brief overview of echo-state networks and the DP prior. In Section 3, we introduce the ES-SB-HMM and derive an efficient truncated variational Bayesian algorithm for model inference. In Section 4, we evaluate our approach considering a number of applications from diverse domains, using benchmark datasets, and we compare it to CRFs, HCRFs, and SB-HMMs. Finally, in the last section, we summarize our results and draw our conclusions.

2. Theoretical Background

2.1. Echo-State Networks

As already discussed, the basic component of ESNs is a discrete-time RNN, called the reservoir. Let us consider an ESN comprising N reservoir neurons. ESN function is described by the following reservoir state update equation:

$$\begin{aligned} \boldsymbol{\varsigma}_{t+1} =& (1-\gamma) h(\boldsymbol{W}\boldsymbol{\varsigma}_t + \boldsymbol{W}_{in}\boldsymbol{x}_{t+1}) \\ &+ \gamma \boldsymbol{\varsigma}_t \end{aligned} \tag{1}$$

where $\boldsymbol{\varsigma}_t$ is the reservoir state at time t (an N-dimensional vector of real numbers), $\boldsymbol{W}$ is the reservoir weight matrix, that is the matrix of the weights of the synaptic connections between the reservoir neurons, $\boldsymbol{x}_t$ is the observed signal fed to the network at time t, $\gamma \geqslant 0$ is the *retainment rate* of the reservoir (with $\gamma > 0$ if leaky integrator neurons are considered), $\boldsymbol{W}_{in}$ are the weights of $\boldsymbol{x}_t$, and $h(\cdot)$ is the activation function of the reservoir. All the weight matrices to the reservoir $(\boldsymbol{W}, \boldsymbol{W}_{in})$ are initialized randomly. The initial state of the reservoir is usually set to zero, $\boldsymbol{\varsigma}_0 = \mathbf{0}$.

An extensively studied subject in the field of ESNs concerns the introduction of appropriate *goodness* measures of the reservoir structure. Indeed, the classical feature that reservoirs should possess is the echo-state property. This property essentially states that the effect of a previous reservoir state and a

previous input on a future state should vanish gradually as time passes, and not persist or even get amplified. However, for most practical purposes, the echo-state property can be easily satisfied by merely ensuring that the *reservoir weight matrix* $\boldsymbol{W}$ is *contractive*, i.e., by scaling the reservoir weight matrix so that its *spectral radius* $\rho(\boldsymbol{W})$ (that is, its largest absolute eigenvalue) is less than one [18]. Indeed, this condition has been proved to be sufficient in practical applications of ESNs; nevertheless, various researchers have also provided more rigorous global asymptotic stability conditions, providing better theoretical guarantees for ESNs to perform well on a physical system (see, e.g., [19]). It has been shown that the maximum possible short-term memory length of an ESN reservoir comprising N neurons is N time points [3].

2.2. Dirichlet process models

Dirichlet process (DP) models were first introduced by Ferguson [20]. A DP is characterized by a base distribution G_0 and a positive scalar α, usually referred to as the innovation parameter, and is denoted as $\mathrm{DP}(G_0, \alpha)$. Essentially, a DP is a distribution placed over a distribution. Let us suppose we randomly draw a sample distribution G from a DP, and, subsequently, we independently draw N random variables $\{\Theta_n^*\}_{n=1}^N$ from G:

$$G|\{G_0, \alpha\} \sim \mathrm{DP}(G_0, \alpha) \tag{2}$$

$$\Theta_n^*|G \sim G, \quad n = 1, \ldots N \tag{3}$$

Integrating out G, the joint distribution of the variables $\{\Theta_n^*\}_{n=1}^N$ can be shown to exhibit a clustering effect. Specifically, given the first $N-1$ samples of G, $\{\Theta_n^*\}_{n=1}^{N-1}$, it can be shown that a new sample Θ_N^* is either (a) drawn from the base distribution G_0 with probability $\frac{\alpha}{\alpha+N-1}$, 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 [21]. Let $\{\Theta_c\}_{c=1}^K$ be the set of distinct values taken by the variables $\{\Theta_n^*\}_{n=1}^{N-1}$. Denoting as f_c^{N-1} the number of values in $\{\Theta_n^*\}_{n=1}^{N-1}$ that equal to Θ_c, the distribution of Θ_N^* given $\{\Theta_n^*\}_{n=1}^{N-1}$ can be shown to be of the form [21]

$$p(\Theta_N^*|\{\Theta_n^*\}_{n=1}^{N-1}, G_0, \alpha) = \frac{\alpha}{\alpha+N-1} G_0 + \sum_{c=1}^{K} \frac{f_c^{N-1}}{\alpha+N-1} \delta_{\Theta_c} \tag{4}$$

where δ_{Θ_c} denotes the distribution concentrated at a single point Θ_c. These results illustrate two key properties of the DP scheme. First, the innovation parameter α plays a key-role in determining the number of distinct parameter values. A larger α induces a higher tendency of drawing new parameters from the base distribution G_0; indeed, as $\alpha \to \infty$ we get $G \to G_0$. On the contrary, as $\alpha \to 0$ all $\{\Theta_n\}_{n=1}^N$ tend to cluster to a single random variable. Second, the more often a parameter is shared, the more likely it will be shared in the future.

A characterization of the (unconditional) distribution of the random variable G drawn from a Dirichlet process $\mathrm{DP}(G_0, \alpha)$ is provided by the *stick-breaking construction* of Sethuraman [22]. Consider two infinite collections of independent random variables $\boldsymbol{v} = (v_c)_{c=1}^{\infty}$, $\{\Theta_c\}_{c=1}^{\infty}$, where the v_c are drawn from the Beta distribution $\mathrm{Beta}(1, \alpha)$, and the Θ_c are independently drawn from the base distribution G_0. The stick-breaking representation of G is then given by [22]

$$G = \sum_{c=1}^{\infty} \pi_c(\boldsymbol{v}) \delta_{\Theta_c} \tag{5}$$

where

$$\pi_c(\boldsymbol{v}) = v_c \prod_{j=1}^{c-1} (1 - v_j) \quad \in [0, 1] \tag{6}$$

$$v_c | \alpha \sim \mathrm{Beta}(1, \alpha) \tag{7}$$

and

$$\sum_{c=1}^{\infty} \pi_c(\boldsymbol{v}) = 1 \tag{8}$$

3. Proposed Approach

3.1. Model Formulation

Let us consider a sequence X of D-dimensional observable data, with $X = \{\boldsymbol{x}_t\}_{t=1}^{T}$. In HMMs, model states are primarily driven by the observation process X. The utility of the latent Markov chain is to provide a set of state transition probabilities for the model that offer additional smoothing. In conventional HMMs, these state transition probabilities are constants, learnt from observed data using some sort of training algorithm. In this work, we want to come up with a non-stationary HMM where the state transition probabilities change over time, driven by the temporal dynamics in the observed sequence. Such an HMM formulation is expected to allow for a significant model performance increase in terms of its pattern recognition capacity, by adopting the configuration of the HMM latent chain to the dynamics of each observed sequence. We are seeking a non-parametric Bayesian formulation for our model, that allows for conducting inference over the number of model parameters (states), thus obviating the need of applying the existing, overfitting-prone, likelihood-based model selection criteria, such as BIC [23].

For this purpose, we first employ an observation-driven ESN reservoir with N neurons to capture the temporal dynamics in the observed sequences at any time point. As discussed in Section 2, at any time point t the employed ESN reservoir, driven by the observed data $\boldsymbol{x}_t$, generates an N-dimensional state vector $\boldsymbol{\varsigma}_t$ that encodes the history of the observed signal values $\{\boldsymbol{x}_\tau\}_{\tau=1}^{t}$, given by (1). Based on the obtained observation history representation, expressed by

means of the reservoir state vectors $\boldsymbol{\varsigma}_t$, we postulate an infinite state HMM, with its state transition probabilities formulated as functions of time, as follows

$$\begin{aligned}\kappa_{ij}(t) \triangleq p(s_t = j|s_{t-1} = i; a_{ij}, \boldsymbol{\varsigma}_t) \propto \\ p(s_t = j|s_{t-1} = i; a_{ij}) p(\boldsymbol{\varsigma}_t|s_t = j; s_{t-1} = i)\end{aligned} \tag{9}$$

$\forall t \geq 2$, where

$$p(s_t = j|s_{t-1} = i; a_{ij}) = a_{ij}(\boldsymbol{v}^A) \tag{10}$$

$$p(\boldsymbol{\varsigma}_t|s_t = j; s_{t-1} = i) = \mathcal{N}(\boldsymbol{\varsigma}_t|\boldsymbol{\varphi}_{ij}, \sigma_{ij}^2) \tag{11}$$

$\mathcal{N}(\cdot|\boldsymbol{\varphi}_{ij}, \sigma_{ij}^2)$ is a spherical Gaussian with mean $\boldsymbol{\varphi}_{ij}$ and variance σ_{ij}^2, the s_t are indicator variables denoting the latent HMM state generating the tth observation $\boldsymbol{x}_t$, and the $a_{ij}(\boldsymbol{v}^A)$ are the probabilities generated by a stick-breaking process with stick-variables $\boldsymbol{v}^A$, such that $\boldsymbol{v}^A = (v_{ij}^A)_{i,j=1}^{\infty}$

$$a_{ij}(\boldsymbol{v}^A) = v_{ij}^A \prod_{k=1}^{j-1} (1 - v_{ik}^A) \tag{12}$$

$$v_{ij}^A \sim \text{Beta}(1, \alpha_{ij}^A) \tag{13}$$

$$\sum_{j=1}^{\infty} a_{ij}(\boldsymbol{v}^A) = 1 \; \forall i \tag{14}$$

Definition (9) comprises the basic structural element of our approach. It shows that, under our approach, the probability of transition between two HMM states does not only depend on the prior assumptions of our model, as they are expressed by means of the postulated stick-breaking priors, but also on a probabilistic model that relates the likelihood of HMM state transition with the state vectors of an employed ESN reservoir. These latter state vectors encode all the history of the observed signal values, thus allowing for us to derive an HMM where state transition probabilities are not constant, but depend on the history of the observed signal values.

Similar to (12), we impose a stick-breaking prior over the initial state prior probabilities π_i of the latent Markov chain, such that $\boldsymbol{v}^\pi = (v_i^\pi)_{i=1}^{\infty}$

$$p(s_1 = i) = \pi_i(\boldsymbol{v}^\pi) \tag{15}$$

$$\pi_i(\boldsymbol{v}^\pi) = v_i^\pi \prod_{k=1}^{i-1} (1 - v_k^\pi) \tag{16}$$

$$v_i^\pi \sim \text{Beta}(1, \alpha_i^\pi) \tag{17}$$

and

$$\sum_{i=1}^{\infty} \pi_i(\boldsymbol{v}^\pi) = 1 \tag{18}$$

We also impose Gamma priors over the innovation parameters of the model, such as

$$\alpha_{ij}^{A} \sim \mathcal{G}(\omega_1, \omega_2) \tag{19}$$

$$\alpha_{i}^{\pi} \sim \mathcal{G}(\varepsilon_1, \varepsilon_2) \tag{20}$$

Finally, we postulate the following state-conditional observation likelihoods

$$\boldsymbol{x}_t | s_t = c; \Theta_c \sim \sum_{m=1}^{M} \varpi_{cm} \mathcal{N}(\boldsymbol{x}_t | \boldsymbol{\mu}_{cm}, \boldsymbol{R}_{cm}) \tag{21}$$

with $\Theta_c = \{\varpi_{cm}, \boldsymbol{\mu}_{cm}, \boldsymbol{R}_{cm}\}_{m=1}^{M}$. In other words, we consider that the state-conditional emitting distributions of the postulated model are simple M-component Gaussian mixture models with means $\boldsymbol{\mu}_{cm}$ and precision matrices $\boldsymbol{R}_{cm}$. Introducing the additional latent variables z_t, with $z_t = m$ if the tth observation $\boldsymbol{x}_t$ is generated from the mth mixture component, we may write

$$\boldsymbol{x}_t | s_t = c, z_t = m; \Theta_c \sim \mathcal{N}(\boldsymbol{x}_t | \boldsymbol{\mu}_{cm}, \boldsymbol{R}_{cm}) \tag{22}$$

with

$$p(z_t = m | s_t = c) = \varpi_{cm} \tag{23}$$

We dub the nonparametric Bayesian model described by Eqs. (9)-(23) the echo-state stick-breaking HMM.

3.2. Variational Bayesian Inference

Let us consider a training sequence X of D-dimensional observable data, with $X = \{\boldsymbol{x}_t\}_{t=1}^{T}$. To conduct Bayesian inference for our model, we need first to impose appropriate prior distributions over its parameters. For convenience, we choose priors of conjugate exponential form, as this selection greatly simplifies inference and interpretability [24]. We let the joint (conjugate exponential) prior on the means and the precisions of the mixture component densities of the ES-SB-HMM hidden states be

$$p(\boldsymbol{\mu}_{cm}, \boldsymbol{R}_{cm}) = \mathcal{NW}(\boldsymbol{\mu}_{cm}, \boldsymbol{R}_{cm} | \lambda_{cm}, \boldsymbol{m}_{cm}, \eta_{cm}, \boldsymbol{\Phi}_{cm}) \tag{24}$$

where $\mathcal{NW}(\boldsymbol{\mu}_{cm}, \boldsymbol{R}_{cm} | \lambda_{cm}, \boldsymbol{m}_{cm}, \eta_{cm}, \boldsymbol{\Phi}_{cm})$ is a Normal-Wishart distribution. We also impose a Dirichlet prior over the mixture weights ϖ_{cm}, yielding

$$p(\boldsymbol{\varpi}_c) = \mathcal{D}(\varpi_{c1}, \ldots, \varpi_{cM} | \nu_{c1}, \ldots, \nu_{cM}) \tag{25}$$

Regarding the parameters of the $p(\boldsymbol{\varsigma}_t | s_t = j; s_{t-1} = i)$, i.e. the set of the means $\boldsymbol{\varphi}_{ij}$ and variances σ_{ij}^2, we choose not to impose a prior over them. These will be optimized as model hyperparameters, as part of the inference algorithm discussed next.

Inference for non-parametric Bayesian models can be conducted by means of variational Bayes (e.g., [25]) or Monte Carlo techniques (e.g., [26]). Here, we prefer a variational Bayesian approach, due to its considerably better scalability

in terms of computational costs, which becomes of a major importance when having to deal with large data corpora. Variational Bayesian inference for the proposed model consists in derivation of a family of variational posterior distributions $q(.)$ which approximate the true posterior distribution over the infinite sets $\boldsymbol{v}^A, \boldsymbol{v}^\pi$, $\{\alpha_{ij}^A, \alpha_i^\pi\}_{i,j=1}^\infty$, and $\{\Theta_c\}_{c=1}^\infty$. Apparently, under this infinite dimensional setting, Bayesian inference is not tractable. For this reason, we employ a common strategy in the literature of nonparametric Bayesian techniques, formulated on the basis of a truncated stick-breaking construction [25]. That is, we fix a value C and we let the variational posterior over the v_{ij}^A and the v_i^π have the property $q(v_{iC}^A = 1) = 1 \; \forall i$, and $q(v_C^\pi = 1) = 1$. In other words, we set $\pi_c(\boldsymbol{v}^\pi)$ and $a_{ic}(\boldsymbol{v}^A)$ equal to zero for $c > C$. Note that, under this setting, the treated nonparametric Bayesian model involves a full stick-breaking prior; truncation is not imposed on the model itself, but only on the variational distribution to allow for a tractable inference procedure. Hence, the truncation level C is a variational parameter which can be freely set, and not part of the prior model specification.

The variational Bayesian treatment of our model is conducted by introducing an arbitrary distribution $q(\Psi)$, and maximizing the variational free energy

$$F(q) = \int \mathrm{d}\Psi q(\Psi) \log \frac{p(X, \Psi)}{q(\Psi)} \tag{26}$$

of the model, which comprises a lower bound of the model log evidence $\log p(X)$ [27], where $\Psi = \left\{\{s_t, z_t\}_{t=1}^T, \{\Theta_c\}_{c=1}^C, \boldsymbol{v}^A, \boldsymbol{v}^\pi, \{\alpha_{ij}^A, \alpha_i^\pi\}_{i,j=1}^C\right\}$. In order to yield a tractable expression for the variational free energy of our model, we assume that $q(\Psi)$ factorizes over the latent variables and the model parameters as

$$q(\Psi) \approx q(\boldsymbol{v}^A) q(\boldsymbol{v}^\pi) \prod_{t=1}^T \left[q(s_t) q(z_t|s_t)\right] \prod_{c=1}^C q(\Theta_c) \prod_{i=1}^C \prod_{j=1}^C q(\alpha_{ij}^A) \prod_{c=1}^C q(\alpha_c^\pi) \tag{27}$$

Factorization of $q(\Psi)$ in the form (27) is a common approach in variational Bayesian inference (see, e.g., [27, 28, 29]). Due to the considered conjugate prior configuration of our model, the variational posteriors comprising $q(\Psi)$ are expected to take the same functional form as their corresponding priors [30]. The expressions of the variational posteriors over the model variables are derived by maximizing $F(q)$ with respect to each one of the factors of $q(\Psi)$ in turn, holding the others fixed, in an iterative manner [31]. By construction, the iterative, consecutive updating of the interdependent distributions of the considered factors of $q(\Psi)$ is guaranteed to monotonically and maximally increase the variational free energy $F(q)$ [30].

Let us denote as $\langle \cdot \rangle$ the posterior expectation of an expression. For completeness sake, the expressions of the posterior expectations found in the derived variational posteriors of our model are provided in the Appendix.

1. For the $q(\boldsymbol{v}^A)$ we obtain

$$q(v_{ij}^A) = \mathrm{Beta}(\tilde{\beta}_{ij}^A, \hat{\beta}_{ij}^A) \tag{28}$$

$$\tilde{\beta}_{ij}^{A} = 1 + \sum_{t=2}^{T} q(s_t = j | s_{t-1} = i) \tag{29}$$

$$\hat{\beta}_{ij}^{A} = \left\langle \alpha_{ij}^{A} \right\rangle + \sum_{\varrho=j+1}^{C} \sum_{t=2}^{T} q(s_t = \varrho | s_{t-1} = i) \tag{30}$$

2. Similar, for the $q(\boldsymbol{v}^{\pi})$ we obtain

$$q(v_i^{\pi}) = \text{Beta}(\tilde{\beta}_i^{\pi}, \hat{\beta}_i^{\pi}) \tag{31}$$

$$\tilde{\beta}_i^{\pi} = 1 + q(s_1 = i) \tag{32}$$

$$\hat{\beta}_i^{\pi} = \left\langle \alpha_i^{\pi} \right\rangle + \sum_{\varrho=i+1}^{C} q(s_1 = \varrho) \tag{33}$$

3. For the $q(\alpha_{ij}^{A})$ we obtain

$$q(\alpha_{ij}^{A}) = \mathcal{G}(\alpha_{ij}^{A} | \tilde{\omega}_{ij}^{A}, \hat{\omega}_{ij}^{A}) \tag{34}$$

where

$$\tilde{\omega}_{ij}^{A} = \omega_1 + 1 \tag{35}$$

$$\hat{\omega}_{ij}^{A} = \omega_2 - \left[\psi(\hat{\beta}_{ij}^{A}) - \psi(\tilde{\beta}_{ij}^{A} + \hat{\beta}_{ij}^{A}) \right] \tag{36}$$

4. For the $q(\alpha_i^{\pi})$ we obtain

$$q(\alpha_i^{\pi}) = \mathcal{G}(\alpha_i^{\pi} | \tilde{\varepsilon}_i^{\pi}, \hat{\varepsilon}_i^{\pi}) \tag{37}$$

where

$$\tilde{\varepsilon}_i^{\pi} = \varepsilon_1 + 1 \tag{38}$$

$$\hat{\varepsilon}_i^{\pi} = \varepsilon_2 - \left[\psi(\hat{\beta}_i^{\pi}) - \psi(\tilde{\beta}_i^{\pi} + \hat{\beta}_i^{\pi}) \right] \tag{39}$$

5. For the $q(\boldsymbol{\mu}_{cm}, \boldsymbol{R}_{cm})$ we obtain

$$q(\boldsymbol{\mu}_{cm}, \boldsymbol{R}_{cm}) = \mathcal{NW}(\boldsymbol{\mu}_{cm}, \boldsymbol{R}_{cm} | \tilde{\lambda}_{cm}, \tilde{\boldsymbol{m}}_{cm}, \tilde{\eta}_{cm}, \tilde{\boldsymbol{\Phi}}_{cm}) \tag{40}$$

where we introduce the notation

$$\xi_t^{cm} \triangleq q(s_t = c) q(z_t = m | s_t = c) \tag{41}$$

$$\bar{\boldsymbol{x}}_{cm} \triangleq \frac{\sum_{t=1}^{T} \xi_t^{cm} \boldsymbol{x}_t}{\sum_{t=1}^{T} \xi_t^{cm}} \tag{42}$$

$$\boldsymbol{\Delta}_{cm} \triangleq \sum_{t=1}^{T} \xi_t^{cm} \left(\boldsymbol{x}_t - \bar{\boldsymbol{x}}_{cm} \right) \left(\boldsymbol{x}_t - \bar{\boldsymbol{x}}_{cm} \right)^{\mathrm{T}} \tag{43}$$

and, we have

$$\tilde{\eta}_{cm} = \eta_{cm} + \sum_{t=1}^{T} \xi_t^{cm} \tag{44}$$

$$\tilde{\boldsymbol{\Phi}}_{cm} = \boldsymbol{\Phi}_{cm} + \boldsymbol{\Delta}_{cm} + \frac{\lambda_{cm}\sum_{t=1}^{T}\xi_t^{cm}}{\lambda_{cm}+\sum_{t=1}^{T}\xi_t^{cm}}\left(\boldsymbol{m}_{cm}-\bar{\boldsymbol{x}}_{cm}\right)\left(\boldsymbol{m}_{cm}-\bar{\boldsymbol{x}}_{cm}\right)^{\mathrm{T}} \tag{45}$$

$$\tilde{\lambda}_{cm} = \lambda_{cm} + \sum_{t=1}^{T}\xi_t^{cm} \tag{46}$$

$$\tilde{\boldsymbol{m}}_{cm} = \frac{\lambda_{cm}\boldsymbol{m}_{cm} + \bar{\boldsymbol{x}}_{cm}\sum_{t=1}^{T}\xi_t^{cm}}{\lambda_{cm}+\sum_{t=1}^{T}\xi_t^{cm}} \tag{47}$$

6. Regarding the mixture weights ϖ_{cm}, we obtain

$$q(\boldsymbol{\varpi}_c) = \mathcal{D}(\varpi_{c1},\ldots,\varpi_{cM}|\tilde{\nu}_{c1},\ldots,\tilde{\nu}_{cM}) \tag{48}$$

where

$$\tilde{\nu}_{cm} = \nu_{cm} + \sum_{t=1}^{T}\xi_t^{cm} \tag{49}$$

7. Regarding the posteriors over the sets of latent indicator variables $S = \{s_t\}_{t=1}^{T}$ and $Z = \{z_t\}_{t=1}^{T}$, we obtain

$$q(S,Z) = \frac{1}{Q}\pi_{s_1}^{*}\prod_{t=1}^{T-1}a_{s_t s_{t+1}}^{*}(\boldsymbol{\varsigma}_{t+1})\prod_{t=1}^{T}\varpi_{s_t z_t}^{*}p^{*}(\boldsymbol{x}_t|s_t,z_t) \tag{50}$$

where

$$\pi_c^{*} \triangleq \exp\left[\langle \log\pi_c(\boldsymbol{v}^{\pi})\rangle\right] \tag{51}$$

$$a_{s_t s_{t+1}}^{*}(\boldsymbol{\varsigma}_{t+1}) \triangleq \exp\left[\left\langle \log a_{s_t s_{t+1}}(\boldsymbol{v}^{A})\right\rangle + \log\mathcal{N}(\boldsymbol{\varsigma}_{t+1}|\boldsymbol{\varphi}_{s_t s_{t+1}},\sigma_{s_t s_{t+1}}^{2})\right] \tag{52}$$

$$\varpi_{cm}^{*} \triangleq \exp\left[\langle \log\varpi_{cm}\rangle\right] \tag{53}$$

$$p^{*}(\boldsymbol{x}_t|s_t=c,z_t=m) \triangleq \exp\left[\langle \log p(\boldsymbol{x}_t|\boldsymbol{\mu}_{cm},\boldsymbol{R}_{cm})\rangle\right] \tag{54}$$

the $\{\boldsymbol{\varsigma}_t\}_{t=1}^{T}$ are the reservoir state vectors corresponding to the input sequence X, Q is a normalizing constant, and $\psi(\cdot)$ is the Digamma function. Based on (50), computation of the probabilities $q(s_t = j|s_{t-1} = i)$, $q(s_1 = i)$, and $q(z_t = m|s_t = c)$, which constitute the variational posterior $q(S, L)$, can be easily performed for our Bayesian model by just employing the classical forward-backward algorithm: we run the forward-backward algorithm for simple HMMs using the posterior expected values π_c^{*}, $a_{s_t s_{t+1}}^{*}(\boldsymbol{\varsigma}_{t+1})$, ϖ_{cm}^{*}, $p^{*}(\boldsymbol{x}_t|s_t = c, z_t = m)$ in (51)-(54) as the parameter values of the algorithm [30].

8. Finally, as already discussed, the parameters $\boldsymbol{\varphi}_{ij}$ and σ_{ij}^2 of the model are optimized as model hyperparameters. This is effected by maximizing the variational free energy of the model over each one of them, eventually yielding

$$\boldsymbol{\varphi}_{ij} = \frac{\sum_{t=2}^{T}q(s_t=j|s_{t-1}=i)\boldsymbol{\varsigma}_t}{\sum_{t=2}^{T}q(s_t=j|s_{t-1}=i)} \tag{55}$$

$$\sigma_{ij}^2 = \frac{\sum_{t=2}^{T}q(s_t=j|s_{t-1}=i)\left|\left|\boldsymbol{\varsigma}_t-\boldsymbol{\varphi}_{ij}\right|\right|^2}{\sum_{t=2}^{T}q(s_t=j|s_{t-1}=i)} \tag{56}$$

3.3. Predictive Density

Let us now consider the task of assigning a predictive probability $p(X')$ to a test sequence $X' = \{\boldsymbol{x}_t^{'}\}_{t=1}^{T}$ with respect to a model trained as described above. To conduct density estimation, we have to estimate the predictive density of the test sequence, which reads

$$p(X') \approx \int \mathrm{d}\Psi q(\Psi) p(X'|\Psi) \tag{57}$$

Using Jensen's inequality, we yield the following *lower bound* for the logarithm of $p(X')$

$$\log p(X') \geq \mathrm{Pred}(X') \tag{58}$$

where

$$\begin{aligned} \mathrm{Pred}(X') = & \sum_{S,Z} q(S,Z) \\ & \times \log \frac{\pi^*_{s_1} \prod_{t=1}^{T-1} a^*_{s_t s_{t+1}}(\boldsymbol{\varsigma}^{'}_{t+1}) \prod_{t=1}^{T} \varpi^*_{s_t z_t} p^*(\boldsymbol{x}^{'}_t|s_t, z_t)}{q(S,Z)} \end{aligned} \tag{59}$$

the $\pi^*_{s_1}, \varpi^*_{s_t z_t}$, and $p^*(\boldsymbol{x}_t|s_t, z_t)$ are given by (51), (53), and (54), respectively, and we now have

$$a^*_{s_t s_{t+1}}(\boldsymbol{\varsigma}^{'}_{t+1}) \triangleq \exp\left[\left\langle \log a_{s_t s_{t+1}}(\boldsymbol{v}^A)\right\rangle + \log \mathcal{N}(\boldsymbol{\varsigma}^{'}_{t+1}|\boldsymbol{\varphi}_{s_t s_{t+1}}, \sigma^2_{s_t s_{t+1}})\right] \tag{60}$$

where the $\{\boldsymbol{\varsigma}'_t\}_{t=1}^{T}$ are the reservoir state vectors corresponding to the input sequence X'. Similar to the estimation of the $q(S,Z)$, computation of $\mathrm{Pred}(X')$ consists in merely employing the forward-backward algorithm, as described in [1]: we run the forward-backward algorithm for simple HMMs using the posterior expected values π^*_c, $a^*_{s_t s_{t+1}}(\boldsymbol{\varsigma}_{t+1})$, ϖ^*_{cm}, $p^*(\boldsymbol{x}_t|s_t = c, z_t = m)$ in (51)-(54) as the parameter values of the algorithm [30].

3.4. Sequence Decoding

Finally, apart from predictive density estimation, another significant problem we have to address is sequence segmentation given a trained ES-SB-HMM. In other words, given a trained ES-SB-HMM model with posterior $q(\Psi)$ and an observed sequence $X = \{\boldsymbol{x}_t\}_{t=1}^{T}$, we want to obtain an optimal assignment of the observed data points $\boldsymbol{x}_t$ to the model states $c = 1, \ldots, C$. This problem essentially comprises optimization of the quantity $q(S)$ over the indicator variables S. Based on the expression (50) of the joint posterior $q(S,Z)$, the posterior $q(S)$ yields

$$q(S) = \frac{1}{\Lambda}\left[\pi^*_{s_1} \prod_{t=1}^{T-1} a^*_{s_t s_{t+1}}(\boldsymbol{\varsigma}_{t+1})\right] \prod_{t=1}^{T} \sum_{z_t} \varpi^*_{s_t z_t} p^*(\boldsymbol{x}_t|s_t, z_t) \tag{61}$$

where Λ is a normalisation constant.

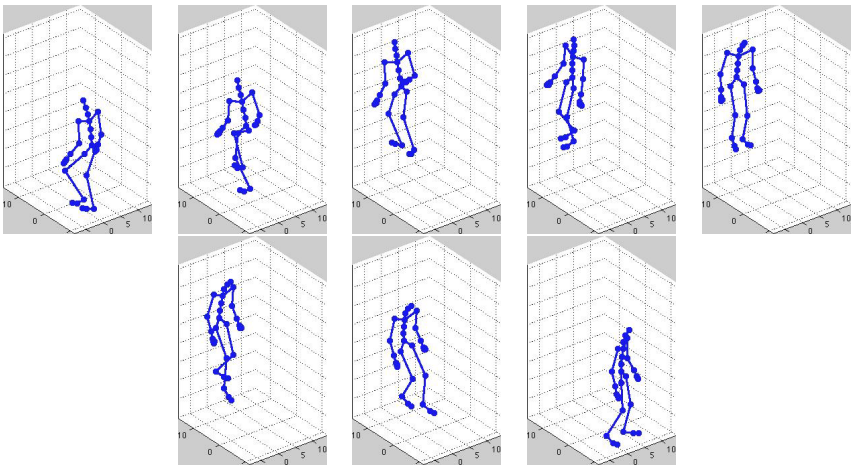


Figure 1: 3-step climbing: Few example frames from a sequence considered in our experiments.

The problem of optimising $q(S)$ over S is essentially a dynamic programming problem. Following, e.g., [32] and [30], it is easy to show that this problem can be resolved by application of a variational Bayesian analogous to the familiar Viterbi algorithm: we run the Viterbi algorithm with the parameter values set equal to the posterior expected values π_c^*, $a_{s_t s_{t+1}}^*(\boldsymbol{\varsigma}_{t+1})$, ϖ_{cm}^*, $p^*(\boldsymbol{x}_t|s_t = c, z_t = m)$ of our model, defined in Eqs. (51)-(54).

4. Experiments

In the following, we experimentally evaluate the performance of the ES-SB-HMM, considering a number of applications. In all our experiments, we used randomly initialized reservoirs, comprising simple analog neurons with $\tanh(\cdot)$ activation functions. The spectral radius of the employed reservoirs was set equal to 0.98, and the connectivity fraction equal to 0.2. For comparison, we also provide the performance of stick-breaking HMMs (SB-HMMs) [15], CRFs [16], and HCRFs [17]. Our source codes were developed in MATLAB R2011b, and made partial use of software provided by Neil Lawrence [33]. Our experiments were executed on a Macintosh platform with a 2.53GHz Intel Core 2 Duo processor, and 4 GB RAM, running OS X 10.7.3.

4.1. Sequence Segmentation Experiments: Experimental setup and results

Here, we consider application of the ES-SB-HMM model to segmentation of human motion video sequences, using the variational Bayesian analog to the Viterbi algorithm described in Section 3.4. For this purpose, we use videos from the CMU motion capture dataset [34]; we consider four different experimental cases, namely *3-step climbing*, *skateboard: stop and go*, *skateboard: push and*

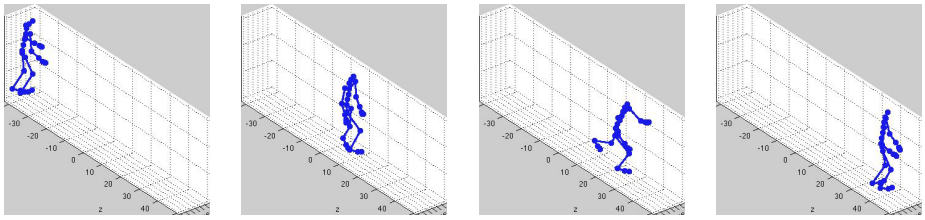


Figure 2: Skateboard: Stop and Go: Few example frames from a sequence considered in our experiments.

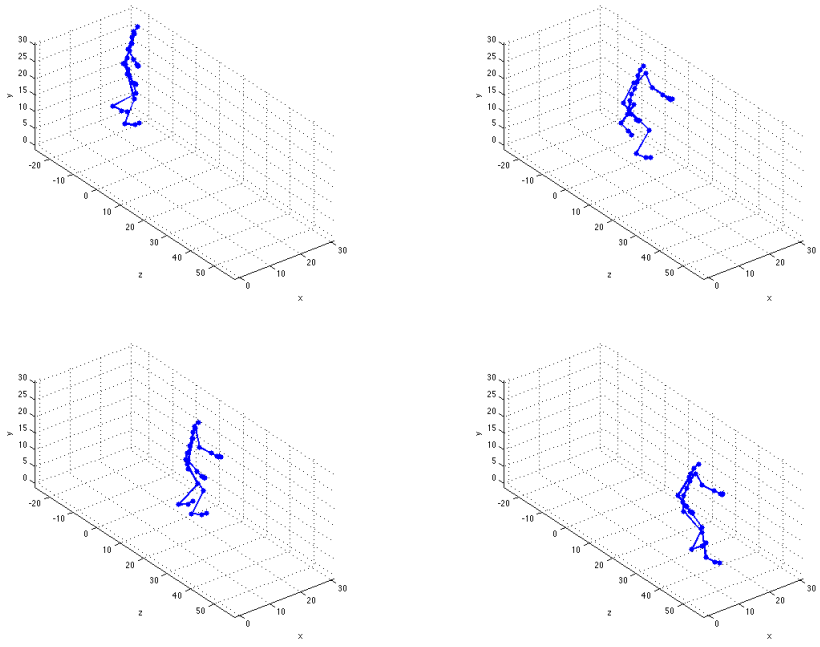


Figure 3: Skateboard: Push and Turn: Few example frames from a sequence considered in our experiments.

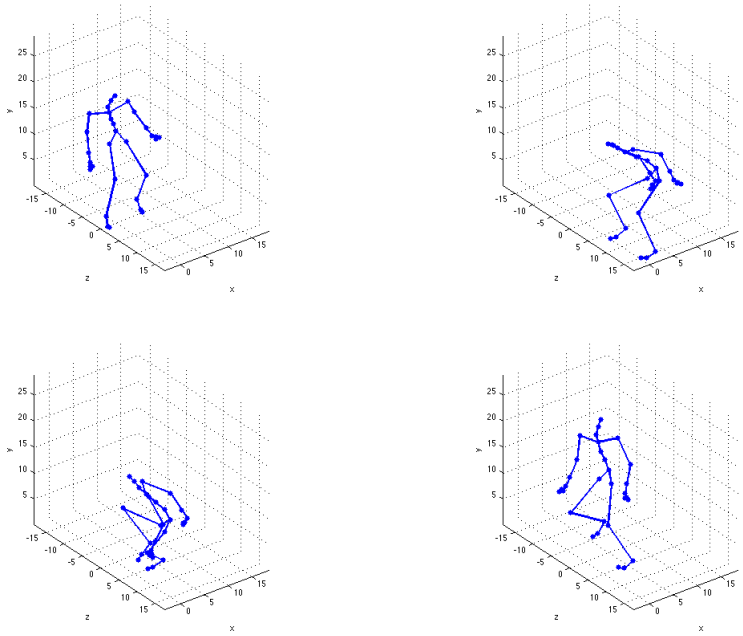


Figure 4: Modeling across subjects: Two subjects run, scramble for last seat: Few example frames from a sequence considered in our experiments.

turn, and *modeling across subjects: two subjects run, scramble for last seat.* Our aim is to use the trained models so as to segment the considered video sequences into meaningful subsequences, that correspond to parts of the depicted behaviors.

As input variables to the evaluated algorithms, we use a simplified version of the skeleton data provided by the CMU motion capture database. Specifically, in the used simplified skeleton, each pose is defined by three global (torso) pose angles, three global (torso) translational velocities, and the provided Euler angles pertaining to the joints: head, lowerneck, thorax, upperback, lowerback, root, upperneck, clavicle, humerus, radius, wrist, femur, tibia, and foot. For each frame, the global velocity is set to the difference between the next and the current frames. The velocity for the last frame is copied from the second to last frame. The used output variables are the activity (class) labels assigned to each video frame. We assume there is an 1-1 correspondence between model states and activity (class) labels; this is a standard procedure in HMM-based sequence labeling (segmentation). Apart from the proposed ES-SB-HMM approach, we also evaluate SB-HMMs and CRFs. The number of reservoir neurons of the evaluated ES-SB-HMM is selected so as to get optimal model performance for minimum model complexity. The evaluated CRFs utilize linear potential functions, and are trained using the L-BFGS algorithm [35].

Apparently, in these experiments, the number of latent states of both the postulated SB-HMM and ES-SB-HMM models is known beforehand, and, hence, their employed nonparametric Bayesian machinery does not come of any utility. However, in our experiments, we use these models employing their full nonparametric Bayesian structure. We set their truncation threshold to the desired number of states $C = 3$, and examine whether they use all these three states to model the data, or less. We have found out that both the SB-HMM and ES-SB-HMM approaches make use of all these three states, which clearly indicates that they are not prone to underestimating the appropriate model size.

In each one of the considered experimental cases, multiple different videos of each movement are used in order to perform cross validation. We also provide the p-metric value of the Student's-t test run on the pairs of performances of the evaluated models. The Student's-t test allows to assess the statistical significance of the performance difference between two evaluated methods, given a set of performance measurements. Generated p-values of the Student's-t test below 0.05 strongly indicate that the means of the obtained performance statistics of the two methods provide a very good assessment of their actual performance difference.

4.1.1. 3-step climbing

In this experimental case, we deal with videos depicting a human subject ascending a short ladder, stepping on a table, making a U-turn on the table, and descending the ladder. In Fig. 1, we provide few characteristic frames from one of the videos used in our experiments. Our aim is to train 3-state models capable of segmenting the videos into three subsequences: (i) ladder ascending; (ii) making U-turn; (iii) ladder descending. Four different videos, comprising

Table 1: Sequence segmentation: *3-step climbing* videos. Error rates obtained by the evaluated methods.

Method	SB-HMM	CRF	ES-SB-HMM
Error Rate (%)	35.50	28.98	29.27
p-value	0.017	0.36	#Neurons=3000

Table 2: Sequence segmentation: *skateboard: stop and go* videos. Error rates obtained by the evaluated methods.

Method	SB-HMM	CRF	ES-SB-HMM
Error Rate (%)	27.30	23.73	22.43
p-value	1.1×10^{-4}	0.005	#Neurons=3000

200-276 frames, from the same subject are used in our experiments to perform (4-fold) leave-one-out cross validation. The obtained performance statistics of the evaluated algorithms are provided in Table 1. As we notice, the ES-SB-HMM model improves considerably the obtained error rate compared to the SB-HMM. We also observe that the ES-SB-HMM yields a slightly lower mean performance compared to the CRF; however, this difference is not statistically significant.

4.1.2. Skateboard: Stop and Go

Here, we consider videos depicting a human subject sliding on a skateboard, then stopping, and subsequently pushing the skateboard back to start sliding again. In Fig. 2, we provide few characteristic frames from one of the videos used in our experiments. The aim in this experiment is to train 3-state models capable of segmenting the videos into 3 parts: (i) sliding on the skateboard; (ii) stopping; (iii) pushing back to resume. Three different videos, comprising 368-474 frames, from the same subject are used in our experiments to perform (3-fold) leave-one-out cross validation. The obtained performance statistics of the evaluated algorithms are provided in Table 2. As we notice, the ES-SB-HMM model improves considerably the obtained error rate compared to the SB-HMM. We also observe that the ES-SB-HMM offers a statistically significant improvement over CRFs.

Table 3: Sequence segmentation: *skateboard: push and turn* videos: Error rates obtained by the evaluated methods.

Method	SB-HMM	CRF	ES-SB-HMM
Error Rate (%)	38.71	29.79	26.46
p-value	2.07×10^{-8}	3.61×10^{-6}	#Neurons=3000

Table 4: Sequence segmentation: *two subjects run, scramble for last seat* videos: Error rates obtained by the evaluated methods.

Method	SB-HMM	CRF	ES-SB-HMM
Error Rate (%)	16.27	13.83	13.65
p-value	5.43×10^{-4}	0.081	#Neurons=500

4.1.3. Skateboard: Push and Turn

Here we consider videos depicting a human subject sliding on a skateboard, subsequently pushing the skateboard back to increase speed, and then turning. In Fig. 3, we provide few characteristic frames from one of the videos used in our experiments. The aim in this experiment is to train 3-state models capable of segmenting the videos into 3 parts: (i) sliding on the skateboard; (ii) pushing back to gain speed; (iii) turning. Four different videos, comprising 202-302 frames, from the same subject are used in our experiments to perform (4-fold) leave-one-out cross validation. The obtained performance statistics of the evaluated algorithms are provided in Table 3. As we notice, in this application the ES-SB-HMM model improves considerably the obtained error rate compared to both the SB-HMM and the CRF model.

4.1.4. Modeling across subjects: Two subjects run, scramble for last seat

Finally, in this experimental case we consider human motion modeling across subjects. Specifically, we train the considered models using two sequences from one subject, and we evaluate them using two sequences from a different subject. Modeling across subjects is a more demanding experimental scenario, thus allowing to better assess the robustness of our algorithm. The aim in this experiment is to train 4-state models capable of segmenting the videos into the following parts: (i) run; (ii) scramble for the seat; (iii) sit or (iv) leave. In Fig. 4, we provide few characteristic frames from one of the videos used in our experiments. Two-fold cross validation is performed to obtain mean performances, as well as the p-value from the Student's-t test to assess the statistical significance of our results. We again observe that ES-SB-HMM offers a clear improvement over SB-HMM. CRFs seem to perform comparably to our approach.

4.2. Sequence Segmentation Experiments: Further discussion

4.2.1. Confusion matrices

To gain further insight into the advantages offered by the proposed ES-SB-HMM, we revisit the previous experiments, providing the confusion matrices of our method for each considered experimental case. For comparison, we also investigate the performance of the SB-HMM. These results are depicted in Tables 5-12. A general observation from these results is that the ES-SB-HMM manages to better distinguish between classes that the SB-HMM has difficulty to recognize, while much less of an improvement is obtained in cases SB-HMM yields higher success rates. Additionally, there is no single case where ES-SB-HMM

Table 5: Sequence segmentation: *3-step climbing* videos. SB-HMM confusion matrix.

	Correct Label (Groundtruth)		
Estimated Label	ladder ascending	making U-turn	ladder descending
ladder ascending	71.97%	20.74%	7.92%
making U-turn	23.10%	50.84%	20.61%
ladder descending	4.93%	28.42%	71.47%

Table 6: Sequence segmentation: *3-step climbing* videos. ES-SB-HMM confusion matrix.

	Correct Label (Groundtruth)		
Estimated Label	ladder ascending	making U-turn	ladder descending
ladder ascending	78.12%	10.99%	7.03%
making U-turn	18.48%	63.28%	16.51%
ladder descending	3.4%	25.73%	76.46%

performed worse than SB-HMM. Another finding we would like to mention is that the SB-HMM tends to yield especially low recognition rates in the beginning of each video. We believe this is due to the fact that the transition matrices of the SB-HMM are static, and probably cannot account for the variability in how each video begins. In contrast, the time-dependent nature of the transition matrices of the ES-SB-HMM allows for accounting for this kind of variability, thus significantly increasing the robustness of the sequence segmentation algorithm.

4.2.2. Effect of the reservoir size on model performance

Here, we examine how the performance of the ES-SB-HMM approach is related to the size of the employed reservoirs (number of neurons). For this purpose, we repeat the previous experiments for various numbers of reservoir neurons, and we illustrate how model performance changes in Figs. 5a-5d. As we observe, for small reservoir sizes, the proposed ES-SB-HMM model yields performance almost identical to the SB-HMM. This fact shows that such small reservoir sizes cannot capture useful information for the HMM algorithm. How-

Table 7: Sequence segmentation: *skateboard: stop and go* videos. SB-HMM confusion matrix.

	Correct Label (Groundtruth)		
Estimated Label	sliding	stopping	resuming
sliding	64.29%	0	28.90%
stopping	32.14%	100%	21.94%
resuming	3.57%	0	49.16%

Table 8: Sequence segmentation: *skateboard: stop and go* videos. ES-SB-HMM confusion matrix.

	Correct Label (Groundtruth)		
Estimated Label	sliding	stopping	resuming
sliding	70.71%	0	21.94%
stopping	29.29%	100%	20%
resuming	0	0	58.06%

Table 9: Sequence segmentation: *skateboard: push and turn* videos. SB-HMM confusion matrix.

	Correct Label (Groundtruth)		
Estimated Label	sliding	pushing back to gain speed	turning
sliding	68.83%	9.09%	33.42%
pushing back to gain speed	31.17%	57.27%	0
turning	0	33.64%	66.58%

Table 10: Sequence segmentation: *skateboard: push and turn* videos. ES-SB-HMM confusion matrix.

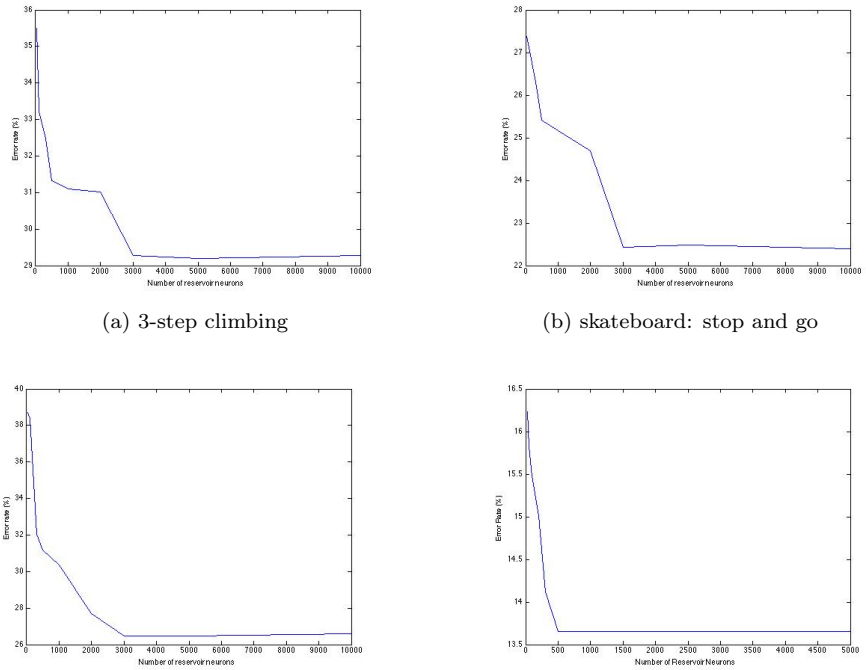
	Correct Label (Groundtruth)		
Estimated Label	sliding	pushing back to gain speed	turning
sliding	75.39%	0	26.84%
pushing back to gain speed	24.61%	71.83%	0
turning	0	28.17%	73.16%

Table 11: Sequence segmentation: *two subjects run, scramble for last seat* videos. SB-HMM confusion matrix.

	Correct Label (Groundtruth)			
Estimated Label	run	scramble for the seat	sit	leave
run	81.22%	0	0	13.37%
scramble for the seat	6.32%	80.70%	4.88%	5.20%
sit	0	9.11%	95.12%	0
leave	12.46%	10.19%	0	81.43%

Table 12: Sequence segmentation: *two subjects run, scramble for last seat* videos. ES-SB-HMM confusion matrix.

	Correct Label (Groundtruth)			
Estimated Label	run	scramble for the seat	sit	leave
run	83.01%	0	0	8.03%
scramble for the seat	5.17%	85.12%	1.35%	3.24%
sit	0	7.83%	98.65%	0
leave	11.82%	7.05%	0	88.73%



(a) 3-step climbing

(b) skateboard: stop and go

(c) skateboard: push and turn

(d) modeling across subjects: two subjects run, scramble for last seat

Figure 5: Error rate of the ES-SB-HMM as a function of the number of reservoir neurons.

ever, we also observe that as the size of the reservoir increases from some tens to one thousand neurons, our approach manifests a notable performance hike, while for reservoir sizes exceeding 3000 neurons, we observe that our method's performance remains largely stable.

4.3. Sequence Classification Experiments: Bimanual Gesture Recognition

4.3.1. Experimental setup and results

Here, we consider the problem of bimanual gesture recognition, experimenting with the American Sign Language gestures for the words: *against, aim, balloon, bandit, cake, chair, computer, concentrate, cross, deaf, explore, hunt, knife, relay, reverse,* and *role*. The used dataset was obtained from four different persons executing each one of these gestures and comprises 40 videos per gesture. 30 of these videos are used for training and the rest for model evaluation. From this dataset, we extracted several features representing the relative position of the hands and the face in the images, as well as the shape of the respective skin regions, by means of the complex Zernike moments [36], as described in [37]. This way, 12-dimensional feature vectors were derived.

Table 13: Gesture recognition: Recognition error rates (%) and p-metrics of the considered methods.

Gesture	SB-HMM	HCRF	ES-SB-HMM	#Neurons
against	0.95 (1.03×10^{-5})	0.83 (0.001)	0.78	300
aim	0.67 (1.38×10^{-5})	0.45 (0.08)	0.49	300
balloon	7.06 (3.44×10^{-9})	6.12 (0.002)	5.80	500
bandit	0.80 (6.28×10^{-4})	0.73 (0.64)	0.76	500
chair	23.67 (5.01×10^{-10})	12.09 (0.001)	9.59	3000
cake	18.29 (3.63×10^{-9})	6.34 (0.48)	7.07	3000
computer	6.84 (7.06×10^{-5})	4.87 (0.002)	4.23	100
concentrate	16.20 (1.44×10^{-7})	7.69 (1.12×10^{-5})	4.82	3000
cross	28.10 (6.88×10^{-9})	14.07 (3.09×10^{-5})	12.44	3000
deaf	4.38 (1.12×10^{-4})	3.66 (0.021)	4.08	300
explore	6.65 (2.12×10^{-4})	5.41 (0.003)	5.01	300
hunt	28.51 (0.98×10^{-9})	13.35 (0.001)	11.22	2000
knife	24.17 (7.12×10^{-5})	14.11 (0.001)	12.36	3000
relay	6.41 (3.48×10^{-4})	5.93 (0.07)	5.90	500
reverse	8.52 (9.09×10^{-5})	6.46 (0.03)	6.17	300
role	1.84 (0.003)	1.62 (0.01)	1.55	200
Average	11.44	6.48	5.77	

For each one of these 16 gestures, we fitted one model of each one of the algorithms SB-HMM and ES-SB-HMM, with truncation threshold set to $C = 100$. We also trained an HCRF model using the available data, as described in [17]. Classification of our test sequences using the SB-HMM and ES-SB-HMM approaches was conducted by computing their predictive densities with respect to each class (e.g., Eq. (59) of our model), and assigning them to the class that yields the highest predictive density. The error rates obtained by application of 10-fold cross-validation are depicted in Table 13. These results were obtained for optimal ES-SB-HMM reservoir size. As we observe, our approach works much better than the SB-HMM, while usually also offering a statistically significant improvement over the HCRF. An interesting finding is that, in most cases where the ES-SB-HMM offers a clear improvement over the competition, especially over the SB-HMM, the hands and head regions are initially separate, then they merge and then they separate again. Although the hand shapes are different, the gesture variability between different or even the same actors certainly creates ambiguity, which seems to be better modeled by means of our approach compared to its competitors.

4.3.2. Varying the number of training data points

To examine the effect of training data availability on model performance, we repeat the previous experiments using limited training data; specifically, we only use 5 videos from each gesture instead of 30. As it is well known, the

Table 14: Gesture recognition: Recognition error rates (%) of the considered methods.

Gesture	SB-HMM	HCRF	ES-SB-HMM	#Neurons
against	3.46	2.42	1.90	300
aim	4.58	2.17	1.52	500
balloon	14.18	10.24	8.88	800
bandit	3.05	2.13	1.59	500
chair	40.12	29.65	26.37	3000
cake	21.71	17.09	12.63	3000
computer	18.38	10.12	9.31	500
concentrate	25.25	14.48	14.92	3000
cross	40.67	28.40	27.60	3000
deaf	15.91	9.08	9.17	300
explore	10.08	7.18	6.42	600
hunt	30.44	19.47	17.58	2000
knife	30.95	30.24	28.23	3000
relay	12.40	7.67	8.08	500
reverse	19.11	12.53	16.26	500
role	2.08	1.98	1.64	800
Average	18.27	12.80	12	

effectiveness of model estimation by means of likelihood (or marginal likelihood) maximization is contingent upon the availability of adequate training data, to prevent the possibility of overfitting. Hence, we expect that limiting the number of training samples should most probably induce an adverse effect on the quality of the fitted models, and, hence, model performance. Our obtained results are provided in Table 14. As we observe, all the methods are affected by the reduction in the number of available training samples. However, we also observe that the ES-SB-HMM seems to be much less affected compared to the SB-HMM.

4.4. Computational Complexity

Finally, let us examine how the computational costs of the reservoir-driven ES-SB-HMM are related to the computational costs of the SB-HMM, that is a baseline HMM formulated in the context of Bayesian nonparametrics. Regarding the costs of the model training algorithms, we note that from a theoretical standpoint, the extra costs incurred by the ES-SB-HMM concern computation of the estimates $\boldsymbol{\varphi}_{ij}$ and σ_{ij} given by (55) and (56), as well as of the likelihoods $\log\mathcal{N}(\boldsymbol{\varsigma}_{t+1}|\boldsymbol{\varphi}_{s_t s_{t+1}}, \sigma^2_{s_t s_{t+1}})$ in (52). These are rather simple computations, expected to induce only a minor computational overhead for the model training algorithm. Similar, regarding the computational costs of the sequence segmentation (Viterbi) algorithm for both models, we observe that our approach requires only the additional computation of the likelihoods $\log\mathcal{N}(\boldsymbol{\varsigma}_{t+1}|\boldsymbol{\varphi}_{s_t s_{t+1}}, \sigma^2_{s_t s_{t+1}})$ in (52), which is not expected to give rise to a significant computational overhead

for our algorithm. The same holds for predictive density computation, under expression (59).

To substantiate our claims, let us consider, for instance, the case of the *skateboard: stop and go* experiment. In this case, 5 iterations of the training algorithm took 4.56 seconds in our unoptimized implementation, while the SB-HMM required 3.93 seconds. Hence, the ES-SB-HMM incurred a 13.82% computational overhead for model training. Similar, segmentation of one test sequence required 0.25 seconds in the case of the ES-SB-HMM and 0.17 seconds in the case of the SB-HMM, that is a 32% increase in the total computational costs of the sequence segmentation algorithm. We believe that these extra costs are absolutely reasonable, given the observed significant benefits of our approach in terms of the obtained pattern recognition performance, as we discussed previously.

5. Conclusions

In this paper, we proposed a novel non-parametric Bayesian approach for sequential data modeling based on the introduction of a non-stationary HMM. The proposed non-stationary stick-breaking HMM formulation is based on the consideration of a novel form of time-dependent state transition priors, the distribution of which jointly depends on a stick-breaking process and an appropriate probabilistic model over the state values of a postulated echo-state network reservoir. We provided efficient algorithms for model training and inference, based on a truncated variational Bayesian approach.

We examined the efficacy of our approach considering a number of applications dealing with human motion modeling and gesture recognition. To provide a thorough account of the merits of our approach, we considered both sequence segmentation and classification applications for our model, and we compared it to stick-breaking HMMs, CRFs, and HCRFs. As we showed, our approach yields significant performance improvements over these state-of-the-art approaches. Note also that the computational costs of our approach are similar to the SB-HMM, since computation of the reservoir states and derivation of the model estimates $\boldsymbol{\varphi}_{ij}$ and σ_{ij}^2 induce negligible additional computational costs for the model training and inference algorithms.

Appendix

Regarding the posterior expectations in Eqs. (28)-(54), we have

$$\langle \alpha_{ij}^A \rangle = \frac{\tilde{\omega}_{ij}^A}{\hat{\omega}_{ij}^A} \tag{62}$$

$$\langle \alpha_i^\pi \rangle = \frac{\tilde{\varepsilon}_i^\pi}{\hat{\varepsilon}_i^\pi} \tag{63}$$

$$\langle \log \varpi_{cm} \rangle = \psi(\tilde{\nu}_{cm}) - \psi\left(\sum_{r=1}^{M} \tilde{\nu}_{cr}\right) \tag{64}$$

$$\begin{aligned} \langle \log p(\boldsymbol{x}_t | \boldsymbol{\mu}_{cm}, \boldsymbol{R}_{cm}) \rangle = & -\frac{D}{2}\log 2\pi + \frac{1}{2}\langle \log |\boldsymbol{R}_{cm}| \rangle \\ & - \frac{1}{2}\left[\left\langle (\boldsymbol{x}_t - \boldsymbol{\mu}_{cm})^{\mathrm{T}} \boldsymbol{R}_{cm} (\boldsymbol{x}_t - \boldsymbol{\mu}_{cm}) \right\rangle\right] \end{aligned} \tag{65}$$

$$\begin{aligned} \left\langle (\boldsymbol{x}_t - \boldsymbol{\mu}_{cm})^{\mathrm{T}} \boldsymbol{R}_{cm} (\boldsymbol{x}_t - \boldsymbol{\mu}_{cm}) \right\rangle = & \tilde{\eta}_{cm} (\boldsymbol{x}_t - \tilde{\boldsymbol{m}}_{cm})^{\mathrm{T}} \left(\tilde{\boldsymbol{\Phi}}_{cm}\right)^{-1} (\boldsymbol{x}_t - \tilde{\boldsymbol{m}}_{cm}) \\ & + \frac{D}{\tilde{\lambda}_{cm}} \end{aligned} \tag{66}$$

$$\langle \log |\boldsymbol{R}_{cm}| \rangle = -\log\left|\frac{\tilde{\boldsymbol{\Phi}}_{cm}}{2}\right| + \sum_{\tau=1}^{D} \psi\left(\frac{\tilde{\eta}_{cm} + 1 - \tau}{2}\right) \tag{67}$$

$$\langle \log \pi_c(\boldsymbol{v}^{\pi}) \rangle = \sum_{c'=1}^{c-1} \langle \log(1 - v_{c'}^{\pi}) \rangle + \langle \log v_c^{\pi} \rangle \tag{68}$$

$$\langle \log v_c^{\pi} \rangle = \psi(\tilde{\beta}_i^{\pi}) - \psi(\tilde{\beta}_i^{\pi} + \hat{\beta}_i^{\pi}) \tag{69}$$

$$\langle \log(1 - v_c^{\pi}) \rangle = \psi(\hat{\beta}_i^{\pi}) - \psi(\tilde{\beta}_i^{\pi} + \hat{\beta}_i^{\pi}) \tag{70}$$

$$\left\langle \log a_{ij}(\boldsymbol{v}^{A}) \right\rangle = \sum_{j'=1}^{j-1} \left\langle \log(1 - v_{ij'}^{A}) \right\rangle + \left\langle \log v_{ij}^{A} \right\rangle \tag{71}$$

$$\left\langle \log v_{ij}^{A} \right\rangle = \psi(\tilde{\beta}_{ij}^{A}) - \psi(\tilde{\beta}_{ij}^{A} + \hat{\beta}_{ij}^{A}) \tag{72}$$

$$\left\langle \log(1 - v_{ij}^{A}) \right\rangle = \psi(\hat{\beta}_{ij}^{A}) - \psi(\tilde{\beta}_{ij}^{A} + \hat{\beta}_{ij}^{A}) \tag{73}$$

Acknowledgment

This work has been funded by the EU FP7 ALIZ-E project (Grant 248116).

[1] L. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (1989) 245–255.

[2] O. Cappé, E. Moulines, T. Rydén, Inference in Hidden Markov Models, Springer Series in Statistics, New York, 2005.

[3] H. Jaeger, The "echo state" approach to analysing and training recurrent neural networks, Tech. Rep. 148, German National Research Center for Information Technology, Bremen (2001).

[4] M. Lukosevicius, H. Jaeger, Reservoir computing approaches to recurrent neural network training, Computer Science Review 3 (2009) 127–149.

[5] D. Verstraeten, B. Schrauwen, M. D'Haene, D. Stroobandt, 2007 special issue: An experimental unification of reservoir computing methods, Neural Networks 20 (3) (2007) 391–403.

[6] W. Maass, T. Natschlaeger, H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, Neural Computation 14 (11) (2002) 2531–2560.

[7] V. N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[8] M. Tipping, Sparse Bayesian learning and the relevance vector machine, Journal of Machine Learning Research 1 (2001) 211–244.

[9] N. Cristianini, J. Shawe-Taylor, An introduction to support vector machines and other kernel-based learning methods, Cambridge University Press, 2000.

[10] S. Walker, P. Damien, P. Laud, A. Smith, Bayesian nonparametric inference for random distributions and related functions, J. Roy. Statist. Soc. B 61 (3) (1999) 485–527.

[11] R. Neal, Markov chain sampling methods for Dirichlet process mixture models, J. Comput. Graph. Statist. 9 (2000) 249–265.

[12] P. Muller, F. Quintana, Nonparametric Bayesian data analysis, Statist. Sci. 19 (1) (2004) 95–110.

[13] C. Antoniak, Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems., The Annals of Statistics 2 (6) (1974) 1152–1174.

[14] M. Beal, Z. Ghahramani, C. Rasmussen, The infinite hidden markov model, Advances in neural information processing systems 1 (2002) 577–584.

[15] I. Paisley, L. Carin, Hidden markov models with stick-breaking priors, IEEE Trans. Signal Process 57 (10) (2009) 3905 – 3917.

[16] C. Sutton, A. McCallum, An introduction to conditional random fields for relational learning, in: L. Getoor, B. Taskar (Eds.), Introduction to Statistical Relational Learning, MIT Press, 2006.

[17] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, T. Darrell, Hidden conditional random fields, IEEE Trans. Pattern Anal. Mach. Intell. 29 (10) (2007) 1848–1853.

[18] H. Jaeger, H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless telecommunication, Science 308 (2004) 78–80.

[19] M. Buehner, P. Young, A tighter bound for the echo state property, IEEE Transactions on Neural Networks 17 (3) (2006) 820–824.

[20] T. Ferguson, A Bayesian analysis of some nonparametric problems, The Annals of Statistics 1 (1973) 209–230.

[21] D. Blackwell, J. MacQueen, Ferguson distributions via Pólya urn schemes, The Annals of Statistics 1 (2) (1973) 353–355.

[22] J. Sethuraman, A constructive definition of the Dirichlet prior, Statistica Sinica 2 (1994) 639–650.

[23] G. Schwarz, Estimating the dimension of a model, The Annals of Statistics 4 (1978) 461–464.

[24] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, New York, 2006.

[25] D. M. Blei, M. I. Jordan, Variational inference for Dirichlet process mixtures, Bayesian Analysis 1 (1) (2006) 121–144.

[26] Y. Qi, J. W. Paisley, L. Carin, Music analysis using hidden Markov mixture models, IEEE Transactions on Signal Processing 55 (11) (2007) 5209–5224.

[27] H. Attias, A variational Bayesian framework for graphical models, in: Proc. Ann. Conf. Neural Information Processing Systems, 2000.

[28] D. MacKay, Ensemble learning for hidden Markov models, Tech. rep., Dept. of Physics, Univ. of Cambridge (1997).

[29] T. Jaakkola, M. Jordan, Bayesian parameter estimation via variational methods, Statistics and Computing 10 (2000) 25–37.

[30] S. P. Chatzis, D. Kosmopoulos, A variational bayesian methodology for hidden Markov models utilizing student's-t mixtures, Pattern Recognition 44 (2) (2011) 295–306.

[31] D. Chandler, Introduction to Modern Statistical Mechanics, Oxford University Press, New York, 1987.

[32] S. Ji, B. Krishnapuram, L. Carin, Variational Bayes for continuous hidden Markov models and its application to active learning, IEEE Trans. Pattern Analysis and Machine Intelligence 28 (4) (2006) 522–532.

[33] N. Lawrence, Gaussian process software: http://staffwww.dcs.shef.ac.uk/people/n.lawrence/software.html.

[34] The CMU MoCap database. [Online]. Available: http://mocap.cs.cmu.edu/.

[35] D. P. Bertsekas, Nonlinear Programming, 2nd Edition, Athena Scientific, 1999.

[36] R. Mukundan, K. R. Ramakrishnan, Moment Functions in Image Analysis: Theory and Applications., World Scientific, 1998.

[37] D. Kosmopoulos, I. Maglogiannis, Hand tracking for gesture recognition tasks using dynamic Bayesian network, International Journal of Intelligent Systems and Applications 1 (3/4) (2006) 359–375.