Convolution based Variational Bayes for Patient Vital Signs Modeling with Factorial HMM

Anonymous authors
Paper under double-blind review

Abstract

We propose a novel convolution based variational distribution and an EM based learning algorithm to scale factorial HMM to long and complex time series. The number of trainable parameters in our model is independent from the length of the input data. Our model is also adapted to the use of arbitrarily complex state emission distribution and can therefore be used in combination with patient physiological models. We show the ability of our model to disentangle independent additive processes from synthetic data. Our experiments also confirm that our algorithm is able to fit real world patient data more accurately when several independent Markov chains are used compared to a single Markov chain with a larger state space. Our model could thus offer a scalable, interpretable and versatile alternative to latent space time series models such as standard HMM.

1 Introduction

Hidden Markov models (HMM) are widely used tools to model observable multidimensional time series data Smit et al. (2021); Honoré et al. (2020). In a HMM, the observed data at time t are assumed to be drawn from emission distributions dependent on the state variable of a single latent K-state Markov chain. Single latent Markov chain models suffer from a limited modeling ability for complex time series. For example, to represent 16 bits of information about the past of a time series, a HMM needs $K = 2^{16}$ states. Learning the transitions between so many states rapidly becomes infeasible and the resulting models are often difficult to interpret. Factorial HMMs (Figure 1) are models with a *distributed* latent space consisting in $M > 1$ independent K-state[1] Markov chains. Equivalent K^M-state HMM exist, but the time complexity for posterior probability computation scales exponentially with M Ghahramani & Jordan (1996). Factorial HMMs allow the decomposition of the observed data into M decoupled processes, readily interpretable by the users. This considerably reduces the amount of state transitions to learn, since only 16 binary state variables are needed to represent 16 bits of information about the past of a time series.

Exact inference algorithms for general probabilistic models exist but are intractable in the case of densely connected graphs such as factorial HMM. For factorial HMM, the intractability arises from the fact that the time complexity of the probability propagation algorithm for a time series of size T is $\mathcal{O}(TMK^{M+1})$, i.e. scales exponentially with the number of independent Markov chains M Ghahramani & Jordan (1996). Instead of exact inference, approximate methods based on sampling and variational inference have been proposed Ghahramani & Jordan (1996); Ng et al. (2016); Mysore & Sahani (2012); Weiss & Ellis (2009). Variational methods often rely on a number of parameters growing linearly with the length of input time series Ghahramani & Jordan (1996); Foti et al. (2014). Also, existing learning algorithms have been designed for specific emission distributions, e.g. Gaussian or linear mixture of Gaussian Ng et al. (2016); Weiss & Ellis (2009). This means that new parameter update rules have to be re-derived when new families of emission distribution are used. This considerably limits the complexity and applicability of the models.

Adverse physiological processes, such as disease onset, are typically reasonably well understood in isolation, but difficult to identify in real time series data of hospitalized patients de Bournonville et al. (2019). The physiological characterization of diseases can often be integrated in *differentiable* compartment models

[1] Assuming all the chains have the same number of states for simplicity.

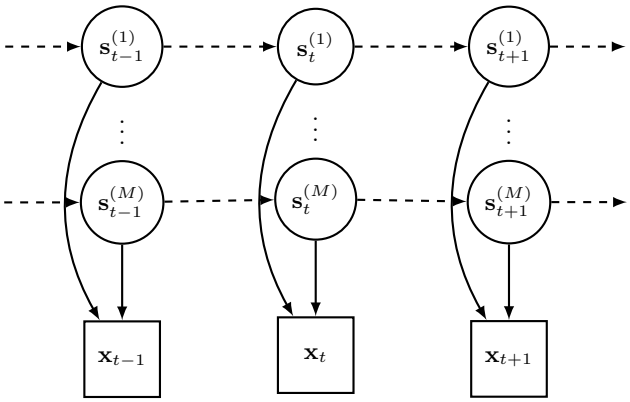

Figure 1: Graphical model of a factorial HMM with M independent Markov chains in the latent space. The unobserved latent space is depicted with circles and the observed data with square boxes. The dashed arrows depict time dependencies modeled with Markov chains (parameter λ) and the full arrows depict dependencies modeled with the emission distribution (parameter ϕ).

describing cardiovascular functions e.g. Guerrero et al. (2022); Ellwein Fix et al. (2018); Albanese et al. (2015). A factorial HMM where each latent Markov chain is associated with a disease model learnt in isolation, could offer a powerful statistical model to disentangle physiological processes from data. Such a model would have interesting practical properties: *interpretable* with a user-defined correspondence between independent Markov chains and physiological models of interest; *versatile* since it could accommodate any differentiable disease models; *scalable* by design with a number of parameters independent from the length of the time series. In this paper, we propose a framework to design such models, and learn their parameters from data.

1.1 Contributions

We propose an approximate variational inference based learning algorithm for factorial HMM. We build an end-to-end learning framework allowing arbitrary differentiable (both wrt their parameters and input) emission distributions using ideas from variational auto-encoders and normalizing flows. Our algorithm is based upon the expectation-maximization (EM) framework, and thus benefits from the same convergence guarantees. We propose a variational distribution built upon fully convolutional neural networks so that (1) the number of variational parameters does not depend on the length of the observed time series and (2) the variational lower bound can be optimized with back-propagation.

2 Methods

2.1 Model definition

Factorial HMMs are dynamical statistical models where time series data are represented on a distributed latent space. The latent space is modeled with a pre-defined number, M, of independent Markov chains. The state variable at time t for the m-th Markov chain, $\mathbf{s}_t^{(m)}$, depends only on the state variable of the same Markov chain at time $t-1$ (Figure 1). The dependence is modeled with transition kernels $p_\lambda(\mathbf{s}_t^{(m)} \mid \mathbf{s}_{t-1}^{(m)})$ where λ is a set of trainable parameters. The parameters related to the latent model can be written $\lambda = \{\pi^{(1)}, \dots, \pi^{(M)}, P^{(1)}, \dots, P^{(M)}\}$, where for $m = 1, \dots, M$, $\pi^{(m)}$ parameterizes the initial state $p_\lambda(\mathbf{s}_1^{(m)})$ and $P^{(m)}$ is the $K \times K$ transition matrix $p_\lambda(\mathbf{s}_t^{(m)} | \mathbf{s}_{t-1}^{(m)})$ for the m-th chain in the latent model. The state variables $\mathbf{s}_t^{(m)}$ are assumed continuous (see section 2.4.4), and lie on the probability simplex $\mathcal{S} = \{\mathbf{y} \in \mathbb{R}_+^K \mid \sum_{k=1}^K \mathbf{y}_k = 1\}$.

An observed d-dimensional data sample at time t, $\mathbf{x}_t \in \mathbb{R}^d$, is assumed to be drawn from an emission distribution that dependents on the current latent state variable $\mathbf{s}_t = [\mathbf{s}_t^{(1)}, \dots, \mathbf{s}_t^{(M)}] \in \mathcal{S}^M$. The emission

distribution is parameterized with a set of parameters ϕ and is denoted $p_\phi(\mathbf{x}_t \mid \mathbf{s}_t)$. The set of parameters of our statistical model is denoted $\theta = \{\lambda, \phi\}$ with λ the set of parameters related to the latent model and ϕ the set of parameters related to the generation model. The joint distribution of a sample time series $\mathbf{x}$ and a series of states $\mathbf{s}$ under a factorial HMM model parameterized with θ' factorizes as follows:

$$p_{\theta'}(\mathbf{x}, \mathbf{s}) = p_{\phi'}(\mathbf{x}_1|\mathbf{s}_1) \prod_{m=1}^{M} p_{\lambda'}(\mathbf{s}_1^{(m)}) \prod_{t=2}^{T} p_{\phi'}(\mathbf{x}_t|\mathbf{s}_t) \prod_{m=1}^{M} p_{\lambda'}(\mathbf{s}_t^{(m)}|\mathbf{s}_{t-1}^{(m)}) \tag{1}$$

2.2 Notations

We denote $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_T] \in \mathbb{R}^{T \times d}$ a multivariate time series of length T and with samples of dimension d. Similarly, $\mathbf{s} = [\mathbf{s}_1, \ldots, \mathbf{s}_T] \in \mathbb{R}^{T \times M \times K}$ denotes a sequence of states of length T, with individual samples $\mathbf{s}_t \in \mathbb{R}_+^{M \times K}$ and such that $\mathbf{s}_t^{(m)} \in \mathcal{S}$ for $m = 1, \cdots, M$. $\odot$ and $\circ$ are the elementwise tensor product and the composition operator respectively. The expectation with respect to (wrt) a certain distribution is noted with the distribution names and parameters, i.e. $\mathbb{E}_{p_{\theta'}(\mathbf{s} \mid \mathbf{x})}[\cdot] = \mathbb{E}_{p_{\theta'}}[\cdot]$. ln denotes the elementwise natural logarithm, exp or e denote the elementwise exponential.

2.3 Learning framework

Learning the statistical model $p_\theta(\mathbf{x})$ consists in maximizing the likelihood wrt parameter θ for some given data $\mathbf{x}$. We learn the statistical model $p_\theta(\mathbf{x})$ in the EM framework Dempster et al. (1977). EM consists in iteratively optimizing the joint distribution of the data and the latent state sequence given our current estimate of the parameters θ'. In other words, EM consists in optimizing the expected value: $\mathbb{E}_{p_{\theta'}}[\ln p_\theta(\mathbf{x}, \mathbf{s})]$ of the joint distribution, wrt our current knowledge of the posterior distribution of the states given the data: $p_{\theta'}(\mathbf{s} \mid \mathbf{x})$. Learning the parameters θ in the EM framework consists in iteratively performing two steps. The E-step: given a current parameter estimate θ', compute $Q(\theta, \theta') = \mathbb{E}_{p_{\theta'}}[\ln p_\theta(\mathbf{x}, \mathbf{s})]$. The M-step: given statistics computed at the previous step, maximize $Q(\theta, \theta')$ wrt θ.

When the statistical model is a HMM, the EM algorithm reduces to the Baum-Welch algorithm Baum & Petrie (1966). The Baum-Welch algorithm provides a set of procedures and update rules to find the maximum likelihood estimator of the HMM parameters. These procedures are however difficult to derive when the underlaying graph is complex and their exact computation scales exponentially with M Ghahramani & Jordan (1996). Moreover, the parameter update rules, based on state posterior statistics, have to be re-derived when the family of distribution changes. We confront these limitations directly, and resort to variational Bayes, an approximate method, to compute the state posterior statistics.

2.4 Variational E-step

2.4.1 Introducing the variational lower bound

We use variational inference to approximate the posterior distribution $p_{\theta'}(\mathbf{s}|\mathbf{x})$. This consists in approximating $p_{\theta'}(\mathbf{s}|\mathbf{x})$ with another distribution $q_\kappa(\mathbf{s}|\mathbf{x})$ with parameters in a set κ. The approximation is obtained by minimizing the Kullback-Leiber (KL) divergence between $q_\kappa(\mathbf{s}|\mathbf{x})$ and $p_{\theta'}(\mathbf{s}|\mathbf{x})$ wrt κ. Following a traditional development Kingma & Welling (2014), it can be shown that minimizing the KL-divergence is equivalent to maximizing the variational lower bound (VLB) $\mathcal{L}(\theta', \kappa; \mathbf{x})$ on $\ln p_{\theta'}(\mathbf{x})$, written:

$$\mathcal{L}(\theta', \kappa; \mathbf{x}) = \mathbb{E}_{q_\kappa(\mathbf{s}|\mathbf{x})}[\ln p_{\theta'}(\mathbf{x}, \mathbf{s}) - \ln q_\kappa(\mathbf{s}|\mathbf{x})]. \tag{2}$$

2.4.2 Computing the variational lower bound

To maximize $\mathcal{L}(\theta', \kappa; \mathbf{x})$ wrt κ, we must be able to differentiate $\mathcal{L}$ wrt κ. The direct optimization of $\mathcal{L}$ is problematic since a naïve Monte Carlo (MC) approximation of the gradient of $\mathcal{L}$ has high variance Paisley et al. (2012). We resort to the local reparameterization trick to define a differentiable variational distribution and optimize an approximation of $\mathcal{L}$.

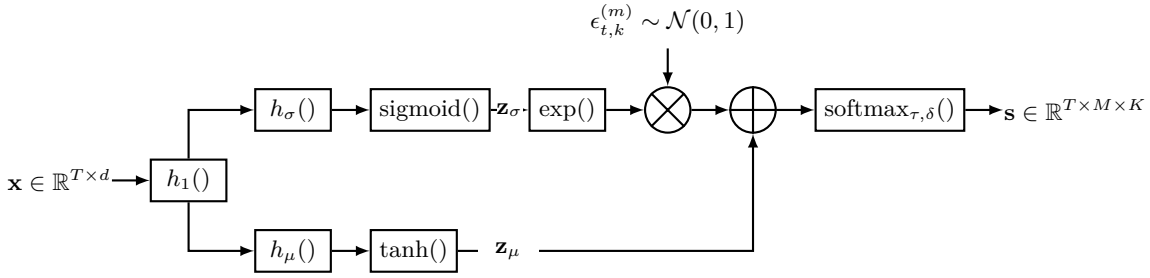

Figure 2: Sampling from the variational distribution $q_\kappa(\mathbf{s}|\mathbf{x})$.

We build $q_\kappa(\mathbf{s}|\mathbf{x})$ such that $\mathbf{s} \sim q_\kappa$ is equivalent to (1) sampling $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$ and (2) computing $\mathbf{s} = g_\kappa(\boldsymbol{\epsilon}; \mathbf{x})$. g_κ is defined as a diffeomorphism wrt $\boldsymbol{\epsilon}$, differentiable wrt κ and $p(\boldsymbol{\epsilon})$ is a fixed prior distribution (Figure 2, details in section 2.4.3). Using this distribution, we can then optimize a MC approximation $\mathcal{L}^A(\theta', \kappa; \mathbf{x})$ of the true lower bound $\mathcal{L}(\theta', \kappa; \mathbf{x})$ defined as:

$$\mathcal{L}^A(\theta', \kappa; \mathbf{x}) = \frac{1}{L}\sum_{l=1}^{L} \ln p_{\theta'}(\mathbf{x}, \mathbf{s}^{(l)}) - \ln q_\kappa(\mathbf{s}^{(l)}|\mathbf{x}), \tag{3}$$

where $\mathbf{s}^{(l)} \sim q_\kappa(\mathbf{s}|\mathbf{x})$ and L is the number of MC samples.

We now focus on the computation of the two terms $\ln p_{\theta'}(\mathbf{x}, \mathbf{s}^{(l)})$ and $\ln q_\kappa(\mathbf{s}^{(l)}|\mathbf{x})$ required to compute $\mathcal{L}^A$.

2.4.3 The variational distribution

We define the variational distribution $q_\kappa(\mathbf{s}|\mathbf{x})$, parameterized by $\kappa = \{\kappa_1, \kappa_\mu, \kappa_\sigma\}$, such that

$$\mathbf{s} \sim q_\kappa(\mathbf{s}|\mathbf{x}) \Leftrightarrow \mathbf{s} = g_\kappa(\boldsymbol{\epsilon}; \mathbf{x}) = \text{softmax}_{\tau,\delta}(\mathbf{z}_\mu + \exp(\mathbf{z}_\sigma) \odot \boldsymbol{\epsilon}), \tag{4}$$

where $\mathbf{z}_\sigma, \mathbf{z}_\mu, \boldsymbol{\epsilon} \in \mathbb{R}^{T\times M\times K}$ with $\epsilon_{t,k}^{(m)} \sim p(\epsilon) = \mathcal{N}(0,1)$. The intermediate variables $\mathbf{z}_\mu, \mathbf{z}_\sigma$ are defined as $\mathbf{z}_\mu = \tanh \circ\, h_\mu \circ\, h_1(\mathbf{x})$ and $\mathbf{z}_\sigma = \text{sigmoid} \circ\, h_\sigma \circ\, h_1(\mathbf{x})$. These variables can be interpreted as a translation and scaling transformations of the variable $\boldsymbol{\epsilon}$ respectively.

The transformations are defined through functions $h_1 : \mathbb{R}^{T\times d} \to \mathbb{R}^{T\times H}$, $h_\mu, h_\sigma : \mathbb{R}^{T\times H} \to \mathbb{R}^{T\times M\times K}$ parameterized with $\kappa_1, \kappa_\mu, \kappa_\sigma$ respectively, and implemented with convolutional neural networks (CNN) Lecun et al. (1998); Ismail Fawaz et al. (2020). The use of deep CNNs enables the extraction of complex non-linear dynamics, with a limited number of trainable parameters. Since q_κ approximates Markov chains, the convolution kernels need not be very large, which further reduces the number of trainable parameters. The function h_1 computes an intermediate representation of the data that is common to the translation and scale transformations. This enables relevant dynamics to be extracted once and shared between the two sub-branches of the model. The sampling process of states from the variational distribution is illustrated in figure 2.

To compute the probability of a drawn state $\mathbf{s}$, we use the fact that g_κ is a diffeomorphism wrt $\boldsymbol{\epsilon}$, together with the change of variable formula:

$$\ln q_\kappa(\mathbf{s}|\mathbf{x}) = \ln p(\boldsymbol{\epsilon}) + \ln \left| \det \frac{\partial g_\kappa(\cdot\,; \mathbf{x})}{\partial \boldsymbol{\epsilon}} \right|^{-1} = \ln p(\boldsymbol{\epsilon}) + \ln \left| \det \frac{\partial \text{softmax}_{\tau,\delta}(\cdot)}{\partial \boldsymbol{\epsilon}} \right|^{-1} - \mathbf{z}_\sigma. \tag{5}$$

The function $f = \text{softmax}_{\tau,\delta}$ was proposed in Potapczynski et al. (2020) and is defined:

$$\text{softmax}_{\tau,\delta}(\mathbf{y})_k = \frac{\exp(y_k/\tau)}{\sum_{j=1}^{K-1} \exp(y_j/\tau) + \delta}, \tag{6}$$

where $\mathbf{y} \in \mathbb{R}^K$, $\delta > 0$ ensures that the function is invertible and $\tau > 0$ is a temperature parameter such that the output tends to a discrete distribution (one-hot vectors) when $\tau \to 0$. Following the development in Potapczynski et al. (2020), the determinant of the Jacobian of this modified softmax is obtained:

$$\ln|\det J_f| = 2(1-K)\ln a + \ln\left|1 - \tau \sum_{k=1}^{K-1} \frac{e^{y_k/\tau}}{y_k}\right| - (K-1)\ln\tau + \sum_{k=1}^{K-1} \ln|y_k| + \frac{y_k}{\tau}, \tag{7}$$

where $a = \delta + \sum_{k=1}^{K-1} e^{y_k/\tau}$. The relaxation to a continuous state space imposes constraints on the use of the transition matrices for the Markov kernels. These constraints are implemented in the computation of the Markov chains transition kernels.

2.4.4 Markov chain transition kernels

In the traditional use of HMM, the state variables $\mathbf{s}_t$ are discrete and this allows the state transition probabilities to be modeled with a finite state transition matrix. A discrete state space is however incompatible with state variables and transition kernels differentiable wrt κ. Instead, we use a continuous state space with state vectors $\mathbf{s}_t \in \mathcal{S}$. The function $\text{softmax}_{\tau,\delta}$ (see Figure 2) at the output of the variational sampling process, allow us to control the continuity of the state variables with parameter τ. The state transitions and the initialization probabilities of chain m are calculated as follows:

$$\begin{aligned} p_{\lambda'}(\mathbf{s}_t^{(m)}|\mathbf{s}_{t-1}^{(m)}) &= {\mathbf{s}_t^{(m)}}^T {P^{(m)}}^T \mathbf{s}_{t-1}^{(m)}, \\ p_{\lambda'}(\mathbf{s}_1^{(m)}) &= {\pi^{(m)}}^T \mathbf{s}_1^{(m)}, \end{aligned} \tag{8}$$

where $P^{(m)}$ and $\pi^{(m)}$ are the m-th chain transition and state initialization parameters respectively. This relaxation ensures that when τ tends to 0, the state variable tends to a one hot encoded vector, and the transition kernels tend to a matrix row and column selection (Bishop, 2006, chap. 13.2).

2.4.5 State emission distribution

We propose to model the state emission distribution by considering that an observed sample at time t is drawn from a multivariate Gaussian distribution with mean and variance that are non-linear functions of the state variable at time t. Specifically, the emission distribution of samples $\mathbf{x}_t \in \mathbb{R}^d$ given $\mathbf{s}_t \in \mathbb{R}^{M\times K}$ is defined:

$$p_\phi(\mathbf{x}_t|\mathbf{s}_t) = \Phi\left(\mathbf{t}(\mathbf{s}_t) + \mathbf{a}(\mathbf{s}_t) \odot \mathbf{x}_t\right), \tag{9}$$

where Φ is the density function of a standard multivariate Normal distribution and $\mathbf{a}, \mathbf{t} : \mathbb{R}^{M\times K} \to \mathbb{R}^d$ are fully connected neural networks with parameters in ϕ. The non-linear functions $\mathbf{a}$ and $\mathbf{t}$ can be arbitrarily complex as long as they are differentiable.

The joint distribution $p_{\theta'}(\mathbf{x}, \mathbf{s})$ (Equation (1)) depends on κ through the sampling process $\mathbf{s} \sim q_\kappa(\mathbf{s}|\mathbf{x})$ and is differentiable wrt κ and ϕ. This gives us all the quantities we need to compute the approximate lower bound $\mathcal{L}^A$ in equation (3). We optimize $\mathcal{L}^A$ wrt κ using back-propagation, as described in algorithm 1.

Algorithm 1 Maximizing $\mathcal{L}^A(\theta', \kappa; \mathbf{x})$ wrt κ

1: **Input:** Data $\mathbf{x}$, number of iterations $n_v \in \mathbb{N}$, learning rate $\eta_v \in \mathbb{R}$, number of Monte Carlo samples $L \in \mathbb{N}$, current variational parameters κ
2: **for** $i = 1, \ldots, n_v$ **do**
3: Draw L samples $\mathbf{s}^{(l)} \sim q_\kappa(\mathbf{s}|\mathbf{x})$
4: $\mathcal{L}^A(\theta', \kappa; \mathbf{x}) = \frac{1}{L}\sum_{l=1}^{L} \ln p_{\theta'}(\mathbf{x}, \mathbf{s}^{(l)}) - \ln q_\kappa(\mathbf{s}^{(l)}|\mathbf{x})$,
5: $\kappa \leftarrow \kappa + \eta_v \partial_\kappa \mathcal{L}^A$
6: **end for**
7: **Output:** $q_\kappa(\mathbf{s}|\mathbf{x}) \approx p_{\theta'}(\mathbf{s}|\mathbf{x})$

2.5 Approximate M-step

The M-step performed in the Baum-Welch algorithm, uses exact expectations obtained from the true posterior $p_{\theta'}(\mathbf{s}|\mathbf{x})$.

Here we use the approximation $q_\kappa(\mathbf{s}|\mathbf{x})$ from the variational E-step to update the Markov chains parameters and the state emission distributions.

2.5.1 Updating the Markov chain parameters

The Markov chain parameters are updated with update rules similar to the Baum-Welch algorithm. The difference here is that the state posteriors are not computed exactly but derived from the learnt variational distributions. The latent space model parameters are updated wrt the approximate posterior $q_\kappa(\mathbf{s}|\mathbf{x})$, as described in algorithm 2.

Algorithm 2 Updating the Markov chains parameters.

1: **Input:** Approximation $q_\kappa(\mathbf{s}|\mathbf{x}) \approx p_{\theta'}(\mathbf{s}|\mathbf{x})$.
2: **for** $m = 1, \dots, M$ **do**
3: $\quad P_{i,j}^{(m)} = \dfrac{\sum_{t=2}^{T} \mathbb{E}_{q_\kappa}[\mathbf{s}_{t,i}^{(m)} \mathbf{s}_{t-1,j}^{(m)}]}{\sum_{t=2}^{T} \mathbb{E}_{q_\kappa}[\mathbf{s}_{t-1,j}^{(m)}]}$, $i, j = 1, \dots, K$,
4: $\quad \pi^{(m)} = \mathbb{E}_{q_\kappa}[\mathbf{s}_1^{(m)}]$
5: **end for**
6: **Output:** Updated markov chain parameters λ

The expectations wrt the approximate posterior distribution in algorithm 2 are computed:

$$\mathbb{E}_{q_\kappa}[\mathbf{s}_t^{(m)}] \approx \frac{1}{L} \sum_{l=1}^{L} g_\kappa(\boldsymbol{\epsilon}^{(l)}; \mathbf{x})_t^{(m)} \text{ and } \mathbb{E}_{q_\kappa}[\mathbf{s}_{t-1}^{(m)} \mathbf{s}_t^{(m)}] \quad \approx \frac{1}{L} \sum_{l=1}^{L} g_\kappa(\boldsymbol{\epsilon}^{(l)}; \mathbf{x})_t^{(m)} g_\kappa(\boldsymbol{\epsilon}^{(l)}; \mathbf{x})_{t-1}^{(m)}, \tag{10}$$

with $\boldsymbol{\epsilon}^{(l)}$ for $l = 1, \dots, L$ sampled independently from $p(\boldsymbol{\epsilon})$.

2.5.2 Updating the emission distribution

The emission distribution parameters are optimized using mini-batch back-propagation on the MC approximation of the joint distribution (see algorithm 3).

Algorithm 3 Updating the emission distribution parameters ϕ

1: **Input:** Data $\mathbf{x}$, $q_\kappa \approx p_{\theta'}$ the variational approximation, $\eta_e \in \mathbb{R}_+$ the learning rate
2: Draw L samples $\mathbf{s}^{(l)} \sim q_\kappa(\mathbf{s}|\mathbf{x})$ {Figure 2}
3: Compute $\mathcal{G}(\phi) \approx \frac{1}{L} \sum_{l=1}^{L} \ln p_{\{\lambda', \phi\}}(\mathbf{x}, \mathbf{s}^{(l)})$
4: $\phi \leftarrow \phi + \eta_e \partial_\phi \mathcal{G}$
5: **Output:** Updated ϕ

2.6 End-to-end training algorithm

The model is initialized with λ, ϕ and κ, the set of parameters for the latent space, the generative model and the variational distribution respectively, $n \in \mathbb{N}$ the total number of training epochs, $n_v \in \mathbb{N}$ and $\eta_v > 0$ the number of iterations and the learning rate in the VLB maximization (see algorithm 1), $\eta_e > 0$ the learning rate for the emission distribution update (see algorithm 3), $L \in \mathbb{N}$ the number of Markov samples. The training procedure is summarized in algorithm 4.

Algorithm 4 Training a factorial HMM

Input: Training sequences $\mathbf{x} \in \mathbb{R}^{T \times d}$, initialized factorial HMM and variational distribution.
for $i = 1, \ldots, n$ **do**
  for $\mathbf{x}$ in the training dataset **do**
    Approximate state posterior distribution {algorithm 1}
    Accumulate Markov chains parameters λ updates {algorithm 2}
    Update Generative model parameters ϕ {algorithm 3}
  end for
  Update Markov chains parameters λ {algorithm 2}
end for
Output: Trained factorial HMM

3 Experiments

We first show a proof of principle of our learning algorithm on synthetic data. The goal of this experiment is to retrieve and identify processes from a data sequence using the sequence of states of independent Markov chains in the model. For ease of visualisation and *interpretation*, we used univariate input data sequences composed of two independent additive processes. We then show that, using a real world dataset, our model has better modeling abilities when M increases with fixed $K = 2$, compared to the case where K increases with fixed $M = 1$. For all our experiments, the number of MC samples was fixed to $L = 10$. To reduce the training time, we pre-warm our training by first initializing 3 models randomly, and then choosing the model with the highest log-likelihood on the training data. We used Pytorch Paszke et al. (2019) to implement our model. The inference time reported in figure 3 are obtained on 1× *Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz*. The rest of the experiments were carried out on a compute server running 2× *Intel(R) Xeon(R) Gold 6230 CPU @ 2.10GHz* and 4× *Tesla V100-SXM2-16GB*.

3.1 Synthetic data

We demonstrate the ability of our system to recover independent processes from a scalar time series on synthetic data. We generated data from two independent additive processes of dimension $d = 1$. The two processes are periodic step functions of height 1 with equal period of 20 samples and width 5 and 10 samples respectively. This means that the signal reaches the value 2 when both processes are active, it has value 1 when only the second process is active, it reaches value 0 when the two processes are inactive. We added white Gaussian noise to the generated sequences, and created 30 sequences of length 60 samples each. An example sequence is depicted in black in figure 4. We chose a generative model where the scaling transform $\mathbf{a}$ is fixed to a constant scalar, and the translation transform $\mathbf{t}$ is linear and trainable.

3.2 Patient monitoring dataset

We then tested our model on a real dataset obtained from neonatal intensive care unit from Philips IntelliVue MX800 patient monitors (Philips Healthcare, Amsterdam, Netherlands). All the patients were born with very low birth weights ($<$ 1500g) and experienced late-onset or early onset sepsis during their recording period. We included both culture positive sepsis and culture negative sepsis if the patient was treated with antibiotics for more than 5 days Persad et al. (2021). The vital signs we obtained were the ECG derived inter-beat-interval (IBI) signal, the oxygen saturation level derived from pulse oximetry and the impedance derived respiration frequency signal. We built non-overlapping time windows of duration 1 hour. For each individual time window, we computed the minimum, maximum, mean, standard deviation, kurtosis and skewness of each vital sign Persad et al. (2021). Additionally, we computed the sample entropy and the sample asymmetry of the IBI signal following Griffin et al. (2003); Joshi et al. (2019). This resulted in 20 features per time window. We also used repetitive body weight data and the postnatal age of patients at the start of the time window. We included patient demographics as constant features: the sex and the birth weight. In total, our dataset comprised 16 patient time series of dimension 24. Overall, this corresponded to 74.9 days (1049 hours) of recording and an average recording length per patient of 66.5 ± 87.6 hours.

We chose a generative model where both the scaling and the translation transforms are two trainable 2-layer fully connected networks. The learning rate for the generative model was fixed to $\eta_e = 0.01$ (see algorithm 3).

3.3 Variational distribution design

We designed the internal architecture of the variational distribution such that: h_1 is (1) a 3-layer CNN with rectified linear unit (ReLU) activations and (2) has inputs processed as single channel $T \times d$ images. The first convolution kernels is of size $5 \times d$, and the last two of size 5×1. We used zero-padding and feature maps of size 10 so that each input to a CNN layer was transformed into a 10-channels $T \times 1$ signal. We did not use pooling layers in order to preserve the length of the time series accross the CNN layers. This is similar to the idea employed in fully convolutional networks to perform image segmentation Long et al. (2015). h_σ and h_μ were 1-layer CNNs with kernels of size 3×1. We chose a smaller kernel size since we wanted to have the largest kernels in the shared transform h_1. The two transforms map the 10 channel output of h_1 to a $M \times K$-dimensional signal of length T (see figure 2). We used *sigmoid* as the activation function for h_σ to restrict the range of the input to the exponential function, and *tanh* for the output of h_μ since there is no positivity constraint. We fixed the learning rate to $\eta_v = 0.01$ and the number of iterations to $n_v = 10$ (see algorithm 1). We chose $\tau = 0.5$ for the $\text{softmax}_{\tau,\delta}$ function. We found that with $\delta > 0$, the learning algorithm does not converge. In practice a traditional non-invertible softmax works well although making the training not theoretically sound. We thus fixed $\delta = 0$.

4 Results

4.1 Model sanity check

On figure 3, we see that the number of parameters does not vary with T but varies linearly with M, and that the inference time and the number of float operations vary linearly with both T and M. This is in line with our target architecture.

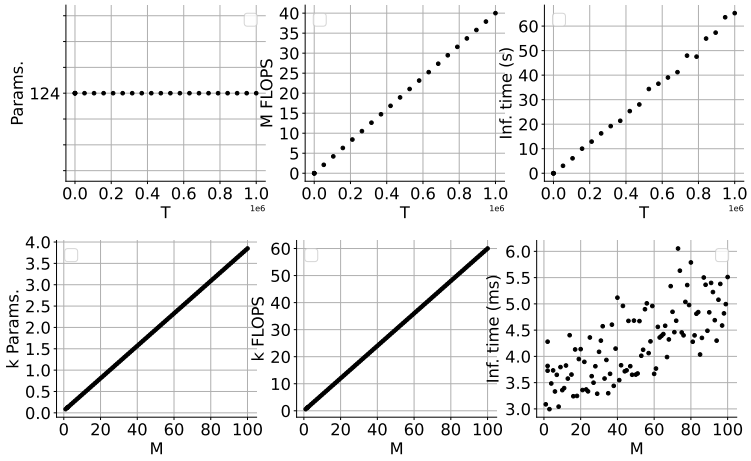

Figure 3: Analysis of the influence of **(row 1)** T and **(row 2)** M on **(column 1)** the number of trainable parameters, **(column 2)** the number of float operations and **(column 3)** the inference time. Each row corresponds to varying one hyper parameter and fixing the other. The default fixed values are $T = 30$, $M = 2$.

4.2 Proof of concept on synthetic data

The results depicted on the LHS in figure 4, show that our learning scheme iteratively increases the VLB (in black) and the expected joint distributions (in green). We observe that the log variational probability (in red) does not decrease as we expected during training. This is probably due our approximation of the variational probability caused by $\delta = 0$. On the RHS figure 4, we observe that the sequence of states of Markov chains

2 and 3 (in red and blue) are enough to retrieve the two independent additive processes in the data. The extra latent chain (Markov chain 1 in green) does not disturb the recovery of the two independent additive processes. We note however a larger variance of the state for Markov chain 1 across Monte Carlo samples. This likely indicates that the model had difficulties to use the extra latent chain efficiently. We plotted the normalized correlation matrix between the sequences of states learnt by the model on the bottom figure 4. We see an anti-correlation between Markov chain 1 and 2, indicating these latent Markov chains are likely learning the process opposite of one another. This in turns means that two independent chains might be enough in our case.

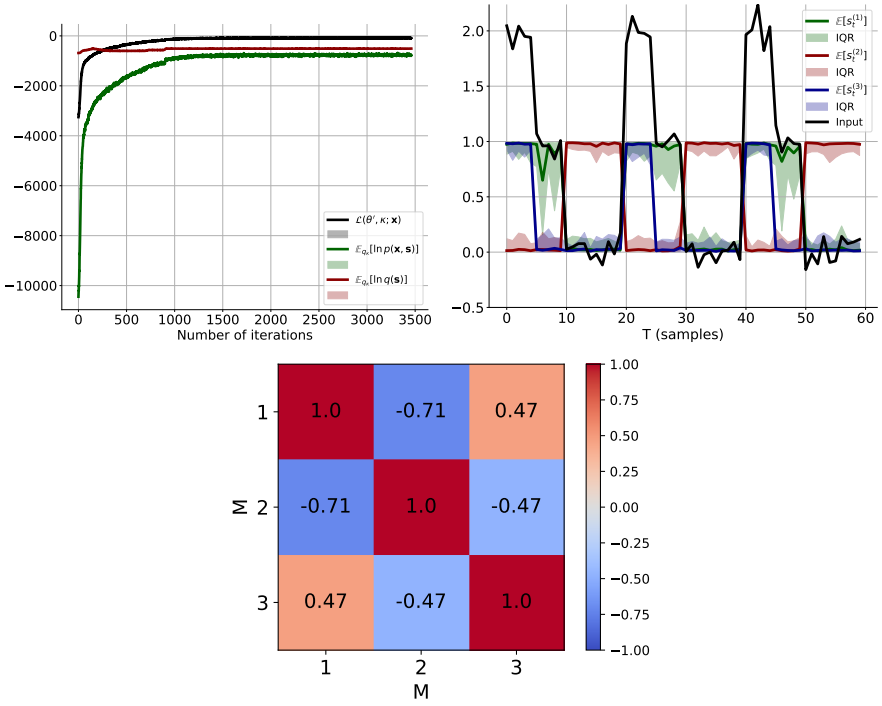

Figure 4: **(Top Left)** Approximate joint distribution (green), expected log variational probability (red) and the resulting VLB (black) over training iterations. **(Top Right)** The input time series data depicted in black results from two independent additive scalar processes. We used a model with $M = 3$ independent Markov chains. The inferred sequences of states for each independent Markov chain in a trained model are depicted in green, red and blue. **(Bottom)** The normalized correlation matrix of the inferred states for a given time series $\mathbf{x}$. The matrix should be diagonal for de-correlated states.

4.3 Modeling patient data

We show the training and validation results of a leave-one-out (LOO) cross validation scheme in table 1. As expected, increasing M with a fixed $K = 2$ is more beneficial to both the training and validation log-likelihood compared to increasing K with fixed $M = 1$. The best median training and validation log-likelihoods are obtained for $M = 9$ and $K = 2$. We note that the IQR of the validation log-likelihoods is multiplied roughly by a factor 5 between $M = 2$ and $M = 9$. This shows that the model overfits the data when the order of the model becomes too large. The log-likelihoods for the standard HMM do not vary significantly as K increases. The performances of this model are consistently lower than that of the factorial HMM.

5 Discussion

In this work, we showed that our variational Bayes EM learning algorithm is able to fit factorial HMM models as we expected. Our proposed model fulfills the design constraints: (1) the number of trainable parameters is

	factorial HMM (K=2)							
M	2	3	4	5	6	7	8	9
Train. LL	4288 (248)	6486 (381)	8655 (539)	10836 (660)	12943 (649)	15386 (874)	17706 (1042)	**19965** (1135)
Val. LL	1686 (3494)	2544 (5066)	3382 (6908)	3914 (6420)	5064 (10235)	5927 (12091)	6769 (13782)	**7624** (15493)
	standard HMM (M=1)							
K	2	3	4	5	6	7	8	9
Train. LL	**2169** (171)	2115 (166)	2114 (132)	2047 (112)	2078 (128)	2025 (152)	2029 (144)	2054 (111)
Val. LL	**840** (1726)	836 (1615)	764 (1248)	820 (1613)	818 (1613)	753 (1235)	809 (1628)	806 (1547)

Table 1: Training and validation log-likelihood for standard and factorial HMM. The results are presented as median (IQR) on the leave-one-out cross validation scheme.

independent from the length of the input time series (figure 3) and (2) the state emission distributions can be chosen arbitrarily complex without the need for new update rules for the model parameters to be derived. This makes our model particularly amenable to situations where complex processes are to be disentangled in long time series. We also showed that our model with several independent Markov chains is able to fit patient vital signs data better than a standard model with a single Markov chain (see table 1).

Sensitivity analysis The sensitivity of our model to hyper-parameters is not quantitatively studied in this work. Hyper parameters related to training (learning rate, number of iterations, regularization parameters) determine the speed of convergence and the fit quality. Other hyper parameters such as the number and size of hidden layers, or convolutional kernel sizes are related to the model architecture. The optimal tuning of these parameters is typically difficult to derive and is problem dependent. It also requires a large enough dataset in order to avoid overfitting. Here we chose the sizes of the convolution kernels on the basis of the model our variational distribution approximates (an order 1 latent Markov chains). Larger convolution kernels could be evaluated to approximate higher order HMM. Other hyper parameters were chosen experimentally. We observed that the training is sensitive to initialisation. This lead us to "pre-warm" our models before training (see section 3). The model training is also sensitive to the learning rate for emission parameters especially in cases where both inference networks and emission distributions are to be learnt. The quantification of these behaviors is left for future studies.

Dependent processes Physiological processes often occur correlated in practice. Some patterns characteristic of different underlying physiological processes might be visible simultaneously. This means that our model should be able to explain the variance in the observed data using several latent Markov chains. Although our model relies on independent latent Markov chains, it is not restricted to learning decorrelated sequences of states (see Figure 4). Ambiguity might however arise when interpreting the prominence of each physiological process, in particular if some variance can be explained by several physiological processes.

Limitations We identified some weaknesses in our approach. Our variational distribution does not explicitly use the Markov chain structure to model the sequence of states. Imposing a strict Markov constraint in our variational distribution could further reduce the amount of parameters. Also, a known problem when modeling transition kernels with transition matrices, is that the probability of a state duration of length D decreases exponentially with D (Bishop, 2006, eq. 13.74). A potential solution to this would be to model the state duration directly, e.g. with Poisson distributions. This would lead to models even more applicable to long time series were the underlying processes are expected to have a long duration.

6 Conclusion

We proposed a novel variational distribution and an EM based learning algorithm to scale factorial HMM to long and complex timeseries. Our model has the property of having a number of trainable parameters independent from the length of the input data, allowing arbitrarily complex state emission distributions. These two characteristics are major steps toward scaling factorial HMM models to long time sequences. We showed that our model was able to recover independent additive processes on synthetic data. We showed that our model outperforms standard HMM, even with a large state space, on modeling patient vital signs data. Finally, our model opens opportunities to incorporate disease computational models into patient monitoring surveillance time series models. This in turn could lead to more reliable automatized predictive monitoring of patient vital signs in hospitals.

References

Antonio Albanese, Limei Cheng, Mauro Ursino, and Nicolas W. Chbat. An integrated mathematical model of the human cardiopulmonary system: Model development. *American Journal of Physiology-Heart and Circulatory Physiology*, 310(7):H899–H921, December 2015. ISSN 0363-6135. doi: 10.1152/ajpheart.00230.2014.

Leonard E. Baum and Ted Petrie. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, December 1966. ISSN 0003-4851, 2168-8990. doi: 10.1214/aoms/1177699147.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 978-0-387-31073-2.

Sébastien de Bournonville, Antoine Pironet, Chris Pretty, J. Geoffrey Chase, and Thomas Desaive. Parameter estimation in a minimal model of cardio-pulmonary interactions. *Mathematical Biosciences*, 313:81–94, July 2019. ISSN 0025-5564. doi: 10.1016/j.mbs.2019.05.003.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data Via the *EM* Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, September 1977. ISSN 00359246. doi: 10.1111/j.2517-6161.1977.tb01600.x.

Laura Ellwein Fix, Joseph Khoury, Russell R. Moores, Lauren Linkous, Matthew Brandes, and Henry J. Rozycki. Theoretical open-loop model of respiratory mechanics in the extremely preterm infant. *PLoS ONE*, 13(6):e0198425, June 2018. ISSN 1932-6203. doi: 10.1371/journal.pone.0198425.

Nick Foti, Jason Xu, Dillon Laird, and Emily Fox. Stochastic variational inference for hidden Markov models. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Zoubin Ghahramani and Michael I. Jordan. Factorial Hidden Markov Models.:. Technical report, Defense Technical Information Center, Fort Belvoir, VA, January 1996.

M. Pamela Griffin, T. Michael O'Shea, Eric A. Bissonette, Frank E. Harrell, Douglas E. Lake, and J. Randall Moorman. Abnormal Heart Rate Characteristics Preceding Neonatal Sepsis and Sepsis-Like Illness. *Pediatric Research*, 53(6):920, June 2003. ISSN 1530-0447. doi: 10.1203/01.PDR.0000064904.05313.D2.

Gustavo Guerrero, Virginie Le Rolle, Corinne Loiodice, Amel Amblard, Jean-Louis Pépin, and Alfredo Hernández. Modeling Patient-Specific Desaturation Patterns in Sleep Apnea. *IEEE Transactions on Biomedical Engineering*, 69(4):1502–1511, April 2022. ISSN 1558-2531. doi: 10.1109/TBME.2021.3121170.

A. Honoré, D. Liu, D. Forsberg, K. Coste, E. Herlenius, S. Chatterjee, and M. Skoglund. Hidden Markov Models for Sepsis Detection in Preterm Infants. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1130–1134, May 2020. doi: 10.1109/ICASSP40776.2020.9054635.

Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F. Schmidt, Jonathan Weber, Geoffrey I. Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. InceptionTime: Finding AlexNet for time series classification. *Data Min Knowl Disc*, 34(6):1936–1962, November 2020. ISSN 1573-756X. doi: 10.1007/s10618-020-00710-y.

Rohan Joshi, Deedee Kommers, Laurien Oosterwijk, Loe Feijs, Carola Van Pul, and Peter Andriessen. Predicting Neonatal Sepsis Using Features of Heart Rate Variability, Respiratory Characteristics and ECG-Derived Estimates of Infant Motion. *IEEE Journal of Biomedical and Health Informatics*, pp. 1–1, 2019. ISSN 2168-2194, 2168-2208. doi: 10.1109/JBHI.2019.2927463.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*, May 2014.

Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-Based Learning Applied to Document Recognition. In *Proceedings of the IEEE*, pp. 2278–2324, 1998.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, June 2015. doi: 10.1109/CVPR.2015.7298965.

Gautham Mysore and Maneesh Sahani. Variational Inference in Non-negative Factorial Hidden Markov Models for Efficient Audio Source Separation. *arXiv:1206.6468 [cs, stat]*, June 2012.

Yin Cheng Ng, Pawel Chilinski, and Ricardo Silva. Scaling Factorial Hidden Markov Models: Stochastic Variational Inference without Messages. *arXiv:1608.03817 [stat]*, October 2016.

John Paisley, David M. Blei, and Michael I. Jordan. Variational Bayesian inference with stochastic search. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pp. 1363–1370, Madison, WI, USA, June 2012. Omnipress. ISBN 978-1-4503-1285-1.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, number 721 in 1, pp. 8026–8037. Curran Associates Inc., Red Hook, NY, USA, December 2019.

Emma Persad, Kerstin Jost, Antoine Honoré, David Forsberg, Karen Coste, Hanna Olsson, Susanne Rautiainen, and Eric Herlenius. Neonatal sepsis prediction through clinical decision support algorithms: A systematic review. *Acta Paediatrica*, 110(12):3201–3226, 2021. ISSN 1651-2227. doi: 10.1111/apa.16083.

Andres Potapczynski, Gabriel Loaiza-Ganem, and John P Cunningham. Invertible gaussian reparameterization: Revisiting the gumbel-softmax. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 12311–12321. Curran Associates, Inc., 2020.

Peter Smit, Sami Virpioja, and Mikko Kurimo. Advances in subword-based HMM-DNN speech recognition across languages. *Computer Speech & Language*, 66:101158, March 2021. ISSN 0885-2308. doi: 10.1016/j.csl.2020.101158.

Ron J. Weiss and Daniel P. W. Ellis. A variational EM algorithm for learning eigenvoice parameters in mixed signals. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 113–116, April 2009. doi: 10.1109/ICASSP.2009.4959533.