Deep parameterizations of pairwise and triplet Markov models for unsupervised classification of sequential data

Hugo Gangloff, Katherine Morales, Yohan Petetin

Deep parameterizations of pairwise and triplet Markov models for unsupervised classification of sequential data

Hugo Gangloff*, Katherine Morales and Yohan Petetin

Samovar, Telecom Sudparis, Institut Polytechnique de Paris, 91011 Évry, France

ARTICLE INFO

Keywords:
Pairwise Markov Chains
Triplet Markov Chains
Deep Neural Networks
Variational Expectation-Maximization
Image Segmentation

ABSTRACT

Hidden Markov models are probabilistic graphical models based on hidden and observed random variables. They are popular to address classification tasks for time series applications such as part-of-speech tagging, image segmentation, genetic sequence analysis. We focus on direct extensions of these models, the pairwise and triplet Markov models. These models aim at relaxing the assumptions underlying the hidden Markov chain by extending the direct dependencies of the involved random variables or by considering the addition of a third latent process. While these extensions define interesting modeling capabilities that have been little explored so far, they also raise new problems such as defining the nature of their core probability distributions and their parameterization. Once the model is fixed, the unsupervised classification task (*i.e.* the estimation of the parameters and next of the hidden random variables) is a critical problem. We address these challenges in this paper. We first show that it is possible to embed recent deep neural networks in these models in order to exploit their full modeling power; we also consider a continuous latent process in triplet Markov chains which aims at estimating the nature of the joint distributions of the hidden and observed random variables, in addition to their parameters. For each model that we introduce, we propose an original Bayesian unsupervised estimation method which can take into account the interpretability of the hidden random variables in terms of signal processing classification. Through unsupervised classification problems on synthetic and real data, we show that the new models outperform hidden Markov chains and their classical extensions usually considered in the literature.

1. Introduction

Let $\boldsymbol{x}_K = (x_0, x_1, \dots, x_K)$, $x_k \in \mathbb{R}^{d_x}$, $\boldsymbol{h}_K = (h_0, h_1, \dots, h_K)$, $h_k \in \Omega = \{\omega_1, \dots, \omega_C\}$ and $\boldsymbol{z}_K = (z_0, z_1, \dots, z_K)$, $z_k \in \mathbb{R}^{d_z}$, be three sequences of observed, hidden and auxiliary latent random variables (r.v.) of length $K+1$, respectively. As far as notations are concerned, we do not distinguish r.v. and their realizations. By hidden, we mean that h_k is an unobserved r.v. that we wish to estimate from $\boldsymbol{x}_K$. It represents an interpretable class associated to x_k contrary to z_k which is an intermediate auxiliary latent r.v. and which may depend on $(\boldsymbol{x}_K, \boldsymbol{h}_K)$. The mathematical expectation of $f(x)$ under the distribution $p(x)$ is denoted as $\mathbb{E}_{p(x)}(f(x))$. Finally, for $k' > k$, we note the sequence $\boldsymbol{x}_{k:k'} = (x_k, \dots, x_{k'})$; note that when the sequence is considered from the beginning, we have $\boldsymbol{x}_k = \boldsymbol{x}_{0:k}$.

1.1. Unsupervised Bayesian classification in hidden Markov models

The estimation of h_k from $\boldsymbol{x}_K$, for all k, $0 \leq k \leq K$, relies on the unknown posterior distribution $p(h_k|\boldsymbol{x}_K)$ and involves several challenges. The first one consists in modeling the unknown joint distribution of $(\boldsymbol{h}_K, \boldsymbol{x}_K)$ by a relevant parametric distribution $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{x}_K)$. It can coincide with the marginal distribution of a distribution in augmented dimension $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{z}_K, \boldsymbol{x}_K)$ through the introduction of an auxiliary process $\boldsymbol{z}_K$. Once a class of distributions $p_{\boldsymbol{\theta}}$ has been chosen, the objective is to estimate the parameter $\boldsymbol{\theta}$ from a realization $\boldsymbol{x}_K$ in an unsupervised way, that is to say without observing $\boldsymbol{h}_K$ and $\boldsymbol{z}_K$. A popular estimate is the Maximum-Likelihood (ML) estimate $\hat{\boldsymbol{\theta}}^{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\boldsymbol{x}_K)$ due to its statistical properties [White, 1982, Douc and Moulines, 2012]. Even if $\hat{\boldsymbol{\theta}}^{\mathrm{ML}}$ is generally not computable, several computational methods have been proposed to approximate it [Dempster et al., 1977, Balakrishnan et al., 2017]. Finally, for a given estimate $\hat{\boldsymbol{\theta}}$, it remains to compute or approximate the posterior distribution $p_{\hat{\boldsymbol{\theta}}}(h_k|\boldsymbol{x}_K)$ and to deduce, for example, the Maximum A Posteriori (MAP) estimates defined as $\arg\max_{h_k \in \Omega} p_{\hat{\boldsymbol{\theta}}}(h_k|\boldsymbol{x}_K)$, for all k. Again, and according to the distribution $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{x}_K)$, the MAP estimates can be computed either exactly or approximately. In

*Corresponding author. Hugo Gangloff is now with IRISA, Université Bretagne Sud, UMR 6074, 56000 Vannes, France.
hugo.gangloff@irisa.fr (H. Gangloff); katherine.morales_quinga@telecom-sudparis.eu (K. Morales); yohan.petetin@telecom-sudparis.eu (Y. Petetin)

summary, the distribution $p_{\boldsymbol{\theta}}$ should properly describe the r.v. introduced for a given application (*i.e.* the relations between the hidden and observed r.v. described by the model are realistic) t should also enable us to compute or approximate the desired estimates at a reasonable computational cost. An additional challenge in our case is that the estimated r.v. h_k, associated to the final model, has to be interpretable in terms of signal processing classification, even if when we only have $\boldsymbol{x}_K$ at our disposal to compute these estimates.

When we deal with sequential data (*e.g.* text, video, music, protein sequences, images) a popular model is the Hidden Markov Chain (HMC) [Rabiner, 1989]. In a HMC, the sequence $\boldsymbol{h}_K$ is a Markov chain and given $\boldsymbol{h}_K$, the observations x_k are independent and only depend on the corresponding h_k. In other words, $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{x}_K)$ satisfies

$$p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{x}_K) = p_{\boldsymbol{\theta}}(h_0) \prod_{k=1}^{K} p_{\boldsymbol{\theta}}(h_k|h_{k-1}) \prod_{k=0}^{K} p_{\boldsymbol{\theta}}(x_k|h_k), \tag{1}$$

where $p_{\boldsymbol{\theta}}(h_k|h_{k-1})$ and $p_{\boldsymbol{\theta}}(x_k|h_k)$ are the distributions representing the transitions of the Markov chain $\boldsymbol{h}_K$ and the relations between the observation and the class, respectively. Because h_k is discrete, the computation of the associated posterior distribution $p_{\boldsymbol{\theta}}(h_k|\boldsymbol{x}_K)$ can be done with the Forward-Backward algorithm [Rabiner, 1989]. More precisely, in some particular HMC models (if $p_{\boldsymbol{\theta}}(x_k|h_k)$ is a Gaussian mixture, for example), it is possible to approximate the ML estimate of $\boldsymbol{\theta}$ with the Baum-Welch algorithm [Rabiner, 1989], an adaptation of the Expectation-Maximation (EM) algorithm [Dempster et al., 1977] for sequential data.

These models and their associated inference algorithms have been extended in multiple directions. In particular, the Pairwise Markov Chain (PMC) model is a direct generalization of (1) where we only assume that the pair $(\boldsymbol{h}_K, \boldsymbol{x}_K)$ is Markovian [Pieczynski, 2003, Le Cam et al., 2008, Morales and Petetin, 2021],

$$p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{x}_K) = p_{\boldsymbol{\theta}}(h_0, x_0) \prod_{k=1}^{K} p_{\boldsymbol{\theta}}(h_k, x_k|h_{k-1}, x_{k-1}). \tag{2}$$

In such a model, $\boldsymbol{h}_K$ is not necessarily a Markov chain and the observations can now be dependent given $\boldsymbol{h}_K$. Finally, PMC models can, in turn, be generalized to Triplet Markov Chain (TMC) models [Pieczynski, 2002, Pieczynski and Desbouvries, 2005] by adding a third latent process $\boldsymbol{z}_K$ and assuming that the triplet $(\boldsymbol{h}_K, \boldsymbol{z}_K, \boldsymbol{x}_K)$ is Markovian,

$$p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{z}_K, \boldsymbol{x}_K) = p_{\boldsymbol{\theta}}(h_0, z_0, x_0) \prod_{k=1}^{K} p_{\boldsymbol{\theta}}(h_k, z_k, x_k|h_{k-1}, z_{k-1}, x_{k-1}). \tag{3}$$

In the case where $\boldsymbol{h}_K$ is discrete, such models have been mainly used with a discrete latent process $\boldsymbol{z}_K$ [Gorynin et al., 2018, Lanchantin et al., 2008, Pieczynski, 2007], whereas a continuous latent process has been used when $\boldsymbol{h}_K$ is also continuous [Lehmann and Pieczynski, 2020].

Models (2) and (3) provide interesting extensions of (1); however, these generalizations involve several issues that we address in this paper. From a modeling point of view, the choice of the transitions distributions is a thorny problem. For example, in (2), the choice of $p_{\boldsymbol{\theta}}(h_k, x_k|h_{k-1}, x_{k-1})$ and so that of $p_{\boldsymbol{\theta}}(h_k|h_{k-1}, x_{k-1})$ is not obvious because of the relation between h_k and x_{k-1}. In addition, the interpretability of h_k, which may be valid in a HMC, is not necessarily satisfied when we introduce the generalization (2) because the observation x_k not only depends on h_k but also on h_{k-1}.

Consequently, and up to our best knowledge, the direct application of these extensions for unsupervised estimation (*i.e.* the joint estimation of $\boldsymbol{\theta}$ and of h_k from $\boldsymbol{x}_K$) has been restricted to a subclass of PMC and TMC models which rely on additional assumptions. First, it has been assumed that $(\boldsymbol{h}_K, \boldsymbol{x}_K)$ (resp. $(\boldsymbol{h}_K, \boldsymbol{z}_K, \boldsymbol{x}_K)$, where z_k is discrete) is a stationary process [Pieczynski, 2003, Gorynin et al., 2018]; so the distribution $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{x}_K)$ (resp. $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{z}_K, \boldsymbol{x}_K)$) is directly described by the initial distribution $p_{\boldsymbol{\theta}}(\boldsymbol{h}_{0:1}, \boldsymbol{x}_{0:1}) = p_{\boldsymbol{\theta}}(\boldsymbol{h}_{0:1}) p_{\boldsymbol{\theta}}(\boldsymbol{x}_{0:1}|\boldsymbol{h}_{0:1})$ (resp. $p_{\boldsymbol{\theta}}(\boldsymbol{h}_{0:1}, \boldsymbol{z}_{0:1}, \boldsymbol{x}_{0:1}) = p_{\boldsymbol{\theta}}(\boldsymbol{h}_{0:1}, \boldsymbol{z}_{0:1}) p_{\boldsymbol{\theta}}(\boldsymbol{x}_{0:1}|\boldsymbol{h}_{0:1}, \boldsymbol{z}_{0:1})$). These distributions coincide with a discrete distribution on $\Omega \times \Omega$ and a conditional continuous one on $\mathbb{R}^{d_x} \times \mathbb{R}^{d_x}$, respectively; they are thus easier to model in the sense that the conditioning does not depend on a continuous r.v. Next, by choosing a classical distribution for the conditional distributions of the observations $(\boldsymbol{x}_{0:1})$ (*e.g.* a Gaussian one whose parameters depend on $\boldsymbol{h}_{0:1}$), they can be easily estimated by popular estimation algorithms [Gorynin et al., 2018]. Finally, it has been observed that under these assumptions, the interpretability of the hidden r.v. h_k has been kept as long as it is satisfied in the underlying HMC model. However, the stationarity assumption is restrictive in practice, the models introduced in this study do not require such hypothesis.

1.2. Classification algorithms based on deep neural networks

In parallel, classification algorithms based on (deep) neural networks (DNNs) have known a regain of interest during the last years. This is mainly due to their great performances for several applications such that speech recognition [Deng et al., 2013, Chan et al., 2016, Abdel-Hamid et al., 2013], image recognition [Fu et al., 2017, Traore et al., 2018, Zheng et al., 2017], natural language processing [Collobert and Weston, 2008, Goldberg, 2017]. Mathematically, a DNN is a parameterized vector-valued function $f_{\boldsymbol{\theta}}(x)$, $x \in \mathbb{R}^{d_x}$, built as the sequential and alternate composition of a linear function and a non linear one. If vector x' represents the input of a given hidden layer, the scalar output of a given neuron is computed as $\sigma(wx' + b)$, where σ is a non linear activation function such as the sigmoid, the ReLU, or the hyberbolic tangent. The parameter $\boldsymbol{\theta}$ of a DNN consists of the weights and the biases which characterize the linear transformations. A major interest of this construction is twofold. First, DNNs can be seen as universal approximators in the sense that $f_{\boldsymbol{\theta}}(x)$ can theoretically approximate any vector-valued function $f(x)$, under some assumptions [Hornik et al., 1989, Pinkus, 1999]. For a classification problem of an observation x, $f_{\boldsymbol{\theta}}(x)$ aims at approximating directly the posterior distribution $p(h|x)$, for all $h \in \Omega$.

The estimation of $\boldsymbol{\theta}$ relies on the observation that the gradient of $f_{\boldsymbol{\theta}}$ w.r.t. $\boldsymbol{\theta}$ can be exactly computed with the backpropagation algorithm [Rumelhart et al., 1986]. Provided that we have at our disposal a labeled training dataset $\epsilon = \{(x^{(i)}, h^{(i)})\}_{i=1}^{n}$, it is possible to minimize a cost function $\mathcal{L}(\epsilon)$ (typically the negative log-likelihood) with a gradient descent approach [Ruder, 2016]. Contrary to the approach described in Section 1.1, note that, this approach does not model the joint probability distribution of the observations and the hidden r.v. and relies on a labeled dataset to estimate the function $f_{\boldsymbol{\theta}}$.

1.3. Contributions of this paper

As we have just seen, PMCs and TMCs have been used under several assumptions which may limit their modeling power. On the other hand, DNNs provide universal approximators but their direct use is limited to supervised classification and do not model a distribution on the observations. The aim of this paper is to propose a general framework for unsupervised signal classification which takes advantage of both of the approaches described above.

Our approach is based on the combination of the probabilistic graphical models described in Section 1.1 and on the DNNs described in Section 1.2. In terms of modeling, this approach has several advantages. First, a direct consequence is that (2)-(3) do not require any additional assumption on the involved distributions. Next, we are able to propose powerful probabilistic PMC or TMC models in the sense that their associated conditional distributions are now parameterized by universal approximators (DNNs) in the spirit of the Variational Auto-Encoders (VAEs) [Kingma and Welling, 2014]. However, while VAEs and their extensions [Chung et al., 2015, Gregor et al., 2015] aim at building powerful generative models (*i.e.* an expressive probability distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}_K)$ on the observations), our objective is to propose an expressive joint distribution $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{x}_K)$ under the constraint that $\boldsymbol{h}_K$ is an interpretable hidden process. Next, a main advantage of embedding the DNN framework into a probabilistic framework is that it is possible to derive unsupervised Bayesian estimation algorithms to jointly estimate $\boldsymbol{\theta}$ and h_k, for all k. The counterpart of this generalization is that the resulting models can be highly parameterized in such a way that the final estimated models can suffer from a lack of interpretability as compared to the simple HMC (1) (*i.e.* the estimated r.v. h_k cannot be interpreted as a physical class associated to x_k). Thus, starting from a simple but interpretable model, we include this constraint in our parameterized models and their associated Bayesian inference algorithms. Our models are based on the declination of the general TMCs (3) in three versions and aim at modeling different kinds of problems:

- first, we consider a model in which we directly parameterize the joint distribution $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{x}_K)$ of a PMC (*i.e.* we consider a TMC (3) without any latent process $\boldsymbol{z}_K$). In this model, we introduce a general parameterized framework which next enables us to use DNNs as a parameterization of the transition distribution $p_{\boldsymbol{\theta}}(h_k, x_k|h_{k-1}, x_{k-1})$ in (2). While we show that it is possible to adapt existing Bayesian inference algorithms, we propose and ad-hoc procedure based on a pretraining of DNNs which aims at transforming a simple and interpretable model such as (1) into a complex probabilistic architecture while keeping this interpretability constraint;

- in our second version of TMCs, we reintroduce a continuous latent process $\boldsymbol{z}_K$. The aim of this continuous process is to learn the nature of the distribution of $(\boldsymbol{h}_K, \boldsymbol{x}_K)$; even if the distributions underlying $p(\boldsymbol{h}_K, \boldsymbol{z}_K, \boldsymbol{x}_K)$ are simple distributions (*e.g.* Gaussian distributions for the continuous r.v.), the implicit marginal one $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{x}_K) = \int p(\boldsymbol{h}_K, \boldsymbol{z}_K, \boldsymbol{x}_K)\mathrm{d}\boldsymbol{z}_K$ can become complex and more relevant than a direct parameterization of $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{x}_K)$. However, due to the continuous nature of the latent process, the distributions of interest cannot be computed exactly;

thus, we modify the variational Bayesian inference framework [Michael et al., 1999] in order to propose a parameter estimation algorithm which takes into account the interpretability constraint of $\boldsymbol{h}_K$ but also the different roles of $\boldsymbol{h}_K$ and $\boldsymbol{z}_K$; we finally propose a Sequential Monte Carlo (SMC) algorithm [Doucet and Johansen, 2009] based on the previous variational framework to obtain the final estimates of h_k;

- in our last version of the TMC model, we propose an alternative use of the latent process $\boldsymbol{z}_K$; here, our objective is to introduce an explicit long dependency on the observations to model the joint process $(\boldsymbol{h}_K, \boldsymbol{x}_K)$. To that end, the latent process $\boldsymbol{z}_K$ becomes deterministic given the observations $\boldsymbol{x}_K$. The resulting TMC model can be interpreted as the combination of the PMC model (2) with a Recurrent Neural Network (RNN) [Rumelhart et al., 1986, Mikolov et al., 2015] in which the distributions of interest can be computed exactly, while preserving the interpretability of h_k.

For each model, we perform simulations to evaluate to what extent our new generalized models lead to a better estimation of the hidden states h_k. Most of the simulations on synthetic and real data are run in the context of unsupervised image segmentation; here, the objective is to estimate the original class h_k (*e.g.* black or white) associated to each pixel x_k of a noisy image $\boldsymbol{x}_K$. We show that our deep parameterizations and the training procedure that we propose always improve the segmentation accuracy. The results then pave the way towards a new and robust approach for unsupervised signal processing with general hidden Markov models.

The paper is organized as follows. In Section 2, we introduce a general parameterization framework for PMC models; we next review the associated Bayesian inference algorithms and we propose a particular estimation algorithm in the case where the parameterizations rely on DNNs. In Section 3, we propose a TMC model with continuous latent r.v. also based on a general parameterization and we review the variational Bayesian inference framework to propose an estimation algorithm adapted to the interpretability constraint. A deep parameterization is also proposed for these models. Finally, we show in Section 4 that it is possible to consider a deterministic latent process $\boldsymbol{z}_K$ to introduce long term dependencies between the r.v. (h_k, x_k) and all the past observations. We show that the particular structure of $\boldsymbol{z}_K$ leads to a direct adaptation of the algorithms derived in Section 2.

2. General Pairwise Markov Chains

In this section, we do not consider any auxiliary latent process $\boldsymbol{z}_K$ and we focus on the PMC model described by (2). The classical HMC (1) is a particular instance of this model. To see this, the transition distribution in (2), can be factorized as

$$p_{\boldsymbol{\theta}}(h_k, x_k | h_{k-1}, x_{k-1}) = p_{\boldsymbol{\theta}}(h_k | h_{k-1}, x_{k-1}) p_{\boldsymbol{\theta}}(x_k | \boldsymbol{h}_{k-1:k}, x_{k-1}). \tag{4}$$

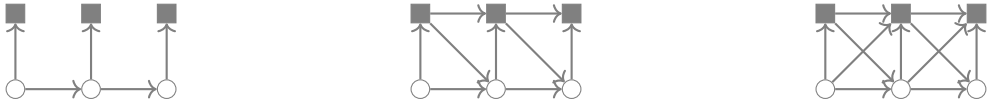
From (4), we deduce two particular instances of the PMC model; the semi PMC (SPMC), where the observation x_k does not depend on h_{k-1}, given $(\boldsymbol{h}_{k-1:k}, x_{k-1})$, *i.e.*

$$p_{\boldsymbol{\theta}}(h_k, x_k | h_{k-1}, x_{k-1}) = p_{\boldsymbol{\theta}}(h_k | h_{k-1}, x_{k-1}) p_{\boldsymbol{\theta}}(x_k | h_k, x_{k-1}); \tag{5}$$

and the HMC (1), in which (h_k, x_k) becomes, in addition, independent of x_{k-1} given h_{k-1},

$$p_{\boldsymbol{\theta}}(h_k, x_k | h_{k-1}, x_{k-1}) = p_{\boldsymbol{\theta}}(h_k | h_{k-1}) p(x_k | h_k). \tag{6}$$

Fig. 1 depicts the graphical representation of the HMM (Fig. 1a), SPMC (Fig. 1b) and PMC (Fig. 1c).



(a) HMC (b) SPMC (c) PMC

Figure 1: Graphical representations of the HMC, SPMC and PMC models. The white circles (resp. gray squares) represent the hidden (resp. observed) r.v. h_k (resp. x_k).

2.1. Bayesian inference for general parameterizations of PMCs

We now introduce a general parameterization of the distribution $p_{\boldsymbol{\theta}}(h_k, x_k|h_{k-1}, x_{k-1})$ underlying PMCs (2). We next show that it remains possible to compute approximately the ML estimate of $\boldsymbol{\theta}$ and to compute exactly the posterior distribution of h_k, for all k. A main advantage of directly parameterizing (4) is that the model does not rely on any stationary assumption contrary to the PMC models in [Derrode and Pieczynski, 2004, 2013, Boudaren and Pieczynski, 2016]. We finally introduce DNNs as particular parameterizations and we derive a procedure to estimate the parameters of the networks which takes into account the interpretability constraint related to h_k.

2.1.1. A general parameterization

Let $f_{\boldsymbol{\theta}}(h_{k-1}, x_{k-1})$ and $g_{\boldsymbol{\theta}}(\boldsymbol{h}_{k-1:k}, x_{k-1})$ be two vector-valued functions of (h_{k-1}, x_{k-1}) and of $(\boldsymbol{h}_{k-1:k}, x_{k-1})$, respectively. $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ are assumed to be differentiable w.r.t. $\boldsymbol{\theta}$. Let also $\lambda(h; v)$ (resp. $\mu(x; v')$) be a probability distribution on Ω (resp. a probability density function (pdf) on $\mathbb{R}^{d_x}$) whose parameters are given by a vector v (resp. v') and which is differentiable w.r.t. v (resp. v'). Then we parameterize the conditional distributions in (4) as

$$p_{\boldsymbol{\theta}}(h_k|h_{k-1}, x_{k-1}) = \lambda(h_k; f_{\boldsymbol{\theta}}(h_{k-1}, x_{k-1})), \tag{7}$$

$$p_{\boldsymbol{\theta}}(x_k|\boldsymbol{h}_{k-1:k}, x_{k-1}) = \mu(x_k; g_{\boldsymbol{\theta}}(\boldsymbol{h}_{k-1:k}, x_{k-1})). \tag{8}$$

In other words, $f_{\boldsymbol{\theta}}$ (resp. $g_{\boldsymbol{\theta}}$) describes the parameters of the (conditional) distribution λ (resp. μ).

As an illustrative example, let us show that this general parameterization includes the classical HMC with independent Gaussian noise (HMC-IN). For the sake of clarity, let us assume that $\Omega = \{\omega_1, \omega_2\}$ and $x_k \in \mathbb{R}$. We denote $\mathcal{N}(x; m; \sigma^2)$ the Gaussian distribution with mean m and variance σ^2 taken at point x, $\mathrm{Ber}(h; v)$ the Bernoulli distribution with parameter v such that $\mathrm{Ber}(\omega_1; v) = v$ and $\mathrm{sigm}(z) = 1/(1 + \exp(-z)) \in [0, 1]$ the sigmoid function. Then the HMC-IN model can be described as

$$\begin{aligned} f_{\boldsymbol{\theta}}(h_{k-1}, x_{k-1}) &= \mathrm{sigm}\left(b_{h_{k-1}}\right), & (9)\\ g_{\boldsymbol{\theta}}(\boldsymbol{h}_{k-1:k}, x_{k-1}) &= \left[d_{h_k}; \sigma_{h_k}\right], & (10)\\ \lambda(h; v) &= \mathrm{Ber}(h, v), & (11)\\ \mu\left(x; v' = \left[v'_1; v'_2\right]\right) &= \mathcal{N}\left(x; v'_1; (v'_2)^2\right), & (12) \end{aligned}$$

Indeed, (9)-(10) only depend on h_{k-1} and on h_k, respectively, so we have $p_{\boldsymbol{\theta}}(h_k = \omega_1|h_{k-1} = \omega_i) = \mathrm{sigm}(b_{\omega_i})$ and $p_{\boldsymbol{\theta}}(x_k|h_k = \omega_j) = \mathcal{N}(x_k; d_{\omega_j}; \sigma^2_{\omega_j})$. Finally, $\boldsymbol{\theta} = (b_{\omega_i}, d_{\omega_j}, \sigma_{\omega_j}|(\omega_i, \omega_j) \in \Omega \times \Omega)$. As a further illustrative example, it is possible to start from this particular parameterization of HMCs to derive a linear and Gaussian PMC model in which we introduce dependencies on x_{k-1} and h_{k-1}. In this case, λ and μ are unchanged but $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ now read

$$\begin{aligned} f_{\boldsymbol{\theta}}(h_{k-1}, x_{k-1}) &= \mathrm{sigm}\left(a_{h_{k-1}} x_{k-1} + b_{h_{k-1}}\right), & (13)\\ g_{\boldsymbol{\theta}}(\boldsymbol{h}_{k-1:k}, x_{k-1}) &= \left[c_{h_{k-1},h_k} x_{k-1} + d_{h_{k-1},h_k};\ \sigma_{h_k,h_{k-1}}\right]. & (14) \end{aligned}$$

The set of parameters is now given by $\boldsymbol{\theta} = (a_{\omega_i}, b_{\omega_i}, c_{\omega_j,\omega_i}, d_{\omega_j,\omega_i}, \sigma_{\omega_j,\omega_i}|(\omega_j, \omega_i) \in \Omega \times \Omega)$. As we will see later, these models play a critical role in the construction of parameterization based on DNNs. Indeed, despite their simple form, they generally provide an interpretable classification.

We now show that under this framework it is possible to derive an unsupervised estimation algorithm which approximates the ML estimate of $\boldsymbol{\theta}$ and which computes exactly the posterior distributions $p_{\boldsymbol{\theta}}(h_k|\boldsymbol{x}_K)$, whatever the parameterization chosen above.

2.1.2. Estimation of $\boldsymbol{\theta}$

Using the differentiability assumptions on $f_{\boldsymbol{\theta}}, g_{\boldsymbol{\theta}}, \lambda$ and μ, we can propose a gradient ascent method on the likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x}_K)$ to approximate the ML estimate of $\boldsymbol{\theta}$. This gradient ascent method is based on the sequential computation of $\alpha_{\boldsymbol{\theta},k}(h_k) = p_{\boldsymbol{\theta}}(h_k, \boldsymbol{x}_k)$, for all k, $0 \leq k \leq K$, from which we deduce the likelihood

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_K) = \sum_{h_K} \alpha_{\boldsymbol{\theta},K}(h_K). \tag{15}$$

Based on the Markovian property of (2) and on the general parameterization (7)-(8), the coefficients $\alpha_{\boldsymbol{\theta},K}(h_K)$ can be computed recursively from [Pieczynski, 2003]

$$\alpha_{\boldsymbol{\theta},k}(h_k) = \sum_{h_{k-1}} \alpha_{\boldsymbol{\theta},k-1}(h_{k-1})\lambda(h_k; f_{\boldsymbol{\theta}}(h_{k-1}, x_{k-1}))\mu(x_k; g_{\boldsymbol{\theta}}(\boldsymbol{h}_{k-1:k}, x_{k-1})). \tag{16}$$

Consequently, the gradient of the likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x}_K)$ (or equivalently that of the log-likelihood) w.r.t. $\boldsymbol{\theta}$ can be computed sequentially by using the decomposition of $\alpha_{\boldsymbol{\theta},k}$ in (16). The estimation of $\boldsymbol{\theta}$ can thus be deduced from an iterative gradient ascent method based on a learning rate ϵ and, for example, on the update

$$\boldsymbol{\theta}^{(j+1)} = \boldsymbol{\theta}^{(j)} + \epsilon \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_K)\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(j)}}. \tag{17}$$

The unsupervised estimation of $\boldsymbol{\theta}$ is summarized in Alg. 1.

Remark 1. *Generally, the parameter estimation procedure for a probabilistic model with hidden r.v. is based on the EM algorithm [Dempster et al., 1977]. It relies on the computation of* $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)}) = \mathbb{E}_{p_{\boldsymbol{\theta}^{(j)}}(\boldsymbol{h}_K|\boldsymbol{x}_K)}\big(\log p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{x}_K)\big)$ *followed by the maximization of* $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)})$ *w.r.t.* $\boldsymbol{\theta}$*. However, for general parameterizations (7)-(8), the maximization step cannot be computed analytically. In this case, it is possible to use a gradient-EM approach to replace the maximization step, but it is then strictly equivalent and computationally more demanding than computing the gradient of the log-likelihood [Xu and Jordan, 1996, Balakrishnan et al., 2017] as we propose in (17). Finally, for particular parameterizations for which the maximization step is computable, the comparison between these two approaches is an open question and is out of scope of this paper.*

2.1.3. Estimation of h_k

Once we have obtained an estimate $\boldsymbol{\theta}^*$ of $\boldsymbol{\theta}$, it remains to compute $p_{\boldsymbol{\theta}^*}(h_k|\boldsymbol{x}_K)$, for all k. Again, this can be done by using the Markovian property of (2) and by introducing the backward coefficients $\beta_{\boldsymbol{\theta}^*,k}(h_k) = p_{\boldsymbol{\theta}^*}(\boldsymbol{x}_{k+1:K}|h_k, x_k)$, for all k, with $\beta_{\boldsymbol{\theta}^*,K}(h_K) = 1$ [Pieczynski, 2003]. They can be computed sequentially from

$$\beta_{\boldsymbol{\theta}^*,k-1}(h_{k-1}) = \sum_{h_k} \beta_{\boldsymbol{\theta}^*,k}(h_k)\lambda(h_k; f_{\boldsymbol{\theta}^*}(h_{k-1}, x_{k-1}))\mu(x_k; g_{\boldsymbol{\theta}^*}(\boldsymbol{h}_{k-1:k}, x_{k-1})); \tag{18}$$

we deduce

$$p_{\boldsymbol{\theta}^*}(\boldsymbol{h}_{k-1:k}|\boldsymbol{x}_K) \propto \alpha_{\boldsymbol{\theta}^*,k-1}(h_{k-1}) \times \beta_{\boldsymbol{\theta}^*,k}(h_k) \times \lambda(h_k; f_{\boldsymbol{\theta}^*}(h_{k-1}, x_{k-1})) \times \mu(x_k; g_{\boldsymbol{\theta}^*}(\boldsymbol{h}_{k-1:k}, x_{k-1})), \tag{19}$$

$$p_{\boldsymbol{\theta}^*}(h_k|\boldsymbol{x}_K) = \sum_{h_{k-1}} p_{\boldsymbol{\theta}^*}(\boldsymbol{h}_{k-1:k}|\boldsymbol{x}_K). \tag{20}$$

The computation of the MAP estimate of h_k is summarized in Alg. 2.

Input: A realization $\boldsymbol{x}_K$, a learning rate ϵ, an initial set of parameters $\boldsymbol{\theta}^{(0)}$
Result: $\boldsymbol{\theta}^*$, a set of estimated parameters

$j = 0$
while *convergence of* $\log p_{\boldsymbol{\theta}^{(j)}}(\boldsymbol{x}_K)$ *is not attained* **do**
  Compute $\log \alpha_{\boldsymbol{\theta}^{(j)},k}(h_k)$ and $\nabla_{\boldsymbol{\theta}} \log \alpha_{\boldsymbol{\theta}^{(j)},k}(h_k)\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(j)}}$, for all $h_k \in \Omega$, for all $0 \leq k \leq K$, with (16)
  Compute $\log p_{\boldsymbol{\theta}^{(j)}}(\boldsymbol{x}_K)$ and $\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}^{(j)}}(\boldsymbol{x}_K)\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(j)}}$, with (15)
  Set $\boldsymbol{\theta}^{(j+1)} = \boldsymbol{\theta}^{(j)} + \epsilon \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_K)\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(j)}}$
  $j \leftarrow j + 1$
end
$\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}^{(j)}$

Algorithm 1: Unsupervised estimation of $\boldsymbol{\theta}$ in general PMC models.

Input: A realization $\boldsymbol{x}_K$, a set of estimated parameters $\boldsymbol{\theta}^*$
Result: $\hat{\boldsymbol{h}}_K$, the estimated hidden r.v.
Compute $\alpha_{\boldsymbol{\theta}^*,k}(h_k)$, for all $h_k \in \Omega$, for all $0 \leq k \leq K$, with (16)
Compute $\beta_{\boldsymbol{\theta}^*,k}(h_k)$, for all $h_k \in \Omega$, for all $0 \leq k \leq K$, with (18)
Compute $p_{\boldsymbol{\theta}^*}(\boldsymbol{h}_{k-1:k}|\boldsymbol{x}_K)$, for all $\boldsymbol{h}_{k-1:k} \in \Omega \times \Omega$, for all $0 \leq k \leq K$, with (19)
Compute $\hat{h}_k = \arg\max p_{\boldsymbol{\theta}^*}(h_k|\boldsymbol{x}_K)$, for all $0 \leq k \leq K$, with (20)

Algorithm 2: Unsupervised estimation of h_k in general PMC models.

2.2. Deep-PMC models

We now introduce a particular parameterization $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ of the distributions λ and μ, respectively. The rationale is as follows. Since DNNs can theoretically approximate any function which satisfies reasonable assumptions [Pinkus, 1999], our objective is to use them to approximate any parameterization of λ and μ. So, from now onwards, $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ are the outputs of two DNNs with (h_{k-1}, x_{k-1}) and $(\boldsymbol{h}_{k-1:k}, x_{k-1})$ as inputs, respectively. $\boldsymbol{\theta}$ now consists of the parameters of these DNNs (weights and biases). Note that a unique DNN is used for $f_{\boldsymbol{\theta}}$ (resp. $g_{\boldsymbol{\theta}}$) overtime.

Since $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ are differentiable w.r.t. $\boldsymbol{\theta}$ and their gradients are computable from the backpropagation algorithm [Rumelhart et al., 1986], Alg. 1 can be directly applied to estimate $\boldsymbol{\theta}$. However, due to the large number of parameters of these architectures, some problems tend to appear in practice. In particular, a random initialization of $\boldsymbol{\theta}$ can lead to convergence issues for the optimization of $\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_K)$. More importantly, the final r.v. h_k learnt by such a model may no longer be interpretable, *i.e.* it is not ensured that h_k coincides with the original class associated to x_k. In other words, a direct application of Alg. 1 tends to return a final model which gives poorer results than the simple models described in Section 2.1.1 in terms of classification.

We introduce a two-step solution based on a constrained output layer and next on a pretraining which aims at initializing properly $\boldsymbol{\theta}$. This solution relies on a simple model such as the linear and Gaussian PMC described in Section 2.1.1 where the linear functions $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ in (13)-(14) can be seen as the output layer of an elementary DNN with no hidden layer. Rather than directly training the DNN associated to $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$, we first estimate the linear PMC model (13)-(14) with Alg. 1 before adding intermediate layers.These layers are next pretrained from the classification obtained with the elementary model, and are finally finely trained with our ML approach.

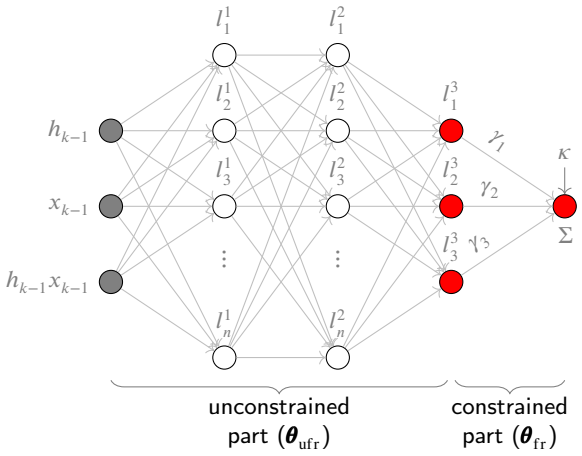
2.2.1. Constrained output layer

The main idea of our constrained training step is to make coincide a subset of $\boldsymbol{\theta}$ with the parameters of an elementary linear (equivalently a non deep) PMC model (13)-(14) which is assumed to provide an interpretable classification. In other words, we first estimate an elementary linear PMC model with Alg. 1 and we denote the set of associated parameters $\boldsymbol{\theta}_{\text{fr}}$, in the sense that these parameters are next *frozen* and will not be further updated. We next consider this linear layer as the output layer of a DNN where the other parameters are denoted $\boldsymbol{\theta}_{\text{ufr}}$, and which are *unfrozen* in the sense that they have not been estimated yet. Fig. 2 describes an example of a constrained DNN architecture for the function $f_{\boldsymbol{\theta}}$ when $\Omega = \{\omega_1, \omega_2\}$ and $\mathbb{R}^{d_x} = \mathbb{R}$, without loss of generality.

2.2.2. Pretraining by backpropagation

It remains to estimate the parameters $\boldsymbol{\theta}_{\text{ufr}}$ of the intermediate hidden layers. The idea is to initialize them in a such way that the initial deep PMC coincides with the elementary one; in other words, and due to the previous step, the output of the newly added hidden layers aims at coinciding with the identity function after the pretraining. After initializing randomly $\boldsymbol{\theta}_{\text{ufr}}$, our pretraining step aims at minimizing cost functions $C_{f_{\boldsymbol{\theta}}}$ and $C_{g_{\boldsymbol{\theta}}}$ which involve the pre-classification $\hat{\boldsymbol{h}}_K^{\text{pre}}$. Typically, the cost function $C_{f_{\boldsymbol{\theta}}}$ is the averaged overtime cross-entropy between the output of the DNN $f_{\boldsymbol{\theta}}$ and $\hat{h}_k^{\text{pre}}$ and $C_{g_{\boldsymbol{\theta}}}$ is the mean square error between the output of $g_{\boldsymbol{\theta}}$ and the parameters of the elementary linear models associated to $\hat{\boldsymbol{h}}_{k-1:k}^{\text{pre}}$ (see (14)). The minimization of these cost functions w.r.t. $\boldsymbol{\theta}_{\text{ufr}}$ is done with the backpropagation algorithm. Finally, once $\boldsymbol{\theta}_{\text{ufr}}$ has been properly initialized, it is fine-tuned with Alg. 1 which approximates the ML estimate of $\boldsymbol{\theta}$. Alg. 3 summarizes the two estimation steps specific to the DNN parameterization.

Remark 2. *In order to estimate the parameters of our deep PMC, we have used a reverse approach w.r.t. the pretraining approaches proposed at the beginning of 2010s to help supervised learning in DNN [Erhan et al., 2010]. Indeed, due to the large number of parameters in these architectures, [Mohamed et al., 2012, Glorot and Bengio, 2010, Hinton*



$\Sigma = f_{\boldsymbol{\theta}}(h_{k-1}, x_{k-1}, h_{k-1}x_{k-1}) = \text{sigm}(\gamma_1 l_1^3 + \gamma_2 l_2^3 + \gamma_3 l_3^3 + \kappa)$, where the last layer parameters $\{\gamma_1, \gamma_2, \gamma_3, \kappa\}$ are frozen to: $\gamma_1 = b_{\omega_2} - b_{\omega_1}, \gamma_2 = a_{\omega_2} - a_{\omega_1}, \gamma_3 = a_{\omega_1}$ and $\kappa = b_{\omega_1}$.

Figure 2: DNN architecture with constrained output layer for $f_{\boldsymbol{\theta}}$ with two hidden layers. The parameters $\boldsymbol{\theta}_{\text{fr}}$ are related to the output layer which computes the function $f_{\boldsymbol{\theta}}$ of the linear PMC model (13). Due to the one-hot encoding of the discrete r.v. h_{k-1} ($h_{k-1} = \omega_1 \leftrightarrow h_{k-1} = 0$ and $h_{k-1} = \omega_2 \leftrightarrow h_{k-1} = 1$), this parameterization is equivalent to that of (13) up to the given correspondence between $\boldsymbol{\theta}_{\text{fr}} = (\gamma_1, \gamma_2, \gamma_3, \kappa)$ and $(a_{\omega_1}, a_{\omega_2}, b_{\omega_1}, b_{\omega_2})$. Linear activation functions are used in the last hidden layer in red.

*et al., 2012] have suggested to first pretrain in an unsupervised way a DNN from a generative probabilistic model which shares common parameters with the original DNN (*e.g. *a Deep Belief Network). The backpropagation algorithm for supervised estimation is next initialized with the (approximated) ML estimate of this probabilistic model. Here, we have started to pretrain our architecture in a supervised way with a pre-classification and next embedded it in our original probabilistic model in which we compute an approximation of the ML estimate.*

Input: $\boldsymbol{x}_K$, the observation
Result: $\hat{\boldsymbol{h}}_K$, the final classification
/* Linear model: initialization of the ouput layer of $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ (§ 2.2.1) */
Initialize randomly $\boldsymbol{\theta}_{\text{fr}}^{(0)}$
Estimate $\boldsymbol{\theta}_{\text{fr}}^{*}$ using Alg. 1 with $\boldsymbol{\theta}_{\text{fr}}^{(0)}$
Estimate $\hat{\boldsymbol{h}}_K^{\text{pre}}$ using Alg. 2 with $\boldsymbol{\theta}_{\text{fr}}^{*}$
/* Pretraining of $\boldsymbol{\theta}_{\text{ufr}}$ (§ 2.2.2) */
$\boldsymbol{\theta}_{\text{ufr}}^{(0)} \leftarrow \text{Backprop}(\hat{\boldsymbol{h}}_K^{\text{pre}}, \boldsymbol{x}_K, \boldsymbol{\theta}_{\text{fr}}^{*}, C_{f_{\boldsymbol{\theta}}}, C_{g_{\boldsymbol{\theta}}})$
/* Complete deep model: fine-tuning */
Compute $\boldsymbol{\theta}_{\text{ufr}}^{*}$ using Alg. 1 with $(\boldsymbol{\theta}_{\text{fr}}^{*}, \boldsymbol{\theta}_{\text{ufr}}^{(0)})$ ($\boldsymbol{\theta}_{\text{fr}}^{*}$ is not updated)
Compute $\hat{\boldsymbol{h}}_K$ using Alg. 2 with $(\boldsymbol{\theta}_{\text{fr}}^{*}, \boldsymbol{\theta}_{\text{ufr}}^{*})$

Algorithm 3: A general estimation algorithm for deep parameterization of PMC models.

2.3. Simulations

We illustrate the gain of our general parameterization w.r.t. an elementary HMC-IN by considering a problem of unsupervised binary image segmentation (so $\Omega = \{\omega_1, \omega_2\}$) from noisy observations. We consider the cattle-type images of the Binary Shape Database[1]. The images are transformed into a 1-D signal $\boldsymbol{x}_K$ with a Hilbert-Peano filling curve [Sagan, 2012]. They are next blurred with a noise which exhibits non-linearities to highlight the ability of the generalized PMC models to learn such a signal corruption[2]. More precisely, we generate an artificial noise by

[1] http://vision.lems.brown.edu/content/available-software-and-databases
[2] The code to reproduce the experiments is available at https://github.com/HGangloff/deep_hidden_markov_models/

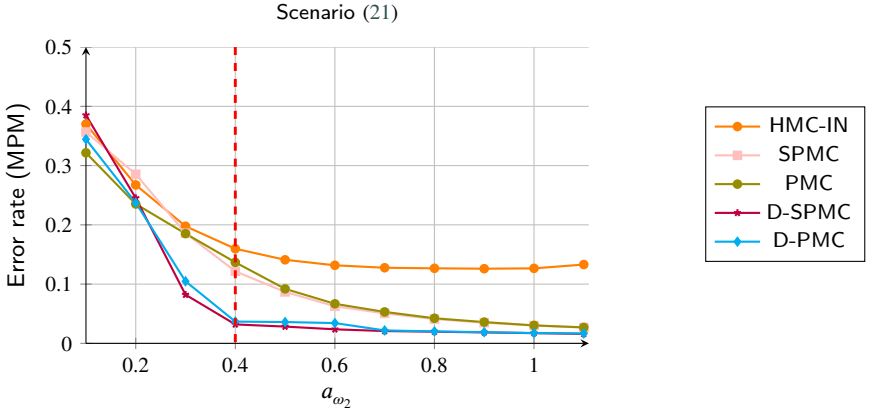
generating x_k according to

$$x_k|h_k, x_{k-1} \sim \mathcal{N}\Big(\sin(a_{h_k} + x_{k-1}); \sigma^2\Big), \tag{21}$$

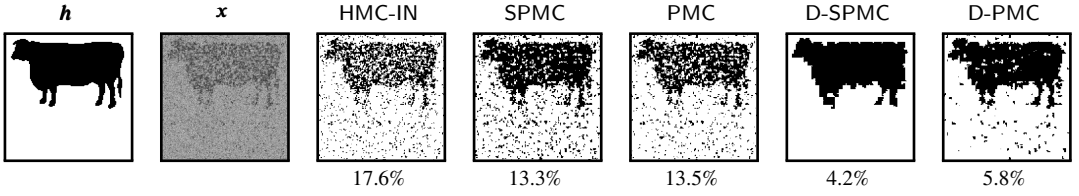
where $a_{\omega_1} = 0$, $\sigma^2 = 0.25$ and a_{ω_2} is a varying parameter.

We next focus on two kinds of parameterizations of distributions λ and μ which coincide with (11)-(12). Each parameterization is applied to the SPMC and PMC models (see Fig. 1). First, we consider a linear parameterization (SPMC and PMC) based on (11)-(14). The second parameterization is a deep one (D-SPMC and D-PMC) and relies on one (unfrozen) hidden layer with 100 neurons and the ReLU activation function. For this architecture, we apply the training constraints discussed in Paragraph 2.2.

In Fig. 3a, we display the averaged error rates for each model over all the selected images as a function of a_{ω_2}. Fig. 3b displays the results of the classifications for a particular image of the database. As it can be observed, although the same Gaussian distribution μ is used both models, the general PMC framework that we introduced leads to a great improvement of the elementary HMC model. Next, the deep parameterized models (D-PMC and D-SPMC) are the most accurate models and are able to capture the complexity by improving the results of their non-deep counterpart. More importantly, note that the gain obtained with our D-PMC and D-SPMC models does not require any further modeling effort in the sense that they are a particular parameterization in our general framework.



(a) Error rate from the unsupervised segmentations with a noise described by (21). Results are averaged on all the *cattle*-type images from the database.



(b) Selected classifications for $a_{\omega_2} = 0.4$ (signaled by the red vertical line in Fig. 3a). Error rates appear below the images.

Figure 3: Unsupervised image segmentation with PMC models.

3. General Triplet Markov Chains

Our previous PMC models rely on a general parameterization of the two distributions λ and μ. However, the choice of these distributions is not obvious in practice and has an impact on the performance of the classification. The goal of this section is to implicitly estimate these distributions in addition to their parameters by the introduction of a third latent auxiliary process $\boldsymbol{z}_K$ which aims at complexifying the distribution $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{x}_K)$. The rationale behind this auxiliary process is the following [Bayer and Osendorfer, 2014]. Assume that a r.v. $x \in \mathbb{R}$ follows an unknown distribution $p(x)$ while $z \in \mathbb{R}$ follows an elementary one $p(z)$ (*e.g.* the Gaussian distribution). Denoting F_X (resp. F_Z) the cumulative density function of x (resp. of z) and observing that the r.v. $F_X(x)$ and $F_Z(z)$ both follow the uniform distribution on the unit interval, then the r.v. $F_X^{-1}(F_Z(z))$ admits $p(x)$ as pdf. In other words, whatever the

distribution of z, it is possible to model an unknown distribution $p(x)$ via an auxiliary r.v. z and a joint distribution $p(x,z) = p(z)p(x|z)$, provided $p(x|z)$ is well chosen and close to $\delta_{F_X^{-1}(F_Z(z))}(x)$.

As we have just seen, the introduction of a continuous latent process $\boldsymbol{z}_K$ is interesting from a modeling point of view but makes Alg. 1 and 2 uncomputable. Indeed, a direct application would involve the computation of intractable integrals in (16) and (18) w.r.t. z_k. Consequently, $p_{\boldsymbol{\theta}}(\boldsymbol{x}_K)$ and $p_{\boldsymbol{\theta}}(h_k|\boldsymbol{x}_K)$ are no longer exactly computable. In order to estimate $\boldsymbol{\theta}$ and h_k, we derive a new estimation algorithm based on variational Bayesian inference which consists in maximizing a lower bound of the likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x}_K)$. After reviewing the principle of variational inference and introducing its extension to TMCs, we propose a general parameterization of these models as well as a parameter estimation algorithm. Our algorithm relies on the optimization of an objective function deduced from the variational inference framework but it also enforces the interpretability of $\boldsymbol{h}_K$ by modifying the classical lower bound used in variational inference.

3.1. Variational Bayesian inference: principle and application to TMCs

For the sake of clarity, let us now denote the triplet $t_k = (h_k, z_k, x_k)$. From a mathematical point of view, the TMC (3) can be seen as a PMC (2) in augmented dimension, *i.e.* a PMC where $(\boldsymbol{h}_K, \boldsymbol{z}_K)$ plays the role of the hidden process. If $\boldsymbol{z}_K$ were a discrete process, it would be possible to apply directly the Bayesian inference framework developed in Section 2.1; however, the continuous nature of z_k involves intractable integrals to compute sequentially the equivalent of (16), *i.e.*,

$$p_{\boldsymbol{\theta}}(h_k, z_k, \boldsymbol{x}_k) = \int \sum_{h_{k-1}} p_{\boldsymbol{\theta}}(t_k|t_{k-1}) p_{\boldsymbol{\theta}}(h_{k-1}, z_{k-1}, \boldsymbol{x}_{k-1}) \mathrm{d}z_{k-1}, \tag{22}$$

and therefore $p_{\boldsymbol{\theta}}(\boldsymbol{x}_K)$. To overcome this issue, we briefly review the general principle of variational Bayesian inference.

Let $q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)$ be a *variational* distribution parameterized by a set of parameters $\boldsymbol{\varphi}$. Observing that the Kullback-Leibler Divergence (KLD) between $q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)$ and $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)$ is positive,

$$\mathrm{D}_{\mathrm{KL}}\left(q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)||p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)\right) = \sum_{\boldsymbol{h}_K} \int q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K) \log\left(\frac{q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)}{p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)}\right) \mathrm{d}\boldsymbol{z}_K \geq 0, \tag{23}$$

we deduce

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_K) \geq \sum_{\boldsymbol{h}_K} \int q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K) \log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{z}_K, \boldsymbol{x}_K)}{q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)}\right) \mathrm{d}\boldsymbol{z}_K = F(\boldsymbol{\theta}, \boldsymbol{\varphi}). \tag{24}$$

Equality holds if $q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K) = p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)$. When the posterior distribution $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)$ is computable, the alternating maximization w.r.t. $\boldsymbol{\theta}$ and $q_{\boldsymbol{\varphi}}$ of the so-called Evidence Lower Bound (ELBO), $F(\boldsymbol{\theta}, \boldsymbol{\varphi})$, coincides with the EM algorithm [Tzikas et al., 2008]. However, here, $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)$ is not computable because $\boldsymbol{z}_K$ is continuous. In this case, variational inference consists in maximizing $F(\boldsymbol{\theta}, \boldsymbol{\varphi})$ w.r.t. $(\boldsymbol{\theta}, \boldsymbol{\varphi})$ for a given class of distributions $q_{\boldsymbol{\varphi}}$. The choice of the variational distribution $q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)$ is critical; $q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)$ should be close to $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)$ but should also be chosen in a such way that the associated ELBO can be exactly computed or easily approximated while remaining differentiable w.r.t. $(\boldsymbol{\theta}, \boldsymbol{\varphi})$. In the context of TMCs with a discrete and continuous latent process, Prop. 1 exploits the observation that

$$p_{\boldsymbol{\theta}}(\boldsymbol{h}_K|\boldsymbol{z}_K, \boldsymbol{x}_K) = p_{\boldsymbol{\theta}}(h_K|\boldsymbol{z}_K, \boldsymbol{x}_K) \prod_{k=1}^{K} p_{\boldsymbol{\theta}}(h_{k-1}|h_k, \boldsymbol{z}_K, \boldsymbol{x}_K) \tag{25}$$

is computable (see App. A.1) and shows that it is optimal (in the sense of the value of the ELBO) to restrict the choice of $q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)$ to that of $q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K|\boldsymbol{x}_K)$.

Proposition 1. *Let us denote $F(\boldsymbol{\theta}, \boldsymbol{\varphi})$, resp. $F^{\mathrm{opt}}(\boldsymbol{\theta}, \boldsymbol{\varphi})$, the ELBO when the variational distribution $q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K) = q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K|\boldsymbol{x}_K) q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K|\boldsymbol{z}_K, \boldsymbol{x}_K)$, resp. $q_{\boldsymbol{\varphi}}^{\mathrm{opt}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K) = q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K|\boldsymbol{x}_K) p_{\boldsymbol{\theta}}(\boldsymbol{h}_K|\boldsymbol{z}_K, \boldsymbol{x}_K)$, is used. Then for any $(\boldsymbol{\theta}, \boldsymbol{\varphi})$, we have*

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_K) \geq F^{\mathrm{opt}}(\boldsymbol{\theta}, \boldsymbol{\varphi}) \geq F(\boldsymbol{\theta}, \boldsymbol{\varphi}), \tag{26}$$

where

$$F^{\text{opt}}(\boldsymbol{\theta},\boldsymbol{\varphi}) = F_0^{\text{opt}}(\boldsymbol{\theta},\boldsymbol{\varphi}) + \sum_{k=1}^{K} F_{k-1,k}^{\text{opt}}(\boldsymbol{\theta},\boldsymbol{\varphi}) + F_K^{\text{opt}}(\boldsymbol{\theta},\boldsymbol{\varphi}), \tag{27}$$

and where

$$F_0^{\text{opt}}(\boldsymbol{\theta},\boldsymbol{\varphi}) = \int \sum_{h_0} q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K|\boldsymbol{x}_K) p_{\boldsymbol{\theta}}(h_0|\boldsymbol{z}_K,\boldsymbol{x}_K) \log p_{\boldsymbol{\theta}}(t_0) \mathrm{d}\boldsymbol{z}_K, \tag{28}$$

$$F_{k-1,k}^{\text{opt}}(\boldsymbol{\theta},\boldsymbol{\varphi}) = \int \sum_{\boldsymbol{h}_{k-1:k}} q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K|\boldsymbol{x}_K) p_{\boldsymbol{\theta}}(\boldsymbol{h}_{k-1:k}|\boldsymbol{z}_K,\boldsymbol{x}_K) \log \left(\frac{p_{\boldsymbol{\theta}}(t_k|t_{k-1})}{p_{\boldsymbol{\theta}}(h_{k-1}|h_k,\boldsymbol{z}_K,\boldsymbol{x}_K) q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K|\boldsymbol{x}_K)} \right) \mathrm{d}\boldsymbol{z}_K, \tag{29}$$

$$F_K^{\text{opt}}(\boldsymbol{\theta},\boldsymbol{\varphi}) = -\int \sum_{h_K} q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K|\boldsymbol{x}_K) p_{\boldsymbol{\theta}}(h_K|\boldsymbol{z}_K,\boldsymbol{x}_K) \log p_{\boldsymbol{\theta}}(h_K|\boldsymbol{z}_K,\boldsymbol{x}_k) \mathrm{d}\boldsymbol{z}_K. \tag{30}$$

A proof of Prop. 1 is given in App. A.1. The practical computation of these integrals will be described later with the modified objective function.

3.2. An estimation algorithm for TMCs with general parameterization

Following the approach that we have developed for PMC models, we extend our general parameterization framework to the distributions of TMC models,

$$p_{\boldsymbol{\theta}}(t_k|t_{k-1}) = p_{\boldsymbol{\theta}}(z_k|t_{k-1}) p_{\boldsymbol{\theta}}(h_k|z_k,t_{k-1}) p_{\boldsymbol{\theta}}(x_k|h_k,z_k,t_{k-1}). \tag{31}$$

We next propose a general estimation method based on a variational distribution $q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K|\boldsymbol{x}_K)$, for estimating $\boldsymbol{\theta}$ and $p_{\boldsymbol{\theta}}(h_k|\boldsymbol{x}_K)$, for all k, which takes into account the interpretability constraint.

3.2.1. A general parameterization of TMCs

As a direct extension of Section 2.1.1, functions $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ can now depend on $\boldsymbol{z}_{k-1:k}$; we also introduce a new parameterized and differentiable function $s_{\boldsymbol{\theta}}$ which depends on t_{k-1}, and a conditional distribution ς on $\mathbb{R}^{d_z}$. We thus parameterize the distributions in (31) as

$$p_{\boldsymbol{\theta}}(z_k|t_{k-1}) = \varsigma\left(z_k; s_{\boldsymbol{\theta}}(t_{k-1})\right), \tag{32}$$

$$p_{\boldsymbol{\theta}}(h_k|z_k,t_{k-1}) = \lambda\left(h_k; f_{\boldsymbol{\theta}}(z_k,t_{k-1})\right), \tag{33}$$

$$p_{\boldsymbol{\theta}}(x_k|h_k,z_k,t_{k-1}) = \mu\left(x_k; g_{\boldsymbol{\theta}}(h_k,z_k,t_{k-1})\right). \tag{34}$$

Remark that if $s_{\boldsymbol{\theta}}$ does not depend on t_{k-1}, and if $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ are independent of $\boldsymbol{z}_{k-1:k}$, the distribution $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K,\boldsymbol{x}_K)$ coincides with that of a PMC built from (7)-(8).

3.2.2. Joint estimation of $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$

Classical variational inference algorithms aim at maximizing the ELBO (27) when the objective is to estimate the parameters of a generative model, *i.e.* a model in which we do not focus on the interpretability of the hidden r.v. but rather on the modeling power of the distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}_K)$. Consequently, in our case, a direct maximization of (27) does not guarantee the interpretability of the r.v. $\boldsymbol{h}_K$. The problem is all the more critical that our hidden process is split into an interpretable one, $\boldsymbol{h}_K$, and an auxiliary one, $\boldsymbol{z}_K$. To that end, we propose an adaptation and an interpretation to the sequential case of two techniques introduced in the machine learning community [Higgins et al., 2017, Kingma et al., 2014]. The first one relies on a reinterpretation of the ELBO (27) as the sum of a reconstruction and a KLD terms; this last one is next penalized. The second technique consists in adding a penalizing term to the resulting ELBO which aims at strengthening the distinct role of $\boldsymbol{h}_K$ and of $\boldsymbol{z}_K$ and exploiting the result of previous classifications obtained with an available model.

The β-ELBO - We first start with an alternative decomposition of the ELBO (27).

Corollary 1. *Let us factorize* $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{z}_K, \boldsymbol{x}_K) = \overline{p}_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)\tilde{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_K|\boldsymbol{h}_K, \boldsymbol{z}_K)$ *with*

$$\tilde{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_K|\boldsymbol{h}_K, \boldsymbol{z}_K) = p_{\boldsymbol{\theta}}(x_0|h_0, z_0)\prod_{k=1}^{K}\mu\left(x_k; g_{\boldsymbol{\theta}}(h_k, z_k, t_{k-1})\right), \tag{35}$$

$$\overline{p}_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K) = p_{\boldsymbol{\theta}}(h_0, z_0)\prod_{k=1}^{K}\varsigma\left(z_k; s_{\boldsymbol{\theta}}(t_{k-1})\right)\lambda\left(h_k; f_{\boldsymbol{\theta}}(z_k, t_{k-1})\right). \tag{36}$$

Then

$$F^{\text{opt}}(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \mathcal{L}_1(\boldsymbol{\theta}, \boldsymbol{\varphi}) + \mathcal{L}_2(\boldsymbol{\theta}, \boldsymbol{\varphi}), \tag{37}$$

where

$$\mathcal{L}_1(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \mathbb{E}_{q_{\boldsymbol{\varphi}}^{\text{opt}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)}\left(\log\tilde{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_K|\boldsymbol{h}_K, \boldsymbol{z}_K)\right), \tag{38}$$

$$\mathcal{L}_2(\boldsymbol{\theta}, \boldsymbol{\varphi}) = -\text{D}_{\text{KL}}\left(q_{\boldsymbol{\varphi}}^{\text{opt}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)||\overline{p}_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)\right). \tag{39}$$

Let us comment on this result. First, this decomposition can be seen as a generalization to the sequential case of the decomposition proposed for the β-VAE in [Higgins et al., 2017]. Indeed, F^{opt} involves the sum of i) a reconstruction term $\mathcal{L}_1$ between $q_{\boldsymbol{\varphi}}^{\text{opt}}$ and $\tilde{p}_{\boldsymbol{\theta}}$ which measures the ability to reconstruct observations $\boldsymbol{x}_K$ according to the conditional likelihood $\tilde{p}_{\boldsymbol{\theta}}$ from the latent r.v. $(\boldsymbol{h}_K, \boldsymbol{z}_K)$ distributed according to $q_{\boldsymbol{\varphi}}^{\text{opt}}$; ii) a KLD term $\mathcal{L}_2$ between the variational distribution and the conditional prior $\overline{p}_{\boldsymbol{\theta}}$. However, contrary to the static case [Higgins et al., 2017], our decomposition involves $\tilde{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_K|\boldsymbol{h}_K, \boldsymbol{z}_K)$ and $\overline{p}_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)$ rather than $p_{\boldsymbol{\theta}}(\boldsymbol{x}_K|\boldsymbol{h}_K, \boldsymbol{z}_K)$ and $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{z}_K)$, respectively. Indeed, except if $K = 0$, the latter two distributions are no longer computable, which makes the classical ELBO decomposition impractical.

The idea underlying our β-ELBO is to penalize the KLD term $\mathcal{L}_2(\boldsymbol{\theta}, \boldsymbol{\varphi})$. To understand why, let us detail the expression of $\mathcal{L}_1(\boldsymbol{\theta}, \boldsymbol{\varphi})$ and of $\mathcal{L}_2(\boldsymbol{\theta}, \boldsymbol{\varphi})$. First, using (35) and (34), $\mathcal{L}_1(\boldsymbol{\theta}, \boldsymbol{\varphi})$ reads

$$\mathcal{L}_1(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \mathbb{E}_{q_{\boldsymbol{\varphi}}^{\text{opt}}(h_0, z_0|\boldsymbol{x}_K)}\left(\log p_{\boldsymbol{\theta}}(x_k|h_0, z_0)\right) + \sum_{k=1}^{K}\mathbb{E}_{q_{\boldsymbol{\varphi}}^{\text{opt}}(h_k, z_k|\boldsymbol{h}_{k-1}, \boldsymbol{z}_{k-1}, \boldsymbol{x}_K)}\left(\log p_{\boldsymbol{\theta}}(x_k|h_k, z_k, t_{k-1})\right). \tag{40}$$

Following this decomposition, it can be seen that at each time step k, the maximization of (40) encourages the model to interpret the latent r.v. (h_k, z_k) as those which explain the best the observation x_k given the past. On the other hand, using (36) and (32)-(33), the maximization of

$$\mathcal{L}_2(\boldsymbol{\theta}, \boldsymbol{\varphi}) = -\text{D}_{\text{KL}}\left(q_{\boldsymbol{\varphi}}^{\text{opt}}(h_0, z_0|\boldsymbol{x}_K)||p_{\boldsymbol{\theta}}(z_0, h_0)\right) - \sum_{k=1}^{K}\text{D}_{\text{KL}}\left(q_{\boldsymbol{\varphi}}^{\text{opt}}(h_k, z_k|\boldsymbol{h}_{k-1}, \boldsymbol{z}_{k-1}, \boldsymbol{x}_K)||p_{\boldsymbol{\theta}}(h_k, z_k|t_{k-1})\right) \tag{41}$$

tends to push the posterior variational distribution at each time step to be close to the conditional prior distribution $p_{\boldsymbol{\theta}}(h_k, z_k|t_{k-1})$. As in [Higgins et al., 2017], we penalize $\mathcal{L}_2(\boldsymbol{\theta}, \boldsymbol{\varphi})$ via the introduction of a scalar β_1. Since a part of the latent r.v. has to be interpretable, and that the interpretability of hidden r.v. is not conditioned by the observations, the interest of this term is to force the posterior distribution $q_{\boldsymbol{\varphi}}^{\text{opt}}$ to take into account the prior term at each time step. In other words, this penalization term aims at limiting the impact of the observations on the interpretability of the hidden r.v., particularly in problems where x_k is a very noisy version of h_k.

Cross-entropy penalization - We finally complete our objective function to guide the estimation process into distinguishing the role of $\boldsymbol{h}_K$ and of $\boldsymbol{z}_K$ in order to obtain better interpretable estimations of h_k. We assume that we have at our disposal a pre-classification $\boldsymbol{h}_K^{\text{pre}}$. Next, introduce the KLD between the empirical distribution deduced from this pre-classification, $p^{\text{emp}}(\boldsymbol{h}_K) = \delta_{\boldsymbol{h}_K^{\text{pre}}}(\boldsymbol{h}_K)$, and the marginal variational distribution $q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K|\boldsymbol{x}_K) = \int q_{\boldsymbol{\varphi}}^{\text{opt}}(\boldsymbol{h}_K, \boldsymbol{z}_K|\boldsymbol{x}_K)\text{d}\boldsymbol{z}_K$ which aims itself at approximating the true posterior distribution $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K|\boldsymbol{x}_K)$. Thus, the objective is to push the variational distribution $q_{\boldsymbol{\varphi}}$ to take into account the interpretable labels obtained from an already interpretable pre-classification through the negative cross-entropy

$$\mathcal{L}_3(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \mathbb{E}_{p^{\text{emp}}(\boldsymbol{h}_K)}\left(\log q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K|\boldsymbol{x}_K)\right) = \log q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K^{\text{pre}}|\boldsymbol{x}_K), \tag{42}$$

see for example [Kingma et al., 2014, Klys et al., 2018, Kumar et al., 2021]. This additional term is next penalized by a scalar β_2 which controls the proximity of the pre-classification with the variational posterior distribution.

Finally, we obtain a new objective function

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\varphi}) = \mathcal{L}_1(\boldsymbol{\theta},\boldsymbol{\varphi}) + \beta_1\mathcal{L}_2(\boldsymbol{\theta},\boldsymbol{\varphi}) + \beta_2\mathcal{L}_3(\boldsymbol{\theta},\boldsymbol{\varphi}), \tag{43}$$

where $\mathcal{L}_1(\boldsymbol{\theta},\boldsymbol{\varphi})$, $\mathcal{L}_2(\boldsymbol{\theta},\boldsymbol{\varphi})$ and $\mathcal{L}_3(\boldsymbol{\theta},\boldsymbol{\varphi})$ are defined in (38), (39) and (42), respectively. If we set $\beta_1 = 1$ and $\beta_2 = 0$, then $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\varphi})$ coincides with the ELBO $F^{\text{opt}}(\boldsymbol{\theta},\boldsymbol{\varphi})$ in (37).

Monte Carlo approximation - It remains to compute and optimize (43) in practice. $\mathcal{L}_1(\boldsymbol{\theta},\boldsymbol{\varphi})$ and $\mathcal{L}_2(\boldsymbol{\theta},\boldsymbol{\varphi})$ coincide with mathematical expectations according to $q_{\boldsymbol{\varphi}}^{\text{opt}}(\boldsymbol{h}_K,\boldsymbol{z}_K|\boldsymbol{x}_K) = q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K|\boldsymbol{x}_K)p_{\boldsymbol{\theta}}(\boldsymbol{h}_K|\boldsymbol{z}_K,\boldsymbol{x}_K)$. Using expressions (40)-(41), expectations according to $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K|\boldsymbol{x}_K,\boldsymbol{z}_K)$ are exactly computable. So $\mathcal{L}_1(\boldsymbol{\theta},\boldsymbol{\varphi})$ and $\mathcal{L}_2(\boldsymbol{\theta},\boldsymbol{\varphi})$ rely on the approximate computation of expectations according to $q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K|\boldsymbol{x}_K)$. It can be also noted that $q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K|\boldsymbol{x}_K) = \mathbb{E}_{q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K|\boldsymbol{x}_K)}(p_{\boldsymbol{\theta}}(\boldsymbol{h}_K|\boldsymbol{z}_K,\boldsymbol{x}_K))$, so $\mathcal{L}_3(\boldsymbol{\theta},\boldsymbol{\varphi})$ also relies on an expectation according to same distribution $q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K|\boldsymbol{x}_K)$ as $\mathcal{L}_1(\boldsymbol{\theta},\boldsymbol{\varphi})$ and $\mathcal{L}_2(\boldsymbol{\theta},\boldsymbol{\varphi})$.

Consequently, Monte Carlo (MC) estimates based on i.i.d. samples $\boldsymbol{z}_K^{(n)} \sim q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K|\boldsymbol{x}_K)$ are natural estimates of $\mathcal{L}_1(\boldsymbol{\theta},\boldsymbol{\varphi})$, $\mathcal{L}_2(\boldsymbol{\theta},\boldsymbol{\varphi})$ and $\mathcal{L}_3(\boldsymbol{\theta},\boldsymbol{\varphi})$. However, since our objective is also to maximize $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\varphi})$ w.r.t. $(\boldsymbol{\theta},\boldsymbol{\varphi})$, the MC approximation $\widehat{\mathcal{L}}(\boldsymbol{\theta},\boldsymbol{\varphi})$ of $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\varphi})$ should remain differentiable w.r.t $(\boldsymbol{\theta},\boldsymbol{\varphi})$. To that end, we include the following constraints on the choice of the variational distribution $q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K|\boldsymbol{x}_K) = q_{\boldsymbol{\varphi}}(z_0|\boldsymbol{x}_K)\prod_{k=1}^K q_{\boldsymbol{\varphi}}(z_k|\boldsymbol{z}_{k-1},\boldsymbol{x}_K)$. First, we set conditional distributions $q_{\boldsymbol{\varphi}}(z_k|\boldsymbol{z}_{k-1},\boldsymbol{x}_K)$ in order to obtain samples according to $q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K|\boldsymbol{x}_K)$ sequentially. Next, $q_{\boldsymbol{\varphi}}(z_k|\boldsymbol{z}_{k-1},\boldsymbol{x}_K)$ is chosen such that it is possible to reparameterize a final sample $\boldsymbol{z}_K^{(n)}$ as a differentiable function of $\boldsymbol{\varphi}$ and of a random sample $\boldsymbol{\epsilon}_K$ independent of $\boldsymbol{\varphi}$. More precisely, the final sample $\boldsymbol{z}_K^{(n)}$ can be written as

$$\boldsymbol{z}_K^{(n)} = \psi\left(\boldsymbol{\varphi},\boldsymbol{\epsilon}_K^{(n)}\right), \tag{44}$$

where $\boldsymbol{\epsilon}_K^{(n)}$ is a sequence of random samples which does not depend on $\boldsymbol{\varphi}$ and ψ is a differentiable function w.r.t. $\boldsymbol{\varphi}$. As an illustratrive example, a sample $z^{(n)}$ according to Gaussian distribution with mean ϕ_1 and standard deviation ϕ_2 can be reparameterized as a differentiable function of (ϕ_1,ϕ_2) via $z^{(n)} = \phi_1 + \phi_2\epsilon^{(n)}$, where $\epsilon^{(n)} \sim \mathcal{N}(0,1)$. This sampling technique is referred to as the *reparameterization trick* [Kingma and Welling, 2014].

Finally, we obtain the following estimate of $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\varphi})$ in (43) given by

$$\widehat{\mathcal{L}}(\boldsymbol{\theta},\boldsymbol{\varphi}) = \widehat{\mathcal{L}}_1(\boldsymbol{\theta},\boldsymbol{\varphi}) + \widehat{\mathcal{L}}_2(\boldsymbol{\theta},\boldsymbol{\varphi}) + \widehat{\mathcal{L}}_3(\boldsymbol{\theta},\boldsymbol{\varphi}), \tag{45}$$

where

$$\widehat{\mathcal{L}}_1(\boldsymbol{\theta},\boldsymbol{\varphi}) = \frac{1}{N}\sum_{n=1}^N \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{h}_K|\boldsymbol{z}_K^{(n)},\boldsymbol{x}_K)}\left(\log\tilde{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_K|\boldsymbol{h}_K,\boldsymbol{z}_K^{(n)})\right), \tag{46}$$

$$\widehat{\mathcal{L}}_2(\boldsymbol{\theta},\boldsymbol{\varphi}) = \frac{1}{N}\sum_{n=1}^N \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{h}_K|\boldsymbol{z}_K^{(n)},\boldsymbol{x}_K)}\left(\log\left(\frac{\overline{p}_{\boldsymbol{\theta}}(\boldsymbol{h}_K,\boldsymbol{z}_K^{(n)}|\boldsymbol{x}_K)}{p_{\boldsymbol{\theta}}(\boldsymbol{h}_K|\boldsymbol{z}_K^{(n)},\boldsymbol{x}_K)q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K^{(n)}|\boldsymbol{x}_K)}\right)\right), \tag{47}$$

$$\widehat{\mathcal{L}}_3(\boldsymbol{\theta},\boldsymbol{\varphi}) = \log\left(\frac{1}{N}\sum_{n=1}^N p_{\boldsymbol{\theta}}(h_K^{\text{pre}}|\boldsymbol{z}_K^{(n)},\boldsymbol{x}_K)\prod_{k=1}^K p_{\boldsymbol{\theta}}(h_{k-1}^{\text{pre}}|h_k^{\text{pre}},\boldsymbol{z}_K^{(n)},\boldsymbol{x}_K)\right), \tag{48}$$

where the remaining expectations are computed from (25) and from (35)-(36) and where samples $\boldsymbol{z}_K^{(n)}$ satisfy (44). The complete estimation algorithm is described in Alg. 4.

3.2.3. Estimation of h_k

Once we have obtained an estimate $\boldsymbol{\theta}^*$ of $\boldsymbol{\theta}$, we focus on the computation of $p_{\boldsymbol{\theta}^*}(h_k|\boldsymbol{x}_K)$,

$$p_{\boldsymbol{\theta}^*}(h_k|\boldsymbol{x}_K) = \int_{\boldsymbol{z}_K} p_{\boldsymbol{\theta}^*}(h_k|\boldsymbol{z}_K,\boldsymbol{x}_K)p_{\boldsymbol{\theta}^*}(\boldsymbol{z}_K|\boldsymbol{x}_K)\mathrm{d}\boldsymbol{z}_K, \tag{50}$$

Input: $\boldsymbol{x}_K$, the data; ϵ, the learning rate; N the number of samples
Result: $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$, sets of estimated parameters
Initialize $(\boldsymbol{\theta}^{(0)}, \boldsymbol{\varphi}^{(0)})$
$t \leftarrow 0$
while *convergence is not attained* **do**
  Sample $z_0^{(n)} \sim q_{\boldsymbol{\varphi}^{(t)}}(z_0|\boldsymbol{x}_K)$, for all $1 \leq n \leq N$
  Sample $z_k^{(n)} \sim q_{\boldsymbol{\varphi}^{(t)}}(z_k|\boldsymbol{z}_{k-1}^{(n)}, \boldsymbol{x}_K)$, for all $1 \leq n \leq N$, for all $1 \leq k \leq K$
  Compute $p_{\boldsymbol{\theta}}(h_{k-1}|h_k, \boldsymbol{z}_K^{(n)}, \boldsymbol{x}_K)$, for all $\boldsymbol{h}_{k-1:k} \in \Omega \times \Omega$, for all $1 \leq n \leq N$, for all $1 \leq k \leq K$
  Evaluate the loss $\widehat{\mathcal{L}}(\boldsymbol{\theta}^{(t)}, \boldsymbol{\varphi}^{(t)})$ from (45)-(48)
  Compute the derivative of the loss function $\nabla_{(\boldsymbol{\theta},\boldsymbol{\varphi})}\widehat{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\varphi})$ from (45)-(48)
  Update the parameters with gradient ascent

$$\begin{pmatrix} \boldsymbol{\theta}^{(t+1)} \\ \boldsymbol{\varphi}^{(t+1)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\theta}^{(t)} \\ \boldsymbol{\varphi}^{(t)} \end{pmatrix} + \epsilon \nabla_{(\boldsymbol{\theta},\boldsymbol{\varphi})}\widehat{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\varphi})\Big|_{(\boldsymbol{\theta}^{(t)},\boldsymbol{\varphi}^{(t)})} \tag{49}$$

  $t \leftarrow t + 1$
end
$\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}^{(t)}$
$\boldsymbol{\varphi}^* \leftarrow \boldsymbol{\varphi}^{(t)}$

Algorithm 4: Parameter estimation in general TMCs.

where $p_{\boldsymbol{\theta}^*}(h_k|\boldsymbol{z}_K, \boldsymbol{x}_K)$ is computable from a direct extension of (16) and (18)-(20) (see the proof of Prop. 1). Since (50) is intractable, we propose an MC estimate $\hat{p}_{\boldsymbol{\theta}}(h_k|\boldsymbol{x}_K)$ deduced from the sequential importance resampling mechanism [Doucet et al., 2001] and based on the observation that $p_{\boldsymbol{\theta}^*}(\boldsymbol{z}_K|\boldsymbol{x}_K) \propto p_{\boldsymbol{\theta}^*}(\boldsymbol{x}_K, \boldsymbol{z}_K)$ is known up to a constant. Indeed, $p_{\boldsymbol{\theta}^*}(\boldsymbol{x}_K, \boldsymbol{z}_K)$ can also be computed from a direct extension of (15)-(16). We thus introduce the estimated variational distribution $q_{\boldsymbol{\varphi}^*}(\boldsymbol{z}_K|\boldsymbol{x}_K) = q_{\boldsymbol{\varphi}^*}(z_0|\boldsymbol{x}_K)\prod_{k=1}^K q_{\boldsymbol{\varphi}^*}(z_k|\boldsymbol{z}_{k-1}, \boldsymbol{x}_K)$ as importance distribution due to its proximity with $p_{\boldsymbol{\theta}^*}(\boldsymbol{z}_K|\boldsymbol{x}_K)$. Finally, rewriting (50) as

$$p_{\boldsymbol{\theta}^*}(h_k|\boldsymbol{x}_K) = \frac{\mathbb{E}_{q_{\boldsymbol{\varphi}^*}(\boldsymbol{z}_K|\boldsymbol{x}_K)}\left(\frac{p_{\boldsymbol{\theta}^*}(h_k|\boldsymbol{z}_K,\boldsymbol{x}_K)p_{\boldsymbol{\theta}^*}(\boldsymbol{z}_K,\boldsymbol{x}_K)}{q_{\boldsymbol{\varphi}^*}(\boldsymbol{z}_K|\boldsymbol{x}_K)}\right)}{\mathbb{E}_{q_{\boldsymbol{\varphi}^*}(\boldsymbol{z}_K|\boldsymbol{x}_K)}\left(\frac{p_{\boldsymbol{\theta}^*}(\boldsymbol{x}_K)}{q_{\boldsymbol{\varphi}^*}(\boldsymbol{z}_K|\boldsymbol{x}_K)}\right)}, \tag{51}$$

we compute the sequential MC sampler [Doucet and Johansen, 2009] presented in Alg. 5 consisting of the sequential application of three elementary steps (sampling, weighting and resampling). Note that any improvement of this sequential MC algorithm can be used [Fearnhead et al., 2010].

3.3. Deep-TMC models

Let us now focus on the particular case where functions $s_{\boldsymbol{\theta}}$, $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ are parameterized with a DNN. We adapt the two-step procedure described in Section 2.2. The main difference with Section 2.2 is that the input of our DNN can now depend on the latent r.v. z_k; in addition, due to the variational inference framework that we have proposed in the previous section, we also consider that the conditional variational distribution $q_{\boldsymbol{\varphi}}(z_k|\boldsymbol{z}_{k-1}, \boldsymbol{x}_K)$ at the core of our estimation algorithm is parameterized by a DNN.

3.3.1. Constrained Ouput Layer

The first step is a direct adaptation of Section 2.2.1 and relies on the preliminary estimation of a non deep TMC model. More precisely, Alg. 5 is applied to estimate the parameter of a linear TMC model (*i.e.* a TMC which is a direct extension of (13)-(14) or equivalently a deep TMC model with no hidden layer). Note that since $\boldsymbol{z}_K$ does not need to be interpretable, $q_{\boldsymbol{\varphi}}(z_k|\boldsymbol{z}_{k-1}, \boldsymbol{x}_K)$ are already parameterized by a DNN in the linear TMC models. Next, the DNNs, which parameterize $s_{\boldsymbol{\theta}}$, $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$, are built according to the same scheme of Fig. 2, except that the input and

Input: $\boldsymbol{x}_K$, the observation; a set of parameters $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$; N, the number of samples
Result: $\hat{\boldsymbol{h}}_K$ the final classification

Sample $z_0^{(n)} \sim q_{\boldsymbol{\varphi}^*}(z_0|\boldsymbol{x}_K)$, compute $w_0^{(n)} = \frac{p_{\boldsymbol{\theta}^*}(z_0^{(n)}, x_0)}{q_{\boldsymbol{\varphi}^*}(z_0|\boldsymbol{x}_K)}$ and $W_0^{(n)} = w_0^{(n)} / \sum_{n=1}^N w_0^{(n)}$, for all $1 \leq n \leq N$
for $k \leftarrow 1$ *to* K **do**
  Sample $z_k^{(n)} \sim q_{\boldsymbol{\varphi}^*}(z_k|\boldsymbol{z}_{k-1}, \boldsymbol{x}_K)$, for all $1 \leq n \leq N$
  Compute $w_k^{(n)} = w_{k-1}^{(n)} \frac{p_{\boldsymbol{\theta}^*}(\boldsymbol{z}_k^{(n)}, \boldsymbol{x}_k)}{p_{\boldsymbol{\theta}^*}(\boldsymbol{z}_{k-1}^{(n)}, \boldsymbol{x}_{k-1}) q_{\boldsymbol{\varphi}^*}(z_k^{(n)}|\boldsymbol{z}_{k-1}^{(n)}, \boldsymbol{x}_K)}$, for all $1 \leq n \leq N$
  Compute $W_k^{(n)} = w_k^{(n)} / \sum_{n=1}^N w_k^{(n)}$, for all $1 \leq n \leq N$
  if *Resampling* **then**
    Sample $l^{(n)} \sim p(l = j) = W_k^{(j)}$, for all $1 \leq n \leq N$
    Set $\boldsymbol{z}_k^{(n)} = \boldsymbol{z}_k^{(l^{(n)})}$ and $W_k^{(n)} = 1/N$ for all $1 \leq n \leq N$
  end
end
Compute $p_{\boldsymbol{\theta}^*}(\boldsymbol{h}_{k-1:k}|\boldsymbol{z}_K^{(n)}, \boldsymbol{x}_K)$, for all $\boldsymbol{h}_{k-1:k} \in \Omega \times \Omega$, for all $1 \leq k \leq K$, using the extension of (19)
Compute $\hat{p}_{\boldsymbol{\theta}^*}(h_k|\boldsymbol{x}_K) = \sum_{n=1}^N W_k^{(n)} p_{\boldsymbol{\theta}^*}(h_k|\boldsymbol{z}_K^{(n)}, \boldsymbol{x}_K)$, for all $h_k \in \Omega$, for all $1 \leq k \leq K$
$\hat{h}_k = \arg\max \hat{p}_{\boldsymbol{\theta}^*}(h_k|\boldsymbol{x}_K)$, for all $1 \leq k \leq K$

Algorithm 5: A Sequential Monte Carlo algorithm for Bayesian classification in general TMC.

the hidden layer before the output also consists of z_{k-1} or of $\boldsymbol{z}_{k-1:k}$. We thus obtain a set of frozen and unfrozen parameters.

3.3.2. Pretraining of the unfrozen parameters

The next step consists in pretraining the unfrozen parameters of the intermediate hidden layers in order to mimic the estimated linear TMC. We use the same approach as the one developed in Section 2.2.2 which relies on a pre-classification $\hat{\boldsymbol{h}}_K^{\text{pre}}$, but we now take into account the fact that z_k is not observed. Since the objective of the r.v. z_k is to encode the corresponding observation x_k through the DNN related to $q_{\boldsymbol{\varphi}}$, we first sample $\boldsymbol{z}_K$ according to the previously estimated variational distribution $q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K|\boldsymbol{x}_K)$; we next use the components $\boldsymbol{z}_{k-1:k}$ or z_k as inputs of the DNNs $s_{\boldsymbol{\theta}}$, $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$. Finally, as in Paragraph 2.2.2, we apply the backpropagation algorithm in order to minimize an adapted cost function w.r.t. $\boldsymbol{\theta}_{\text{ufr}}$ which depends on $\hat{\boldsymbol{h}}_K^{\text{pre}}$. Fig. 4 summarizes our pretraining procedure for function $f_{\boldsymbol{\theta}}$ and the final estimation procedure is described in Alg. 6.

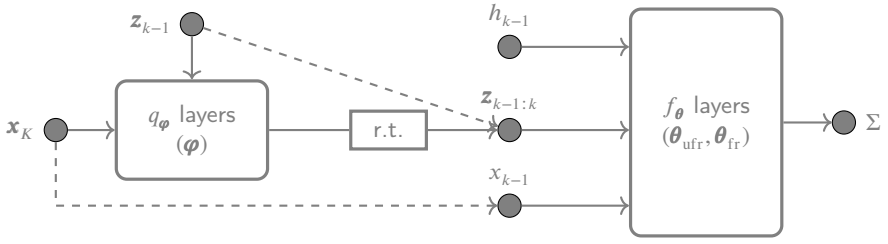


Figure 4: Graphical and condensed representation of the parameterization of $f_{\boldsymbol{\theta}}$ in the D-TMC models. *r.t.* stands for reparameterization trick. The dashed arrows represent the fact that some variables are copied. For clarity, we do not represent the block $f_{\boldsymbol{\theta}}$ which is similar to Fig. 2, up to the introduction of $\boldsymbol{z}_{k-1:k}$.

3.4. Simulations

We continue to illustrate the performance of our models with the same binary image segmentation problem as Section 2.3. Since Section 2.3 was devoted to the evaluation of deep parameterizations, we focus our experiments on the relevance of the latent process $\boldsymbol{z}_K$. To that end, we focus on a particular TMC model in which the role of the latent

Input: $\boldsymbol{x}_K$, the observation; $q_{\boldsymbol{\varphi}}$ a class of variational distribution
Result: $\hat{\boldsymbol{h}}_K$ the final classification
/* Initialization of the output layer of $s_{\boldsymbol{\theta}}$, $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ */
Estimate $(\boldsymbol{\theta}^*_{\text{fr}}, \tilde{\boldsymbol{\varphi}})$ and $\hat{\boldsymbol{h}}^{\text{pre}}_K$ with Alg. 4-5, using the related non-deep TMC model
/* Pretraining of $\boldsymbol{\theta}_{\text{ufr}}$ */
$\boldsymbol{\theta}^{(0)}_{\text{ufr}} \leftarrow \text{Backprop}(\hat{\boldsymbol{h}}^{\text{pre}}_K, \boldsymbol{x}_K, \boldsymbol{\theta}^*_{\text{fr}}, \tilde{\boldsymbol{\varphi}}, C_{s_{\boldsymbol{\theta}}}, C_{f_{\boldsymbol{\theta}}}, C_{g_{\boldsymbol{\theta}}})$
/* Fine-tuning of the complete model */
Compute $(\boldsymbol{\theta}^*_{\text{ufr}}, \boldsymbol{\varphi}^*)$ with Lines 2-13 of Alg. 4
Compute $\hat{\boldsymbol{h}}_K$ with Alg. 5

Algorithm 6: A general estimation algorithm for deep parameterizations of TMC models

process $\boldsymbol{z}_K$ is to complexify the conditional distribution μ of the noise but not λ. We first present the particular model and next the results.

3.4.1. The Minimal TMCs

In order to highlight the role of $\boldsymbol{z}_K$ w.r.t. the other characteristics of our models, we introduce the Minimal TMC (MTMC) model which exhibits a reduced number of direct dependencies. In this model, $\boldsymbol{z}_K$ is an independent process and given $\boldsymbol{z}_K$, $(\boldsymbol{h}_K, \boldsymbol{x}_K)$ is a HMC where only the observations depend on $\boldsymbol{z}_k$; in other words, $s_{\boldsymbol{\theta}}$ in (32) does not depend on t_{k-1}, $f_{\boldsymbol{\theta}}$ in (33) only depends on (h_{k-1}) and $g_{\boldsymbol{\theta}}$ in (34) only depends on (z_k, h_k). The joint distribution of $\boldsymbol{t}_K$ can be rewritten as

$$p_{\boldsymbol{\theta}}(\boldsymbol{t}_K) = \underbrace{\prod_{k=0}^{K} \varsigma(z_k; s_{\boldsymbol{\theta}})}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}_K)} \underbrace{p_{\boldsymbol{\theta}}(h_0) \prod_{k=1}^{K} \lambda(h_k; f_{\boldsymbol{\theta}}(h_{k-1}))}_{p_{\boldsymbol{\theta}}(\boldsymbol{h}_K|\boldsymbol{z}_K)=p_{\boldsymbol{\theta}}(\boldsymbol{h}_K)} \underbrace{\prod_{k=0}^{K} \mu(x_k; g_{\boldsymbol{\theta}}(z_k, h_k))}_{p_{\boldsymbol{\theta}}(\boldsymbol{x}_K|\boldsymbol{h}_K, \boldsymbol{z}_K)}, \tag{52}$$

With this model, the latent process $\boldsymbol{z}_K$ affects the conditional distribution of the observations.

We next consider three instances of MTMCs. The first one is the continuous linear MTMC in which $z_k \in \mathbb{R}$ are distributed according to standard normal distribution (so ς is the Gaussian distribution and $s_{\boldsymbol{\theta}} = [0; 1]$), $f_{\boldsymbol{\theta}}$, $g_{\boldsymbol{\theta}}$, λ and μ coincide with our first illustrative example in Section 2.1.1, see (9)-(10), up to the dependency in z_k. We also consider a deep version of the MTMC (D-MTMC) in which $g_{\boldsymbol{\theta}}$ is parameterized by a DNN (with one hidden layer of 100 neurons and ReLU activation function). For both continuous versions of the MTMC, we use the variational distribution

$$q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K|\boldsymbol{x}_K) = \prod_{k=1}^{K} q_{\boldsymbol{\varphi}}(z_k|z_{k-1}, x_k) = \prod_{k=1}^{K} \mathcal{N}(z_k; \nu_{\boldsymbol{\varphi}}(z_{k-1}, x_k)). \tag{53}$$

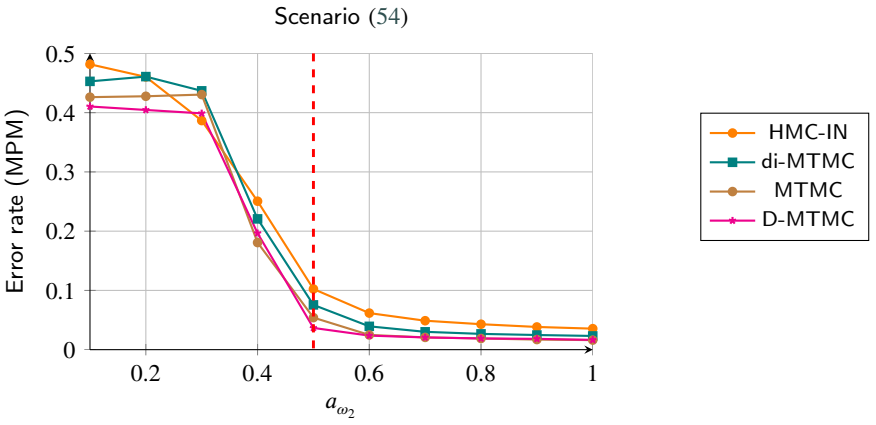
where $\nu_{\boldsymbol{\varphi}}(z_{k-1}, x_k)$ is parameterized by a DNN with one hidden layer of 100 neurons and a ReLU activation function. Finally, we consider a discrete version of the MTMC (di-MTMC) in which $z_k \in \{\nu_1, \nu_2\}$ is discrete [Gorynin et al., 2018, Li et al., 2019, Chen and Jiang, 2020]. For this model, Alg. 1 and 2 can be directly applied in the augmented space $\{\omega_1, \omega_2\} \times \{\upsilon_1, \upsilon_2\}$.

3.4.2. Experiments and results

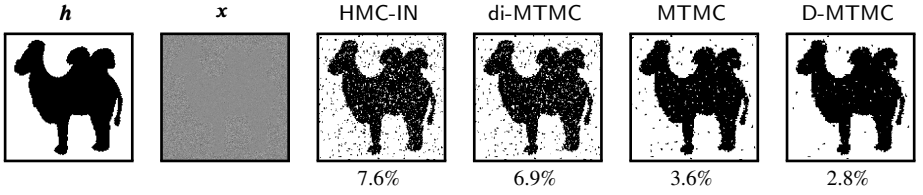
We now consider two scenarios in which binary images are corrupted with non elementary noises. In the first scenario, the hidden images $\boldsymbol{h}_K$ are the *camel*-type images of the Binary Shape Database and are corrupted with a stationary multiplicative noise,

$$x_k|h_k, z_k \sim \mathcal{N}\left(a_{h_k}; b^2_{h_k}\right) * z_k, \tag{54}$$

where $z_k \sim \mathcal{N}(0, 1)$, $a_{\omega_1} = 0, a_{\omega_2}$ is a varying parameter and $b_{\omega_1} = b_{\omega_2} = 0.2$. Fig. 5a displays the results for the setting $\beta_1 = 5, \beta_2 = 1$ in our variational approach. Scalar β_1 can be interpreted as enforcing the standardized Gaussian prior on the learnt latent variables, which is seemingly favorable on this example because of the way $\boldsymbol{z}_K$ is generated. β_2 is also needed and seems to guide the optimization so that the estimated $\hat{\boldsymbol{h}}_K$ corresponds to the desired segmentation. A particular classification is also displayed in Fig. 5b. As we see, our MTMC models improve the performance (up to a 7%-point improvement) of the HMC-IN. This comparison illustrates the interest of the third latent process $\boldsymbol{z}_K$. A



(a) Error rate from the unsupervised segmentations of Scenario (54). Results are averaged on all the *camel*-type images from the database.



(b) Selected illustrations for $a_{\omega_2} = 0.5$ (signaled by the red vertical line on Fig. 5a). Error rates appear below the images.

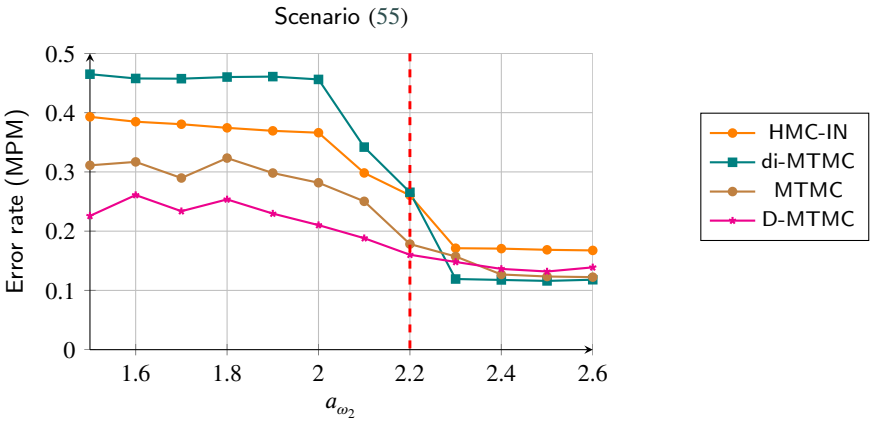
Figure 5: Unsupervised image segmentation with General Triplet Markov Chains (Scenario (54)).

slight advantage goes to the models with continuous $\boldsymbol{z}$ (MTMC and D-MTMC) over the di-MTMC which still performs better than the HMC-IN model. Note that in the case where we optimize directly the ELBO (*i.e.* $\beta_1 = 1$ and $\beta_2 = 0$), it has been observed that the classification obtained is not interpretable. This observation validates experimentally our strategy to adapt the objective function.

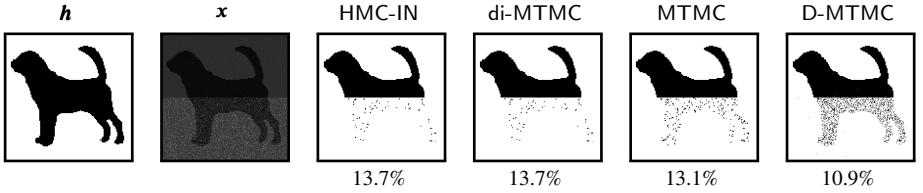
In the second scenario, the hidden images $\boldsymbol{h}_K$ are the *dog*-type images of the Binary Shape Database. They are corrupted by a non-stationary general noise,

$$\begin{cases} x_k|h_k \sim \mathcal{N}\left(a_{h_k}; \sigma^2\right), \text{ if } k \in \left\{1, \ldots, \left\lfloor \frac{K}{2} \right\rfloor\right\}, \\ x_k|h_k \sim a_{h_k} + \mathcal{E}\left(\lambda\right), \text{ if } k \in \left\{\left\lfloor \frac{K}{2} \right\rfloor + 1, \ldots, K\right\}, \end{cases} \tag{55}$$

where $\mathcal{E}(\lambda)$ is the exponential probability distribution of parameter λ, $a_{\omega_1} = 0$, a_{ω_2} is a varying parameter, $\sigma = 0.2$ and $\lambda = 1.4$. The main difficulty of this scenario is that the images are corrupted by two different noises with a relatively low level for both areas and have to be fitted in a unique model. For this scenario, we set $\beta_1 = 0.1$ and $\beta_2 = 0$. A small value of β_1 can be interpreted as a way to better fit the observations. Indeed, more flexibility seems to be needed to learn such a complex non-stationary noise. The reason why β_2 is set to 0 is that the pre-classification obtained with the HMC-IN is poor and should not be used to learn the parameters in the MTMC. It has been observed that other values deteriorate the final classification obtained with MTMC models. The results are displayed in Fig. 6a and Fig. 6b displays a particular classification. It is clear that the TMC models with a continuous auxiliary latent r.v. (MTMC and D-MTMC) offer a greater flexibility and are able to learn this complex multi-stationary noise. On the other hand the average classification provided by the di-MTMC or the HMC-IN models are irrelevant as soon as $a_{\omega_2} < 2$. This experiment illustrates the interest of a continuous auxiliary latent r.v. over discrete auxiliary latent r.v.; the latter being the only option that has been considered in the literature so far [Gorynin et al., 2018, Li et al., 2019, Chen and Jiang, 2020]. These experiments show the interesting capabilities of the generalized models to provide results in presence of very general noises. Coupled to the deep parameterization, a continuous third latent process enables our models to bypass the need of an explicit expression of the conditional distribution of the noise.



(a) Error rate from the unsupervised segmentations of Scenario (55). Results are averaged on all the *dog*-type images from the database.



(b) Selected illustrations for $a_{\omega_2} = 2.2$ (signaled by the red vertical line on Fig. 6a). Error rates appear below the images.

Figure 6: Unsupervised image segmentation with General Triplet Markov Chains (Scenario (55)).

4. Long memory dependency for PMCs

In this section, we propose a particular class of TMC which aims at extending the PMC model proposed in Section 2. The main motivation underlying this particular model is to introduce a long dependency on the past observations $\boldsymbol{x}_{k-1}$ of the r.v. (h_k, x_k), for all k. This dependency is introduced through the continuous latent process $\boldsymbol{z}_K$ and enables us to build an explicit joint distribution $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{x}_K)$ which does not satisfy the Markovian property of the PMC (2). The main difference with Section 3 is that $\boldsymbol{z}_K$ is now a conditional deterministic latent process. The resulting model is called a Partially PMC (PPMC). As we will see, this particular construction enables us to use directly the Bayesian inference framework developed in Section 2. Finally, since PMCs appears as particular TMCs, the pretraining of deep parameterized PPMCs is a direct adaptation of Section 3.3.

4.1. Partially Pairwise Markov Chains as deterministic TMCs

Let us focus on a particular case of the TMC (31)-(34). From now on, we consider that the conditional distribution ς coincides with the Dirac distribution δ, and that function $s_{\boldsymbol{\theta}}$ only depends on (z_{k-1}, x_{k-1}). Thus, z_k becomes deterministic given (z_{k-1}, x_{k-1}),

$$z_k = s_{\boldsymbol{\theta}}(z_{k-1}, x_{k-1}). \tag{56}$$

Each variable z_k can be interpreted as a summary of all the past observations $\boldsymbol{x}_{k-1}$. Consequently, it is easy to see that (33) and (34) now coincide with $p_{\boldsymbol{\theta}}(h_k|h_{k-1}, \boldsymbol{x}_{k-1})$ and $p_{\boldsymbol{\theta}}(x_k|\boldsymbol{h}_{k-1:k}, \boldsymbol{x}_{k-1})$, respectively, and marginalizing (3) w.r.t. $\boldsymbol{z}_K$ gives the explicit distribution of $(\boldsymbol{h}_K, \boldsymbol{x}_K)$,

$$p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{x}_K) = p_{\boldsymbol{\theta}}(h_0, x_0) \prod_{k=1}^{K} \underbrace{\lambda(h_k; f_{\boldsymbol{\theta}}(\boldsymbol{z}_{k-1:k}, h_{k-1}, x_{k-1}))}_{p_{\boldsymbol{\theta}}(h_k|h_{k-1}, \boldsymbol{x}_{k-1})} \underbrace{\mu(x_k; g_{\boldsymbol{\theta}}(\boldsymbol{z}_{k-1:k}, \boldsymbol{h}_{k-1:k}, x_{k-1}))}_{p(x_k|\boldsymbol{h}_{k-1:k}, \boldsymbol{x}_{k-1})}, \tag{57}$$

where z_k satisfies (56). It can noted that $(\boldsymbol{h}_K, \boldsymbol{x}_K)$ is no longer Markovian.

This kind of parameterization has an advantage in terms of Bayesian inference. Since z_k is a deterministic function of (z_{k-1}, x_{k-1}) (and so of $\boldsymbol{x}_{k-1}$, by induction), the conditional posterior distribution $p_{\boldsymbol{\theta}}(\boldsymbol{z}_k|z_{k-1}, \boldsymbol{x}_K)$ reduces to

$\delta_{s_{\boldsymbol{\theta}}(z_{k-1},x_{k-1})}$. Consequently, Alg. 1 and Alg. 2 can be directly applied to estimate $\boldsymbol{\theta}$ and h_k, for all k, by introducing the dependency in $\boldsymbol{z}_{k-1:k}$ in functions $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ of Section 2.1. An alternative point of view is that when z_k is deterministic, Alg. 4 can be seen as a particular instance of Alg. 1 in which we have set $q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K|\boldsymbol{x}_K) = p_{\boldsymbol{\theta}}(\boldsymbol{z}_K|\boldsymbol{x}_K)$, $\beta_1 = 1$ and $\beta_2 = 0$. Indeed, for this particular setting the objective function (45) coincides with the ELBO but also with the log-likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x}_K)$.

4.2. Deep-PPMC models

As previous models, we consider the case where PPMCs (57) are parameterized with DNNs. Such models will be referred to as D-PPMCs. In the particular case of PPMCs, $s_{\boldsymbol{\theta}}$ can be seen as a RNN, *i.e.* a neural network which admits the output of the network at previous time $k-1$ as input at time k [Hochreiter and Schmidhuber, 1997]. It is thus possible to directly combine our models with powerful RNN architectures such as Long Short Term Memory (LSTM) RNNs or Gated Recurrent Unit (GRU) RNNs which have been developed to introduce long term dependencies for sequential problems. Note that the gradient of $s_{\boldsymbol{\theta}}$ w.r.t. $\boldsymbol{\theta}$ can also be computed with a version of the backpropagation algorithm adapted to RNNs [Hochreiter and Schmidhuber, 1997, Chung et al., 2014].

The pretraining of this deep architecture is direct. The constrained output layer step is an application of Paragraph 3.3.1 with $q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K|\boldsymbol{x}_K) = p_{\boldsymbol{\theta}}(\boldsymbol{z}_K|\boldsymbol{x}_K)$, $\beta_1 = 1$ and $\beta_2 = 0$; so it can be seen as the step described for PMCs in Paragraph 2.2.1 up to the additional input $\boldsymbol{z}_{k-1:k}$.

The second step of our pretraining procedure of Paragraph 3.3.2 can also be simplified. Since in this particular case we have implicitly computed the optimal conditional variational distribution $q^{\text{opt}}_{\boldsymbol{\varphi}}(z_k|\boldsymbol{z}_{k-1},\boldsymbol{x}_K) = \delta_{s_{\boldsymbol{\theta}}(z_{k-1},x_{k-1})}(z_k)$, the reparameterized sample $\boldsymbol{z}_{k-1:k}$ of Fig. 4 is now deterministic and coincides directly with the output of $s_{\boldsymbol{\theta}}$, as shown in Fig. 7. Note that the parameters of $s_{\boldsymbol{\theta}}$ are unfrozen. The training process is summarized in Alg. 7.

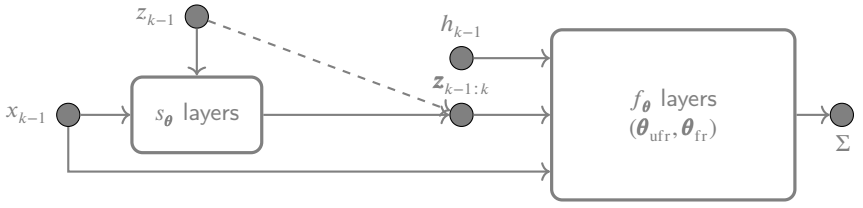


Figure 7: Graphical and condensed representation of the parameterization of $f_{\boldsymbol{\theta}}$ in the D-PPMC model. The dashed arrows represent the fact that some variables are copied.

Input: $\boldsymbol{x}_K$, the observation
Result: $\hat{\boldsymbol{h}}_K$, the final classification
/* Initialization of the output layer of $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ */
1 Estimate $\boldsymbol{\theta}^*_{\text{fr}}$ and $\hat{\boldsymbol{h}}^{\text{pre}}_K$ with Lines (1)-(3) of Alg. 3
/* Pretraining of $\boldsymbol{\theta}_{\text{ufr}}$ */
2 $\boldsymbol{\theta}^{(0)}_{\text{ufr}} \leftarrow \text{Backprop}(\hat{\boldsymbol{h}}^{\text{pre}}_K, \boldsymbol{x}_K, \boldsymbol{\theta}^*_{\text{fr}}, C_{f_{\boldsymbol{\theta}}}, C_{g_{\boldsymbol{\theta}}})$
/* Fine-tuning of the complete model */
3 Update all the models parameters (except $\boldsymbol{\theta}_{\text{fr}}$) with Alg. 1
4 Compute $\hat{\boldsymbol{h}}_K$ with Alg. 2

Algorithm 7: A general estimation algorithm for deep parameterizations of PPMC models.

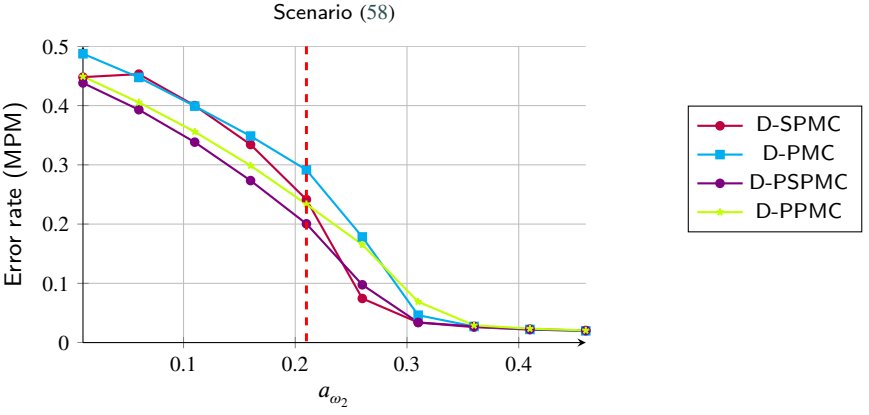
4.3. Simulations

We start again with the same experiments as those in Section 2.3, but we use an alternative noise which aims at introducing longer dependencies on the observations. We now set

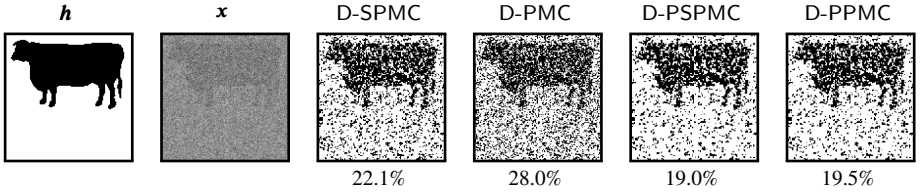
$$x_k|h_k, \boldsymbol{x}_{k-2:k-1} \sim \mathcal{N}\Big(\sin(a_{h_k} + 0.2(x_{k-1} + x_{k-2})); \sigma^2\Big). \tag{58}$$

where $a_{\omega_1} = 0$, $\sigma^2 = 0.25$ and a_{ω_2} is a varying parameter. We compare the deep models of Section 2 (D-SMPC and D-PMC) with their natural extensions developed in this section (D-PSPMC and D-PPMC).

Fig. 8a illustrates the results involving the models we have just introduced. For $s_{\boldsymbol{\theta}}$ we use two independent standard RNNs with ReLU activation function, i.e. $z_k = [z_k^1, z_k^2] = [s_{\boldsymbol{\theta}}^1(z_{k-1}^1, x_{k-1}), s_{\boldsymbol{\theta}}^2(z_{k-1}^2, x_{k-1})]$; $f_{\boldsymbol{\theta}}$ (resp. $g_{\boldsymbol{\theta}}$) depends on $\boldsymbol{z}_{k-1:k}^1$ (resp. $\boldsymbol{z}_{k-1:k}^2$). In this setting, we found that the models worked the best when the dimensions of z_k^1 and of z_k^2 is 5. We can see that the more general parameterizations embedded in D-PSPMC and D-PPMC lead to an improvement of the D-PMC models; each D-PPMC model leading to a better accuracy than its D-PMC counterpart. The ability to model long term dependencies proves to be important to better solve the correlated noise. This experiment illustrates a way to take advantage of a deterministic auxiliary process: by strengthening the sequential dependencies between the hidden random variables.



(a) Error rate from the unsupervised segmentations of Scenario (58). Results are averaged on all the *cattle*-type images from the database.



(b) Selected illustrations for $a_{\omega_2} = 0.21$ (signaled by the red vertical line on Figure 8a). Error rates appear below the images.

Figure 8: Unsupervised image segmentation with Partially Pairwise Markov Chains.

5. Experiments on real datasets

We finally experiment our models on two real datasets. The first one is devoted to a medical images. The main challenge of this kind of data is that the noise associated to such images is unknown and non usual; that is why we introduce our TMCs to measure the impact of the third latent process. The next dataset is related to human activity recognition. For this problem, the dependencies between the r.v. (the class and the observed r.v.) are critical; that is why we focus on the impact of our PMCs.

5.1. Unsupervised segmentation of biomedical images

We first illustrate the potential of the generalized TMC models on real biomedical data. The taks consists in the segmentation of micro-computed tomography X-ray scans of human arteries containing a metallic stent biomaterial[3]. These images are reminiscent of the synthetic experiment of Scenario (55): some regions exhibit a particular type of correlated noise (because of of the beam hardening artifacts caused by the interactions between X-rays and the metallic stent) and some regions do not.

Table 1 and Fig. 9 summarize the experiment. It can be seen that the classical models (HMC-IN and di-MTMC) are unable to handle the non-stationarity of the noise. The di-MTMC model even fail to provide any improvement

[3]The authors want to acknowledge Dr. Salomé Kuntz (GEPROMED, Strasbourg, France) for the acquisition of the micro-computed tomography images.

Slice	HMC-IN	di-MTMC	MTMC	D-MTMC
Average	8.6	8.6	7.6	**6.5**

Table 1
Averaged error rates (%) in unsupervised image segmentation with all the generalized TMCs assessed on ten micro-computed tomography slices. The detailed scores are given in App. A.2.

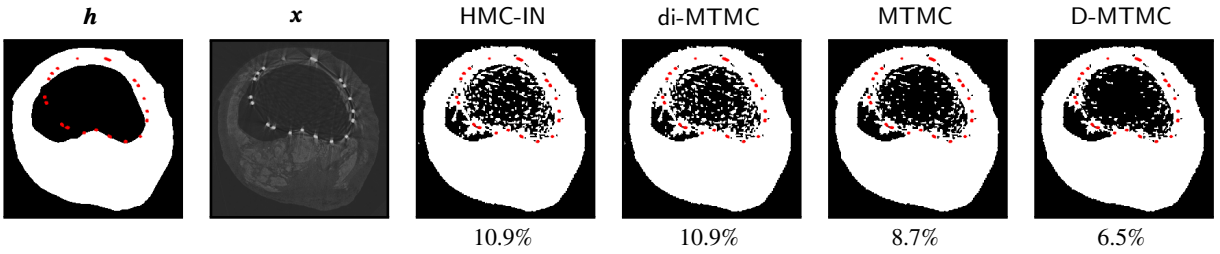


Figure 9: Illustration of the unsupervised segmentation of slice B, as reported in Table 1. The D-MTMC appears to better fit the non-stationary noise, offering a 4%-point improvement in the error rate. The stent components appearing in red are segmented beforehand with a thresholding technique and are considered as image borders during the segmentation using the probabilistic models.

Data	HMC-IN	SPMC	D-SPMC	D-PSPMC	PMC	D-PMC	D-PPMC
Average	25.2	21.3	16.8	**16.7**	17.1	16.8	16.8

Table 2
Averaged error rates (%) in the binary clustering of the first twenty raw entries of the HAPT dataset [Reyes-Ortiz et al., 2016]. The detailed scores are given in App. A.2.

over the HMC-IN model. On the other hand, major improvements can be seen when using the TMC models with a continuous auxiliary process, suggesting that the latter model offers more flexibility and that our parameter estimation algorithm enables to take advantage of it. These results on real-world data corroborates the results found in the synthetic experiment given in Section 3.4.2. Note that, in this case, we set $\beta_1 = 5$, $\beta_2 = 1$ and used the HMC-IN classification as a pre-segmentation. The network configurations are the same as in Section 3.4.2.

5.2. Unsupervised clustering for human activity recognition

We now illustrate the performances of classical PMC models, deep PMC models and deep PPMC models on a real clustering task linked with human activity recognition. We use the Human Activity and Postural Transition (HAPT) dataset described in [Reyes-Ortiz et al., 2016][4]. It consists of three-dimensional time series that we wish to cluster into two classes: *movement* and *no movement*.

The results are given in Table 2 for models sharing the same configurations with the models in Section 2.3 and 4.2. First of all, the modelization using the pairwise models seems very relevant in this application since we notice up to a 9%-point improvement over the HMC-IN model. In the case of the SPMCs, we clearly see the advantage of using deep parameterizations over the shallow models. The advantage of the deep parameterization is less significant in the PMC case. The contributions of the D-PSPMC and D-PPMC models are also less significant. The absence of gains in error rate when using the most complex models might be related to the limited length of the training sequences in this application (sequences of length between 15000 and 20000).

6. Conclusion

In this paper, we have proposed a general framework for PMC and TMC models which fully exploits the modeling power offered by such models for unsupervised signal processing. Contrary to previous work on TMCs, we have

[4] http://archive.ics.uci.edu/ml/datasets/smartphone-based+recognition+of+human+activities+and+postural+transitions

introduced a continuous latent process. For these models, we have derived Bayesian inference algorithms for estimating their parameters and the associated hidden r.v. and we have emphasized the case where the parameterization relies on DNNs. Our algorithms rely on an objective function deduced from the variational Bayesian inference but which has been modified in order to include the interpretability of the discrete hidden r.v.

This contribution enables us to propose an efficient answer to three recurrent questions linked with the practical applications of complex probabilistic graphical models for sequential data: which probability distributions to choose, how to parameterize them and how to estimate their parameters in an unsupervised way. For several applications, it has indeed been showed that our global procedure leads to new models which consistently performs better than the classical ones. Importantly, the ability of these models to tackle more complex noises comes without no additional effort from the signal processing point of view. Our experiments also suggest that it possible to model complex noises by using the universal approximating properties of DNNs and by training them in an unsupervised way with the new algorithms that we propose.

References

O. Abdel-Hamid, L. Deng, and D. Yu. Exploring convolutional neural network structures and optimization techniques for speech recognition. In *Interspeech 2013*. ISCA, August 2013.

S. Balakrishnan, M. J. Wainwright, B. Yu, et al. Statistical guarantees for the EM algorithm: From population to sample-based analysis. *Annals of Statistics*, 45(1):77–120, 2017.

J. Bayer and C. Osendorfer. Learning stochastic recurrent networks. *CoRR*, abs/1411.7610, 2014. URL `http://arxiv.org/abs/1411.7610`.

M. E.-Y. Boudaren and W. Pieczynski. Dempster–Shafer fusion of evidential pairwise Markov chains. *IEEE Transactions on Fuzzy Systems*, 24(6): 1598–1610, 2016.

W. Chan, N. Jaitly, Q. Le, and O. Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964. IEEE, 2016.

S. Chen and X. Jiang. Modeling repayment behavior of consumer loan in portfolio across business cycle: A triplet Markov model approach. *Complexity*, 2020, 2020.

J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems*, pages 2980–2988, 2015.

R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference*, volume 307 of *ACM International Conference Proceeding Series*, pages 160–167. ACM, 2008.

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, 1977.

L. Deng, G. Hinton, and B. Kingsbury. New types of deep neural network learning for speech recognition and related applications: An overview. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 8599–8603. IEEE, 2013.

S. Derrode and W. Pieczynski. Signal and image segmentation using pairwise Markov chains. *IEEE Transactions on Signal Processing*, 52(9): 2477–2489, 2004.

S. Derrode and W. Pieczynski. Unsupervised data classification using pairwise Markov chains with automatic copulas selection. *Computational Statistics & Data Analysis*, 63:81–98, 2013.

R. Douc and E. Moulines. Asymptotic properties of the maximum likelihood estimation in misspecified hidden Markov models. *Annals of Statistics*, 40(5):2697–2732, 2012.

A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

A. Doucet, N. de Freitas, and N. Gordon. An introduction to Sequential Monte carlo methods. In *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science, pages 3–14. Springer, 2001.

D. Erhan, A. Courville, Y. Bengio, and P. Vincent. Why does unsupervised pre-training help deep learning? In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS*, volume 9 of *JMLR Proceedings*, pages 201–208. JMLR.org, 2010.

P. Fearnhead, D. Wyncoll, and J. Tawn. A sequential smoothing algorithm with linear computational cost. *Biometrika*, 97(2):447–464, 2010.

J. Fu, H. Zheng, and T. Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4438–4446, 2017.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS*, volume 9 of *JMLR Proceedings*, pages 249–256. JMLR.org, 2010.

Y. Goldberg. Neural network methods for natural language processing. *Synthesis lectures on human language technologies*, 10(1):1–309, 2017.

I. Gorynin, H. Gangloff, E. Monfrini, and W. Pieczynski. Assessing the segmentation performance of pairwise and triplet Markov Models. *Signal Processing*, 145:183–192, 2018.

K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra. DRAW: A recurrent neural network for image generation. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1462–1471. JMLR.org, 2015.

I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. Beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR*. OpenReview.net, 2017.

G. E. Hinton, L. Deng, D. Yu, G. E. Dahl, Mohamed A-R., N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29 (6):82–97, 2012.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *2nd International Conference on Learning Representations, ICLR*, 2014.

D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems*, pages 3581–3589, 2014.

J. Klys, J. Snell, and R. Zemel. Learning latent subspaces in variational autoencoders. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems*, pages 6445–6455, 2018.

S. Kumar, J. Pradeep, and H. Zaidi. Learning robust latent representations for controllable speech synthesis. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3562–3575. Association for Computational Linguistics, 2021.

P. Lanchantin, J. Lapuyade-Lahorgue, and W. Pieczynski. Unsupervised segmentation of triplet Markov chains hidden with long-memory noise. *Signal Processing*, 88(5):1134–1151, 2008.

S. Le Cam, F. Salzenstein, and C. Collet. Fuzzy pairwise Markov chain to segment correlated noisy data. *Signal processing*, 88(10):2526–2541, 2008.

F. Lehmann and W. Pieczynski. Suboptimal Kalman filtering in triplet markov models using model order reduction. *IEEE Signal Processing Letters*, 27:1100–1104, 2020.

H. Li, S. Derrode, and W. Pieczynski. An adaptive and on-line IMU-based locomotion activity classification method using a triplet markov model. *Neurocomputing*, 362:94–105, 2019.

I. J. Michael, G. Zoubin, T. S. Jaakola, and L. K. Saul. An introduction to variational methods for graphical models. In *MACHINE LEARNING*, pages 183–233. MIT Press, 1999.

T. Mikolov, A. Joulin, S. Chopra, M. Mathieu, and M.-A. Ranzato. Learning longer memory in recurrent neural networks. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015.

A. Mohamed, G. E. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech and Langage Processing*, 20(1):14–22, January 2012.

K. Morales and Y. Petetin. Variational Bayesian inference for pairwise markov models. In *2021 IEEE Statistical Signal Processing Workshop (SSP)*, pages 251–255, 2021.

W. Pieczynski. Chaines de Markov triplet. *Comptes Rendus de l'Academie des Sciences - Mathematiques*, 335:275–278, 2002. in French.

W. Pieczynski. Pairwise Markov chains. *IEEE Transactions on pattern analysis and machine intelligence*, 25(5):634–639, 2003.

W. Pieczynski. Multisensor triplet Markov chains and theory of evidence. *International Journal of Approximate Reasoning*, 45(1):1–16, 2007.

W. Pieczynski and F. Desbouvries. On triplet Markov chains. In *International Symposium on Applied Stochastic Models and Data Analysis, (ASMDA)*, 2005.

A. Pinkus. Approximation theory of the MLP model in neural networks. *Acta numerica*, 8:143–195, 1999.

L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita. Transition-aware human activity recognition using smartphones. *Neurocomputing*, 171:754–767, 2016.

S. Ruder. An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747, 2016. URL `http://arxiv.org/abs/1609.04747`.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

H. Sagan. *Space-filling curves*. Springer, 2012.

B. B. Traore, B. Kamsu-Foguem, and F. Tangara. Deep convolution neural network for image recognition. *Ecological Informatics*, 48:257–268, 2018.

D. G. Tzikas, A. C. Likas, and N. P. Galatsanos. The variational approximation for Bayesian inference. *IEEE Signal Processing Magazine*, 25(6): 131–146, 2008.

H. White. Maximum likelihood estimation of misspecified models. *Econometrica*, 50(1):1–25, 1982.

L. Xu and M. I. Jordan. On convergence properties of the EM algorithm for Gaussian mixtures. *Neural computation*, 8(1):129–151, 1996.

H. Zheng, J. Fu, T. Mei, and J. Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 5209–5217, 2017.

A. Appendix

A.1. Proof of Proposition 1

The ELBO

$$F(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \sum_{\boldsymbol{h}_K} \int q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K, \boldsymbol{z}_K | \boldsymbol{x}_K) \log \left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{z}_K, \boldsymbol{x}_K)}{q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K, \boldsymbol{z}_K | \boldsymbol{x}_K)} \right) \mathrm{d}\boldsymbol{z}_K \tag{59}$$

can be decomposed as

$$\begin{aligned} F(\boldsymbol{\theta}, \boldsymbol{\varphi}) = & \int \overbrace{\left[\sum_{\boldsymbol{h}_K} q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K | \boldsymbol{z}_K, \boldsymbol{x}_K) \right]}^{1} q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K | \boldsymbol{x}_K) \log \left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{z}_K, \boldsymbol{x}_K)}{q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K | \boldsymbol{x}_K)} \right) \mathrm{d}\boldsymbol{z}_K \\ & - \int q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K | \boldsymbol{x}_K) \mathrm{D}_{\mathrm{KL}} \left(q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K | \boldsymbol{z}_K, \boldsymbol{x}_K) || p_{\boldsymbol{\theta}}(\boldsymbol{h}_K | \boldsymbol{z}_K, \boldsymbol{x}_K) \right) \mathrm{d}\boldsymbol{z}_K, \qquad (60) \\ \leq & \int q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K | \boldsymbol{x}_K) \log \left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{z}_K, \boldsymbol{x}_K)}{q_{\boldsymbol{\varphi}}(\boldsymbol{z}_K | \boldsymbol{x}_K)} \right) \mathrm{d}\boldsymbol{z}_K = F^{\mathrm{opt}}(\boldsymbol{\theta}, \boldsymbol{\varphi}). \qquad (61) \end{aligned}$$

We have $F(\boldsymbol{\theta}, \boldsymbol{\varphi}) = F^{\mathrm{opt}}(\boldsymbol{\theta}, \boldsymbol{\varphi})$ when the KLD term in (60) is null, *i.e.* when $q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K | \boldsymbol{z}_K, \boldsymbol{x}_K) = p_{\boldsymbol{\theta}}(\boldsymbol{h}_K | \boldsymbol{z}_K, \boldsymbol{x}_K)$. It remains to compute $F^{\mathrm{opt}}(\boldsymbol{\theta}, \boldsymbol{\varphi})$. Starting again from (59) where we set $q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K, \boldsymbol{z}_K | \boldsymbol{x}_K) = q_{\boldsymbol{\varphi}}(\boldsymbol{h}_K | \boldsymbol{x}_K) p_{\boldsymbol{\theta}}(\boldsymbol{h}_K | \boldsymbol{z}_K, \boldsymbol{x}_K)$, the Markovian structure of $p_{\boldsymbol{\theta}}(\boldsymbol{h}_K, \boldsymbol{z}_K, \boldsymbol{x}_K)$ and the additive property of the logarithm function give the decomposition (28)-(30).

Note that the computation of $F^{\mathrm{opt}}(\boldsymbol{\theta}, \boldsymbol{\varphi})$ via (28)-(30) relies on $p_{\boldsymbol{\theta}}(\boldsymbol{h}_{k-1:k} | \boldsymbol{z}_K, \boldsymbol{x}_K)$. It can be computed from a direct extension of the intermediate quantities $\alpha_{\boldsymbol{\theta},k}$ and $\beta_{\boldsymbol{\theta},k}$ which are now defined as $\alpha_{\boldsymbol{\theta},k}(h_k) = p_{\boldsymbol{\theta}}(h_k, \boldsymbol{z}_k, \boldsymbol{x}_k)$ and $\beta_{\boldsymbol{\theta},k}(h_k) = p_{\boldsymbol{\theta}}(\boldsymbol{z}_{k+1:K}, \boldsymbol{x}_{k+1:K} | h_k, z_k, x_k)$. Their computation is similar to (16) and (18), except that they now involve the transition $p(h_k, z_k, x_k | h_{k-1}, z_{k-1}, x_{k-1})$ rather than $p(h_k, x_k | h_{k-1}, x_{k-1})$.

A.2. Detailed error rates for experiments 5.1 and 5.2

This section provides the full results of the real world experiments of Section 5.

Slice	HMC-IN	di-MTMC	MTMC	D-MTMC
A	8.5	8.5	6.5	5.4
B	10.9	10.9	8.7	6.5
C	6.9	7.0	6.0	5.2
D	10.0	10.1	8.3	6.1
E	6.5	6.3	6.2	5.4
F	11.5	11.5	10.8	9.3
G	4.6	4.6	3.9	3.7
H	8.6	8.6	8.5	7.7
I	11.5	11.5	10.1	9.2
J	7.2	7.2	6.9	6.5
Average	8.6	8.6	7.6	**6.5**

Table 3
Detailed error rates (%) in unsupervised image segmentation with all the generalized TMCs assessed on ten micro-computed tomography slices. See Section 5.1.

Data	HMC-IN	SPMC	D-SPMC	D-PSPMC	PMC	D-PMC	D-PPMC
acc_exp01_user01	15.0	29.0	20.9	17.8	20.9	19.9	20.1
acc_exp02_user01	16.0	20.3	13.3	12.4	13.1	18.2	14.6
acc_exp03_user02	25.7	16.1	11.7	9.8	11.7	5.6	12.7
acc_exp04_user02	24.3	15.2	10.9	11.5	10.9	5.6	11.7
acc_exp05_user03	21.1	28.8	23.2	15.3	22.4	22.7	23.4
acc_exp06_user03	26.3	15.6	12.9	11.0	12.3	19.9	14.2
acc_exp07_user04	23.3	19.2	14.4	13.4	23.3	21.9	14.6
acc_exp08_user04	26.3	17.1	13.1	12.3	12.9	10.4	12.9
acc_exp09_user05	24.3	19.0	14.9	12.3	14.7	12.3	15.5
acc_exp10_user05	25.8	48.3	24.5	25.4	24.3	27.6	24.3
acc_exp11_user06	27.7	15.1	12.7	10.9	12.7	12.6	11.9
acc_exp12_user06	36.9	43.5	42.8	43.2	42.8	42.1	41.5
acc_exp13_user07	26.1	18.2	14.6	16.5	14.4	13.9	13.9
acc_exp14_user07	26.0	18.5	14.5	21.9	14.4	18.9	13.6
acc_exp15_user08	22.2	16.7	12.9	9.0	12.8	10.0	13.0
acc_exp16_user08	26.2	19.4	16.5	14.7	16.5	15.8	14.3
acc_exp17_user09	25.6	17.0	13.1	17.9	12.9	14.0	11.0
acc_exp18_user09	24.8	13.8	10.9	11.3	10.8	8.1	12.3
acc_exp19_user10	26.1	13.3	10.4	21.4	10.3	8.0	15.2
acc_exp20_user10	34.9	22.1	27.2	26.8	27.1	29.1	25.9
Average	25.2	21.3	16.8	**16.7**	17.1	16.8	16.8

Table 4
Detailed Error rates (%) in the binary clustering of the first twenty raw entries of the HAPT dataset [Reyes-Ortiz et al., 2016].