A GENERAL PARAMETRIZATION FRAMEWORK FOR PAIRWISE MARKOV MODELS: AN APPLICATION TO UNSUPERVISED IMAGE SEGMENTATION

Hugo Gangloff, Katherine Morales, Yohan Petetin

A GENERAL PARAMETRIZATION FRAMEWORK FOR PAIRWISE MARKOV MODELS: AN APPLICATION TO UNSUPERVISED IMAGE SEGMENTATION

Hugo Gangloff, Katherine Morales, Yohan Petetin

Samovar, Telecom Sudparis, Institut Polytechnique de Paris

ABSTRACT

Probabilistic graphical models such as Hidden Markov models have found many applications in signal processing. In this paper, we focus on a particular extension of these models, the Pairwise Markov models. We propose a general parametrization of the probability distributions describing the Pairwise Markov models which enables us to combine them with recent architectures from machine learning such as deep neural networks. In order to evaluate the power of these combined architectures, we focus on the unsupervised image segmentation problem which is particularly challenging and we propose a new parameter estimation algorithm. We show that our models with their associated estimation algorithm outperforms the classical probabilistic models for the task of unsupervised image segmentation.

Index Terms— Pairwise Markov Chains, Image Segmentation, Neural Networks, Gradient Expectation-Maximization.

1. INTRODUCTION

Let $\boldsymbol{X} = (X_1, \cdots, X_K)$ (resp. $\boldsymbol{H} = (H_1, \cdots, H_K)$) be a sequence of observed (resp. latent) random variables (r.v.), where $X_k \in \mathbb{R}$ and $H_k \in \Omega = \{\omega_1, \cdots, \omega_C\}$, for all k, $1 \leq k \leq K$. The joint distribution of $(\boldsymbol{H}, \boldsymbol{X})$ is denoted $p(\boldsymbol{h}, \boldsymbol{x})$. In this paper, we focus on the estimation of H_k from a realization $\boldsymbol{X} = \boldsymbol{x}$ through the posterior distribution $p(h_k|\boldsymbol{x})$, for all k, $1 \leq k \leq K$. Image segmentation is a critical application of this Bayesian estimation problem. In this context, the hidden r.v. H_k is associated to the class (e.g., black or white) of the k-th pixel of an image, while X_k represents a noisy observation (e.g., a grayscale observation) of the pixel (see Fig. 4). This problem relies on a relevant probabilistic model $p_{\boldsymbol{\theta}}(\boldsymbol{h}, \boldsymbol{x})$ which should be able to model jointly the hidden classes and the observed signal but in which it is also possible to perform (unsupervised) Bayesian inference, that is to say to estimate $\boldsymbol{\theta}$ from a realization $\boldsymbol{X} = \boldsymbol{x}$, and next to compute or to approximate the marginal smoothing distribution $p_{\boldsymbol{\theta}}(h_k|\boldsymbol{x})$, for all k, $1 \leq k \leq K$.

Hidden Markov Chains (HMCs) define an important family of probabilistic graphical models and have been the subject of many investigations for the Bayesian image segmentation application [1]. In an HMC, $\boldsymbol{H}$ is a Markov chain and given $\boldsymbol{H}$, the observations $\boldsymbol{X}$ are independent and X_k only depends on H_k. These models have been generalized by the introduction of the Pairwise Markov Chains (PMCs) [2] which only satisfy the assumption that the pair $(\boldsymbol{H}, \boldsymbol{X})$ is a Markov chain, while keeping the computational properties of HMCs. They rely on a transition distribution $p_{\boldsymbol{\theta}}(h_k, x_k|h_{k-1}, x_{k-1})$ and have also received a particular attention for image segmentation, see e.g. [3, 4, 5].

By contrast, artificial neural networks do not model the observations with a probabilistic model. However, as universal approximators $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ of an unknown function $f(\boldsymbol{x})$ [6], deep neural networks (DNNs) have gained in popularity due to their excellent performances in many tasks such as classification. The parameters $\boldsymbol{\theta}$ underlying to $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ are estimated in a supervised way by the backpropagation algorithm from a labeled training dataset [7]. These architectures have been recently combined with probabilistic graphical models in order to provide powerful generative models [8, 9, 10] which aim at modeling an unknown distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ of observations.

Let us turn now to the contributions of this paper. As stated above, PMCs are very general models for unsupervised and sequential Bayesian classification. However, and up to our best knowledge, their application have been restricted to stationary PMCs [3, 11], i.e. models defined from a joint distribution $p_{\boldsymbol{\theta}}(h_{k-1}, h_k, x_{k-1}, x_k)$ which does not depend on k. This additional assumption may be motivated by the fact that the choice of a transition distribution $p_{\boldsymbol{\theta}}(h_k, x_k|h_{k-1}, x_{k-1})$ for a given problem is not obvious. In this paper, we relax this underlying assumption and we first propose a general parametrization of PMC models and an associated Bayesian inference algorithm. Next, we focus on the unsupervised image segmentation problem; to that end, we exploit our general framework by considering a parametrization of our PMC models with DNNs, in the spirit of the Variational Auto Encoder [8]. For this application, and contrary to the Variational Auto Encoder, we take into account that the latent r.v. H_k has to be interpretable, since it is associated to the class of pixel k; we thus propose a particular tuning of these architectures.

The rest of the paper is organized as follows. In section 2, we briefly review the rationale of PMC models and we introduced a general parametrization of these models. In Section

3, we combine PMC models with DNN architectures and we propose an estimation algorithm for the unsupervised image segmentation problem. We finally compare our new models with the classical ones on some experiments.

2. GENERAL PAIRWISE MARKOV MODELS

2.1. A brief review of PMC models

As recalled in the Introduction, the HMC is a popular model which satisfies

$$p(\boldsymbol{h},\boldsymbol{x}) \overset{\text{HMC}}{=} p(h_1,x_1)\prod_{k=2}^{K} p(h_k|h_{k-1})p(x_k|h_k). \quad (1)$$

In other words, $\boldsymbol{H}$ is a Markov chain, and $p(\boldsymbol{x}|\boldsymbol{h}) = \prod_{k=1}^{K} p(x_k|h_k)$; the graphical representation of the model is given in Fig. 1a. Actually, in model (1), the pair $(\boldsymbol{H},\boldsymbol{X})$ is a Markov chain in which the transition distribution satisfies a particular factorization; in other words, model (1) can be rewritten as

$$p(\boldsymbol{h},\boldsymbol{x}) \overset{\text{PMC}}{=} p(h_1,x_1)\prod_{k=2}^{K} p(h_k,x_k|h_{k-1},x_{k-1}), \quad (2)$$

where

$$p(h_k,x_k|h_{k-1},x_{k-1}) \overset{\text{HMC}}{=} p(h_k|h_{k-1})p(x_k|h_k). \quad (3)$$

So from now on, we will consider the general PMC model (2) in which the transition distribution is the most general,

$$p(h_k,x_k|h_{k-1},x_{k-1}) \overset{\text{PMC}}{=} p(h_k|h_{k-1},x_{k-1})p(x_k|h_k,h_{k-1},x_{k-1}). \quad (4)$$

The graphical representation of this model is given in Fig. 1c; as we see, this model also generalizes some recent Stochastic Recurrent Neural architectures [12].

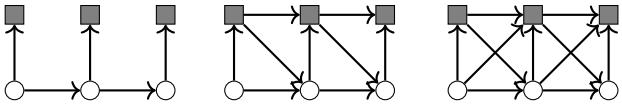
Finally, we also introduce the semi PMC (SPMC), an intermediate model between the PMC and the HMC In the SPMC, the observation X_k no longer depends on H_{k-1} given (H_{k-1}, X_{k-1}, H_k),

$$p(h_k,x_k|h_{k-1},x_{k-1}) \overset{\text{SPMC}}{=} p(h_k|h_{k-1},x_{k-1})p(x_k|h_k,x_{k-1}). \quad (5)$$

The graphical representation of the SPMC is given in Fig. 1b.

2.2. A general parametrization of PMC models

We now introduce a function $f_{\boldsymbol{\theta}}(h_{k-1},x_{k-1})$ (resp. $g_{\boldsymbol{\theta}}(h_k, h_{k-1},x_{k-1})$) of (h_{k-1},x_{k-1}) (resp. of (h_k,h_{k-1},x_{k-1})) which depends on an unknown parameter $\boldsymbol{\theta}$; we also assume that these functions are differentiable w.r.t. $\boldsymbol{\theta}$. The transition distribution of the PMC (4) is parametrized through $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$,



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

Fig. 1: Graphical representations of the HMC, SPMC and PMC models of section 2.1. The white circles (resp. gray squares) represent the hidden (resp. observed) r.v. As we will see in section 3.1, the transition distributions of these models can be parametrized by a DNN, leading to *Deep* PMC models.

$$p_{\boldsymbol{\theta}}(h_k|h_{k-1},x_{k-1}) = \lambda(h_k; f_{\boldsymbol{\theta}}(h_{k-1},x_{k-1})), \quad (6)$$
$$p_{\boldsymbol{\theta}}(x_k|h_k,h_{k-1},x_{-1}) = \mu(x_k; g_{\boldsymbol{\theta}}(h_k,h_{k-1},x_{k-1})), \quad (7)$$

where $\lambda(h;z)$ (resp. $\mu(x;z')$) is a probability distribution on Ω (resp. a probability density function on $\mathbb{R}$) whose parameters are function of z (resp. z') and which is differentiable w.r.t z (resp. w.r.t. z'). We give below an example of a Gaussian PMC to illustrate the notations and the general parametrization introduced.

Example 1 *Let us consider the case where Ω consists of two classes, i.e. $\Omega = \{\omega_1,\omega_2\}$, λ is a Bernoulli distribution and μ a Gaussian one. In addition, the parameters of these distributions are linear, so we have*

$$f_{\boldsymbol{\theta}}(h_{k-1},x_{k-1}) = a_{h_{k-1}}x_{k-1} + b_{h_{k-1}}, \quad (8)$$
$$g_{\boldsymbol{\theta}}(h_k,h_{k-1},x_{k-1}) = c_{h_k,h_{k-1}}x_{k-1} + d_{h_k,h_{k-1}}, \sigma_{h_k,h_{k-1}}] \quad (9)$$
$$\lambda(h=\omega_1;z) = \text{sigm}(z) = \frac{1}{1+\exp(-z)}, \quad (10)$$
$$\mu(x;z'=(z_1',z_2')) = \mathcal{N}(x;z_1';(z_2')^2), \quad (11)$$

($\mathcal{N}(x;m;\sigma^2)$ is the Gaussian distribution with mean m, variance σ^2 taken at point x). Here,

$$\boldsymbol{\theta} = \{(a_{\omega_i}, b_{\omega_i}, c_{\omega_j,\omega_i}, d_{\omega_j,\omega_i}, \sigma_{\omega_j,\omega_i}) | (i,j) \in \{1,2\}^2\}.$$

The equivalent SPMC is obtained when the parameters in (9) *do not depend on h_{k-1}; the HMC satisfies in addition $a_{h_{k-1}} = 0$.*

As we have seen in the example above, $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ are linear functions. Of course, our general parametrization allows more general functions. In section 3, we will focus on functions parametrized by DNNs in Section 3.1.

2.3. Bayesian estimation

Let us now discuss on the computation of $p_{\boldsymbol{\theta}}(h_k|\boldsymbol{x})$ for all k in model (2) satisfying (6)-(7). The general expressions derived for the PMCs in [2] and which are a direct extension

of the Forward-Backward algorithm [13] are still valid here. More precisely, let us set $\alpha_{\boldsymbol{\theta},k}(h_k) = p_{\boldsymbol{\theta}}(x_1, \cdots, x_k, h_k)$ and $\beta_{\boldsymbol{\theta},k}(h_k) = p_{\boldsymbol{\theta}}(x_{k+1}, \cdots, x_K | h_k, x_k)$, $\beta_{\boldsymbol{\theta},K}(h_K) = 1$ and remember that

$$p_{\boldsymbol{\theta}}(h_k, x_k | h_{k-1}, x_{k-1}) = \lambda(h_k; f_{\boldsymbol{\theta}}(h_{k-1}, x_{k-1})) \times \mu(x_k; g_{\boldsymbol{\theta}}(h_k, h_{k-1}, x_{k-1}));$$

then, using the Markovian property of $p_{\boldsymbol{\theta}}(\boldsymbol{h}, \boldsymbol{x})$ in (4), we have for all k, $1 \leq k \leq K$,

$$\alpha_{\boldsymbol{\theta},k}(h_k) = \sum_{h_{k-1}} \alpha_{\boldsymbol{\theta},k-1}(h_{k-1}) p_{\boldsymbol{\theta}}(h_k, x_k | h_{k-1}, x_{k-1}), \quad (12)$$

and next for all k, $K > k \geq 1$,

$$\beta_{\boldsymbol{\theta},k-1}(h_{k-1}) = \sum_{h_k} \beta_{\boldsymbol{\theta},k}(h_k) p_{\boldsymbol{\theta}}(h_k, x_k | h_{k-1}, x_{k-1}). \quad (13)$$

We finally deduce

$$p_{\boldsymbol{\theta}}(h_{k-1}, h_k | \boldsymbol{x}) \propto \alpha_{\boldsymbol{\theta},k-1}(h_{k-1}) \times \beta_{\boldsymbol{\theta},k}(h_k) \times p_{\boldsymbol{\theta}}(h_k, x_k | h_{k-1}, x_{k-1}), \quad (14)$$

and so

$$p_{\boldsymbol{\theta}}(h_k | \boldsymbol{x}) = \sum_{h_{k-1}} p_{\boldsymbol{\theta}}(h_{k-1}, h_k | \boldsymbol{x}). \quad (15)$$

An estimate of H_k is deduced from the maximum posterior mode criterion [14].

2.4. Parameter estimation

In practice, $\boldsymbol{\theta}$ is unknown and has to be estimated from a realization $\boldsymbol{X} = \boldsymbol{x}$. Unlike the method proposed in [2] for stationary PMC models, we propose a maximum likelihood estimation approach with a variant of the Expectation-Maximisation (EM) algorithm [15]. For a given parameter $\boldsymbol{\theta}^{(j)}$, it relies on the computation and the maximization w.r.t. $\boldsymbol{\theta}$ of

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)}) = \mathrm{E}_{\boldsymbol{\theta}^{(j)}}(\log(p_{\boldsymbol{\theta}}(\boldsymbol{h}, \boldsymbol{x})) | \boldsymbol{x}),$$

which reads (up to the initial distribution of the PMC)

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)}) = \sum_{k=2}^{K} \sum_{h_{k-1}, h_k} p_{\boldsymbol{\theta}^{(j)}}(h_{k-1}, h_k | \boldsymbol{x}) \times \log(\lambda(h_k; f_{\boldsymbol{\theta}}(h_{k-1}, x_{k-1})) \mu(x_k; g_{\boldsymbol{\theta}}(h_k, h_{k-1}, x_{k-1}))). \quad (16)$$

Note that $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)})$ can be exactly computed from (14) regardless of the parametrization (6)-(7).

It remains to maximize $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)})$ w.r.t. $\boldsymbol{\theta}$. In general PMC models, computing $\arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)})$ is not possible, except for simple model such as the linear and Gaussian HMC discussed in Example 1. When the exact maximization is not possible, we resort to the Gradient EM (GEM) algorithm [16]; since $\lambda(h; z)$ and $\mu_x(x; z')$ (resp. $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$) are differentiable w.r.t. z and z' (resp. w.r.t. $\boldsymbol{\theta}$), it is possible to compute the gradient of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)})$ w.r.t. $\boldsymbol{\theta}$. Finally, introducing a learning rate ϵ, and according to [16], $\boldsymbol{\theta}^{(j)}$ is updated as

$$\boldsymbol{\theta}^{(j+1)} = \boldsymbol{\theta}^{(j)} + \epsilon \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)}) \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(j)}}. \quad (17)$$

Algorithm 1 summarizes the inference and parameter estimation processes for our general PMC models.

Data: A realization $\boldsymbol{X} = \boldsymbol{x}$, a learning rate ϵ
Result: $\hat{\boldsymbol{h}}$, the estimated hidden r.v.
$\boldsymbol{\theta}^*$, a set of estimated parameters

Initialize randomly $\boldsymbol{\theta}^{(0)}$
$j = 0$
while *convergence of* $p_{\boldsymbol{\theta}^{(j)}}(\boldsymbol{x})$ *is not attained* **do**
  Set $\alpha_{\boldsymbol{\theta}^{(j)},1} = p_{\boldsymbol{\theta}^{(j)}}(h_1, x_1)$
  for $k \leftarrow 2$ *to* K **do**
    Compute $\alpha_{\boldsymbol{\theta}^{(j)},k}(h_k)$ with (12)
  end
  Compute $p_{\boldsymbol{\theta}^{(j)}}(\boldsymbol{x}) = \sum_{h_K} \alpha_{\boldsymbol{\theta}^{(j)},K}(h_K)$
  Set $\beta_{\boldsymbol{\theta}^{(j)},K}(h_K) = 1$, for all h_K
  for $k \leftarrow K - 1$ *to* 1 **do**
    Compute $\beta_{\boldsymbol{\theta}^{(j)},k}(h_k)$ with (13)
  end
  Compute $p_{\boldsymbol{\theta}^{(j)}}(h_k, h_{k+1} | \boldsymbol{x})$ with (14), for all k
  Compute $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)})$ with (16)
  Set $\boldsymbol{\theta}^{(j+1)} = \boldsymbol{\theta}^{(j)} + \epsilon \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)}) \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(j)}}$
  $j \leftarrow j + 1$
end
$\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}^{(j)}$
$\hat{h}_k = \arg\max p_{\boldsymbol{\theta}^*}(h_k | \boldsymbol{x})$, for all k, with (15)

Algorithm 1: Unsupervised estimation in general PMC models.

3. DEEP PMC MODELS FOR UNSUPERVISED IMAGE SEGMENTATION

In this section, we now focus on a particular parametrization of functions $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ by (deep) neural networks, giving rise to deep PMC architectures. We discuss on the practical implementation of the Algo. 1 in this particular case, while taking into account that the hidden r.v. H_k is interpretable as the class associated to the observation X_k. We thus give a particular attention to the tuning of our models and to their initialization.

The contribution of our deep parametrization of PMC models are discussed from simulations in which we perform unsupervised binary image segmentation ($\Omega = \{\omega_1, \omega_2\}$) extracted from the Binary Shape Database [1]. Note that the

[1] http://vision.lems.brown.edu/content/available-software-and-databases

considered images are transformed into a 1-D signal $\boldsymbol{x}$ with a Hilbert-Peano filling curve [17].

3.1. Deep PMC architecures

3.1.1. The models

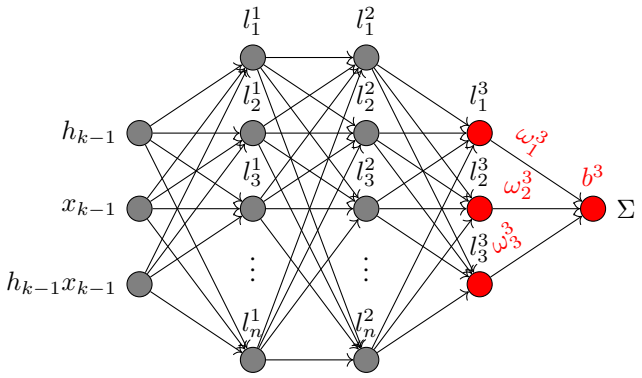
From now on, we will consider two classes of PMC models.

The first one is the class of non-deep PMC models and consists of the linear and Gaussian PMC and SPMC models described in Example 1. We easily show that the parameters related to $g_{\boldsymbol{\theta}}$ can be updated exactly from (16) (the expressions are not presented due to limited space) while those related to $f_{\boldsymbol{\theta}}$ are updated according to the gradient update rule (17).

The second class is the class of PMC models parametrized by (deep) neural networks and consists of the deep PMC (D-PMC) and the deep SPMC (D-SPMC) models. For both models, λ and μ are the distributions given in Exemple 1, but $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ are now parametrized by neural networks (e.g., see Fig. 2) with rectified linear activation functions for intermediate hidden layers and linear or square function for the ouput layer. The parameters associated to the last linear layers will be denoted $\boldsymbol{\theta}_{\text{fr}}$ while the other parameters are denoted $\boldsymbol{\theta}_{\text{ufr}}$ (the meaning of this notation will be clarified later). For the update rule, we use (17); the gradient of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)})$ w.r.t. $\boldsymbol{\theta}$ is indeed deduced from those of $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ which are computable with the backpropagation algorithm [18]. While such architectures are promising for modeling a large variety of PMC models, practical problems may appear when it comes to using them for unsupervised and interpretable estimation, due to the large number of parameters characterizing these architectures. In practice, when the parameters of the neural networks $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ are initialized randomly, we may encounter convergence issues to optimize $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)})$. More importantly, we are not ensured that the latent r.v. H_k which is learnt coincides with the original class associated to the observation X_k. To deal with this challenging optimization problem, we propose a solution in two steps that we now explain and which gives promising results in practice.

3.1.2. Constrained deep PMC architectures

We first constrain the parameters of the last layers associated to $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ to coincide with the parameters of the non deep and linear version of the equivalent model described by Example 1, which can be seen as neural networks without hidden layer. It means that the size of the last hidden layer (before the output layer) coincides with that of the input layer and that the associated weights and biases $\boldsymbol{\theta}_{\text{fr}}$ are deduced from the application of Algo. 1 with the non deep model (8)-(9). These parameters are next *frozen*. Fig. 2 describes an example of constrained DNN for the function $f_{\boldsymbol{\theta}}$ in (6). This step provides a pre-segmentation $\hat{\boldsymbol{h}}_{\text{pre}}$ which aims at helping the tuning of the remaining parameters $\boldsymbol{\theta}_{\text{ufr}}$ as we now see.



with $\Sigma = f_{\boldsymbol{\theta}}(h_{k-1}, x_{k-1}, h_{k-1}x_{k-1}) = \omega_1^3 l_1^3 + \omega_2^3 l_2^3 + \omega_3^3 l_3^3 + b^3$.

Fig. 2: Example of the proposed constrained architecture for $f_{\boldsymbol{\theta}}$: once the parameters of the last linear $\boldsymbol{\theta}_{\text{fr}}$ have been estimated, they are frozen.

3.1.3. Pretraining PMC models with deep architectures

In order to ensure that the final model indeed associates H_k to the class of X_k despite the large number of parameters, the *unfrozen* parameters $\boldsymbol{\theta}_{\text{ufr}}$ of $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ are initialized with some iterations of a supervised backpropagation algorithm which uses $\boldsymbol{x}$ as inputs and the latent variables $\hat{\boldsymbol{h}}_{\text{pre}}$ estimated by the constrained step above. The cost function of the backpropagation algorithm is the cross entropy for $f_{\boldsymbol{\theta}}$ and the mean square error for $g_{\boldsymbol{\theta}}$. The parameters $\boldsymbol{\theta}_{\text{ufr}}$ are next fine tuned (in an unsupervised way) with steps $2-19$ of Algo. 1 In practice, we have found that such a pretraining leads to a better initial point in terms of likelihood, and next to a faster and easier optimization.

The final Algo. 2 takes into account the specific constraints of PMC models parametrized by deep architectures.

Data: $\boldsymbol{x}_I$, the observed image
Result: $\hat{\boldsymbol{h}}_I$ the segmented image
$\boldsymbol{x} = (x_1, \ldots, x_K) \leftarrow \text{Peano\_transform}(\boldsymbol{x}_I)$
Estimate $\boldsymbol{\theta}_{\text{fr}}^*$ and $\hat{\boldsymbol{h}}_{\text{pre}}$ with Algo. 1
$\boldsymbol{\theta}_{\text{ufr}}^{(0)} \leftarrow \text{Backprop}(\hat{\boldsymbol{h}}_{\text{pre}}, \boldsymbol{x}, \boldsymbol{\theta}_{\text{fr}}^*)$
Compute $\boldsymbol{\theta}_{\text{ufr}}^*$ and $\hat{\boldsymbol{h}}$ with steps $2-19$ of Algo. 1
$\hat{\boldsymbol{h}}_I \leftarrow \text{invert\_Peano\_transform}(\hat{\boldsymbol{h}})$

Algorithm 2: Unsupervised image segmentation with PMC models based on Deep Neural Networks architectures.

3.2. Simulations

3.2.1. Non-linear and correlated noise

In this section we propose to blur our images with a noise which exhibits non-linearities to highlight the ability of the

generalized models to learn such a signal corruption. The parameters introduced in the following examples are considered unknown to meet the case of unsupervised segmentation. In the following examples, our neural networks consist of one unfrozen hidden layer with 100 neurons and one frozen hidden layer whose size coincides with that of the input of the neural network according to the constraint discussed in section 3.1.2. Two scenarios are considered.

Scenario 1 *The hidden image* $\boldsymbol{h}$ *is the* dude*-type image of the Binary Shape Database. Each observation is simulated as*

$$X_k \sim \mathcal{N}\Big(a_{h_{k-1},h_k}\cos(x_{k-1}); (\sigma + |\cos(x_{k-1})|)^2\Big). \quad (18)$$

σ will be a varying parameter and we set $a_{\omega_1,\omega_1} = 0, a_{\omega_1,\omega_2} = 0.3$, $a_{\omega_2,\omega_1} = 0.9, a_{\omega_2,\omega_2} = 0.7$.

Scenario 2 *The hidden image* $\boldsymbol{h}$ *is the* horse*-type images from the database. Each observation is simulated as*

$$X_k \sim b_{h_k} + 0.5\cos x_{k-1} + R \cdot U, \quad (19)$$

where R is a Rademacher r.v. and U is an uniformly distributed r.v. on $[0, 1]$. We set $b_{\omega_1} = 0$ and $b_{\omega_2} = \beta$, where β will be a varying parameter.

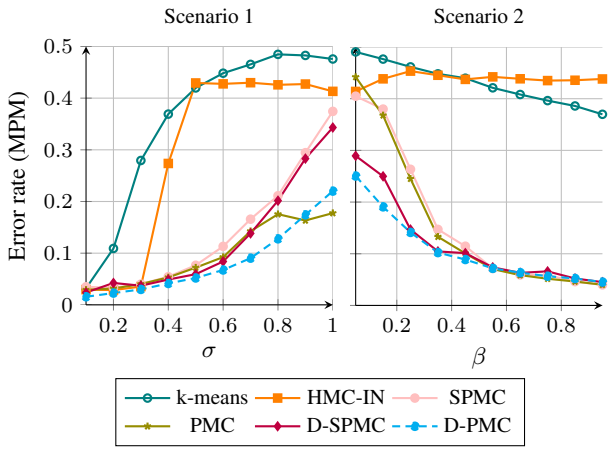


Fig. 3: Error rate in the unsupervised segmentations of Section 3. Results are averaged on all the *dude* (Scenario 1) or *horse*-type (Scenario 2) images from the database.

Fig. 3 illustrates the error rate associated to our unsupervised segmentations with varying noise level. It is clear that the D-PMC and D-SPMC models are the best performing models for almost all noise levels, offering up to a 10%-point improvement over the non-deep PMCs. The gain obtained with our D-PMC and D-SPMC models is available without any further modeling effort. Indeed, the PMC models based on deep architectures seem to be able to capture the non-linear correlated noises. Moreover, the training strategies proposed

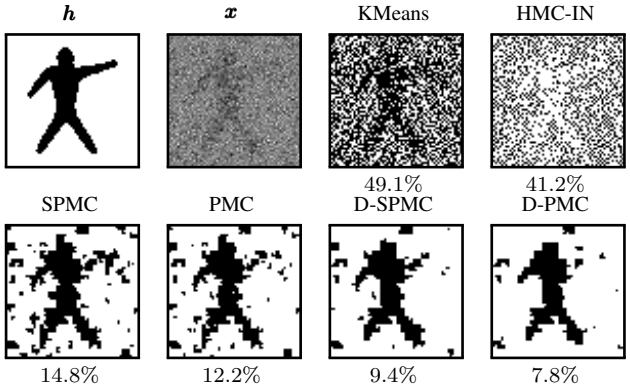


Fig. 4: Illustration of an unsupervised segmentation of an image from the Example 1 described in Sec. 3.2.1, for $\sigma = 0.7$. The error rates appear below the images.

in Section 3.1 offer stability and consistent results for the training of the generalized models during all the experiments. A graphical example of unsupervised segmentations with the different models considered in this paper is given in Fig. 4.

Finally, other configurations with more hidden layers and/or more neurons on each layer only slightly improved the error rate; it means that for this kind of noise, the proposed configuration is sufficient.

3.2.2. Real Data

Finally, our models were tested on a real dataset taken from fluorescence microscopy images [19]. The annotated dataset is available online. Again, the DNNs using in the D-PMCs are set to one unfrozen hidden layer with 100 neurons. These images are particularly challenging because of the fuzzy contours of the objects in the fluorescence images which leads to a complex unsupervised segmentation task. The results of the experiments are given in Fig. 5. In this experiment, the HMC and non deep PMC models give similar results and were outperformed by the our new generalized PMCs. Note that the D-PMC model seems less suited for this kind of noise than the D-SPMC model. A possible reason is that the fact that X_k does not depend on H_{k-1} given (h_k, x_{k-1}), contrary to the D-PMC, produces an unsupervised segmentation more robust in the sense that H_k is more interpretable as the class associated to X_k in the D-SPMC.

4. CONCLUSION

In this paper, we have proposed a general parametrization of PMC models. From this general framework, we have deduced PMC architectures based on DNN and we have proposed a parameter estimation algorithm to train these models for unsupervised image segmentation. Our experiments have indeed shown that substantial gains can be attained over the classical models.

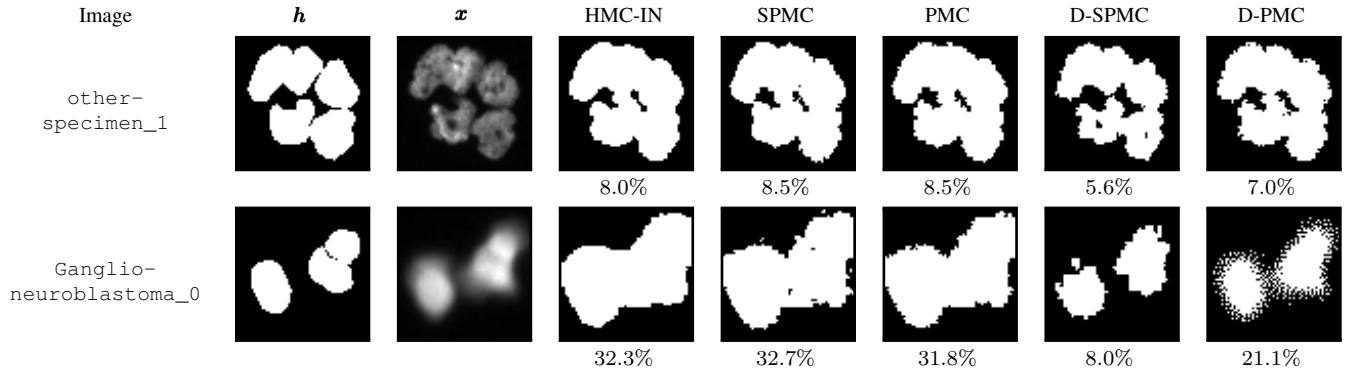


Fig. 5: Illustration of unsupervised segmentations of real-world data from [19]. The error rates appear below the images.

5. REFERENCES

[1] P. Dymarski, *Hidden Markov Models: Theory and Applications*, IntechOpen, 2011.

[2] W. Pieczynski, "Pairwise Markov chains," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 25, no. 5, pp. 634–639, 2003.

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

[4] J.-B. Courbot, V. Mazet, E. Monfrini, and C. Collet, "Pairwise Markov fields for segmentation in astronomical hyperspectral images," *Signal Processing*, vol. 163, pp. 41–48, 2019.

[5] H. Gangloff, J.-B. Courbot, E. Monfrini, and C. Collet, "Unsupervised image segmentation with gaussian pairwise Markov fields," *Computational Statistics & Data Analysis*, vol. 158, pp. 107178, 2021.

[6] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[8] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[9] M. J. Johnson, D. Duvenaud, A. B. Wiltschko, S. R. Datta, and R. P. Adams, "Composing graphical models with neural networks for structured representations and fast inference," *arXiv preprint arXiv:1603.06277*, 2016.

[10] K. Tran, Y. Bisk, A. Vaswani, D. Marcu, and Kevin Knight, "Unsupervised neural Hidden Markov Models," *arXiv preprint arXiv:1609.09007*, 2016.

[11] S. Derrode and W. Pieczynski, "Signal and image segmentation using pairwise Markov chains," *IEEE Transactions on Signal Processing*, vol. 52, no. 9, pp. 2477–89, 2004.

[12] J. Bayer and C. Osendorfer, "Learning stochastic recurrent networks," *arXiv preprint arXiv:1411.7610*, 2014.

[13] L. R. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[14] J. Marroquin, S. Mitter, and T. Poggio, "Probabilistic solution of ill-posed problems in computational vision," *Journal of the American statistical association*, vol. 82, no. 397, pp. 76–89, 1987.

[15] 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)*, vol. 39, no. 1, pp. 1–22, 1977.

[16] S. Balakrishnan, M. J. Wainwright, B. Yu, et al., "Statistical guarantees for the EM algorithm: from population to sample-based analysis," *Annals of Statistics*, vol. 45, no. 1, pp. 77–120, 2017.

[17] H. Sagan, *Space-filling curves*, Springer, 2012.

[18] K. Gurney, *An Introduction to Neural Networks*, Taylor & Francis, Inc., USA, 1997.

[19] F. Kromp, E. Bozsaky, F. Rifatbegovic, L. Fischer, M. Ambros, M. Berneder, T. Weiss, D. Lazic, W. Dörr, A. Hanbury, et al., "An annotated fluorescence image dataset for training nuclear segmentation methods," *Scientific Data*, vol. 7, no. 1, pp. 1–8, 2020.