Triplet Markov Fields for the Classification of Complex Structure Data

Juliette Blanchet and Florence Forbes

Abstract—We address the issue of classifying complex data. We focus on three main sources of complexity, namely, the high dimensionality of the observed data, the dependencies between these observations, and the general nature of the noise model underlying their distribution. We investigate the recent *Triplet Markov Fields* and propose new models in this class designed for such data and in particular allowing very general noise models. In addition, our models can handle the inclusion of a learning step in a consistent way so that they can be used in a supervised framework. One advantage of our models is that whatever the initial complexity of the noise model, parameter estimation can be carried out using state-of-the-art Bayesian clustering techniques under the usual simplifying assumptions. As generative models, they can be seen as an alternative, in the supervised case, to discriminative Conditional Random Fields. Identifiability issues underlying the models in the nonsupervised case are discussed while the models performance is illustrated on simulated and real data, exhibiting the mentioned various sources of complexity.

Index Terms—Triplet Markov model, supervised classification, conditional independence, complex noise models, high-dimensional data, EM-like algorithms.

1 Introduction

Statistical methods that were once restricted to specialist statisticians such as multivariate discrimination and classification are now widely used by individual scientists, engineers, and social scientists, aided by statistical packages. However, these techniques are still restricted by necessary simplifying assumptions such as precise measurement and independence between observations, and it long ago became clear that in many areas, such assumptions can be both influential and misleading. There are several generic sources of complexity in data that require methods beyond the commonly understood tools in mainstream statistical packages. In this paper, we consider more specifically classification problems in which observations have to be grouped into a finite number of classes. We propose a unified Markovian framework for classifying unlabeled observed data into these classes. We focus on three sources of complexity. We consider data exhibiting (complex) dependence structures, having to do for example with spatial or temporal association, family relationship, and so on. Markov models or more generally hidden Markov models are used to handle dependencies. Observations are associated to sites or items at various locations. These locations can be irregularly spaced. This goes beyond the regular lattice case traditionally used in image analysis and requires some adaptation. A second source of complexity is connected with the measurement process such as having multiple measuring instruments or computations generating high-dimensional data. There are not so many one-dimensional (1D) distributions for continuous variables that generalize to multidimensional ones except when considering the product of 1D independent components. The multivariate Gaussian distribution is most commonly used, but it suffers from significant limitations when it comes to modeling real data sets. For very high-dimensional data, the general covariance matrix model involves the estimation of too many parameters, leading to intractable computations or singularity issues. Solutions have been proposed based on so-called *parsimonious models* [1], but they are not specifically designed for high-dimensional data. They do not take into account the fact that real data points are often confined to a region of the space having lower effective dimensionality so that the data actually live on a smaller dimensional manifold embedded within the high-dimensional space. Other approaches consider reducing the dimensionality of the data as a preprocessing step possibly using Principal Component Analysis or variable selection methods. In a classification context, this may not be satisfactory as relevant information may be lost that can help separating the classes. For these reasons, we rather consider a more recent approach developed for independent Gaussian mixtures [2]. We extend this approach to our Markov models and maintain this way their efficiency and tractability for high-dimensional data. Both dependencies between sites and dependencies between components of the multidimensional observations are modeled, while the number of parameters to be estimated remains tractable. Another limitation of Gaussian distribution is that a single Gaussian distribution is unable to capture nonunimodal structures. Therefore, the main contribution of the paper is to address a third major source of complexity related to the structure of the *noise* model or the distribution linking the unknown labels to the observations. The models we propose are able to deal with very general noise models far beyond the modeling capabilities of traditional hidden Markov models. In the hidden Markov field (HMF) framework, a strong assumption of conditional independence of the observed data is generally

• *The authors are with the MISTIS team, INRIA Rhône-Alpes, ZIRST, 655 av. de l'Europe, Montbonnot, 38334 Saint-Ismier Cedex, France. E-mail: {Juliette.Blanchet, Florence.Forbes}@inrialpes.fr.*

Manuscript received 2 Aug. 2007; revised 20 Nov. 2007; accepted 17 Dec. 2007; published online 22 Jan. 2008.
Recommended for acceptance by S-C. Zhu.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2007-08-0473.
Digital Object Identifier no. 10.1109/TPAMI.2008.27.

used for tractability. This assumption combined with the Markovianity of the hidden field has the advantage to lead to a distribution of the labels given the observations (the *posterior distribution*), which is Markovian. This last property is essential in all Markov model-based clustering algorithms. However, conditional independence is too restrictive for a large number of applications such as textured or nonstationary image segmentation. For this reason, various Markov models have been proposed in the literature, including Gaussian Markov fields [3] and, more recently, *Pairwise Markov models* [4]. The latter are based on the observation that the conditional independence assumption is sufficient but not necessary for the Markovianity of the conditional distribution to hold. A further generalization has then been proposed in [5] through the *Triplets Markov models* with larger modeling capabilities. In practice, the Triplet models illustrated in applications (see [5] and [6]) satisfy particular assumptions. In this paper, we consider Triplet models different from those in [5] and [6]. Our more general noise models allow class and site dependent mixtures of distributions and, more specifically, mixtures of Gaussians, which provide a richer class of density models than the single Gaussians. In addition, our models were originally designed for supervised classification issues in which training sets are available and correspond to data for which data exemplars have been grouped into classes. Nontrivial extensions are required to include a learning step while preserving the Markovian modeling of the dependencies. In this context, we propose a class of Triplet Markov models that have the advantage to account for very general noise models while still allowing standard processing, as regards classification and parameter estimation. We illustrate our models using an Expectation Maximization framework and a mean-field-like approximation procedure developed in [7] for the standard HMF case. Any other choice (Iterative Conditional Estimation, Stochastic gradient, etc.) would have been possible, but it is not the purpose of this paper to provide a comparison of all these techniques. We adapt the approach in [7] to the use of our Triplet models, including a learning and test stages in the supervised case. We consider the issue of selecting the best model with regards to the observed data using a criterion based on the Bayesian Information Criterion (BIC). Although more general, it is important to note that our Triplet models are simpler to deal with and have greater modeling capabilities in a supervised case. In the nonsupervised case, general noise models can lead to nonidentifiability issues. In addition, it is important to specify the relationship between the Triplet Models and the Conditional Random Fields (CRF) [8], which have been widely and successfully used in applications, including text processing, bioinformatics, and computer vision. CRF's are *discriminative models* in the sense that they model directly the posterior or conditional distribution of the labels given the observations, which is the one needed in classification issues. Explicit models of the joint distribution of the labels and observations or of the noise distribution are not required. However, even in classification contexts, approaches that model the joint distribution of the labels and observations are considered. They are known as *generative models*. Triplet Markov models belong to this class. Such generative models are certainly more demanding in terms of modeling, but they have the advantage to provide a model of the observed data (the likelihood), allowing this way better access to theoretical properties of the estimators and to approaches for outliers detection. In this work, we show that Triplet Markov models can then be seen as an alternative to CRFs with good modeling capabilities. As generative models, they better model the structure of the data. They can be used with standard Bayesian techniques and probabilistic clustering tools requiring no more algorithmic effort than CRFs. They allow theoretically well-based studies and, in particular, model selection to guide the user to specific modeling choices consistent with the observed data.

To outline the structure of the paper, the hidden Markov model approach is recalled in Section 2, which presents basic tools and points out some limitations when dealing with complex data. Section 3 introduces our *Triplet Markov Field (TMF)* model in the context of supervised segmentation. A general scheme and procedure based on EM-like algorithms for parameter estimation is proposed in Section 4. The automatic selection of Triplet Markov models is addressed in Section 5. As an illustration, the simulations of a simple Triplet model are shown in Section 6. In the same section, experiments on synthetic data are made to compare the performance of our TMF model with the standard HMF approach. In Section 7, we consider a texture recognition task that involves real complex data. Section 8 ends the paper with elements for discussion and further work.

2 Hidden Markov Model-Based Clustering

Hidden structure models and, more specifically, Gaussian mixture models are among the most statistically mature methods for clustering. A clustering or labeling problem is specified in terms of a set of sites S and a set of labels $\mathcal{L}$. A site often represents an item, a point or a region in the euclidean space such as an image pixel or an image feature. A set of sites may be categorized in terms of their regularity. Sites on a lattice are considered as spatially regular (for example, the pixels of a 2D image). Sites that do not present spatial regularity are considered as irregular. This is the usual case when sites represent geographic locations or features extracted from images at a more abstract level such as *interest points* (see Section 7). It can also be that the sites correspond to items (for example, genes) that are related to each other through a distance or dissimilarity measure [9] or simply to a collection of independent items. A label is an event that may happen to a site. We will consider only the case where a label assumes a discrete value in a set of L labels. In the following developments, it is convenient to consider $\mathcal{L}$ as the set of L-dimensional indicator vectors $\mathcal{L} = \{e_1, \ldots, e_L\}$ where each e_l has all its components being 0 except the lth, which is 1. The labeling problem is to assign a label from a label set $\mathcal{L}$ to each of the sites. If there are n sites, the set $\mathbf{y} = \{y_1, \ldots, y_n\}$ with $y_i \in \mathcal{L}$ for all $i \in S$ is called a labeling of the sites in S in terms of the labels in $\mathcal{L}$. Our approach of the labeling problem aims at modeling dependencies or taking into account contextual information. It is based on hidden Markov models. We consider cases where the data naturally divide into observed data $\mathbf{x} = \{x_1, \ldots, x_n\}$ and unobserved or missing membership data $\mathbf{y} = \{y_1, \ldots, y_n\}$, both considered as realizations of random variables denoted, respectively, by $\mathbf{X} = \{X_1, \ldots, X_n\}$ and $\mathbf{Y} = \{Y_1, \ldots, Y_n\}$. Each X_i takes its values in $\mathbb{R}^d$, and each Y_i takes its values in $\mathcal{L}$. The goal is to estimate $\mathbf{Y}$ from the observed $\mathbf{X} = \mathbf{x}$. When the Y_i's are independent, the model reduce to a standard mixture model. When the Y_is

are not independent, the interrelationship between sites can be maintained by a so-called neighborhood system usually defined through a graph. Two neighboring sites correspond to two nodes of the graph linked by an edge. The dependencies between the Y_is are then modeled by further assuming that the joint distribution of $Y_1, \ldots, Y_n$ is a discrete Markov Random Field (MRF) on this specific graph defined by

$$P(\mathbf{y}) = W^{-1} \exp(-H(\mathbf{y})), \tag{1}$$

where W is a normalizing constant, and H is a function assumed to be of the following form (we restrict to pairwise interactions), $H(\mathbf{y}) = \sum_{i \sim j} V_{ij}(y_i, y_j)$, where the V_{ij}s are functions referred to as pair potentials. We write $i \sim j$ when sites i and j are neighbors on the graph, so that the sum above is only over neighboring sites. We consider pair potentials $V_{ij}(y_i, y_j)$ that depend on y_i and y_j but also possibly on i and j. Since the y_is can only take a finite number of values, for each i and j, we can define a $L \times L$ matrix $\mathbb{W}_{ij} = (\mathbb{W}_{ij}(k,l))_{1 \leq k,l \leq L}$ and write, without loss of generality, $V_{ij}(y_i, y_j) = -\mathbb{W}_{ij}(k,l)$ if $y_i = e_k$ and $y_j = e_l$. Using the indicator vector notation and denoting y_i^t, the transpose of vector y_i, it is equivalent to write $V_{ij}(y_i, y_j) = -y_i^t \mathbb{W}_{ij} y_j$. This latter notation has the advantage to still make sense when the vectors are arbitrary and not necessarily indicators. This will be useful when describing the algorithms of the Appendix. If for all i and j, $\mathbb{W}_{ij} = \beta \times I_L$, where β is a scalar, and I_L is the $L \times L$ identity matrix, the model parameters reduce to a single scalar interaction parameter β, and we get the Potts model traditionally used for image segmentation [10]. Note that this model is most of the time appropriate for classification since, for positive β, it tends to favor neighbors that are in the same class. In practice, these parameters can be tuned according to expert or a priori knowledge, or they can be estimated from the data. In the latter case, the part to be estimated is usually assumed independent of the indices i and j, so that in what follows the Markov model parameters will reduce to a single matrix $\mathbb{W}$. Note that formulated as such, the model is not identifiable in the sense that different values of the parameters, namely, $\mathbb{W}$ and $\mathbb{W} + \alpha \mathbb{1}$ (where $\mathbb{1}$ denotes the $L \times L$ matrix with all its components being 1) lead to the same probability distribution. This issue is generally easily handled by imposing some additional constraint such as $\mathbb{W}(k,l) = 0$ for one of the components (k,l).

When $\mathbf{Y}$ is assumed to be Markovian (1) with respect to a neighboring system, $(\mathbf{X}, \mathbf{Z})$ is said to be a HMF. HMF models have been widely used for a number of classification tasks. Most applications are related to image analysis, but other examples include population genetics, bioinformatics, etc. Note that the Markovianity of $\mathbf{Y}$ is not strictly necessary (see, for instance, [5]). In a segmentation or classification context, it has the advantage to provide some insight and control on the segmentation regularity through a meaningful and easy to understand parametric model, but it also somewhat reduces the modeling capabilities of the approach. In the following developments, we will consider more general cases. In labeling problems, most approaches then fall into two categories. The first ones focus on finding the best $\mathbf{y}$ using a Bayesian decision principle such as Maximum A Posteriori (MAP) or Maximum Posterior Mode (MPM) rules. This explicitly involves the use of $P(\mathbf{y}|\mathbf{x})$ and uses the fact that the conditional field denoted by $\mathbf{Y}|\mathbf{X} = \mathbf{x}$ is a Markov field. This includes methods such as ICM [10] and Simulated Annealing [11], which differ in the way they deal with the intractable $P(\mathbf{y}|\mathbf{x})$ and use its Markovianity. The second type of approach is related to a missing data point of view. Originally, the focus is on estimating parameters when some of the data are missing (the y_is here). The reference algorithm in such cases is the Expectation-Maximization (EM) algorithm [12]. In addition to providing estimates of the parameters, the EM algorithm provides also a classification $\mathbf{y}$ by offering the possibility to restore the missing data. However, when applied to HMFs, the algorithm is not tractable and requires approximations. This approach includes the Gibbsian EM in [13], the MCEM algorithm and a generalization of it [14], the PPL-EM algorithm in [14], and various Mean-Field-like approximations of EM [7]. Such approximations are also all based on the Markovianity of $\mathbf{Y}|\mathbf{X} = \mathbf{x}$. This property appears as a critical requirement for any further developments. When $\mathbf{Y}$ is Markovian, a simple way to guarantee the Markovianity of $\mathbf{Y}|\mathbf{X} = \mathbf{x}$ is to further assume that

$$P(\mathbf{x}|\mathbf{y}) = \prod_{i \in S} P(x_i|y_i). \tag{2}$$

Indeed, (1) and (2) imply that $(\mathbf{X}, \mathbf{Y})$ is an MRF, which implies that $\mathbf{Y}|\mathbf{X} = \mathbf{x}$ is an MRF too. This standard and widely used case is referred to, in [5], as the HMF-IN model for HMF with Independent Noise. Equation (2) is a conditional independence or noncorrelated noise condition. In addition, in such a setting, the class dependent distribution $P(.|y_i)$ is usually a standard distribution, typically a Gaussian distribution $\mathcal{N}(.|\theta_{y_i})$, where the y_i subscript in θ_{y_i} indicates that the distribution parameters depends on the specific value of y_i. More generally, HMF-IN parameters are denoted by $\mathbf{\Psi} = (\Theta, \mathbb{W})$ with $\Theta = \{\theta_1, \ldots, \theta_L\}$. In the 1D Gaussian case, $\theta_{y_i} = (\mu_{y_i}, \sigma^2_{y_i})$, the mean and variance parameters. This corresponds to $P(\mathbf{x}|\mathbf{y})$ proportional to

$$\exp\left(-\frac{1}{2} \sum_{i \in S} \sigma_{y_i}^{-2} (x_i - \mu_{y_i})^2 \right). \tag{3}$$

However, the Gaussian assumption is not satisfactory whenever the goal is to partition data into nonhomogeneous class for which the distribution of individuals in the class is very unlikely to be Gaussian and more generally unimodal. As an example, the last two assumptions (2) and (3) are too simple to allow one to take into account such class distributions, for which it may be critical to capture spatial noise correlations. In particular, for texture modeling, an alternative hypothesis is that textures are realizations of a Gaussian MRF [3]. For illustration, in the 1D case, $P(\mathbf{x}|\mathbf{y})$ is proportional to

$$\exp\left(-\sum_{i \sim j} \alpha_{y_i y_j} x_i x_j - \sum_{i \in S} \left(\alpha_{y_i y_i} x_i^2 + \gamma_{y_i} x_i \right) \right). \tag{4}$$

Note the additional double terms $\alpha_{y_i y_j} x_i x_j$ when comparing to (3). If the cardinality of S is n, the later corresponds to a multidimensional Gaussian distribution with an n-dimensional diagonal covariance matrix, while (4) corresponds to a more general covariance matrix. When defined by (4), the X_is are not conditionally independent given $\mathbf{Y}$,

but the trouble with (4) is that except in particular cases, neither $\mathbf{Y}|\mathbf{X}=\mathbf{x}$ nor $(\mathbf{X},\mathbf{Y})$ is Markovian. Note that if $(\mathbf{X},\mathbf{Y})$ is an MRF, then $\mathbf{Y}|\mathbf{X}=\mathbf{x}$ is an MRF too, but the reverse is not necessarily true. Different strategies can then arise. It appears that a lot of theoretical and computational tools have been developed for a Bayesian treatment (MAP or MPM) so that there are significant advantages both theoretically and practically in adapting new models to this framework. The TMF in [5] were designed for this purpose. In what follows, we build new TMF models that are appropriate for Bayesian supervised segmentation of complex data. They can be seen as particular TMF. The generally used strong assumption of conditional independence of the observed data is relaxed. We consider a generative framework, but some aspects of this work are similar to the discriminative CRFs approach that models the conditional distribution $P(\mathbf{y}|\mathbf{x})$ directly [8]. Such CRFs are related to the Pairwise Markov random fields (PMF) in [4]. PMFs consist in modeling the joint distribution $P(\mathbf{x},\mathbf{y})$ as an MRF, which implies that $P(\mathbf{y}|\mathbf{x})$ is an MRF too without modeling explicitly the likelihood $P(\mathbf{x}|\mathbf{y})$ or assuming that $P(\mathbf{y})$ is Markovian. The TMF approach is based on the introduction of a third field $\mathbf{Z}$ so that $(\mathbf{X},\mathbf{Y},\mathbf{Z})$ is Markovian and, therefore, $P(\mathbf{z},\mathbf{y}|\mathbf{x})$ is an MRF as a consequence, while $P(\mathbf{y}|\mathbf{x})$ is not necessarily one, generalizing this way the CRF approach. More details are given in Section 3. We then show in Section 4 how we can use algorithms developed for HMF-IN for inference in these more general models.

3 Designing Triplet Markov Fields for Supervised Segmentation of Complex Data

3.1 Supervised Segmentation

We first focus on data that exhibit some complexity due to the general nature of the noise that does not necessarily satisfy usual standard assumptions such as being Gaussian and noncorrelated. Doing so, we propose models that can handle nonunimodal or non-Gaussian class dependent distributions and this also for high-dimensional data, as specified in Section 3.2.

When starting from the standard HMF-IN models (Section 2) to cluster data, a natural idea to extend the modeling capabilities of this approach is to decompose each class, given by the y_is, into subclasses, allowing more general class dependent distributions. However, introducing such subclasses in a mathematically consistent way is not straightforward. Let us first assume that each of the L classes is decomposed into K subclasses so that we can introduce additional variables $\{Z_1,\ldots,Z_n\}$, indicating the subclasses and then consider class and subclass dependent distributions $P(.|y_i,z_i)$ that depend on some parameters $\theta_{y_iz_i}$. The $\theta_{y_iz_i}$s belong to a set $\Theta=\{\theta_{lk},l=1\ldots L,k=1\ldots K\}$. In a supervised framework, we assume that learning data sets are available and can be used to first estimate the component parameters $\Theta=\{\theta_{lk},l=1\ldots L,k=1\ldots K\}$. However, how to include a learning step when dealing with Markovian dependent data is not straightforward. In a learning stage, typically, observations $\mathbf{x}$ are given together with their ground truth $\mathbf{y}$. This means that both $\mathbf{x}$ and $\mathbf{y}$ are known and that, when considering maximum likelihood criterion, the model parameters $\mathbf{\Psi}=(\Theta,\mathbb{W})$ has to be estimated by maximizing the joint distribution, $P(\mathbf{x},\mathbf{y}|\mathbf{\Psi})$ over $\mathbf{\Psi}$. It is easy to see that estimating parameters Θ is done independently of the assumptions made on $P(\mathbf{y})$. In particular, if (2) holds, whatever the condition on $P(\mathbf{y})$ (Markovianity, etc.), the parameters will be estimated as if the sites i were independent. Relaxing assumption (2) is therefore essential when considering a supervised framework. For textures, this has been used in [15], but the approach based on independent Gaussian mixtures and used for texture recognition cannot be extended in a consistent way. If the class dependent distributions are assumed to be standard mixture of Gaussians, the model used in learning each class is an independent mixture model and does not account for dependencies between the sites. We could deal with learning data as independent mixtures, but this will mean dealing with two different models, one for the images in the learning set and one for the images in the test set. As an alternative, in the following, we consider less straightforward but consistent extensions of HMF-IN. We propose to define the distribution $P(\mathbf{x}|\mathbf{y})$ in a more general way. Equation (3) defines the distribution of $\mathbf{X}|\mathbf{Y}=\mathbf{y}$ as a n-dimensional Gaussian distribution with a diagonal covariance matrix due to the conditional independence assumption. We generalize (3) by introducing an additional field $\mathbf{Z}=(Z_i)_{i\in S}$ with $Z_i\in\mathcal{K}=\{e'_1,\ldots,e'_K\}$, where the e'_k are K-dimensional indicator vectors. For all $\mathbf{y}\in\mathcal{L}^n$, we can write

$$P(\mathbf{x}|\mathbf{y})=\sum_{\mathbf{z}\in\mathcal{K}^n}P(\mathbf{z}|\mathbf{y})P(\mathbf{x}|\mathbf{y},\mathbf{z})=\sum_{\mathbf{z}\in\mathcal{K}^n}\Pi_{\mathbf{yz}}\ f_{\boldsymbol{\theta}_{\mathbf{yz}}}(\mathbf{x}).$$

The distribution of $\mathbf{X}|\mathbf{Y}=\mathbf{y}$ can be seen as a mixture of K^n distributions where the mixing proportions, denoted by $\Pi_{\mathbf{yz}}$, are the $P(\mathbf{z}|\mathbf{y})$s, and the mixed distributions are denoted by $f_{\boldsymbol{\theta}_{\mathbf{yz}}}(\mathbf{x})=P(\mathbf{x}|\mathbf{y},\mathbf{z})$. More specifically, we will consider Gaussian $P(\mathbf{x}|\mathbf{y},\mathbf{z})$ with independence between the components, that is,

$$f_{\boldsymbol{\theta}_{\mathbf{yz}}}(\mathbf{x})=\prod_{i\in S}f_{\theta_{y_iz_i}}(x_i)=\prod_{i\in S}P(x_i|y_i,z_i),\tag{5}$$

where $\{f_{\theta_{lk}},l\in\{1,\ldots,L\},k\in\{1,\ldots,K\}\}$ are d-dimensional Gaussian distributions with parameters $\theta_{lk}=(\mu_{lk},\Sigma_{lk})$. In particular, it follows that for all $i\in S$, $P(x_i|y_i)=\sum_{z_i\in\mathcal{K}}P(z_i|y_i)f_{\theta_{y_iz_i}}(x_i)$, which is a mixture of K Gaussians, depending on y_i and whose mixture coefficients $P(z_i|y_i)$ also depend on the site i. Equation (8) below shows that this latter dependence is one of the key and main differences with standard independent mixtures of Gaussians in which all items are independently and identically distributed. To comment further on this choice of Gaussian mixture components, it provides some flexibility in that any arbitrary probability distribution on $\mathbb{R}^d$ can be approximated by a Gaussian mixture. See Section 6 for an illustration of this flexibility in a simple image segmentation task.

As we do not assume a specific Markovian form for $\mathbf{Y}$, in order to consistently define the full model, that is, the joint distribution of $(\mathbf{X},\mathbf{Y},\mathbf{Z})$, we need to define $P(\mathbf{z},\mathbf{y})$. We choose a Markovian distribution

$$P(\mathbf{z},\mathbf{y})\propto\exp\left(\sum_{i\sim j}V_{ij}(z_i,y_i,z_j,y_j)\right),\tag{6}$$

where the $V_{ij}(z_i,y_i,z_j,y_j)$ are pair potentials. These potentials could be written in terms of a $KL\times KL$ matrix $\mathbb{W}$, as specified in Section 2, but we rather write it as

$$V_{ij}(z_i,y_i,z_j,y_j)=z_i^t\mathbb{B}_{y_iy_j}z_j+y_i^t\mathbb{C}y_j,\tag{7}$$

where $\{\mathbb{B}_{ll'}, l, l' \in \{1, \ldots, L\}\}$ are symmetric matrices of size $K \times K$ so that $\mathbb{B}_{ll'} = \mathbb{B}_{l'l}$ (there is thus $L(L+1)/2$ different matrices), and $\mathbb{C}$ is an additional symmetric matrix of size $L \times L$ that does not depend on the z_is. This is simply looking at $KL \times KL$ matrix $\mathbb{W}$ as an $L \times L$ matrix of $K \times K$ bloc matrices. Note that the term in (7) involving $\mathbb{C}$ could have been directly included in the term just before in (7). The reason for such a parameterization is made clearer below. The (k, k') term in matrix $\mathbb{B}_{ll'}$ can be interpreted as measuring the compatibility between subclass k of class l and subclass k' of class l'. The larger this term, the more likely are neighboring sites to be in such subclasses. Similarly, the (l, l') term in $\mathbb{C}$ also has to do with the compatibility between classes l and l'.

It follows from (5) and (6) that variable $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ is Markovian and consists then in a TMF, as defined in [5]. From (5) and (6), it comes clearly that, as $\mathbf{U} = (\mathbf{Y}, \mathbf{Z})$ is Markovian, the pair $(\mathbf{X}, \mathbf{U})$ is a Gaussian HMF-IN with KL hidden classes. EM-like algorithms (for example, [7]) or, more generally, any other algorithms for inference in HMF-IN can then be applied in particular to provide estimates of the Θ_{lk}s. However, defined as such, the model still suffers from some identifiability issue due to the possibility of label switching. The problem known as the *label switching* problem in a Bayesian framework [16] is due to the fact that mixtures of components belonging to the same parametric family are invariant under the permutation of the component labels. Intuitively, at the θ_{lk}s level, there is some ambiguity when trying to assign each component (subclass) to its class. In our case, the aim is to estimate $\mathbf{y}$ from the observed $\mathbf{x}$ using the posterior probability $P(\mathbf{y}|\mathbf{x})$. When considering the Triplet $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ defined above, this probability is not directly available but only through the marginalization (sum over the $\mathbf{z}$s) of $P(\mathbf{y}, \mathbf{z}|\mathbf{x})$. In practice, to compute $P(\mathbf{y}|\mathbf{x})$ then requires to sum over the right terms, that is, to know the permutation of the estimates of the $\{\theta_{lk}, l = 1 \ldots L, k = 1 \ldots K\}$. This interchanging of labels is generally handled by the imposition of an appropriate constraint on the parameters but none of the usual ones would be general enough and make sense in our context. Other proposals can be found in [17] in a clustering context. They are based on the intuition that components in the same cluster ought to be relatively close to each other, which is not true in general (for example, texture model). Possibly relabeling techniques using a likelihood or Loss function criterion, as proposed in [18], could be considered, but this would require enumerating about $(KL)!$ permutations at each iteration and would be time consuming even for not so large values of K and L. The TMF defined above are then not adapted to an unsupervised framework, at least when considering components $f_{\theta_{lk}}$s from the same parametric family, which is often the case when no additional a priori knowledge is available. In a supervised framework, this issue disappears, as soon as the $\{\theta_{lk}\}$s can be learned in a way that allows to group them according to values of k, $\{\theta_{lk}, l = 1 \ldots L\}$. The TMF above are appropriate for learning. It follows from (6) that $P(\mathbf{z}|\mathbf{y})$ is Markovian, too

$$\Pi_{\mathbf{yz}} = P(\mathbf{z}|\mathbf{y}) = \frac{1}{W(\mathbf{y})} \exp\left(\sum_{i \sim j} z_i^t \mathbb{B}_{y_i y_j} z_j \right), \tag{8}$$

where $W(\mathbf{y})$ is a normalizing constant that depends on $\mathbf{y}$. Note that matrix $\mathbb{C}$ disappears in (8). This will result in some variations between the learning and classification steps of Section 4.

Equation (5) means that the X_is are conditionally independent given the Y_is and the Z_is. In the whole model definition, it acts in a similar way as (2). The keypoint in introducing $\mathbf{Z}$ this way is that given (5) and (8), $\mathbf{X}, \mathbf{Z}|\mathbf{Y} = \mathbf{y}$ is an HMF-IN. This property will be useful in the learning stage, while the fact that the pair $(\mathbf{X}, \mathbf{U})$ with $\mathbf{U} = (\mathbf{Y}, \mathbf{Z})$ is an HMF-IN will be useful in the classification stage. More specifically, combining (5) and (8), we have

$$\begin{aligned} P(\mathbf{x}, \mathbf{z}|\mathbf{y}) &= P(\mathbf{x}|\mathbf{y}, \mathbf{z}) P(\mathbf{z}|\mathbf{y}) \\ &= \frac{1}{W(\mathbf{y})} \exp\left(\sum_{i \sim j} z_i^t \mathbb{B}_{y_i y_j} z_j + \sum_{i \in S} \log f_{\theta_{y_i z_i}}(x_i) \right), \end{aligned} \tag{9}$$

which shows that $P(\mathbf{x}, \mathbf{z}|\mathbf{y})$ does not generally factorize and results then in a different model than the TMFs that [5, p. 483] and [6] suggest to use in practical applications. The estimation procedures suggested in [5] that use the factorization of $P(\mathbf{x}, \mathbf{z}|\mathbf{y})$ cannot then be applied straightforwardly, but we will propose one in the Appendix.

As mentioned before, the triplet model is described above for $Z_i \in \mathcal{K}$, meaning implicitly that the number of subclasses is K for each of the L classes. In practice, it is important to handle the more general case of varying numbers of subclasses. This requires to specify some modifications but does not fundamentally change the procedure.

3.2 High-Dimensional Data

Using Gaussian distributions for the $f_{\theta_{lk}}$s in (5) has the advantage to admit a straightforward formulation of the model for high-dimensional data. However, estimating full covariances matrices is not always possible and advisable beyond small dimensions. A common solution is to consider diagonal covariance matrices, but this is assuming independence between the observations components and is usually not satisfying. As an alternative, we propose to use specific parameterizations described in [2]. The authors propose new Gaussian models of high-dimensional data for clustering purposes based on the idea that high-dimensional data live around subspaces with a dimension lower than the one of the original subspace. Low-dimensional class-specific subspaces are introduced in order to limit the number of parameters. The covariance matrix Σ_{lk} of each class is reparameterized in its eigenspaces. Denoting by Q_{lk} the orthogonal matrix with the eigenvectors of Σ_{lk} as columns, the class conditional covariance matrix D_{lk} is therefore defined in the eigenspace of Σ_{lk} by $D_{lk} = Q_{lk}^t \Sigma_{lk} Q_{lk}$. The matrix D_{lk} is a diagonal matrix that contains the eigenvalues of Σ_{lk}. It is further assumed that the diagonal of D_{lk} is made of d_{lk} (with $d_{lk} < d$) first values, $a_{lk}^1, \ldots, a_{lk}^{d_{lk}}$, and $d - d_{lk}$ other values all fixed to some value b_{lk} with, for all $j = 1, \ldots, d_{lk}$, $a_{lk}^j > b_{lk}$. Notation d denotes the dimension of the original space, and $d_{lk} \in \{1, \ldots, d-1\}$ is unknown, but in practice, it is much smaller that d when d is large. See [2] for additional details and further interpretation of such decompositions. In the present work, we recast this approach into the EM-based procedure described in the Appendix. When dealing with high-dimensional data, this reduces the number of parameters to be estimated significantly and tends to avoid numerical problems with singular matrices while allowing to go beyond the standard diagonal covariance matrices and the usual independence assumptions between dimensions.

4 The Supervised Classification Scheme

More than an algorithm, we describe a general scheme to deal with complex data as specified. With regards to parameter estimation, we consider a soft clustering approach and use an algorithm based on EM and *mean-field*-like approximations [7], described in the Appendix. We implemented it to illustrate the performance of the models we propose, but other algorithms could be considered. Its actual use in our supervised classification framework requires two stages that are described in Sections 4.1 and 4.2. The algorithm was originally developed for HMF-IN (Section 2), but we show below that it can be used to deal with more general models such as those in Section 3. For such models, starting from a description of a supervised clustering issue in terms of L *complex* classes corresponding to some nonstandard noise model, the learning step can be decomposed into L simpler issues, each involving an HMF-IN with K classes. The following classification step can then be reduced to an inference problem for a standard HMF-IN model with KL classes. It follows that the computational complexity of the TMF models may vary depending on the algorithm chosen for inference but is equivalent to that of usual HMF-IN models.

4.1 Learning Step

We consider a supervised framework in which part of the information is available through learning data. It is assumed that for a number of individuals, we both observe x_i and its corresponding class y_i. Using the triplet model defined in Section 3, it remains that the z_i are missing. It follows that by considering variables $\mathbf{X}$ and $\mathbf{Z}|\mathbf{Y}=\mathbf{y}$, we can apply the algorithm described in the Appendix to the HMF-IN $(\mathbf{X},\mathbf{Z}|\mathbf{Y}=\mathbf{y})$ (see (9)) to provide estimates of the model parameters, which are the $\{\mathbb{B}_{ll'}, l, l' \in \{1,\dots,L\}\}$ and the $\{\theta_{lk}, l=1,\dots L, k=1,\dots,K\}$. As mentioned in Section 3, estimating the later parameters is especially important to solve identifiability issues when dealing with our triplets Markov fields in the classification step. To estimate the θ_{lk}s, it is necessary that all L classes are sufficiently represented in the learning data. In practice, the learning data are often divided in a number of separate data sets (for example, Section 7) so that the learning procedure actually consists of a number of separate runs of the estimation algorithm. Regarding the $\mathbb{B}_{ll'}$s estimated in the learning stage, we do not necessarily need to keep them for the classification step. However, for complex data, it may be that learning also the $\mathbb{B}_{ll'}$s or at least part of them is a better choice in terms of modeling capabilities. We illustrate and explain such cases in more details in Section 7. This Section also presents a case where among the $\mathbb{B}_{ll'}$s, only the $\mathbb{B}_{ll}$ can be learned due to the specificity of the learning data. Typically, if the underlying neighborhood structure is such that there exists no neighbors in classes l and l', then $\mathbb{B}_{ll'}$ cannot be estimated since terms involving $\mathbb{B}_{ll'}$ will not appear in the model formulas. More generally, if the number of pairs in classes l and l' is too small, the estimation of $\mathbb{B}_{ll'}$ is likely not to be good, and in this case, it would be better to ignore it.

When choosing to use in the subsequent classification step, all or part of the $\mathbb{B}_{ll'}$s learned, considering separate runs for the estimation of the $\mathbb{B}_{ll'}$s may raise identifiability issues. The model (8) used in each run is identifiable only up to a constant, which may then vary from one run to another. The issue appears when grouping all estimations in a single model for the classification stage since various equivalent inferences under model (8) could lead to nonequivalent inference under model (7). However, the explicit introduction of matrix $\mathbb{C}$ in (7) and the fact that its estimation is postponed to the classification step prevents this issue. In addition, constraints in the form of $\mathbb{C}$ can be easily imposed (for example, Potts-like constraint) to make the estimated parameters unique in the classification stage.

4.2 Classification Step

At this stage, $\mathbf{Y}$ and $\mathbf{Z}$ are missing, and only the observations $\mathbf{X}$ are available. Considering $\mathbf{X}$ and $\mathbf{U}=(\mathbf{Y},\mathbf{Z})$, $(\mathbf{X},\mathbf{U})$ is an HMF-IN ((6) and (5)), and we can apply again the algorithm of the Appendix. The parameters are the $K\times K$ dimensional matrices $\{\mathbb{B}_{ll'}, l, l' \in \{1,\dots,L\}\}$ and the $\{\theta_{lk}, l=1\dots L, k=1\dots K\}$ as before with, in addition, the $L\times L$ dimensional matrix $\mathbb{C}$, that is, parameters $\{\mathbb{C}_{ll'}, l, l' \in \{1,\dots,L\}\}$.

The θ_{lk}s are considered as fixed to the values computed in the learning stage. For the $\mathbb{B}_{ll'}$s, different strategies arise depending on the available learning data and the goal in mind, in particular the type of interactions we want to account for. In practice, we propose to use specified or learned values for all $\mathbb{B}_{ll'}$s. See Section 7 for an example. In most cases then, regarding parameters, the classification step consists of estimating $\mathbb{C}$. Note that as mentioned in the previous paragraph, the matrix $\mathbb{C}$ cannot be estimated in the learning step since it disappears from (8), which is used in this latter step. The classification step differs then in that $\mathbb{C}$ plays an important role in modeling interaction. One possibility is to specify $\mathbb{C}$ to be of the Potts form, that is, to consider diagonal $\mathbb{C}$, denoted by $\mathbb{C}=[\beta_l]$ when the diagonal terms are arbitrary or $\mathbb{C}=[\beta]$ when they are all equal to some value β. More complex knowledge on the classes could be incorporated through other definitions of $\mathbb{C}$, but this simple case appears satisfying in a number of applications. Although $\mathbf{Y}$ is not Markovian, this acts as a regularizing term favoring homogeneous regions of the same class. This is an important feature of our classification step.

5 Selecting Triplet Markov Models

Choosing the probabilistic model that best accounts for the observed data is an important first step for the quality of the subsequent estimation and classification stages. In statistical problems, a commonly used selection criterion is the BIC in [19]. The BIC is computed given the data $\mathbf{x}$ and a model $\mathcal{M}$ with parameters $\mathbf{\Psi}$. It is defined by $BIC(\mathcal{M}) = 2\log P(\mathbf{x}\mid\mathbf{\Psi}^{ml}) - \delta\log n$, where $\mathbf{\Psi}^{ml}$ is the maximum likelihood estimate of $\mathbf{\Psi}$, $\mathbf{\Psi}^{ml}=\arg\max_{\mathbf{\Psi}} P(\mathbf{x}\mid\mathbf{\Psi},\mathcal{M})$, δ is the number of free parameters in model $\mathcal{M}$, and n is the number of observations. The selected model is the one with the maximum BIC. BIC allows the comparison of models with differing parameterizations. In this study, we consider the number of subclasses (cardinality of the Z_is state space, possibly varying) as fixed to focus more specifically on the Markov model and on the Gaussian models. For the Markov model, as defined in (6) and (7), model selection is performed in two steps. We first select the best models for the matrices $\mathbb{B}_{ll'}$s. This can be done in the learning stage, while finding the best model for matrix $\mathbb{C}$ can only be done in the test stage. We will in general only consider specific forms for $\mathbb{C}$ (Potts like). Regarding matrices $\mathbb{B}_{ll'}$s, omitting the subscripts, we will consider the decomposition of each

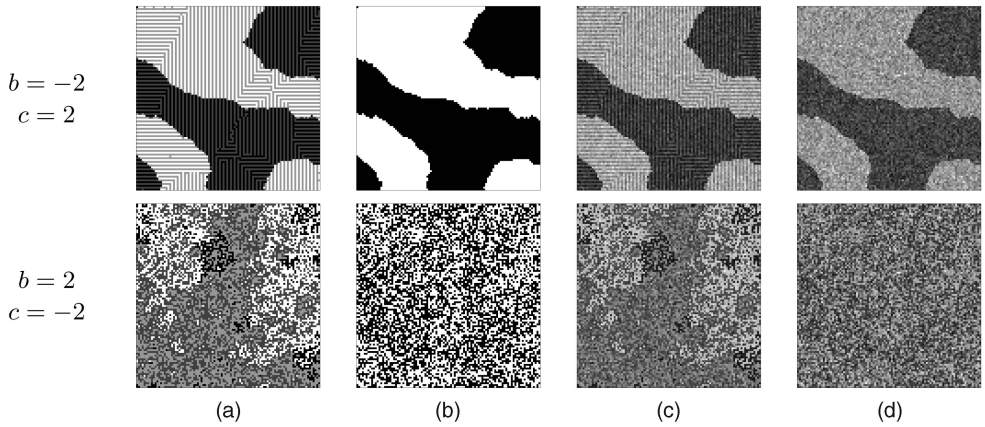

Fig. 1. Simulations of two parameter (b and c) TMF defined by (10) when $L = 2$ and $K = 2$ with, respectively, $b = -2$, $c = 2$ (first row) and $b = 2$, $c = -2$ (second row): (a) realizations of $(\mathbf{Y}, \mathbf{Z})$, (b) realizations of $\mathbf{Y}$, (c) realizations of $\mathbf{X}$, and (d) realizations of a HMF-IN built by adding to images in (b) some Gaussian noise with 0 mean and standard deviation equal to 0.3. Note that in the images, each of the four possible values of (y_i, z_i) has been associated to a gray level for visualization.

of the $K \times K$ matrix into $\mathbb{B} = \Delta + \Gamma$, where similar to Section 4, Δ is a diagonal matrix denoted by $\Delta = [\beta]$ if all diagonal terms are equal to a single value β and $\Delta = [\beta_k]$ if the diagonal terms are arbitrary. Conversely, for the second matrix Γ, all diagonal terms are 0. We then compare four possible models, namely, $\mathbb{B} = [\beta]$ (standard Potts model), $\mathbb{B} = [\beta_k]$ (generalized Potts model with class-dependent interaction parameters), $\mathbb{B} = [\beta] + \Gamma$, and $\mathbb{B} = [\beta_k] + \Gamma$ (unconstrained or full model). For multivariate Gaussian subclass specific distributions, there exists a number of different choices for the Σ_{lk}s. See [1] for a description of the examples of such forms and their meaning. The simplest models are those for which the Σ_{lk}s are diagonal matrices. We then compare this choice to the parameterizations described in Section 3.2 for high-dimensional data.

However, for HMFs and for TMFs as well, the exact computation of BIC is not tractable due to the dependence structure induced by the Markov modeling. When focusing on the Gaussian parameters, a possibility is to compute BIC for independent mixture models, forgetting any spatial information, but this would not make sense when choosing among various $\mathbb{B}$ models. We then propose to use mean-field-like approximations of BIC proposed in [20], which is based on principles similar to that presented in the Appendix. In what follows, this approximated BIC will be denoted by BIC_{MF}. Examples of model selection results are shown in Section 7. Before that, as part of our experiments on simulated data, we mention and illustrate in the next section, a problem of phase transition, which can occur for the underlying Markov field (7) and can affect parameter estimation.

6 Simulated TMF and Illustration on Synthetic Data

The goal of this section is to provide a brief study to illustrate the difference between our new TMF model and the standard HMF-IN model and to emphasize the general interest of the former model with respect to the latter.

Let us consider the Markov field $(\mathbf{Y}, \mathbf{Z})$ with pair potentials parameterized by $b, c \in \mathbb{R}$

$$V_{ij}(z_i, y_i, z_j, y_j) = b z_i^t z_j \, y_i^t y_j + c y_i^t y_j, \tag{10}$$

which is (7) with $\mathbb{B}_{ll'} = 0_L$ (the $L \times L$ zero matrix) if $l \neq l'$, $\mathbb{B}_{ll} = bI_K$ (where I_K denotes the $K \times K$ identity matrix), and $\mathbb{C} = cI_L$.

Figs. 1a and 1b show realizations of $(\mathbf{Y}, \mathbf{Z})$ and the corresponding realizations of $\mathbf{Y}$ for $K = L = 2$ and for varying values of the two parameters. Note that each of the four possible values of (y_i, z_i) is associated to a gray level. Fig. 1c shows simulated data $\mathbf{X}$ using Triplet Markov models when the Gaussian distributions in (5) are 1D with standard deviation equal to 0.3. For comparison, Fig. 1d shows the realizations of the Gaussian HMF-IN models obtained using the images in Fig. 1b and adding some Gaussian noise with 0 mean and standard deviation equal to 0.3.

We then compare the performance of our TMF model with that of the HMF-IN model on synthetic images obtained as follows: Starting from the two-class image of Fig. 2, we consider several noise models. A first noisy image is generated by considering that for the first class in Fig. 2, continuous observations are obtained by simulating a mixture of two Gaussians both with variance 0.25 and with mean, respectively, 0 and 4. For the second class, observations are obtained by simulating a mixture of two Gaussians both with variance 0.25 and with mean, respectively, 0.5 and 4.5. The corresponding noisy image is that in Fig. 2a. Similarly, a second noisy image is generated by replacing for the first class, the above mixture distribution by a Gamma distribution with a scale parameter equal to 1 and shape parameter equal to 2, while for the second class, observations are generated by simulating realizations from an Exponential distribution with parameter 1 and adding 1 to the simulated values. It follows that observations in both classes correspond to distributions with the same mean equal to 2. The corresponding noisy image is that in Fig. 2c. Two other noisy images are obtained easily from the previous two by replacing each simulated observation by the mean of this later value and that of its four nearest neighbor pixels. It follows images such as shown, respectively, in Figs. 2b and 2d.

The segmentations results using, respectively, an HMF-IN model and a TMF model are shown in the second and third rows in Fig. 2. For both models, the number of classes L is set to 2, while we use our BIC_{MF} criterion to select the value of K for the TMF model. The corresponding selected values of K are given in the last row in Fig. 2. The classification rates,

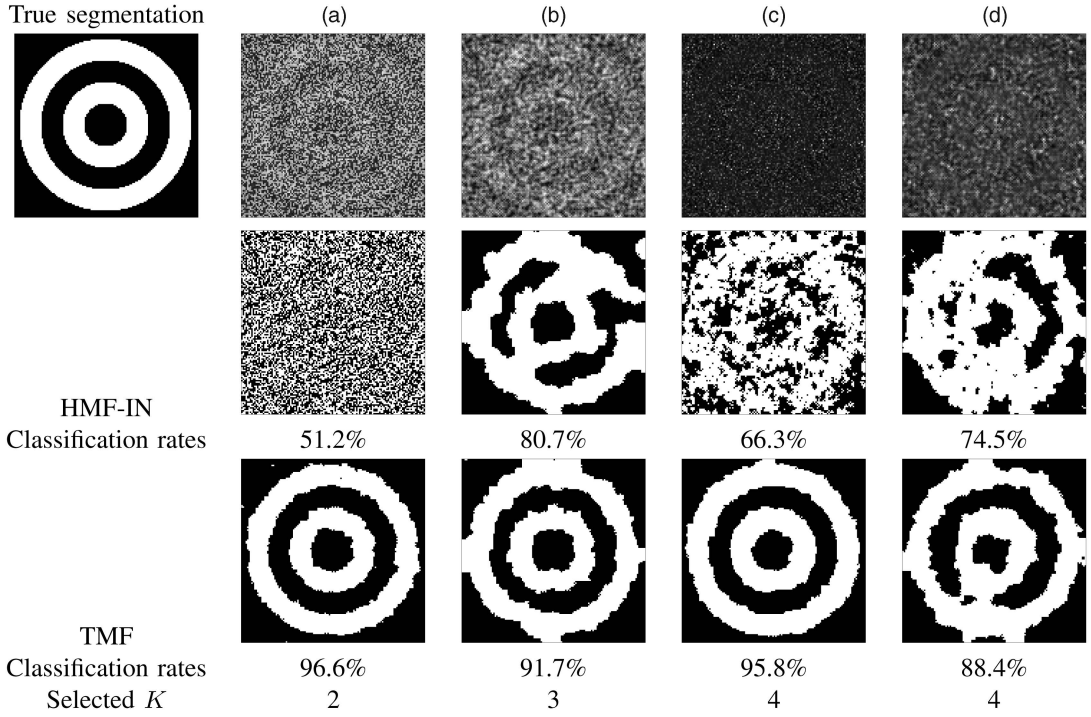

Fig. 2. Synthetic image segmentations using an HMF-IN model (second row) and our TMF model (third row): the true two-class segmentation is the image in the upper left corner, and four different noise models are considered. In (a), class distributions are mixtures of two Gaussians. In (c), observations from class 1 are generated from a Gamma(1, 2) distribution, and observations from class 2 are obtained by adding 1 to the realizations of an Exponential distribution with parameter 1. In (b) and (d), the noisy images are obtained by replacing each pixel value, respectively, in (a) and (c) by its average with its four nearest neighbors. Classification rates are given below each segmentation results. In the TMF model case, the selected K values using our BIC_{MF} criterion is given in the last row.

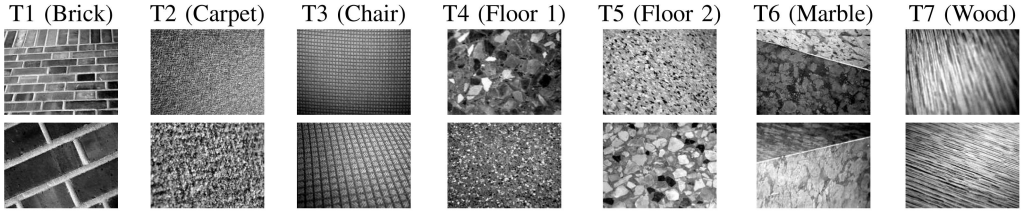

Fig. 3. Samples of the texture classes used in the experiments.

that is, the percentages of well-classified pixels are given below each segmented image.

It appears that our TMF model always gives better classification rates (more than 11 percent higher). As expected, the best rate is obtained when the noise is simulated using mixtures of Gaussians (column (a)). This case is the closest to our model assumptions. The segmentation is also satisfying when other distributions are considered showing that the TMF model is able to better deal with more general class distributions than the HMF-IN model, which assumes Gaussian class distributions. In addition, the TMF model is better when dealing with correlated noise, as shown in Figs. 2b and 2d, although it appears that in these latter cases, the HMF-IN model gives better results than in cases Figs. 2a and 2c. This is partly due to the averaging over neighboring pixels, which reduces the noise variance, as can be seen on images Figs. 2b and 2d.

7 Application to Texture Recognition

Texture analysis plays an important role in many applications. Various feature extraction methods have been proposed. Some approaches are based on local properties of the image (for example, Gaussian Markov Random Fields (GMRF) [21], local binary pattern operator (LBP) [22], and higher order local autocorrelation (HLAC) features [23]). Other use frequency representations (for example, wavelet transform [24] and Gabor features [25]). Our aim is not to discuss the performance of the numerous existing approaches besides the abovementioned. In this section, we focus on texture recognition as a good application for our model. Texture recognition identifies the texture class for an image location, whereas texture classification determines the texture class of an entire image. The issue is a supervised clustering issue involving complex data. The data set is made of 140 single texture images and 63 multiple texture images. Images have been gathered over a wide range of viewpoints and scale changes. The data set contains $L = 7$ different textures illustrated in Fig. 3. For each of the seven textures, we have 20 single texture images from which 10 are kept for the learning set. The data set is then divided into a learning set containing 70 single texture images and a test set containing 70 other single texture images and 63 multiple texture images.

As mentioned before, traditional Gaussian MRFs modeling 1D gray-level intensity images cannot easily handle such

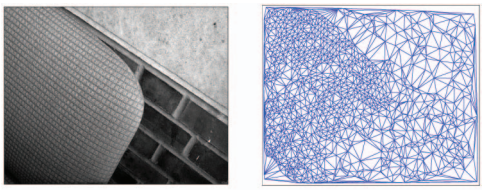

Fig. 4. (a) Multitexture image and its associated (b) Delaunay graph built from the detected interest points.

viewpoint and scale varying images. More and more high-level image analysis go beyond the traditional regular grids of pixels and 1D gray-level intensities. Our images are then rather described by local affine-invariant descriptors and their spatial relationships. A graph is associated to an image with the nodes representing feature vectors describing image regions and the edges joining spatially related regions (see [26] and the references therein for preliminary work on such data). For the feature extraction stage, we follow the texture representation method described in [15]. It is based on an interest point detector (Laplace detector) that leads to a sparse representation selecting the most perceptually salient regions in an image and on a *shape selection* process that provides affine invariance. Each detected region is then described by a feature vector (descriptor). The descriptors we use are 128-dimensional SIFT descriptors [27]. A graph is then built from the detected interest points by considering their Delaunay graph. Other choices are possible [26], but Delaunay graphs tend to provide more regular graphs, where nodes all have a reasonable number of neighbors, with the possibility to put a threshold on too long edges. An illustration is given in Fig. 4 that shows a multitexture image and the corresponding Delaunay graph.

We assume then that descriptors are random variables with a specific probability distribution in each texture class. For comparison, we consider three aspects in building various models: 1) The first one is the nature of the class dependent distributions, Gaussian or not. 2) The second one is the nature of the interactions between descriptors, which are considered as independent or not. 3) The third aspect is related to the parameterization choice for each class distribution, which can be specific to high-dimensional data or not.

More specifically, we first consider two cases in the first aspect. The simplest assumes that each class distribution is a single Gaussian, while the second case consists in introducing Gaussian subclasses for each class distribution. For simplicity, the number of subclasses to describe each class distribution is set to $K = 10$ for each texture. Selecting K using BIC is also possible, but we did not observe significantly better recognition results. Note that the Mixtures of Gaussians have been extensively used for density estimation so that this case can be viewed as an attempt to account for general class distributions (see Section 6 for an illustration).

Then, for each of these two cases, we consider in the second aspect, two alternatives depending on the use or not of the graph or, equivalently, of the interaction modeling mentioned above. To forget dependencies between descriptors corresponds to use a standard independent mixture model. Among these models, we consider therefore two families, the independent Gaussian mixture models that mix L Gaussian components and mixture models built by mixing L component distributions that are themselves mixtures of K Gaussians. The first family will be referred to as "Mixture," while the second will be referred to as "Mixture of Mixtures." As a second alternative, we consider approaches incorporating dependencies as given by the graph. More specifically, we compare our proposed TMF model with the standard HMF-IN model. This last model is a particular TMF with $\mathbf{Y} = \mathbf{Z}$ or, equivalently, $K = 1$. It can also be seen as a generalization of the independent Gaussian mixture model.

Eventually, regarding the third aspect, for the 128-dimensional class (Mixtures and HMF-IN) or subclass (Mixtures of Mixtures and TMF) dependent Gaussian distributions, we consider two possibilities: diagonal covariance matrices or specific parameterization of the covariance matrices, as described in Section 3.2. When dealing with high-dimensional data, this reduces the number of parameters to be estimated significantly and tends to avoid numerical problems with singular matrices.

To focus more on our TMF approach, as regards the Markov model, we consider that matrix $\mathbb{C}$ is fixed to a Potts form, that is, to a diagonal $[\beta]$ or $[\beta_l]$. Results are reported for the latter choice, but the first one gives similar results. For matrices $\mathbb{B}_{ll'}$, the nature of the learning data set, including only single texture images, does not allow to estimate the $\mathbb{B}_{ll'}$s for $l \neq l'$. We therefore set them to 0, which is consistent with the fact that we aim at recovering homogeneous regions of the same texture. An alternative is to postpone their estimation in the classification step, but in practice, test images do not generally include samples of all textures so that most of the $\mathbb{B}_{ll'}$ could not be estimated due to a lack of relevant information. For the $\mathbb{B}_{ll}$s, we consider the possibilities described in Section 5 and use a mean field approximation of BIC to select the best model for each texture l with $l = 1, \ldots, L$. Again, the estimation of the $\mathbb{B}_{ll}$s could be postponed to the classification step, but this would mean estimating simultaneously on each test images, L matrices of size $K \times K$ (the bloc diagonal of a $KL \times KL$ dimensional matrix). Considering the number of detected points in each image (from few hundreds to few thousands), the estimation could be reasonably carried out only for very simple models such as diagonal models and would then greatly reduce the model flexibility. As an alternative, learning each texture separately involves less parameters and more data points, allowing more complex models to be estimated accurately. Tables 1 and 2 report BIC_{MF} values for various models of $\mathbb{B}_{ll}$ in two cases corresponding to diagonal Σ_{lk}s (Table 1) and the more general Σ_{lk}s described in Section 3.2, referred to as *High Dim* Σ_{lk}s. It appears that models with *High Dim* Σ_{lk}s are always better, in terms of BIC_{MF}, whatever the $\mathbb{B}_{ll}$ model. For such Σ_{lk}s (Table 2), the selected $\mathbb{B}_{ll}$ model depends on the texture class. It appears that for the wood texture, the simplest model is selected, whereas the more general model is selected only for the Floor 2 and Marble textures.

To illustrate and compare the various models performance, Table 3 shows recognition results for individual regions that is the fraction of all individual regions in the test images that were correctly classified. These results are obtained using only the single texture images. As mentioned, results for eight models are reported. The "Mixture of Mixtures" rows, for instance, refer to the method that assumes an independent Gaussian mixture for each image in the learning and classification steps. The two possible choices for the covariance matrices are considered. The EM algorithm is used for estimation and classification. The

TABLE 1
$\mathbb{B}_{ll}$ Model Selection for Each Texture Class when Covariance Matrices Are Assumed to Be Diagonal: The Bold Numbers Indicate the Model Selected According to Our BIC_{MF} Criterion

Diagonal Σ_{lk}'s	$\mathbb{B}_{ll}$ Model	Brick	Carpet	Chair	Floor 1	Floor 2	Marble	Wood
BIC_{MF}	$[\beta]$	1739730	2403890	3141300	2556280	3147610	2967630	2092090
	$[\beta_k]$	1739840	2403970	3141440	2556410	**3147900**	2967770	2092030
	$[\beta]+\Gamma$	1740010	2404450	3141930	2556630	3145570	**2968290**	1946380
	$[\beta_k]+\Gamma$	**1740080**	**2404600**	**3142170**	**2556730**	3145600	2968280	**2092760**

TABLE 2
$\mathbb{B}_{ll}$ Model Selection for Each Texture Class when Covariance Matrices Are Parameterized to Account for High-Dimensional Data: The Bold Numbers Indicate the Model Selected According to Our BIC_{MF} Criterion

High Dim Σ_{lk}'s	$\mathbb{B}_{ll}$ Model	Brick	Carpet	Chair	Floor 1	Floor 2	Marble	Wood
BIC_{MF}	$[\beta]$	1902700	2515860	3516630	2696700	3290280	3172210	**2263160**
	$[\beta_k]$	1882040	**2524590**	**3529860**	**2697180**	3286890	3172450	2260870
	$[\beta]+\Gamma$	**1905800**	2518320	3521420	2692730	3292960	3177140	2260650
	$[\beta_k]+\Gamma$	1876510	2518430	3495890	2691310	**3293230**	**3178150**	2262300

TABLE 3
Percentage of Individual Regions Correctly Classified for the Single Texture Images of the Test Set

Dependencie Model	Covariance Model	Brick	Carpet	Chair	Floor 1	Floor 2	Marble	Wood
Mixture	Diagonal Σ_{lk}	34.08	27.63	43.70	27.41	33.80	26.27	29.78
Mixture	High Dim Σ_{lk}	42.12	35.11	52.05	29.37	46.42	28.44	31.06
HMF-IN	Diagonal Σ_{lk}	36.03	29.96	43.80	31.14	39.58	29.15	32.48
HMF-IN	High Dim Σ_{lk}	42.46	35.65	52.65	33.06	48.34	29.83	34.91
Mixture of Mixtures	Diagonal Σ_{lk}	77.58	31.60	58.26	28.26	58.79	33.87	58.56
Mixture of Mixtures	High Dim Σ_{lk}	81.18	56.94	62.48	35.64	67.43	37.05	65.02
TMF	Diagonal Σ_{lk}	96.59	80.70	83.60	82.69	83.90	46.05	95.18
TMF	High Dim Σ_{lk}	99.33	98.61	99.28	97.36	99.57	56.24	99.28
TMF-BIC		**99.37**	**98.71**	**99.30**	**98.16**	**99.62**	**56.77**	**99.52**

Rows correspond to different models. The bold numbers indicate the higher percentages.

"TMF" rows refer to our method when the more general model ($[\beta_k] + \Gamma$]) is used for all $\mathbb{B}_{ll}$s with the two possible cases for the covariance matrices. The "TMF-BIC" row then refers to the case where the form of the covariance matrices, and the $\mathbb{B}_{ll}$ models are selected according to BIC (Tables 1 and 2). As expected, the results in Table 3 show that the rates for the "Mixture" and "HMF-IN" cases are rather poor. They also show that the rates improve significantly on the independent "Mixture of Mixtures" rates (19 percent at the minimum) when our TMF model, with the *High Dim* parameterization of the Σ_{kl}s, is used. For this latter case, the rates are all very good (98 percent and above) except for the Marble texture. For this texture, the images available for learning are very heterogeneous in terms of lightning. On the same Marble image, some parts can be very badly lit and appear as very dark, while others appear as very light. This prevents a good learning mainly due to the descriptor quality that cannot properly handle such variations.

For multiple texture images, significant improvement is also observed on all images. The rates increase about 53 percent in average between the "Mixture of Mixtures" and diagonal Σ_{lk}'s case and the *TMF-BIC* case. An illustration is given in Fig. 5 with more details regarding the various possible models.

Rates for our TMF-BIC approach are all above 90 percent. It happens in very few images that using TMF with the most complex $\mathbb{B}_{ll}$ models, instead of selecting them with BIC_{MF}, gives slightly better results (from 1 or 2 percent). In the general case model, selection leads to a larger gain (larger than 6 percent).

As mentioned before (Table 3), the Marble texture suffers from lower recognition rates due to the nature of the learning data set. The high variability of the Marble images in this set makes learning a model for this texture very difficult. To illustrate the behavior of our method in this case, global recognition rates are still over 90 percent, but the errors mainly come from points in the Marble texture being misclassified. As regards the other methods, whose rates are not reported here, we observe similar results. Classification rates are greatly improved with our TMF-BIC method.

8 Discussion

We considered particular cases of TMFs by designing them to include a learning stage and to adapt to general noise models or equivalently to general class dependent distributions. Starting from a traditional hidden data model for which various estimation procedures exist, a *subclass* variable $\mathbf{Z}$ is introduced in addition to the usual observed and missing variables $\mathbf{X}$ and $\mathbf{Y}$. The supervised problem is recasted as an unsupervised problem, which allows traditional treatment. In particular, our approach allows modeling of Markovian dependencies on the sites and their effect on the noise parameter estimation. In a way similar to [5], introducing an extra $\mathbf{Z}$ allows to keep the same computational properties while increasing modeling capabilities.

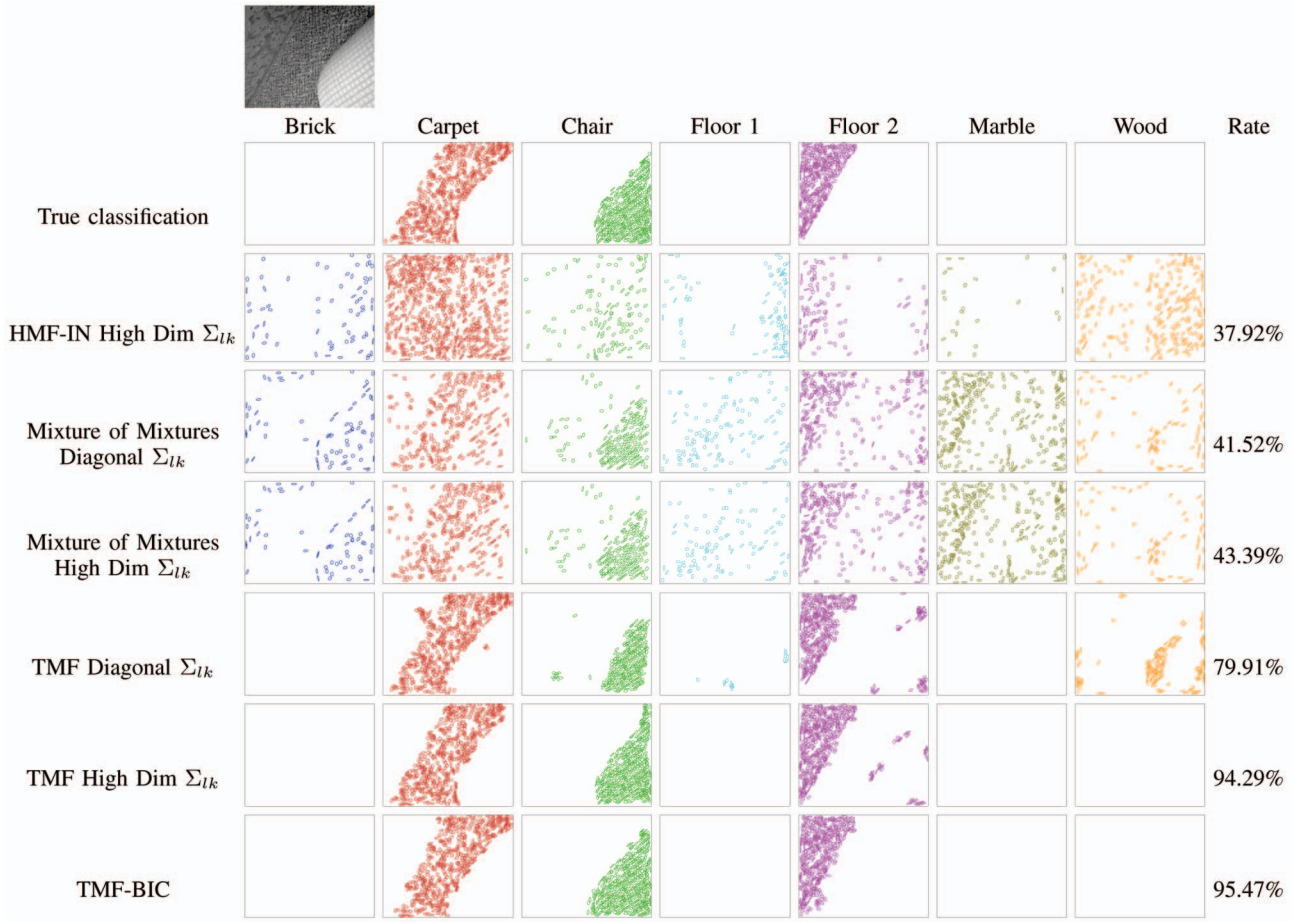

Fig. 5. Three-texture (Carpet, Chair, and Floor 2) image shown at the top-left corner: the first row shows the true classification, while each following row corresponds to a different model. Columns show the interest points classified in each of the seven texture classes. The last column reports the classification rate.

In practice, our choice of building class distributions from mixtures of Gaussians was satisfying because of the ability of such mixtures to capture adequately a large class of probability distributions. However, other choices can be developed using the same framework. More generally, it would be interesting to investigate how we could adapt our model to the use of *generalized mixture models* in which the exact nature of each mixture component is not known but can be estimated [28].

The supervised framework was dictated by the type of applications in mind (for example, texture recognition). The TMF model has shown its relevance in unsupervised frameworks too [5], [6], but our particular TMFs differ from the ones investigated in these papers in that some factorization properties do not hold. What limits our present study is the identifiability issue inherent to our model, and the way we solve it by making a full benefit of the learning data. Alternative ways to deal with the identifiability issue in order to consider our TMF models in unsupervised cases would be interesting to investigate. This includes ideas related to the *relabeling algorithm* described in [29]. Our model is not limited to regular graphs. An interesting question that was not addressed in this paper involves the choice of the neighborhood structure. This choice may depend on the application. Indeed, for irregular lattices, the points relative displacements do not follow a predictable pattern, and their linkage are not always obvious from their geometry so that a lot of possible spatial structures can be generated. With regards to Markov models, the automatic neighborhood selection has not been really addressed in the literature except very recently in [30]. In our experiments, it appears that graphs with similar numbers of neighbors for each sites give more satisfying results. Directions of research for neighborhood selection can be found in [30].

Appendix

Parameter Estimation Procedure

In this section, we describe the main features of the algorithm used for estimation in the two stages described in Sections 4.1 and 4.2. The algorithm was originally developed for standard HMFs referred to as HMF-IN (Section 2). To distinguish this particular case from the more general TMF cases considered above, we will denote by $\mathbf{O} = \{O_1, \ldots, O_n\}$ the observed variables and by $\mathbf{M} = \{M_1, \ldots, M_n\}$ the missing variables such that $(\mathbf{O}, \mathbf{M})$ is an HMF-IN, that is, $\mathbf{M}$ is a Markovian on a discrete state space with G members $\{e_1, \ldots, e_G\}$, and the conditional independence assumption (2) is satisfied (with $\mathbf{x}$ replaced by $\mathbf{o}$ and $\mathbf{y}$ replaced by $\mathbf{m}$). As mentioned earlier, the learning stage (Section 4.1) is somewhat recasted as an unsupervised case so that the estimation procedures we consider were originally developed for unsupervised segmentation. We focus on soft clustering approaches and more specifically on EM-based approaches. We consider recent

procedures combining an EM approach with *mean-field*-like approximations [7].

Briefly, these algorithms can be presented as follows: (see [7]) they are based on the EM algorithm, which is an iterative algorithm aiming at maximizing the log-likelihood (for the observed variables $\mathbf{o}$) of the model under consideration by maximizing at each iteration the expectation of the complete log-likelihood (for the observed and hidden variables $\mathbf{O}$ and $\mathbf{M}$) knowing the data and a current estimate of the model parameters. When the model is an HMF with parameters $\mathbf{\Psi}$, there are two difficulties in evaluating this expectation. Both the normalizing constant W in (1) and the conditional probabilities $P(m_i \mid \mathbf{o}, \mathbf{\Psi})$ and $P(m_i, m_j \mid \mathbf{o}, \mathbf{\Psi})$ for j in the neighborhood $N(i)$ of i cannot be computed exactly. Informally, the mean field approach consists in approximating the intractable probabilities by neglecting fluctuations from the mean in the neighborhood of each site i. More generally, we talk about mean-field-like approximations when the value for site i does not depend on the value for other sites that are all set to constants (not necessarily to the means) independently of the value for site i. These constant values denoted by $\tilde{m}_1, \ldots, \tilde{m}_n$ are not arbitrary but satisfy some appropriate consistency conditions (see [7]). Let $m_{N(i)}$ denote the set of variables $\{m_j, j \in N(i)\}$ associated to the set $N(i)$ of neighbors of i. It follows that $P(m_i \mid \mathbf{o}, \mathbf{\Psi})$ is approximated by

$$P(m_i|\mathbf{o}, \tilde{m}_{N(i)}, \mathbf{\Psi}) \propto f(o_i|m_i^t\Theta) \exp\left(m_i^t\left(\mathbb{W}\sum_{j\in N(i)}\tilde{m}_j\right)\right),$$

where Θ is considered as a vector of parameters. The normalizing constant is not specified, but its computation is not an issue. Then, for all $j \in N(i)$, $P(m_i, m_j \mid \mathbf{o}, \mathbf{\Psi})$ is approximated by $P(m_i \mid \mathbf{o}, \tilde{m}_{N(i)}, \mathbf{\Psi})\, P(m_j \mid \mathbf{o}, \tilde{m}_{N(j)}, \mathbf{\Psi})$. Both approximations are easy to compute. Using such approximations leads to algorithms, which in their general form consist in repeating two steps. At iteration q, we have the following:

- **Step 1.** Create from the data $\mathbf{o}$ and some current parameter estimates $\mathbf{\Psi}^{(q-1)}$ a configuration $\tilde{m}_1^{(q)}$, $\ldots \tilde{m}_n^{(q)}$. Replace the Markov distribution $P(\mathbf{m})$ defined, as in (1), by the factorized distribution $\prod_{i=1}^n P(m_i|\tilde{m}_{N(i)}^{(q)})$.

  It follows that the joint distribution $P(\mathbf{o}, \mathbf{m}|\mathbf{\Psi})$ can also be approximated by a factorized distribution, and the two problems encountered when considering the EM algorithm with the exact joint distribution disappear. The following is therefore the second step:
- **Step 2.** Apply the EM algorithm for this factorized model with starting values $\mathbf{\Psi}^{(q-1)}$ to get updated estimates $\mathbf{\Psi}^{(q)}$ of the parameters.

Note that, in practice, performing one EM iteration is usually enough.

In particular, the *mean field* and *simulated field* algorithms correspond to two different ways of performing Step 1. The *mean field* algorithm consists in updating the $\tilde{m}_i{}^{(q)}$s by setting, for all $i = 1, \ldots, n$, $\tilde{m}_i{}^{(q)}$ to the mean of distribution $P(m_i \mid \mathbf{o}, \tilde{m}_{N(i)}^{(q)}, \mathbf{\Psi}^{(q-1)})$. Note that as M_i is an indicator vector, the mean value $\tilde{m}_i{}^{(q)}$ is a vector made of the respective probabilities to be in each of the G classes. In the *simulated field* algorithm, $\tilde{m}_i{}^{(q)}$ is simulated from $P(m_i \mid \mathbf{o}, \tilde{m}_{N(i)}^{(q)}, \mathbf{\Psi}^{(q-1)})$. The HMRF estimation provides us with estimations for the means and covariance matrices of the G Gaussian distributions, namely, μ_g and Σ_g, for $g = 1, \ldots, G$, but also for the hidden field parameters, matrix $\mathbb{W}$. It follows easily the approximations of the $P(M_i = e_g|\mathbf{o}, \mathbf{\Psi})$s required to classify each site using the MPM or MAP principles.

REFERENCES

[1] G. Celeux and G. Govaert, "Gaussian Parsimonious Clustering Models," *Pattern Recognition,* vol. 28, pp. 781-793, 1995.

[2] C. Bouveyron, S. Girard, and C. Schmid, "High Dimensional Data Clustering," *Computational Statistics and Data Analysis,* 2007.

[3] G.R. Cross and A.K. Jain, "Markov Random Fields Texture Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 5, no. 1, p. 2539, 1983.

[4] W. Pieczynski and A. Tebbache, "Pairwise Markov Random Fields and Segmentation of Textured Images," *Machine Graphics and Vision,* vol. 9, pp. 705-718, 2000.

[5] D. Benboudjema and W. Pieczynski, "Unsupervised Image Segmentation Using Triplet Markov Fields," *Computer Vision and Image Understanding,* vol. 99, pp. 476-498, 2005.

[6] D. Benboudjema and W. Pieczynski, "Unsupervised Statistical Segmentation of Non Stationary Images Using Triplet Markov Fields," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 8, pp. 367-1378, Aug. 2007.

[7] G. Celeux, F. Forbes, and N. Peyrard, "EM Procedures Using Mean Field-Like Approximations for Markov Model-Based Image Segmentation," *Pattern Recognition,* vol. 36, no. 1, pp. 131-144, 2003.

[8] S. Kumar and M. Hebert, "Discriminative Random Fields," *Int'l J. Computer Vision,* vol. 68, no. 2, pp. 179-201, 2006.

[9] M. Vignes and F. Forbes, "Gene Clustering via Integrated Markov Models Combining Individual and Pairwise Features," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* 2007.

[10] J. Besag, "On the Statistical Analysis of Dirty Pictures," *J. Royal Statistical Soc. B,* vol. 48, no. 3, pp. 259-302, 1986.

[11] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, pp. 721-741, 1984.

[12] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc. B,* vol. 39, pp. 1-38, 1977.

[13] B. Chalmond, "An Iterative Gibbsian Technique for Reconstruction of m-ary Images," *Pattern Recognition,* vol. 22, no. 6, pp. 747-761, 1989.

[14] W. Qian and M. Titterington, "Estimation of Parameters in Hidden Markov Models," *Philosophical Trans. Royal Soc. A,* vol. 337, pp. 407-428, 1991.

[15] S. Lazebnik, C. Schmid, and J. Ponce, "Affine-Invariant Local Descriptors and Neighborhood Statistics for Texture Recognition," *Proc. IEEE Int'l Conf. Computer Vision,* 2003.

[16] G. Celeux, "Bayesian Inference for Mixtures: The Label Switching Problem," *Proc. Computational Statistics,* pp. 227-232, 1998.

[17] J. Li, "Clustering Based on a Multi-Layer Mixture Model," *J. Computational and Graphical Statistics,* vol. 14, no. 3, 2005.

[18] M. Stephens, "Dealing with Label Switching in Mixture Models," *J. Royal Statistical Soc. B,* vol. 62, pp. 795-809, 2000.

[19] G. Schwarz, "Estimating the Dimension of a Model," *Annals of Statistics,* vol. 6, pp. 461-464, 1978.

[20] F. Forbes and N. Peyrard, "Hidden Markov Random Field Model Selection Criteria Based on Mean Field-Like Approximations," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 8, Aug. 2003.

[21] R. Chellapa and S. Chatterjee, "Classification of Textures Using Gaussian Markov Random Fields," *IEEE Trans. Acoustics Speech and Signal Processing,* pp. 959-963, 1985.

[22] T. Ojala, M. Pietikainen, and T. Manpaa, "Multiresolution Gray-Scale and Rotation-Invariant Texture Classification with Local Binary Patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, pp. 971-987, 2002.

[23] T. Toyoda and O. Hasegawa, "Texture Classification Using Extended Higher Order Local Autocorrelation Features," *Proc. Fourth Int'l Workshop Texture Analysis and Synthesis,* pp. 131-136, 2005.

[24] J.S. De Bonet and P. Viola, "Texture Recognition Using a Non Parametric Multi-Scale Statistical Model," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1998.
[25] B.S. Manjunath and W.Y. Ma, "Texture Features for Browsing and Retrieval of Image Data," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, pp. 837-842, 1996.
[26] J. Blanchet, F. Forbes, and C. Schmid, "Markov Random Fields for Textures Recognition with Local Invariant Regions and Their Geometric Relationships," *Proc. British Machine Vision Conf.,* Sept. 2005.
[27] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.
[28] Y. Delignon, A. Marzouki, and W. Pieczynski, "Estimation of Generalized Mixtures an Its Application in Image Segmentation," *IEEE Trans. Image Processing,* vol. 6, pp. 1364-1375, 1997.
[29] A. Jasra, C.C. Holmes, and D.A. Stephens, "Markov Chain Monte Carlo Methods and the Label Switching Problem in Bayesian Mixture Modeling," *Statistical Science,* vol. 20, no. 1, 2005.
[30] S. Le Hegarat-Mascle, A. Kallel, and X. Descombes, "Ant Colony Optimization for Image Regularization Based on a Nonstationary Markov Modeling," *IEEE Trans. Image Processing,* vol. 16, no. 3, pp. 865-879, 2007.

Juliette Blanchet received the PhD degree in applied mathematics, specializing in statistics, from the Joseph Fourier University, Grenoble, France, in 2007. She is currently carrying out a postdoctorate at the Swiss Federal Institute for Snow and Avalanche Research (SLF), Davos, Switzerland. Her research activities include clustering, hidden Markov fields, missing data, and extreme events.

Florence Forbes received the MSc degree in computer science and applied mathematics from the Ecole National Supérieure Informatique et Mathématiques Appliquées de Grenoble in 1993 and the PhD degree in applied probabilities from the Joseph Fourier University, Grenoble, France, in 1996. She is a research scientist at the Institut National de Recherche en Informatique et Automatique (INRIA). She joined the IS2 research team at INRIA Rhône-Alpes in 1998 and is currently the head of the MISTIS team since 2003. Her research activities include Bayesian image analysis, Markov models, and hidden structure models.