Unsupervised Learning of Gaussian Mixtures Based on Variational Component Splitting

Constantinos Constantinopoulos and Aristidis Likas, *Senior Member, IEEE*

***Abstract*—In this paper, we present an incremental method for model selection and learning of Gaussian mixtures based on the recently proposed variational Bayes approach. The method adds components to the mixture using a Bayesian splitting test procedure: a component is split into two components and then variational update equations are applied only to the parameters of the two components. As a result, either both components are retained in the model or one of them is found to be redundant and is eliminated from the model. In our approach, the model selection problem is treated locally, in a region of the data space, so we can set more informative priors based on the local data distribution. A modified Bayesian mixture model is presented to implement this approach, along with a learning algorithm that iteratively applies a splitting test on each mixture component. Experimental results and comparisons with two other techniques testify for the adequacy of the proposed approach.**

***Index Terms*—Clustering, mixture models, model selection, variational Bayes methods.**

I. Introduction

GAUSSIAN mixture models are a valuable statistical tool for modeling densities. They are flexible enough to approximate any given density with high accuracy, and, in addition, they can be interpreted as a soft clustering solution. Thus, they have been widely used in both supervised and unsupervised learning, and have been extensively studied [1] and applied in several domains (for example, [2] and [3]). The parameters of a mixture can be estimated using the expectation–maximization (EM) algorithm [4], [5]. This is the standard training approach, although it exhibits some drawbacks. At first, the EM converges to a local maximum of the likelihood that depends on the initial parameter values. Regarding likelihood, we have to avoid unbounded maxima due to singular covariance matrices [1]. To tackle this, we can set constraints on the covariance matrices or impose a proper prior. Another issue that arises in the training of mixture models is the specification of the appropriate number of mixture components for a given data set. Due to the importance of the model selection problem a number of methods have been proposed to address it.

The most straightforward model selection approach is fitting a number of mixtures with varying number of components using EM for likelihood maximization, and evaluating the solutions using a suitable criterion. The criterion is usually in the form of a likelihood term plus a penalty term that penalizes mixtures with higher number of components. Examples of such criteria are the Akaike's information criterion, the Bayesian inference criterion, the Laplace empirical criterion, and the minimum message length (MML) criterion (see [1] for a review and comparisons of the criteria). In the same spirit, in [6], the use of the cross-validated likelihood has been proposed; mixtures of varying complexity are fitted to the training data, and are evaluated using the likelihood on a separate test set. A test that is based on stability has been proposed in [7]. The proposed stability measure constitutes an upper bound to the cross-validation classification error, and has been extended for clustering problems.

Methods that simultaneously train the mixture and adjust the number of components have also been proposed. A top–down algorithm has been proposed in [8] based on suitable statistical tests, which check the tendency for clustering and the significant number of clusters in the data. In [9], the MML criterion has been integrated in the likelihood function, and is optimized gradually using EM. Starting with a large number of components, the optimization method progressively removes components, and the MML criterion is used to suggest the best model. Moreover, this approach is less sensitive to initialization than standard EM training. Another method for maximizing a weighted likelihood function has been proposed in [10]. The proposed empirical objective function is maximized using an EM algorithm which incorporates a rival penalization mechanism. This mechanism forces the components to compete for responsibility over the data, and the losers are penalized and eventually fade out.

A fully Bayesian approach has been proposed in [11], where the number of components is treated as a random variable, and the reversible jump Markov chain Monte Carlo method is used for sampling, however, this method is computationally demanding. To deal with the intractable integrations appearing in the Bayesian approach, the use of the variational approximation [12]–[14] has been proposed that yields an iterative method similar to the formulation of EM that has been proposed in [15]. This general optimization method called *variational Bayes* (VB) has been employed in a number of recent works. An online version of VB has been proposed in [16]. VB for fitting mixture models has been used in [17]–[19]. Also in [20], the VB method has been used in conjunction with the split-and-merge EM algorithm [21]. They derived an objective function that allows the estimation of the parameters and the number of components simultaneously, and applied

Manuscript received March 3, 2006; revised August 9, 2006 and October 18, 2006; accepted October 24, 2006. This work was supported in part by the European Union in the framework of the program "Heraklitos" of the "Operational Program for Education and Initial Vocational Training" of the 3rd Community Support Framework of the Hellenic Ministry of Education, funded by 25% from national sources and 75% from the European Social Fund (ESF).

The authors are with the Department of Computer Science, University of Ioannina, GR 45110 Ioannina, Greece (e-mail: ccostas@cs.uoi.gr; arly@cs.uoi.gr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2006.891114

split-and-merge EM for fitting a Gaussian mixture, as well as a mixture of experts for regression. The VB has also been used in [13] to estimate the parameters and the number of components for a mixture of factor analyzers. They utilize a birth/death operation on components to treat the model selection problem.

One of the most interesting approaches for the determination of the number of components is suggested in [19]. It is a VB method for optimizing the marginal likelihood given the mixing coefficients. It starts with a large number of components, and progressively removes those that reside in the same region of the data space. We have found the method to be quite effective, but the results are affected by the parameters of the priors. As it will be discussed later, although the method does not allow for several components covering the same cluster of data, if the prior on the precision matrix (inverse covariance) of the components is not properly chosen, it frequently reaches a solution where a component covers more than two clusters. In addition, although the method is deterministic, its convergence point depends on the initial specification of the component parameters.

In this paper, we propose a Bayesian method for Gaussian mixture learning that is deterministic, does not depend on the initialization, and resolves adequately the model selection problem. The method is an incremental one: it starts with one component and progressively adds components to the model. The procedure for component addition is based on a *splitting test* applied to each of the existing mixture components. According to this test, a component is replaced by two subcomponents and, then, VB update equations are applied to the specific pair of components, while the rest components remain "fixed." Due to the introduction of priors on the parameters of the Gaussians, a competition takes place between the subcomponents. If the data distribution in the region of the tested component strongly suggests the existence of more than one clusters, then both subcomponents will "survive" and the number of model components will be increased. Otherwise, the competition among the two subcomponents will cause one of them to be eliminated and the initial component will be recovered. This strategy of incremental component addition also facilitates the specification of the parameters of the priors, since it can be based on the parameters of the component to be split. In order to apply this idea, a modification of the Bayesian mixture model is required that we will describe later.

In Section II, we describe in short the VB method. In Section III, we present a modified Bayesian model and, in Section IV, we derive update equations for mixture parameters based on the maximization of a variational bound of the marginal likelihood. In Section V, the splitting test is described along with the proposed incremental training algorithm. Experimental results are presented in Section VI, and, finally, in Section VII, we provide conclusions and directions for future research.

II. Variational Bayesian Model Selection

A convenient way to control complexity in mixture models is by adjusting the values of the mixing coefficients. A component is removed from the model if its mixing coefficient is set to zero. Consequently, it is possible to consider a mixture model with a large number of components and maximize a suitable objective function with respect to the mixing coefficients. In this way, the redundant components would be eliminated, as their mixing coefficients would be set equal to zero. The typical maximum likelihood estimation approach through EM is not a viable choice for this kind of model selection, since a diminishing mixing coefficient results in a component whose covariance has diminishing eigenvalues, i.e., it leads to the formation of singular components. The Bayesian approach provides a solution to this problem since it constrains the component parameters through the introduction of priors, thus it does not encourage the formation of singular components. In this manner, only insignificant components are removed from the model. The difficulties of the Bayesian method stem from the computation of Bayesian integrals. However, on several occasions, the variational approximation has provided a viable solution to inference in Bayesian models. In the rest of the section, we briefly describe and discuss the variational Bayes approach proposed in [19].

Let $X = \{x_n\}$ be a set of N observations, where each $x_n \in \Re^d$ is a feature vector. Let also f be a mixture with J Gaussian components

$$f(x) = \sum_{j=1}^{J} \pi_j \, \mathcal{N}(x|\mu_j, T_j) \tag{1}$$

where $\pi = \{\pi_j\}$ are the mixing coefficients (weights), $\mu = \{\mu_j\}$ the means (centers) of the components, and $T = \{T_j\}$ the precision (inverse covariance) matrices.

Modeling the data using f implies the assumption that for each observation x_n there exists a hidden variable z_n denoting the component that generated x_n. Let $Z = \{z_n\}$ denote the set of these hidden variables. z_n can be represented as a J-dimensional binary vector, such that if the jth component is responsible for x_n, then $z_{jn} = 1$; otherwise, $z_{jn} = 0$. Thus, the constraints $\sum_{j=1}^{J} z_{jn} = 1$ hold for each n. Consequently, the density of x_n given z_n is $\mathcal{N}(x_n|\mu_j, T_j)$, assuming $z_{jn} = 1$ for some component j.

A Bayesian mixture model is obtained by imposing priors on the parameters π, μ and T of the components. Typically, conjugate priors are used; that is, Dirichlet, Gaussian, and Wishart, correspondingly. The Wishart prior for T is

$$\begin{aligned} p(T) &= \prod_{j=1}^{J} \mathcal{W}(T_j|\nu, V) \\ &= \prod_{j=1}^{J} \frac{|T_j|^{(\nu-d-1)/2} \exp \operatorname{tr}\left\{-\frac{1}{2} V T_j\right\}}{2^{\nu d/2} \pi^{d(d-1)/4} |V|^{-n/2} \prod_{i=1}^{d} \Gamma((\nu+1-i)/2)} \end{aligned} \tag{2}$$

where the scalar ν denotes the degrees of freedom, V is the scale matrix, and the expected value of T_j is $\langle T_j \rangle = \nu V^{-1}$. However, in [19], a Bayesian model has been proposed that does not assume a prior over the mixing weights, which are treated as parameters and not as random variables. The graphical model for this approach is depicted in Fig. 1(a).

In [19], the explicit estimation of means and covariances has been suggested, using the expected values of their respective variational posterior distributions. If we also assume a prior over

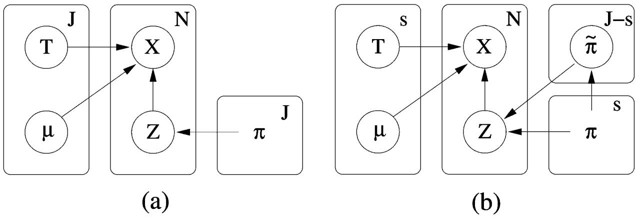


Fig. 1 (a) Graphical model proposed in [19]. The plates represent repetitions of the included random variables and the exact number of repetitions is depicted in the upper right corner of each plate. We do not circle π to denote their special treatment as parameters without priors. (b) Proposed graphical model adapted for local model selection.

the mixing coefficients, and estimate them using the expectations with respect to the variational posterior, then these estimates are going to be biased towards nonzero values. We avoid this bias by omitting the prior over the mixing coefficients. We are going to exploit these alternative choices for the development of our method.

Bayesian model selection is obtained through maximization of the marginal likelihood $p(X|\pi)$ that results by integrating out the variables $\theta = \{Z, \mu, T\}$ from the joint density $p(X, \theta|\pi)$

$$p(X|\pi) = \int p(X, \theta|\pi)\,\mathrm{d}\theta \tag{3}$$

and treating the mixing weights as parameters. The variational approximation of the VB method suggests the maximization of a lower bound of the logarithmic marginal likelihood

$$\mathcal{L}[q, \pi] = \int q(\theta) \log \frac{p(X, \theta|\pi)}{q(\theta)}\,\mathrm{d}\theta \leq \log p(X|\pi) \tag{4}$$

where the variational posterior $q(\theta)$ is an arbitrary distribution approximating the posterior $p(\theta|X)$. During maximization, the *mean-field* approximation [13], [14], [17]–[19] is adopted, namely that

$$q(\theta) = q(Z)q(\mu)q(T).$$

A notable property of the method is that during maximization of $\mathcal{L}$, if some of the components fall in the same region in the data space, then there is a strong tendency in the model to eliminate the redundant components, once the data in this region are sufficiently explained by fewer components. An interpretation of this competition between components is obtained from the following decomposition of the variational bound:

$$\mathcal{L} = \int q(\theta) \log p(X|\theta)\,\mathrm{d}\theta - \int q(\theta) \log \frac{q(\theta)}{p(\theta|\pi)}\,\mathrm{d}\theta. \tag{5}$$

The first term corresponds to the expected log-likelihood, with respect to $q(\theta)$. The second term is the Kullback–Leibler divergence of the prior $p(\theta|\pi)$ from $q(\theta)$. Due to the mean-field approximation, the divergence is a sum of three terms

$$\begin{aligned}\int q(\theta) \log \frac{q(\theta)}{p(\theta|\pi)}\,\mathrm{d}\theta &= \int q(Z) \log \frac{q(Z)}{p(Z|\pi)}\,\mathrm{d}\theta \\ &\quad + \int q(\mu) \log \frac{q(\mu)}{p(\mu)}\,\mathrm{d}\mu \\ &\quad + \int q(T) \log \frac{q(T)}{p(T)}\,\mathrm{d}T. \end{aligned} \tag{6}$$

If the adopted conjugate priors factorize over the number of components, then the divergence is a sum over the number of components, for example

$$\int q(T) \log \frac{q(T)}{p(T)}\,\mathrm{d}T = \sum_{j=1}^{J} \int q(T_j) \log \frac{q(T_j)}{p(T_j)}\,\mathrm{d}T_j. \tag{7}$$

If only a few data are available for the estimation of the posterior $q(T_j)$, then this estimation is dominated by the prior $p(T_j)$. Therefore, as the number of data that are available to the jth component tends to zero, the corresponding term of (7) also tends to zero. Similar results hold for the rest of the terms in (6). Consequently, a redundant component that covers few data favors the decrease of divergence, and the variational bound (5) increases as the component is eliminated. On the other hand, if there is a strong evidence from the data, then the component is retained, and the bound increases dominated by the increase of the expected log-likelihood term in (5). In [18], it is discussed how the Bayesian information criterion and the MML criterion can emerge as a limiting case of the variational maximization of the marginal likelihood.

The competition between components suggests a natural approach for addressing the model selection problem: fit a mixture initialized with a large number of components and let the competition eliminate the redundant. This is an effective method, and in general provides the correct solution; however, it exhibits some weaknesses. The method depends on the initial parameters of the mixture, and this affects model selection especially if initially fits a small number of components. Also, if the mixture is initialized with a large number of components, it is computationally expensive for large data sets in high dimensions. Apart from these, the most serious difficulty to be addressed is related with the specification of the Wishart prior imposed on the precision matrix. More specifically, the prior knowledge captured by the scale matrix V affects the results of model selection (see, for an example, Fig. 2). The method of [19] (we refer to it as VBgmm) has been applied on an artificial data set with 208 two-dimensional (2-D) points that form 15 Gaussian clusters. Three different scale matrices ($V = \beta\mathcal{I}$ with $\beta = 1, 0.25, 0.025$) were tested for fitting a mixture with 40 initial components, resulting in solutions with 9, 13, and 15 components, respectively.

We have observed that this is a consistent trend of the method: the more narrow scale matrices are adopted, the more components are used in the final solution. Here, we use the term narrow to denote a scale matrix with comparably small eigenvalues. However, it does not seem possible to determine in advance a good value for the scale matrix. This issue becomes more important in the case of data sets that contain both large and small clusters. If a broad scale matrix is selected, then many small clusters will be covered by a single component. If a narrow scale matrix is selected, then large clusters will be covered by more than one component. It must be noted that we have not observed analogous sensitivity to the prior over the centers, which is set to be broad and uninformative.

Concluding, the disadvantage of the method is that by using an arbitrarily broad scale matrix, it is not possible to take into account the characteristics of the data in the region where the

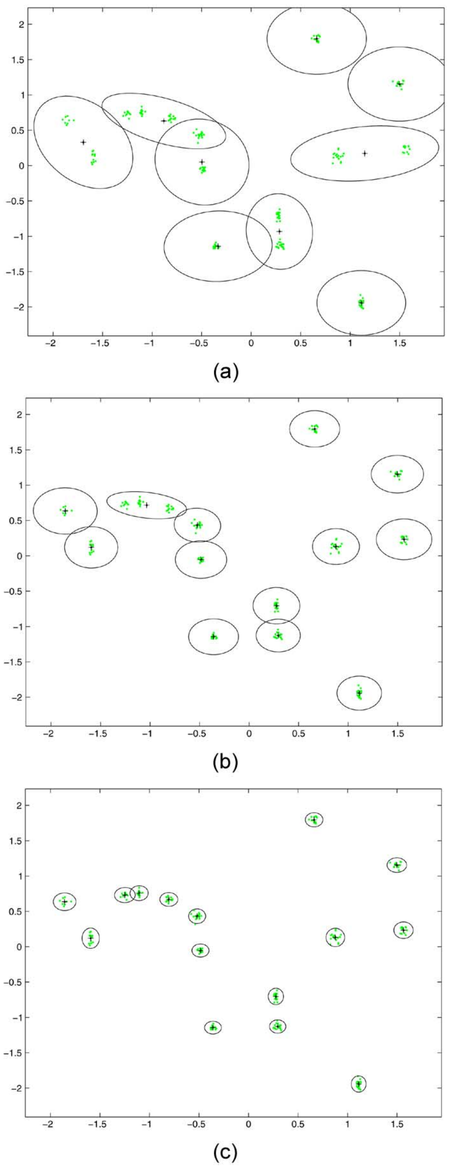


Fig. 2. Fitting artificial data using VBgmm with different prior parameters. From (a) through (c), the results are shown using narrower scale matrices.

competition between components takes place. In other words, the method operates with a *global precision prior* that is difficult to be correctly determined *a priori*, while a *local precision prior* seems to be desirable. The proper choice of a different V for each component seems to be a very difficult and complicated task. The use of a hierarchical Bayesian model, as in [15], could offer a solution. Namely, we could impose a distinct prior over the precision matrix of each component and a suitable hyperprior over the parameters of these priors, and estimate the parameters of the priors maximizing a proper variational bound. However, we propose an incremental method for building the mixture that allows us to define V in a more explicit way. At each step, learning is restricted in the data region occupied by a specific mixture component j, thus a local precision prior can be specified based on the precision matrix T_j. In order to achieve this behavior, a modification to the generative model used in VBgmm is needed that restricts the competition in a subset of the components only. This idea of *local model selection* is presented in Section III.

III. Bayesian Framework for Local Model Selection

Consider a set of observations $X = \{x_n \in \Re^d | n = 1, \ldots, N\}$ and a mixture model f with J Gaussian components

$$f_J(x) = \sum_{j=1}^{J} \pi_j \mathcal{N}(x|\mu_j, T_j). \tag{8}$$

Suppose that a number $J - s$ of the components fit the data well in their corresponding region of influence; then, the question is: can we further optimize the parameters of the remaining s components and also impose a model selection mechanism? In other words, the problem is how to adapt the model of Section II so that the competition among components is restricted in a specific subset of them, while the rest remain "fixed."

This means that we partition the components in two groups, the "fixed" components and the "free" ones, and we estimate only the parameters of the latter. However, before proceeding to such an estimation, it is necessary to impose a suitable prior on the mixing coefficients of the "fixed" components (called "fixed" mixing coefficients), thus preventing their elimination from the mixture model. Following the introduction of this prior, the "fixed" mixing coefficients are treated as random variables and are integrated out, leading to a marginal likelihood that depends only on the "free" mixing coefficients. Maximizing the marginal likelihood with respect to the "free" mixing coefficients, we restrict the search for the redundant components to the corresponding components.

The proposed graphical model is illustrated in Fig. 1(b). It can be observed that it is similar to the model in Fig. 1(a) with the difference that a prior has been imposed on the $J - s$ "fixed" mixing coefficients $\tilde{\pi}$. As before, given the set of hidden variables $Z = \{z_{jn}\}$, it holds that

$$p(X|Z, \mu, T) = \prod_{n=1}^{N} \prod_{j=1}^{J} [\mathcal{N}(x_n|\mu_j, T_j)]^{z_{jn}} \tag{9}$$

assuming independent identically distributed (i.i.d.) observations. The prior distribution of Z assuming i.i.d. hidden variables is a product of multinomials

$$p(Z|\pi, \tilde{\pi}) = \prod_{n=1}^{N} \prod_{j=1}^{s} \pi_j^{z_{jn}} \prod_{j=s+1}^{J} \tilde{\pi}_j^{z_j n} \tag{10}$$

given the subset $\tilde{\pi} = \{\tilde{\pi}_j\}$ of "fixed" mixing coefficients and the subset $\pi = \{\pi_j\}$ of "free" mixing coefficients. For notational convenience and assuming J mixing components, we can always rearrange the indices so that the first s components are the "free" ones. The subsets of the mixing coefficients are disjoint, and their values are restricted to be nonnegative and sum to unit: $\sum_{j=1}^{s} \pi_j + \sum_{j=s+1}^{J} \tilde{\pi}_j = 1$. We also note that in the boundary case where coefficient π_j becomes zero, it is necessary that $z_{jn} = 0$ for all n, and, consequently, all the corresponding factors become unit.

The typical Bayesian framework assumes conjugate Dirichlet priors over the entire set of mixing coefficients. However, in order to apply our idea, it is necessary to define the conditional joint distribution $p(\tilde{\pi}|\pi)$ of the "fixed" coefficients given the "free" ones. Thus, we define a Dirichlet prior over all the mixing coefficients

$$p(\tilde{\pi}, \pi) = \frac{\Gamma\left(\sum_{j=1}^{J} \alpha_j\right)}{\prod_{j=1}^{J} \Gamma(\alpha_j)} \prod_{j=1}^{s} \pi_j^{\alpha_j - 1} \prod_{j=s+1}^{J} \tilde{\pi}_j^{\alpha_j - 1} \tag{11}$$

where $a_j = 1$ for $j = 1, \ldots, s$ and $a_j > 1$ for $j = (s+1), \ldots, J$. Consequently, the first s factors are reduced to a "uniform distribution" that allows some of the "free" coefficients to become zero, while the "fixed" ones have zero probability to become zero. It is known that if the joint distribution of a set of variables is Dirichlet, then the marginal joint distribution of a subset of the variables is also Dirichlet (see [22]). Using Bayes theorem, the conditional joint distribution $p(\tilde{\pi}|\pi)$ can be derived, which is a nonstandard Dirichlet

$$p(\tilde{\pi}|\pi) = \left(1 - \sum_{k=1}^{s} \pi_k\right)^{-J+s} \frac{\Gamma\left(\sum_{j=s+1}^{J} \alpha_j\right)}{\prod_{j=s+1}^{J} \Gamma(\alpha_j)} \cdot \prod_{j=s+1}^{J} \left(\frac{\tilde{\pi}_j}{1 - \sum_{k=1}^{s} \pi_k}\right)^{\alpha_j - 1} \tag{12}$$

and constitutes a conjugate prior of the "fixed" coefficients. The expected values of $\tilde{\pi}_j$ and $\log \tilde{\pi}_j$ given π are

$$\langle \tilde{\pi}_j \rangle_\pi = \left(1 - \sum_{k=1}^{s} \pi_k\right) \frac{\alpha_j}{\sum_{k=s+1}^{J} \alpha_k} \tag{13}$$

$$\langle \log \tilde{\pi}_j \rangle_\pi = \log\left(1 - \sum_{k=1}^{s} \pi_k\right) + \psi(\alpha_j) - \psi\left(\sum_{k=s+1}^{J} \alpha_k\right) \tag{14}$$

where $\psi(x)$ is the digamma function, $\psi(x) = (\mathrm{d}/\mathrm{d}x) \log \Gamma(x)$.

Completing the specification of our Bayesian model, we assume Gaussian and Wishart priors for μ and T, respectively

$$p(\mu) = \prod_{j=1}^{J} \mathcal{N}(\mu_j | 0, \beta\, \mathcal{I}) \tag{15}$$

$$p(T) = \prod_{j=1}^{J} \mathcal{W}(T_j | \nu, V). \tag{16}$$

In Section IV, we derive a learning method for this model, based on the maximization of the marginal likelihood.

IV. Variational Learning With Local Model Selection

Learning in the Bayesian framework can be achieved through maximization of the marginal likelihood of the data which is obtained by integrating out the hidden variables of the model. In our case, the marginal likelihood of X given π is obtained by integrating out $\theta = \{Z, \mu, T, \tilde{\pi}\}$ as follows:

$$p(X|\pi) = \sum_Z \int p(X, Z, \mu, T, \tilde{\pi}|\pi)\, \mathrm{d}\mu\, \mathrm{d}T\, \mathrm{d}\tilde{\pi}. \tag{17}$$

Following the VB methodology [13], [14], [17]–[19], which aims to maximize a lower bound $\mathcal{L}$ of the logarithmic marginal likelihood $\log p(X|\pi)$, we maximize

$$\mathcal{L}[q, \pi] = \sum_Z \int q(Z, \mu, T, \tilde{\pi}) \times \log \frac{p(X, Z, \mu, T, \tilde{\pi}|\pi)}{q(Z, \mu, T, \tilde{\pi})}\, \mathrm{d}\mu\, \mathrm{d}T\, \mathrm{d}\tilde{\pi} \tag{18}$$

where q is an arbitrary distribution that approximates the posterior distribution $p(Z, \mu, T, \tilde{\pi}|X, \pi)$. The choices that affect q are explicitly the mean-field constraint that we describe in the following, and implicitly the conjugate priors that we impose on the variables of the graphical model. The maximization of $\mathcal{L}$ is performed in an iterative way, where at each iteration two steps take place: first, maximization of the bound with respect to q, and, subsequently, maximization of the bound with respect to π.

To implement the maximization with respect to q, we have adopted the mean-field approximation [13], [14], [17]–[19], and we consider that q is constrained to be a product of the form

$$q(\theta) = q_Z(Z) q_\mu(\mu) q_T(T) q_{\tilde{\pi}}(\tilde{\pi}).$$

The method does not assume any specific form for the factors of q; instead, it maximizes $\mathcal{L}$ with respect to the functional form of q_Z, q_μ, q_T, and $q_{\tilde{\pi}}$. The standard variational analysis optimization involves the Euler equation and constraints of Lagrange multiplier type to ensure that the solutions are density functions (see [23]). The solution for each $\vartheta \in \theta$ is

$$q_\vartheta(\vartheta) = \frac{\exp\langle \log\, p(X, \theta|\pi) \rangle_{\theta - \vartheta}}{\int \exp\langle \log\, p(X, \theta|\pi) \rangle_{\theta - \vartheta}\, \mathrm{d}\vartheta} \tag{19}$$

where the expectation $\langle \cdot \rangle_{\theta - \vartheta}$ is computed with respect to all the variables except ϑ. Applying (19), the result is the following set of densities:

$$q_Z(Z) = \prod_{n=1}^{N} \prod_{j=1}^{s} r_{jn}^{z_{jn}} \prod_{j=s+1}^{J} \rho_{jn}^{z_{jn}} \tag{20}$$

$$q_\mu(\mu) = \prod_{j=1}^{J} \mathcal{N}(\mu_j | m_j, S_j) \tag{21}$$

$$q_T(T) = \prod_{j=1}^{J} \mathcal{W}(T_j | \eta_j, U_j) \tag{22}$$

$$q_{\tilde{\pi}}(\tilde{\pi}) = \left(1 - \sum_{k=1}^{s} \pi_k\right)^{-J+s} \frac{\Gamma\left(\sum_{j=s+1}^{J} \tilde{\alpha}_j\right)}{\prod_{j=s+1}^{J} \Gamma(\tilde{\alpha}_j)} \cdot \prod_{j=s+1}^{J} \left(\frac{\tilde{\pi}_j}{1 - \sum_{k=1}^{s} \pi_k}\right)^{\tilde{\alpha}_j - 1}. \tag{23}$$

The parameters of the densities are the following:

$$r_{jn} = \frac{\tilde{r}_{jn}}{\sum_{k=1}^{s} \tilde{r}_{kn} + \sum_{k=s+1}^{J} \tilde{\rho}_{kn}}, \qquad \text{for } j = 1, \ldots, s \tag{24}$$

$$\rho_{jn} = \frac{\tilde{\rho}_{jn}}{\sum_{k=1}^{s} \tilde{r}_{kn} + \sum_{k=s+1}^{J} \tilde{\rho}_{kn}}, \qquad \text{for } j = s+1, \ldots, J \tag{25}$$

$$\tilde{r}_{jn} = \pi_j \exp\Bigg\{\frac{1}{2}\langle \log |T_j| \rangle - \frac{1}{2}\mathrm{tr} \times \left\{\langle T_j \rangle \left(x_n x_n^T - x_n \langle \mu_j \rangle^T + \langle \mu_j \rangle x_n^T + \langle \mu_j \mu_j^T \rangle\right)\right\}\Bigg\} \tag{26}$$

$$\tilde{\rho}_{jn} = \exp\Bigg\{\langle \log \tilde{\pi}_j \rangle + \frac{1}{2}\langle \log |T_j| \rangle - \frac{1}{2}\mathrm{tr} \times \left\{\langle T_j \rangle \left(x_n x_n^T - x_n \langle \mu_j \rangle^T + \langle \mu_j \rangle x_n^T + \langle \mu_j \mu_j^T \rangle\right)\right\}\Bigg\} \tag{27}$$

$$m_j = S_j^{-1} \langle T_j \rangle \sum_{n=1}^{N} \langle z_{jn} \rangle x_n \tag{28}$$

$$S_j = \beta \mathcal{I} + \langle T_j \rangle \sum_{n=1}^{N} \langle z_{jn} \rangle \tag{29}$$

$$\eta_j = \nu + \sum_{n=1}^{N} \langle z_{jn} \rangle \tag{30}$$

$$U_j = V + \sum_{n=1}^{N} \langle z_{jn} \rangle \left(x_n x_n^T - x_n \langle \mu_j \rangle^T - \langle \mu_j \rangle x_n^T + \langle \mu_j \mu_j^T \rangle\right) \tag{31}$$

$$\tilde{\alpha}_j = \alpha_j + \sum_{n=1}^{N} \langle z_{jn} \rangle. \tag{32}$$

The expectations with respect to $q(\theta)$ used in (24)–(32) satisfy the following: $\langle T_j \rangle = \eta_j U_j^{-1}$, $\langle \log |T_j| \rangle = \sum_{i=1}^{d} \psi(0.5(\eta_j + 1 - i)) + d \ln 2 - \ln |U_j|$, $\langle \mu_j \rangle = m_j$, and $\langle \mu_j \mu_j^T \rangle = S_j^{-1} + m_j m_j^T$. Concerning the responsibilities (posteriors) of component j with respect to x_n, we have that for the "free" components $\langle z_{jn} \rangle = r_{jn}$ (for $j = 1, \ldots, s$), and for the "fixed" it holds that $\langle z_{jn} \rangle = \rho_{jn}$, (for $j = s+1, \ldots, J$). Thus, according to (13) and (14), we get

$$\langle \tilde{\pi}_j \rangle = \left(1 - \sum_{k=1}^{s} \pi_k\right) \frac{\sum_{n=1}^{N} \rho_{jn} + \alpha_j}{\sum_{k=s+1}^{J} \left(\sum_{n=1}^{N} \rho_{kn} + \alpha_k\right)} \tag{33}$$

$$\langle \log \tilde{\pi}_j \rangle = \log\left(1 - \sum_{k=1}^{s} \pi_k\right) + \psi\left(\sum_{n=1}^{N} \rho_{jn} + \alpha_j\right) - \psi\left(\sum_{k=s+1}^{J} \sum_{n=1}^{N} \rho_{kn} + \alpha_k\right). \tag{34}$$

It can be observed that the densities are coupled through their expectations, thus an iterative estimation of the parameters is needed. However, in practice, a single pass seems to be sufficient for this step.

After the maximization of $\mathcal{L}$ with respect to q, the second step of each iteration of the training method requires maximization of $\mathcal{L}$ with respect to π, leading to the following update equation:

$$\pi_j = \left(1 - \sum_{k=s+1}^{J} \langle \tilde{\pi}_k \rangle\right) \frac{\sum_{n=1}^{N} r_{jn}}{\sum_{k=1}^{s} \sum_{n=1}^{N} r_{kn}}. \tag{35}$$

Comparing (33) and (35), we can see how the imposed prior over the "fixed" coefficients affects their estimation. In contrast to the "free" coefficients, the Dirichlet prior hinders the zeroing of the "fixed" ones.

The previous update equations are applied iteratively until convergence, which can be monitored through inspection of the variational bound. During the optimization some of the "free" coefficients converge to zero. Although these coefficients eventually will become exactly zero, we can eliminate the components with very small coefficients. We used as a threshold the value 10^{-10}, as we verified experimentally that any value in this scale can be used without affecting the obtained results. In Section V, we present an algorithm that incorporates the proposed local model selection method to solve the problem globally.

V. Incremental Learning Based on Component Splitting

We have exploited the local model selection method to develop an incremental algorithm for Bayesian mixture model learning. In our approach, mixture components are sequentially added to the mixture model using the following component splitting procedure: one of the mixture components is selected and is appropriately split into two components. We treat the resulting two components as "free" and the rest as "fixed," according to the terminology introduced in Section IV. Next, we set the precision prior $p(T)$ based on the characteristics of the split component, and apply variational learning with local model selection as described in Section IV. There are three possible cases for the outcome of the local model selection. In the first case, where the two "free" components provide a much better fit to the data in their region, both of them are retained in the mixture model. In the second case, where one of them is redundant, during the optimization, it is eliminated and the other is retained. There is also a rare third case, where both components are eliminated because the split component is insignificant (with a very low mixing coefficient). It happens that such a component is responsible for a few outliers, in the vicinity of a significant component. After splitting, the dominant component also gets the responsibility for the outliers, and the new components are both eliminated. In the proposed algorithm, we do not accept this split because it may lead the method to an infinite loop, so we restore the split component. However, it is possible for one to remove the outlier components after termination of the method. An obvious heuristic is to set very small mixing coefficients equal to zero, and check if these changes increase the variational bound. Although it is possible for the proposed algorithm to overestimate the number of components, we do not adopt a global pruning mechanism, e.g., as given in [21]. We are solely based on the capability of the splitting test to prune locally the redundant components. Experimental results supported our choice, as we did not detect a systematic overestimation. In order to apply the proposed method, the mixture model must consist of two components at least. To take into account the possibility that the data set has been generated by a single component, we initially apply the VBgmm method to a two-component model, using as precision prior the inverse covariance of the data set. If learning yields a single-component model, we terminate; otherwise, we start applying splitting tests to the resulting two-component model.

The splitting test is applied sequentially to all components and the method terminates when all mixture components have been unsuccessfully tested for splitting. In the case where a successful split is encountered, the number of mixture components increases and a new round of splitting tests for all components is initialized. A description of the proposed algorithm is summarized in the following.

```
1) Set $\beta := 1\mathrm{e}-10$, and $\nu := d$.
2) Initialize $J := 2$, $V := \mathrm{Cov}\{X\}$ and train a
   mixture model using VBgmm.
3) If after convergence there is only one
   component, then stop.
4) Let $C$ be the set of $J$ components that
   form the mixture model $f_J$.
5) Sort the elements of $C$ in descending
   order, according to $|U_j|$.
6) For each component $c \in C$ do the
   following.
   a) Split $c$ in $c_1$ and $c_2$, according to
      (36)-(39), and form $f_{J+1}$.
   b) Let $F = \{c_1, c_2\}$ be the set of "free"
      components, and $\bar{F}$ the set of
      "fixed" components with elements the
      components of $f_{J+1}$ except $c_1$ and $c_2$.
   c) Set $\alpha_j := \sum_{n=1}^{N} \langle z_{jn} \rangle$ for $c_j \in \bar{F}$, and
      $V := \nu\lambda\mathcal{I}$ where $\lambda$ is the maximum
      eigenvalue of $U_c/\eta_c$.
   d) Apply iteratively (20)-(35) on
      the parameters of $f_{J+1}$, and
      after convergence form $f_{J'}$ with $J'$
      components.
   e) If both components in $F$ have been
      removed, then
        i) register the failure of the
           split;
       ii) continue with the next component
           in $C$ [go to step 6a)].
   f) If one of the components in $F$ has
      been removed, then register the
      failure of the split.
   g) Set $J := J'$ and $f_J := f_{J'}$.
7) If all splits failed, stop; otherwise, go
   to step 4).
```

To illustrate the details of the splitting process, assume that component $\hat{\jmath}$ has to be split, with density $\mathcal{N}(x|\mu_{\hat{\jmath}}, T_{\hat{\jmath}})$. The idea is that in order to form the new mixture, we remove component $\hat{\jmath}$ and insert two new components with densities $\mathcal{N}(x|\mu_{\hat{\jmath}1}, T_{\hat{\jmath}1})$ and $\mathcal{N}(x|\mu_{\hat{\jmath}2}, T_{\hat{\jmath}2})$, respectively. We have selected to place the centers of the two components along the dimension of the principal axis of the covariance $T_{\hat{\jmath}}^{-1}$ and at opposite directions with respect to the center $\mu_{\hat{\jmath}}$. The mixing coefficients of the two components are set equal $\pi_{\hat{\jmath}1} = \pi_{\hat{\jmath}2} = \pi_{\hat{\jmath}}/2$ and their parameters are set according to

$$\mu_{\hat{\jmath}1} = \mu_{\hat{\jmath}} + \sqrt{\lambda}\, u \tag{36}$$

$$\mu_{\hat{\jmath}2} = \mu_{\hat{\jmath}} - \sqrt{\lambda}\, u \tag{37}$$

$$T_{\hat{\jmath}1} = T_{\hat{\jmath}} \tag{38}$$

$$T_{\hat{\jmath}2} = T_{\hat{\jmath}} \tag{39}$$

where λ is the maximum eigenvalue of $T_{\hat{\jmath}}^{-1}$ and u the corresponding eigenvector. It must be noted that we have selected a simple and sensible choice for placing the centers of the two components which has also been used in other methods involving cluster splitting, e.g., see [24] and [25] for advanced methods on how to specify the splitting direction. Other options could also be tested (e.g., random direction selection [13]), as well as multiple splitting tests with different initializations of the two components.

An important issue in the proposed method is the specification of the scale parameter V of the prior $\mathcal{W}(\nu, V)$ over the precision matrices, based on the split component. We set $\nu = d$ (which is the minimum allowed value), and wish the mean νV^{-1} of the precision prior to be comparable to the precision matrix $T_{\hat{\jmath}}$. However, we have empirically found that when setting $V = \nu T_{\hat{\jmath}}^{-1}$ there is a tendency to accept more splits than necessary and better results are obtained if we specify the scale matrix to be somewhat broader. In this spirit, we have selected to set $V = \nu\lambda\mathcal{I}$, where λ is the highest eigenvalue of $T_{\hat{\jmath}}^{-1}$. An example of splitting and setting the prior is illustrated in Fig. 3.

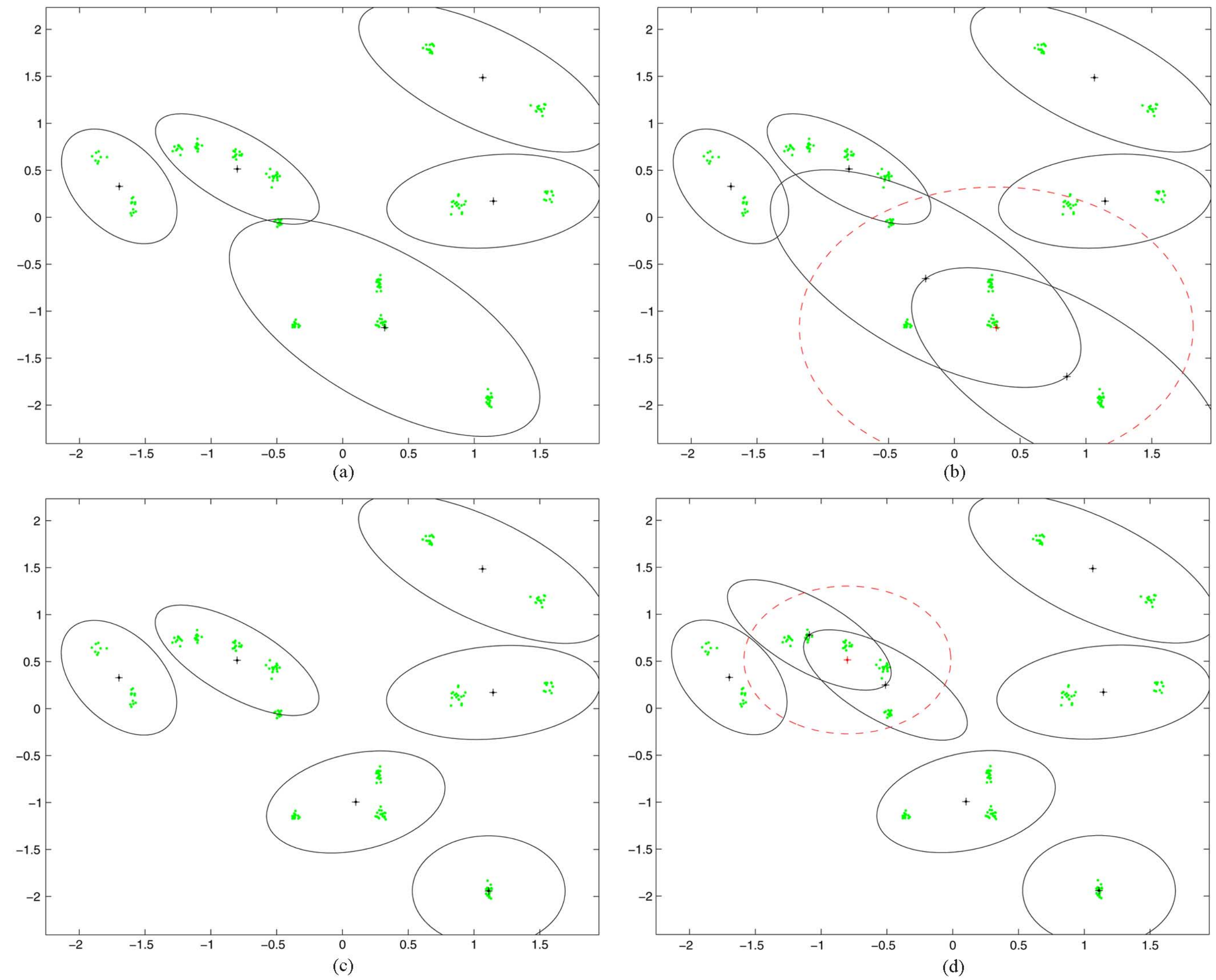
Another aspect of the method deals with the order that components are sequentially selected for the splitting test. We have found that the method performs faster (makes fewer unsuccessful splits) if we give priority to the broader components with broadness measured by the determinant of the covariance matrix of the component. It must be noted that, depending on the structure of the data set, it is possible that the final result is affected by this selection order (order effect). If the clusters are well separated, we have empirically observed that the ordering has no effect on the final solution. For more difficult data sets, it is expected that the results could be sensitive to the selection order. However, it seems very difficult to experimentally assess the importance of this issue.

In what concerns the effect of outliers, it is possible that outliers could affect our estimate of the covariance matrix that is used to specify the prior at each splitting step. This issue can be treated only by using some preprocessing method for outlier removal. Also, it is possible that, due to outliers in the data, outlier components will exist in our final solution. However, as we previously mentioned, it is possible to remove the outlier components after termination of the method, since they have very small mixing coefficients. We could set equal to zero those mixing coefficients that are very small and check if this change increases the variational bound.

A. Complexity Issues

In the following, we briefly discuss the time complexities of the proposed algorithm (we refer to it as VBgmmSplit) and VBgmm. For both algorithms, if the number of components is fixed to J, then the time complexity of the update equations is $O(N\,J\,d^2 + J\,d^3)$, similar to the EM case. However, as the number of components changes, the total execution time is affected differently by the two alternative model selection approaches. The VBgmm algorithm benefits from the adopted bottom-up approach, because during optimization J decreases. It is obvious that for both algorithms the execution time depends on the estimated final number of components. If it is high, then the VBgmmSplit algorithm has to execute a large number of splitting tests, although each test is fast, as it is applied to two components only. On the other hand, if the estimated number of components is low, then the VBgmm algorithm suffers because the mixture is initialized with a large number of components.

In order to speed up each split test, we propose to keep fixed the estimations of the mean vectors and the covariance matrices



(a) (b) (c) (d)

Fig. 3. Four instances of the training procedure. The expected covariance with respect to the Wishart prior is depicted with a dashed line. (a) Intermediate solution with five components. (b) One component is split into two. (c) Mixture after variational learning. (d) Another component is selected and split.

of the "fixed" components. Namely, at each iteration of step 6), instead of the maximization of the variational bound with respect to (21) and (22), we propose the following partial updates:

$$q_\mu(\mu) = C_\mu \prod_{j=1}^{s} \mathcal{N}(\mu_j | m_j, S_j) \tag{40}$$

$$q_T(T) = C_T \prod_{j=1}^{s} \mathcal{W}(T_j | \eta_j, U_j) \tag{41}$$

where $C_\mu = \prod_{j=s+1}^{J} \mathcal{N}(\mu_j | m_j, S_j)$ and $C_T = \prod_{j=s+1}^{J} \mathcal{W}(T_j | \eta_j, U_j)$ are fixed. The values of C_μ and C_T are set according to the results of the previous splitting test.

As a concluding remark, we observed that VBgmm is in general faster than VBgmmSplit. However, an advantage of VBgmmSplit is that it suggests a deterministic initialization of the mixture, thus we do not have to resort to multiple restarts or similar time consuming techniques.

VI. Experimental Results

We evaluated the proposed VBgmmSplit algorithm for learning mixtures using artificial and real data sets. For comparison, we fitted the same data using VBgmm and two more methods: an MML-based method [9] (we refer to it as MMLgmm[1]) and the variational Bayesian mixture of factor analyzers method [13] (we refer to it as VBmfa[2]). During training with VBmfa, we set the maximum intrinsic dimensionality of each factor analyzer equal to the original dimension of the data, so that it can capture sufficiently the covariance matrix. We initialized VBgmm, VBmfa, and MMLgmm with 50 components.

The first test of VBgmmSplit was using artificial data that form Gaussian clusters, so that the resulting mixture can be interpreted as a clustering solution. The first data set (the same that was used in Section II) consists of 208 2-D points forming

[1] Software available at http://www.lx.it.pt/mtf/mixturecode.zip

[2] Software available at http://www.cse.buffalo.edu/faculty/mbeal/software/vbmfa/vbmfa.tar.gz

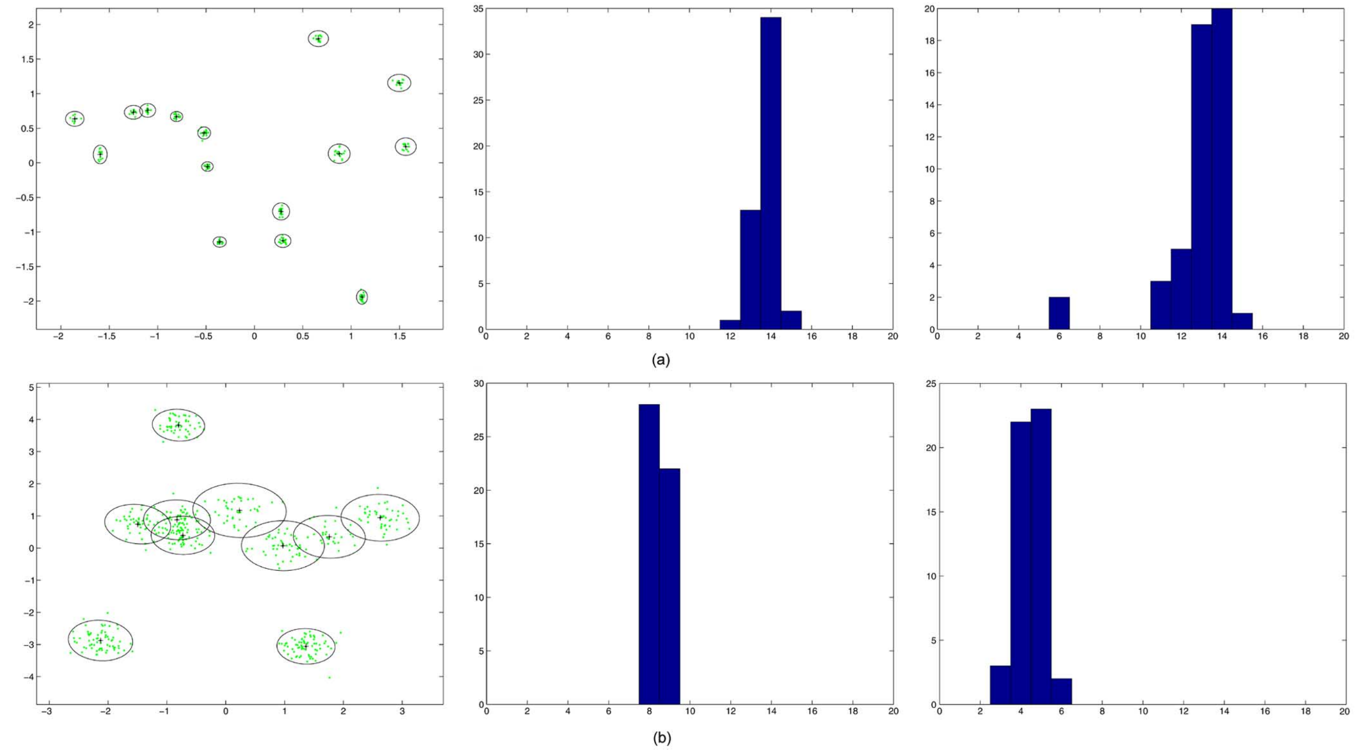


Fig. 4. From left to right: the data clustered using VBgmmSplit, histogram of the number of components found by MMLgmm, and histogram of the number of components found by VBmfa. (a) Artificial data forming 15 clusters. (b) Artificial data forming ten clusters.

15 Gaussian clusters. Fig. 4(a) depicts the result of VBgmmSplit. We applied MLgmm and VBmfa 50 times each, and most of the times they fitted the data using 14 components. Fig. 4(a) illustrates histograms of the number of components found by those methods. VBmfa exhibits higher variance in the results than MMLgmm, and also two times found solutions with only six components. For the next experiment, we used 505 ten-dimensional points generated from ten Gaussian clusters. Fig. 4(b) depicts the result of VBgmmSplit projected on the two first principal axes, and histograms of the number of components found by MMLgmm and VBmfa for 50 runs. MMLgmm did better, using eight or nine components, while VBmfa found four or five components most of the times.

To test the effect of separation of the clusters on the performance of the algorithms, we conducted a series of tests using artificial data with varying degree of separation [26] among the clusters. The degree c of separation for a data set generated from a Gaussian mixture (1) means that for each pair of components (i, j) it holds that

$$||\mu_i - \mu_j||^2 \geq c \max_{(i,j)} \left\{ \mathrm{tr}\left\{T_i^{-1}\right\}, \mathrm{tr}\left\{T_j^{-1}\right\} \right\}. \tag{42}$$

For each value of $c \in \{1, 1.5, 2, 2.5, 3\}$, a data set with 1000 ten-dimensional points was created by sampling from a ten-component Gaussian mixture with equal mixing weights. For each covariance matrix, we constrained the ratio of the largest eigenvalue to its smallest eigenvalue to be less than ten. To check the dependence of the training algorithms on the initial conditions, we repeated the clustering of each data set 20 times. We present the average estimated number of components for each data set in Table I. During training with VBgmm, we set V equal to the covariance matrix of the data set. We found this algorithm to be the most affected by the separation of the clusters.

TABLE I
AVERAGE ESTIMATED NUMBER OF COMPONENTS AND THE STANDARD DEVIATION IN PARENTHESES FOR SEVERAL VALUES OF THE DEGREE OF SEPARATION c

c	VBgmmSplit	VBgmm	VBmfa	MMLgmm
1.0	8	25.1 (2.0)	6.6 (0.8)	8.2 (0.9)
1.5	9	18.7 (2.1)	7.1 (0.7)	8.8 (0.3)
2.0	10	11.6 (0.8)	7.7 (1.3)	9.9 (0.3)
2.5	10	11.6 (0.7)	8.2 (0.8)	9.8 (0.4)
3.0	10	9.2 (0.5)	6.8 (0.6)	10.0 (0.0)

The performance of the algorithms was also tested on real data. More specifically, we applied VBgmmSplit to a set of handwritten digits [27] [see Fig. 5(a) for some examples]. Each digit is an image of 8 × 8 pixels, with 256 grayscale intensities. We created a data set using 700 cases for each of the digits 0–4, and clustered the data with a single mixture model. The data set was preprocessed, so that the mean was zero and the variance unit in each dimension. The mean vectors of the components found by VBgmmSplit are illustrated in Fig. 5(b). After fitting a single mixture on the data with each algorithm, we compared them in terms of the "classification" error using previously unseen test cases (200 for each digit). To compute this error, we assigned each training case to the component with maximum responsibility (posterior). After this hard clustering, we labeled each component with the class of the majority of its data. In order to classify a test case, its responsibilities with respect to each component were computed, and the class label of the component with maximum responsibility was assigned to it. VBgmmSplit provided a solution with 14 components and classification error

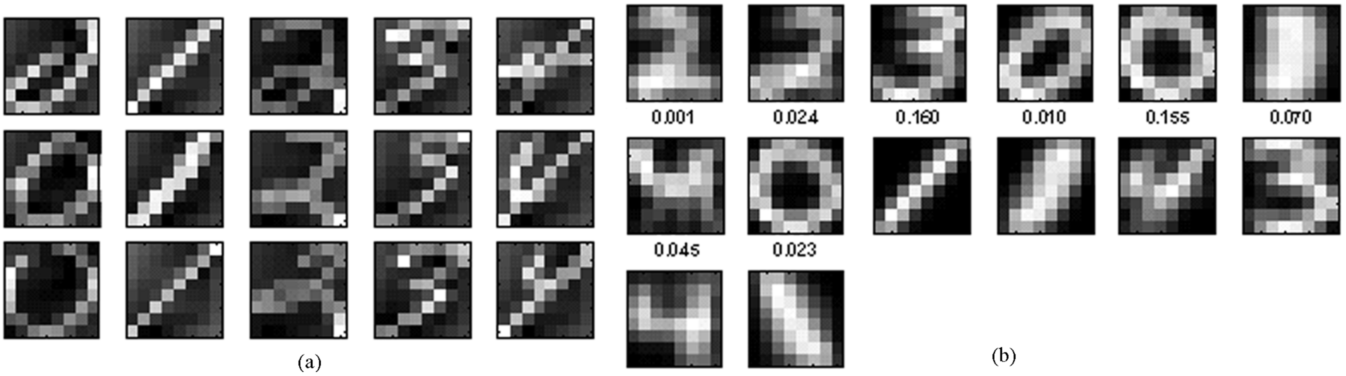


Fig. 5. (a) Some examples of the digits data set. (b) Mean vectors for each component of the mixture fitted with VBgmmSplit. Above each mean the mixing weight of the component is displayed.

TABLE II
EXPECTED NUMBER OF COMPONENTS AND CLASSIFICATION ERROR (STANDARD DEVIATION IN PARENTHESIS) OF THE VBGMM METHOD FOR THE DIGITS DATA. THE RESULTS OBTAINED USING A WISHART PRIOR WITH SCALE MATRIX $\beta\mathcal{I}$

β	components	% error	β	components	% error
1	50.0 (0.0)	5.0 (1.0)	60	9.4 (1.5)	4.8 (3.9)
10	47.8 (1.6)	2.9 (1.3)	70	7.8 (0.8)	10.0 (4.3)
20	40.8 (1.6)	2.5 (1.0)	80	6.8 (0.8)	13.2 (8.4)
30	27.8 (2.7)	3.6 (3.6)	90	6.8 (0.4)	21.8 (10.5)
40	17.0 (1.7)	2.0 (0.6)	100	6.0 (0.0)	28.7 (4.1)
50	10.8 (0.8)	6.2 (4.2)	200	3.2 (0.4)	50.6 (6.1)

1.9%. The VBgmm method was also tested using a Wishart prior with various scale matrices, and the results are summarized in Table II. The expectations were computed after five trials for each scale matrix. We trained a mixture with 50 components initially, and the best solution had average classification error 2.0% and the average number of components was 17.

Due to the high dimensionality and sparsity of the data set, the MMLgmm was able to provide acceptable results under the assumption of a common full covariance for all mixture components. For five trials, the average classification error was 11.9% and the average number of components was 19.8. When a separate diagonal covariance for each component was assumed, the average error on five runs was 35.9% and the average number of components was 7. In both experiments, the initial mixture had 50 components.

VBmfa was also used to fit a mixture of factor analyzers. It must be noted that in this data set the VBmfa method exhibited sensitivity on the value of maximum intrinsic dimensionality that had to be specified in advance. To obtain results comparable with VBgmmSplit in terms of execution time and number of components, the maximum intrinsic dimensionality of each component was set to ten after experimentation. For five trials, the average error was 11.4% and the average number of components was 14. The result was not satisfactory, as some of the components were responsible for data of more than one class. The error could be further improved if we decreased the maximum intrinsic dimensionality of the factor analyzers, although in this way the number of components and the execution time would increase.

VII. CONCLUSION

We have proposed an incremental approach for model selection and learning of Gaussian mixtures. The method improves the Bayesian approach proposed in [19], which provides an elegant mechanism to allow for the competition among components residing in the same region of the data space and the elimination of the redundant ones. However, apart from the initialization problem, this approach exhibits sensitivity on the parameters of the prior of the precision (inverse covariance) matrix. As we have shown, it is difficult to specify appropriate values for these parameters, especially in the case of problems containing clusters of different sizes.

The proposed method ameliorates these difficulties by sequentially adding components to the mixture using a Bayesian splitting test procedure where a component is split into two components and then variational update equations are applied only to the parameters of the two components. As a result, either both components are retained in the model or one of them is found to be redundant and is eliminated. Our approach allows for the specification of a different *local precision prior* for each splitting test, whose parameters can be specified by taking into account the characteristics of the precision matrix of the component that is tested for splitting. In addition, the proposed method is deterministic and does not depend on parameter initialization as it happens with the other methods. As indicated by the experimental results and comparisons with two other powerful methods, the proposed approach seems to adequately address the model selection problem in Gaussian mixtures.

Future work will focus on refining the method by elaborating on and testing two issues. The first is to explore alternative ways to specify the local precision prior. Apart from this, it is possible to perform multiple splitting tests for the same component, with the scale matrix gradually increasing in order to obtain a measure of robustness for the splitting test. The second issue is to consider alternative ways to initialize the means of the two subcomponents during split (e.g., in [13] split direction is selected randomly). Also, it is possible to perform multiple splitting tests for a specific component with different subcomponent initializations each time. Finally, other issues to be considered are the scalability of the method, the possibility to concurrently perform splitting tests for many components, and its use in several application domains (e.g., image segmentation). It is also possible to examine the applicability of this approach for supervised training using mixture models [28].

References

[1] G. McLachlan and D. Peel, *Finite Mixture Models*. New York: Wiley, 2000.

[2] H. Hajji, "Statistical analysis of network traffic for adaptive faults detection," *IEEE Trans. Neural Netw.*, vol. 16, no. 5, pp. 1053–1063, Sep. 2005.

[3] L. A. G. N. P. Blekas, K. and I. E. Lagaris, "A spatially constrained mixture model for image segmentation," *IEEE Trans. Neural Netw.*, vol. 16, no. 2, pp. 494–498, Mar. 2005.

[4] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood estimation from incomplete data via the EM algorithm," *J. Roy. Statist. Soc., Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[5] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York: Wiley, 1997.

[6] P. Smyth, "Model selection for probabilistic clustering using cross-validated likelihood," *Statist. Comput.*, vol. 10, no. 1, pp. 63–72, 2000.

[7] T. Lange, M. L. Braun, V. Roth, and J. M. Buhmann, "Stability-based model selection," in *Advances in Neural Information Processing Systems 15*. Cambridge, MA: MIT Press, 2002.

[8] M. Hareven and V. L. Brailovsky, "Probabilistic validation approach for clustering," *Pattern Recognit. Lett.*, vol. 16, pp. 1189–1196, 1995.

[9] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 381–396, Mar. 2002.

[10] Y. M. Cheung, "Maximum weighted likelihood via rival penalized EM for density mixture clustering with automatic model selection," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 750–761, Jun. 2005.

[11] S. Richardson and P. Green, "On Bayesian analysis of mixtures with unknown number of components," *J. Roy. Statist. Soc., Ser. B*, vol. 59, no. 4, pp. 731–792, 1997.

[12] T. Jaakkola, "Tutorial on variational approximation methods," in *Advanced Mean Field Methods: Theory and Practice*, M. Opper and D. Saad, Eds. Cambridge, MA: MIT Press, 2000, pp. 129–160.

[13] Z. Ghahramani and M. Beal, "Variational inference for Bayesian mixtures of factor analysers," in *Advances in Neural Information Processing Systems 12*. Cambridge, MA: MIT Press, 2000, pp. 449–455.

[14] ——, "Propagation algorithms for variational Bayesian learning," in *Advances in Neural Information Processing Systems 13*. Cambridge, MA: MIT Press, 2001, pp. 507–513.

[15] R. Neal and G. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*, M. I. Jordan, Ed. Norwell, MA: Kluwer, 1998, pp. 355–370.

[16] M. Sato, "On-line model selection based on the variational Bayes," *Neural Comput.*, vol. 13, no. 7, pp. 1649–1681, 2001.

[17] H. Attias, "Inferring parameters and structure of latent variable models by variational Bayes," in *Proc. 15th Conf. Uncertainty Artif. Intel.*, 1999, pp. 21–30.

[18] ——, "A variational Bayesian framework for graphical models," in *Advances in Neural Information Processing Systems 12*. Cambridge, MA: MIT Press, 2000, pp. 209–215.

[19] A. Corduneanu and C. M. Bishop, "Variational Bayesian model selection for mixture distributions," in *Artificial Intelligence and Statistics 2001*. San Mateo, CA: Morgan Kaufmann, 2001, pp. 27–34.

[20] N. Ueda and Z. Ghahramani, "Bayesian model search for mixture models based on optimizing variational bounds," *Neural Netw.*, vol. 15, no. 10, pp. 1223–1241, 2002.

[21] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton, "SMEM algorithm for mixture models," *Neural Comput.*, vol. 12, no. 9, pp. 2109–2128, 2000.

[22] S. Kotz, N. Balakrishnan, and N. L. Johnson, *Continuous multivariate distributions*, 2nd ed. New York: Wiley, 2000, vol. 1.

[23] G. B. Arfken and H. J. Weber, *Mathematical Methods For Physicists*, 6th ed. New York: Elsevier, 2005.

[24] K. Fukumizu, S. Akaho, and S. Amari, "Critical lines in symmetry of mixture models and its application to component splitting," in *Advances in Neural Information Processing Systems 15*. Cambridge, MA: MIT Press, 2003, pp. 865–872.

[25] Z. Zhang, C. Chen, J. Sun, and K. L. Chan, "EM algorithms for Gaussian mixtures with split-and-merge operation," *Pattern Recognit.*, vol. 36, no. 9, pp. 1973–1983, 2003.

[26] S. Dasgupta, "Learning mixtures of Gaussians," in *Proc. 40th Symp. Found. Comput. Sci.*, 1999, pp. 634–644.

[27] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, May 1994.

[28] C. Constantinopoulos and A. Likas, "An incremental training method for the probabilistic RBF network," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 966–974, Jul. 2006.

Constantinos Constantinopoulos received the B.Sc., M.Sc., and Ph.D. degrees from the Computer Science Department, University of Ioannina, Ioannina, Greece, in 2000, 2002, and 2006, respectively.

He is currently a Research Associate at the Computer Science Department, University of Ioannina. His research interests include neural networks, machine learning, and Bayesian reasoning.

Aristidis Likas (S'91–M'96–SM'03) received the Diploma degree in electrical engineering and the Ph.D. degree in electrical and computer engineering from the National Technical University of Athens, Greece, in 1990 and 1994, respectively.

Since 1996, he has been with the Department of Computer Science, University of Ioannina, Greece, where he is currently an Associate Professor. His research interests include neural networks, machine learning, statistical signal processing, and bioinformatics.