Variational Learning for Gaussian Mixture Models

Nikolaos Nasios and Adrian G. Bors, *Senior Member, IEEE*

***Abstract*—This paper proposes a joint maximum likelihood and Bayesian methodology for estimating Gaussian mixture models. In Bayesian inference, the distributions of parameters are modeled, characterized by hyperparameters. In the case of Gaussian mixtures, the distributions of parameters are considered as Gaussian for the mean, Wishart for the covariance, and Dirichlet for the mixing probability. The learning task consists of estimating the hyperparameters characterizing these distributions. The integration in the parameter space is decoupled using an unsupervised variational methodology entitled variational expectation–maximization (VEM). This paper introduces a hyperparameter initialization procedure for the training algorithm. In the first stage, distributions of parameters resulting from successive runs of the expectation–maximization algorithm are formed. Afterward, maximum-likelihood estimators are applied to find appropriate initial values for the hyperparameters. The proposed initialization provides faster convergence, more accurate hyperparameter estimates, and better generalization for the VEM training algorithm. The proposed methodology is applied in blind signal detection and in color image segmentation.**

***Index Terms*—Bayesian inference, expectation–maximization algorithm, Gaussian mixtures, maximum log-likelihood estimation, variational training.**

I. Introduction

THIS PAPER develops a new statistically based training methodology for Gaussian mixtures. Gaussian mixtures constitute a widely used model in science and technology due to its modeling and approximation properties [1]–[3]. Gaussian mixtures have been embedded into radial basis function (RBF) neural networks and used in several applications [4]–[8]. RBF networks can be trained in two stages, namely: 1) unsupervised training for the hidden unit parameters and 2) supervised training for the output parameters [5], [9]. Learning mixtures of Gaussians is unsupervised in its most general form and corresponds to the first training stage in RBF networks. Well-defined statistical properties for mixtures of Gaussians determine the use of a training algorithm relying onto statistical estimation. In statistical parameter estimation, we can identify three different approaches, namely: 1) maximum likelihood; 2) maximum *a posteriori* (MAP); and 3) Bayesian inference [10], [11].

Maximum-likelihood algorithms estimate the model parameters such that it maximizes a likelihood function. The best known algorithm that finds maximum-likelihood estimates in parametric models for incomplete data is the expectation–maximization (EM) algorithm [3], [12], [13]. The EM is an iterative two-step algorithm. The E-step calculates the conditional expectation of the complete data log-likelihood given the observed data and parameter estimates. The M-step finds the parameter estimates that maximize the complete data log-likelihood from the E-step. The EM algorithm has been used for training RBF networks [7], [8] and for parameter estimation of Gaussian mixtures [6], [14]. Extensions of the EM algorithm for pattern recognition include a split and merge methodology [15]–[17]. MAP estimation consists of finding the parameters that correspond to the location of the MAP density function and is used when this density cannot be computed directly [15].

Unlike the MAP method, where we estimate a set of characteristic parameters, Bayesian inference aims to model entirely the *a posteriori* parameter probability distribution. Bayesian inference assumes that the parameters are not uniquely described, but instead are modeled by probability density functions (pdfs) [10]. This introduces an additional set of hyperparameters modeling distributions of parameters. The *a posteriori* probability of a data set is obtained by integrating over the probability distribution of the parameters. Bayesian approaches do not suffer from overfitting and have good generalization capabilities [10], [11], [18]–[24]. Prior knowledge can be easily incorporated and uncertainty manipulated in a consistent manner. Another advantage of Bayesian methods over maximum-likelihood ones is that they achieve models of lower complexity. Main tasks in algorithms using Bayesian inference consist of defining proper distribution functions for modeling the parameters as well as in deriving the appropriate algorithms for integrating over the entire parameter space. The latter task can be computationally heavy and may result in intractable integrals. The following Bayesian approaches, which integrate over the entire parameter distributions, have been adopted to date: the Laplacian method [22], Markov chain Monte Carlo (MCMC) [24], and variational learning [18], [19], [25]–[29]. The Laplacian approach employs an approximation, using Taylor expansion, for the expression of the integrals [22]. However, in high dimensions, this approximation becomes computationally expensive and can provide poor approximation results. MCMCs are a class of algorithms for sampling from probability distributions based on constructing a Markov chain that has the desired distribution as its stationary distribution [24]. A selection mechanism can be used in order to validate a certain sample draw. A reversible jump MCMC was used for the Bayesian learning of Gaussian mixtures [30] and RBF networks [31]. The disadvantage of MCMC methods is the difficulty to select appropriate distributions and to design sampling techniques in order to draw appropriate data samples. Due to their stochastic nature, MCMC algorithms may require a long time to converge [24]. The variational Bayesian (VB) approach consists of converting the complex inferring

Manuscript received February 10, 2005; revised October 25, 2005 and November 2, 2005. This paper was recommended by Associate Editor E. Santos.

The authors are with the Department of Computer Science, University of York, YO10 5DD York, U.K. (e-mail: nn@cs.york.ac.uk; adrian.bors@cs.york.ac.uk).

Digital Object Identifier 10.1109/TSMCB.2006.872273

problem into a set of simpler calculations, characterized by decoupling the degrees of freedom in the original problem [20]. This decoupling is achieved at the cost of including additional parameters that must fit the given model and data. Variational training has been applied in the Bayesian estimation of time series [32], [33], while mixed variational and MCMC have been used in [34].

Variational algorithms are guaranteed to provide a lower bound on the approximation error [18]–[20], [25], [26], [28], [35]. The most used graphical models trained by variational learning include mixtures of Gaussians [27], [37], mean field methods [20], independent component analysis [23], [36], mixtures of factor analyzers [35], linear models [29], [33], and mixtures of products of Dirichlet and multinomial distributions [28].

In most approaches, parameter initialization is considered random, defined in a given range of values. However, an inappropriate initialization could lead to overfitting and poor generalization [33], [36], [38]. In [4], it was shown how the RBF network performance is improved by using a suitable initialization. k-means clustering was used in [4] for initializing the hidden unit centers of the RBF network, implemented by Gaussian means. The posterior of a set of parameters, modeling an autoregressive model, is initialized using a maximum-likelihood approximation in [33]. In our study, we employ a maximum log-likelihood estimation procedure for hyperparameter initialization for variational mixtures of Gaussians. The proposed variational method is called the variational expectation–maximization (VEM) algorithm. The initialization of the proposed approach relies on the dual EM algorithm. The parameters estimated from multiple runs of a first-stage EM are used as inputs for the second-stage EM. Data samples are used for initializing the second EM algorithm whose aim is to provide appropriate hyperparameter initialization [38], [39]. The number of mixture components is chosen according to the Bayesian information criterion (BIC) [27], [33], [40] that corresponds to the negative of the minimum description length criterion (MDL) [41]. The VEM algorithm is applied to source separation in phase- and amplitude-modulated signals when assuming intersymbol and interchannel interference, and in color image segmentation.

The variational inference methodology is introduced in Section II. Section III provides the selection of the probability distributions that are used for modeling the parameters in the context of Gaussian mixtures. In Section IV, the dual EM algorithm and the maximum log-likelihood initialization procedure are outlined. The variational algorithm is described in Section V. In Section VI, we provide a set of experimental results for the proposed methodology, while in Section VII the conclusions of the present study are drawn.

II. Variational Inference Framework

In signal processing or pattern classification problems, we are usually provided only with a subset of data to be modeled. The data samples that are provided, $\mathbf{x} = \{\mathbf{x}_i, i = 1, \ldots, M\}$, where M is the number of data samples, correspond to the observed data, while other data that are consistent with the underlying probability are unseen or hidden. One of the aims in parametric estimation is to represent both the observed and the hidden data by means of a pdf characterized by a set of parameters. The set of parameters is found by means of a training algorithm [2]. The ability of an algorithm trained on incomplete data to model the hidden data is called generalization. We want to estimate the probability $p(\mathbf{x}, \mathbf{z}, \mathbf{t})$, where $\mathbf{x}$ is the given data, $\mathbf{z}$ represents the unseen data, and $\mathbf{t}$ represents the modeling parameters. For the given data, we can define the marginal log-likelihood or the evidence as

$$L_{\mathbf{t}}(\mathbf{x}) = \log \iint p(\mathbf{x}, \mathbf{z}, \mathbf{t}) d\mathbf{z} d\mathbf{t}. \tag{1}$$

Most machine learning algorithms estimate the model parameters by maximizing the likelihood of the given data set. Often, the training algorithm may get stuck in a local minima or may provide a biased solution to the likelihood maximization. Overfitting and poor generalization are other causes of concern when using maximum-likelihood techniques [18], [22], [27], [32], [36]. The most known algorithm implementing the maximization of the log-likelihood is the EM algorithm [3], [6]–[8], [12], [13], [15].

In Bayesian learning, direct estimation of the model parameters is replaced by the inference of a set of distributions modeling those parameters. Bayesian learning is used to approximate posterior densities using distributions of parameters. Let us consider a variational *a posteriori* probability distribution $q(\mathbf{z}, \mathbf{t}|\mathbf{x}, \theta)$, approximating the true posterior $p(\mathbf{z}, \mathbf{t}|\mathbf{x})$, where θ refers to a set of hyperparameters that model distributions of parameters. By using Jensen's inequality, from the true log-likelihood defined in (1), we can derive [18], [27]

$$\begin{aligned} L_{\mathbf{t}}(\mathbf{x}) &= \log \iint q(\mathbf{z}, \mathbf{t}|\mathbf{x}, \theta) \frac{p(\mathbf{x}, \mathbf{z}, \mathbf{t})}{q(\mathbf{z}, \mathbf{t}|\mathbf{x}, \theta)} d\mathbf{z} d\mathbf{t} \\ &\geq \iint q(\mathbf{z}, \mathbf{t}|\mathbf{x}, \theta) \log \frac{p(\mathbf{x}, \mathbf{z}, \mathbf{t})}{q(\mathbf{z}, \mathbf{t}|\mathbf{x}, \theta)} d\mathbf{z} d\mathbf{t} \\ &= F_\theta(\mathbf{x}) \end{aligned} \tag{2}$$

where $F_\theta(\mathbf{x})$ represents the variational free energy. We can observe that $F_\theta(\mathbf{x})$ consists of a lower bound for the marginal log-likelihood.

According to Bayes theorem, we have

$$p(\mathbf{x}, \mathbf{z}|\mathbf{t}) = \frac{p(\mathbf{x}, \mathbf{z}, \mathbf{t})}{p(\mathbf{z}, \mathbf{t}|\mathbf{x})}. \tag{3}$$

After applying the logarithm and multiplying both the numerator and the denominator with $q(\mathbf{z}, \mathbf{t}|\mathbf{x}, \theta)$, we can define the variational log-likelihood $L_\theta(\mathbf{x})$ as

$$\begin{aligned} L_\theta(\mathbf{x}) = &\iint q(\mathbf{z}, \mathbf{t}|\mathbf{x}, \theta) \log p(\mathbf{x}, \mathbf{z}, \mathbf{t}) d\mathbf{z} d\mathbf{t} \\ &- \iint q(\mathbf{z}, \mathbf{t}|\mathbf{x}, \theta) \log p(\mathbf{z}, \mathbf{t}|\mathbf{x}) d\mathbf{z} d\mathbf{t}. \end{aligned} \tag{4}$$

We can rewrite the above expression of the log-likelihood function as

$$\begin{aligned} L_\theta(\mathbf{x}) &= F_\theta(\mathbf{x}) + \mathrm{KL}\left(q(\mathbf{x}), p(\mathbf{x})\right) \\ &= \iint q(\mathbf{z},\mathbf{t}|\mathbf{x},\theta) \log \frac{p(\mathbf{x},\mathbf{z},\mathbf{t})}{q(\mathbf{z},\mathbf{t}|\mathbf{x},\theta)} d\mathbf{z}d\mathbf{t} \\ &\quad + \iint q(\mathbf{z},\mathbf{t}|\mathbf{x},\theta) \log \frac{q(\mathbf{z},\mathbf{t}|\mathbf{x},\theta)}{p(\mathbf{z},\mathbf{t}|\mathbf{x})} d\mathbf{z}d\mathbf{t}. \end{aligned} \quad (5)$$

The first term $F_\theta(\mathbf{x})$ consists of the variational free energy, while the second term $\mathrm{KL}(q(\mathbf{x}), p(\mathbf{x}))$ is the Kullback–Leibler (KL) divergence between the variational *a posteriori* probability density $q(\mathbf{z},\mathbf{t}|\mathbf{x},\theta)$ and the true posterior $p(\mathbf{z},\mathbf{t}|\mathbf{x})$, i.e.,

$$\mathrm{KL}\left(q(\mathbf{x}), p(\mathbf{x})\right) = \iint q(\mathbf{z},\mathbf{t}|\mathbf{x},\theta) \log \frac{q(\mathbf{z},\mathbf{t}|\mathbf{x},\theta)}{p(\mathbf{z},\mathbf{t}|\mathbf{x})} d\mathbf{z}d\mathbf{t}. \quad (6)$$

We can observe from the inequality (2) that the variational free energy is a lower bound on the true log-likelihood function. The equality occurs if the approximate posterior is equal to the true posterior. VB learning aims to maximize this lower bound and therefore brings the approximate posterior as close as possible to the true posterior. What a learning algorithm needs to do is to find an appropriate variational posterior $q(\mathbf{z},\mathbf{t}|\mathbf{x},\theta)$ and to estimate its parameters θ such that it maximizes $L_\theta(\mathbf{x})$. It can be observed from (6) that $\mathrm{KL}(q(\mathbf{x}), p(\mathbf{x}))$ is a similarity measure between the true *a posteriori* and its variational approximate $q(\mathbf{z},\mathbf{t}|\mathbf{x},\theta)$. When the two distributions are identical, then the KL divergence is zero. This divergence represents an error function, expressing the difference between $L_\theta(\mathbf{x})$ and its bound $F_\theta(\mathbf{x})$.

III. Bayesian Modeling of Mixtures of Gaussians

Due to their excellent approximation properties, mixtures of Gaussians have been used in various applications [4]–[7], [9], [14], [15], [17], [21], [36], [39], [46]. A mixture of Gaussians approximates a given probability density as

$$p(\mathbf{x}) = \sum_{i=1}^{N} \frac{\alpha_i}{\sqrt{(2\pi)^d |\Sigma_i|}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu_i)^T \mathbf{\Sigma}_i^{-1}(\mathbf{x}-\mu_i)\right] \quad (7)$$

where d is the data dimension, $\mathbf{t} = \{\alpha, \mathbf{\Sigma}, \mu\}$ represents a set of parameters, and N corresponds to the number of components. A Gaussian mixture is implemented by an RBF network, where the hidden unit activation function is Gaussian [4]–[6], [31]. The number of mixture components corresponds to the number of hidden units required by the RBF network. In order to achieve probability normalization, in Gaussian mixtures we impose a constraint onto the mixing parameters

$$\sum_{i=1}^{N} \alpha_i = 1. \quad (8)$$

In a probability estimation problem, we want to estimate all the parameters such that our model approximates as closely as possible the true probability. Various algorithms can be used for finding the Gaussian mixture parameters. Supervised training algorithms include gradient descent [4], while for unsupervised classification we can mention robust learning vector quantization approaches [5], [9], stochastic algorithms such as MCMC [31], as well as the EM algorithm [7], [15].

As described in the previous section, in the VB framework, we replace the true *a posteriori* distribution $p(\mathbf{z},\mathbf{t}|\mathbf{x})$ with an approximation $q(\mathbf{z},\mathbf{t}|\mathbf{x},\theta)$. In the case of a Gaussian mixture, as in (7), this means to calculate a distribution for each model parameter and to integrate over the entire parameter space as in (4). For the sake of algebraic convenience, we choose the posterior $q(\mathbf{z},\mathbf{t}|\mathbf{x},\theta)$ as the conjugate prior of the given parameterized data distribution $p(\mathbf{x},\mathbf{z},\mathbf{t})$ [27], [29]. This choice ensures that both distributions $p(\mathbf{x},\mathbf{z},\mathbf{t})$ and $q(\mathbf{z},\mathbf{t}|\mathbf{x},\theta)$ belong to the same family of distributions.

In the case of the parameters $\mathbf{t}$, characterizing mixtures of Gaussians, the following conjugate priors are considered: a joint Normal–Wishart distribution for mean and inverse covariance matrix, and a joint Dirichlet distribution for the mixing parameters. We assume that the joint Normal–Wishart distribution can be factorized into a Normal and a Wishart distribution for each component mean and covariance, respectively, as

$$\mathcal{NW}(\mu, \mathbf{\Sigma}|\mathbf{m}, \mathbf{S}, \nu, \beta) = \mathcal{N}(\mu|\mathbf{m}, \beta\mathbf{S})\mathcal{W}(\mathbf{\Sigma}|\nu, \mathbf{S}). \quad (9)$$

The Normal distribution for the mean is $\mathcal{N}(\mu|\mathbf{m}, \beta\mathbf{S})$, where β is a scaling factor and $\mathbf{m}$ is the hypermean of the mean distribution. This distribution is parameterized as

$$\begin{aligned} \mathcal{N}(\mu|\mathbf{m}, \beta\mathbf{S}) \sim & \frac{1}{\sqrt{(2\pi)^d |\beta\mathbf{S}|}} \\ & \times \exp\left[-\frac{1}{2}(\mu-\mathbf{m})^T(\beta\mathbf{S})^{-1}(\mu-\mathbf{m})\right]. \end{aligned} \quad (10)$$

A Wishart distribution $\mathcal{W}(\mathbf{\Sigma}|\nu, \mathbf{S})$ is the conjugate prior for the inverse covariance matrix

$$\mathcal{W}(\mathbf{\Sigma}|\nu, \mathbf{S}) \sim \frac{|\mathbf{S}|^{-\frac{\nu}{2}} |\mathbf{\Sigma}|^{\frac{(\nu-d-1)}{2}}}{2^{\frac{\nu d}{2}} \pi^{\frac{d(d-1)}{4}} \Pi_{k=1}^{d} \Gamma\left(\frac{\nu+1-k}{2}\right)} \exp\left[-\frac{Tr(\mathbf{S}^{-1}\mathbf{\Sigma})}{2}\right] \quad (11)$$

where ν are the degrees of freedom, Tr denotes the trace of the resulting matrix (the sum of the diagonal elements), and $\Gamma(\cdot)$ represents the Gamma function

$$\Gamma(x) = \int_0^\infty \tau^{x-1} \exp(-\tau) d\tau. \quad (12)$$

The mixing probabilities, after considering the normalization condition (8), are modeled as a joint Dirichlet distribution

$$\mathcal{D}(\alpha|\lambda_1, \ldots, \lambda_N) = \frac{\Gamma\left(\sum_{j=1}^{N} \lambda_j\right)}{\Pi_{j=1}^{N} \Gamma(\lambda_j)} \Pi_{i=1}^{N} \alpha_i^{\lambda_i - 1}. \quad (13)$$

As we have observed in (2), we have to find a likelihood function that is as close as it can get to the lower bound provided

by the free energy $F_\theta(\mathbf{x})$. The expression $L_\theta(\mathbf{x})$ from (4) can be seen as a penalized likelihood, where the penalty is KL divergence between the variational posterior and prior distributions. In order to simplify the calculations, we consider that the approximating *a posteriori* distribution can be factorized into a distribution of the hidden variables and a distribution of the parameters. Moreover, we consider that we can factorize the latter distribution into its individual components

$$\begin{aligned} q(\mathbf{z},\mathbf{t}|\mathbf{x},\theta) &= q(\mathbf{z}|\mathbf{x},\theta)q(\mathbf{t}|\mathbf{x},\theta) \\ &= q(\mathbf{z}|\mathbf{x},\theta)\prod_{i=1}^{N}\mathcal{D}(\alpha_i|\lambda_1,\ldots,\lambda_N) \\ &\quad \times \mathcal{W}(\Sigma_i|\nu,\mathbf{S})\mathcal{N}(\mu_i|\mathbf{m},\beta\mathbf{S}) \end{aligned} \tag{14}$$

where i denotes each Gaussian mixture component. This factorization is derived from the mean field analysis [20] and assumes that the hyperparameter posterior distributions are independent of each other and of those of the hidden variables $\mathbf{z}$.

The estimation of the distribution from (14) is done in two steps. In the first step, the *a posteriori* probabilities are evaluated. In the second step, the parameters $\mathbf{t}$ are calculated and used to derive the hyperparameters θ such that they maximize the *a posteriori* probabilities. The two steps are iterated until the variation in the likelihood function $L_\theta(\mathbf{x})$, from one iteration to another, becomes very small. Alternatively, the KL distance can be used for assessing the effectiveness of estimating $q(\mathbf{z},\mathbf{t}|\mathbf{x},\theta)$, by comparing the variational likelihood $L_\theta(\mathbf{x})$ from one iteration to another. Hyperparameter estimation can be done using the VB algorithm [25], [29], [37].

In our approach, we use an initialization procedure for the hyperparameters based on estimating the maximum log-likelihood from parameter distributions. The model parameters $\mathbf{t}$ can be estimated from the maximum log-likelihood for the given data set. The EM algorithm finds an approximation to the maximum log-likelihood solution. However, the EM algorithm is known to be sensitive to the initialization of its parameters [6]. In variational training, the number of parameters, called hyperparameters, is larger when compared to those required by the EM algorithm, and this makes the issue of initialization even more sensitive [33], [36], [39]. In Section IV, we describe a hyperparameter initialization procedure using the dual EM algorithm.

The activation region of a hidden unit in a Gaussian network is ellipsoidal as modeled by the covariance matrix [4], [5], [7]. In the case of the proposed Bayesian approach, the activation region is given by the integration over the distributions of parameters as provided in (10), (11), and (13). In this context, the hypermean $\mathbf{m}$ can be seen as the center of the activation region corresponding to the mean distribution, while the covariance matrix $\mathbf{S}$ provides its shape in d dimensions. The covariance of the Gaussian function can be represented geometrically with a shape that is identical with that of the mean distribution covariance, while the two shapes differ in size by a scaling factor $\beta < 1$. While $\mathbf{S}$ characterizes the maximum likelihood of the inverse of Wishart distribution, $\beta\mathbf{S}$ shows the precision

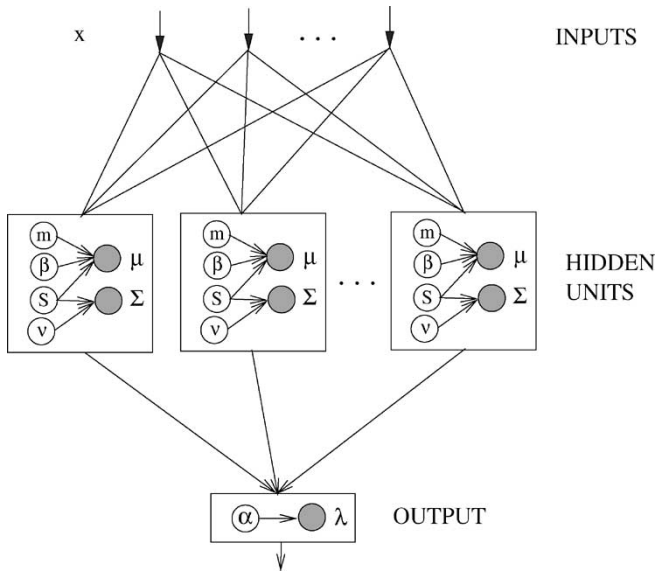


Fig. 1. Topology of the variational Gaussian network.

of estimation for the hypermean and represents geometrically an ellipsoidal activation region [38].

The topology of the variational Gaussian neural network is provided in Fig. 1. The number of Gaussian mixture components is equal to the number of hidden units. The parameters are implicit, while the entire hyperparameter set $\theta = \{\mathbf{m},\mathbf{S},\beta,\nu,\lambda\}$ is estimated using the VEM algorithm, which is described in the following sections. Variational learning is expected to provide better data estimation by taking into account the uncertainty in the parameter estimation. On the other hand, better generalization is expected while maintaining the good localization and modeling capabilities. Variational learning for Gaussian networks is also expected to provide models that require fewer hidden units.

IV. Maximum Log-Likelihood Hyperparameter Initialization

Most iterative algorithms used in pattern recognition and for training neural networks assume a random initialization, in a certain range of values, for the model parameters. However, algorithms such as EM may not converge properly due to an unsuitable initialization. Moreover, the solution at convergence is not optimal and may provide overfitting and poor generalization. When increasing the number of model parameters, we are facing an even more challenging problem in choosing their initial values. In this paper, we adopt a hierarchical maximum log-likelihood initialization approach to the hyperparameter estimation problem. At each level of the hierarchy, we employ algorithms that run onto the model parameters estimated by the algorithm at the previous level. In the first stage, we employ the EM algorithm using a set of random initializations. After several runs of the EM algorithm on the same data set, we form distributions of the estimated model parameters. Eventually, a maximum log-likelihood criterion is applied onto distributions of parameters in order to initialize the hyperparameters for the VEM algorithm.

For the sake of notational simplicity, let us denote

$$D(\mathbf{x};\mu_i,\mathbf{\Sigma}_i)=-\frac{1}{2}(\mathbf{x}-\mu_i)^T\mathbf{\Sigma}_i^{-1}(\mathbf{x}-\mu_i). \tag{15}$$

For the initialization of the hyperparameters corresponding to the mean distribution, we employ a dual-EM algorithm [38]. During the first stage, the EM algorithm for Gaussian mixtures is applied on the given data set $\{\mathbf{x}_i, i=1,\ldots,M\}$ [3], [7], [12], [15]. In the E-step, the *a posteriori* probabilities are estimated as

$$\hat{P}^I_{\mathrm{EM}}(i|\mathbf{x}_j)=\frac{\hat{\alpha_i}|\hat{\mathbf{\Sigma}}_i|^{-\frac{1}{2}}\exp\left[D(\mathbf{x}_j;\hat{\mu}_i,\hat{\mathbf{\Sigma}}_i)\right]}{\sum_{k=1}^{N}\hat{\alpha_k}|\hat{\mathbf{\Sigma}}_k|^{-\frac{1}{2}}\exp\left[D(\mathbf{x}_j;\hat{\mu}_k,\hat{\mathbf{\Sigma}}_k)\right]} \tag{16}$$

where $D(\mathbf{x}_j;\mu_i,\mathbf{\Sigma}_i)$ is provided by (15). In the M-step, we update the parameters of the Gaussian mixture as

$$\hat{\alpha}_i=\frac{\sum_{j=1}^{M}\hat{P}^I_{\mathrm{EM}}(i|\mathbf{x}_j)}{M} \tag{17}$$

$$\hat{\mu}_{i,\mathrm{EM}}=\frac{\sum_{j=1}^{M}\mathbf{x}_j\hat{P}^I_{\mathrm{EM}}(i|\mathbf{x}_j)}{\sum_{j=1}^{M}\hat{P}^I_{\mathrm{EM}}(i|\mathbf{x}_j)} \tag{18}$$

$$\hat{\mathbf{\Sigma}}_i=\frac{\sum_{j=1}^{M}\hat{P}^I_{\mathrm{EM}}(i|\mathbf{x}_j)(\mathbf{x}_j-\hat{\mu}_{i,\mathrm{EM}})(\mathbf{x}_j-\hat{\mu}_{i,\mathrm{EM}})^T}{\sum_{j=1}^{M}\hat{P}^I_{\mathrm{EM}}(i|\mathbf{x}_j)} \tag{19}$$

where M is the number of data samples. We run the EM algorithm L times, by considering various random initializations, each time for the same number of iterations. The total number of estimated Gaussian means, corresponding to the given model, by EM is LN, where N is the number of components. All the parameters estimated in each of the runs are stored, forming data sample distributions for each Gaussian network parameter. We assume that these distributions can be characterized parametrically by a set of hyperparameters. The parametric description of these probabilities is given by (10) for the means μ, by (11) for the covariance matrices $\mathbf{\Sigma}$, and by (13) for the mixing probabilities α.

The next step consists of estimating the hyperparameters characterizing the distributions formed in the previous step. This estimation corresponds to a second level of embedding. The distribution of the means provided by the EM algorithm, as calculated by (18), can be modeled again as a mixture of Gaussians, with an identical number of components N. We apply a second EM algorithm onto the distributions of parameters provided by successive runs of the first EM. In the second EM, we consider randomly selected data samples as the initial starting points for the hypermeans. The second EM equations are similar with those from (16)–(19). The E-step corresponds to the calculation of the *a posteriori* probabilities for parameters

$$\hat{P}^{II}_{\mathrm{EM}}(i|\hat{\mu}_j)=\frac{\hat{a}_i|\hat{\mathbf{S}}_i|^{-\frac{1}{2}}\exp\left[D(\hat{\mu}_j;\hat{\mathbf{m}}_i,\hat{\mathbf{S}}_i)\right]}{\sum_{k=1}^{N}\hat{a}_k|\hat{\mathbf{S}}_k|^{-\frac{1}{2}}\exp\left[D(\hat{\mu}_j;\hat{\mathbf{m}}_k,\hat{\mathbf{S}}_k)\right]} \tag{20}$$

where $D(\hat{\mu}_j;\hat{\mathbf{m}}_i,\hat{\mathbf{S}}_i)$ is provided by (15). In the M-step of the dual EM, we calculate the hyperparameters of the Gaussian network as

$$\hat{a}_i=\frac{\sum_{j=1}^{LN}\hat{P}^{II}_{\mathrm{EM}}(i|\hat{\mu}_j)}{LN} \tag{21}$$

$$\hat{\mathbf{m}}_{i,\mathrm{EM}}=\frac{\sum_{j=1}^{LN}\hat{\mu}_j\hat{P}^{II}_{\mathrm{EM}}(i|\hat{\mu}_j)}{\sum_{j=1}^{LN}\hat{P}^{II}_{\mathrm{EM}}(i|\hat{\mu}_j)} \tag{22}$$

$$\hat{\mathbf{S}}_{i,\mathrm{EM}}=\frac{\sum_{j=1}^{LN}\hat{P}^{II}_{\mathrm{EM}}(i|\hat{\mu}_j)(\hat{\mu}_j-\hat{\mathbf{m}}_{i,\mathrm{EM}})(\hat{\mu}_j-\hat{\mathbf{m}}_{i,\mathrm{EM}})^T}{\sum_{j=1}^{LN}\hat{P}^{II}_{\mathrm{EM}}(i|\hat{\mu}_j)}. \tag{23}$$

The EM algorithm corresponds to the maximum log-likelihood estimate of the given model when representing the given data set [3], [12]. In order to find the number of mixture components, we use the BIC [40] that corresponds to the negative of the MDL criterion [15], [41]. The maximization of the BIC criterion, when varying N, provides the appropriate number of hidden units.

The hyperparameters of the VEM are initialized as follows. The hypermeans $\hat{\mathbf{m}}(0)$ are calculated by averaging the resulting means as they are provided in (22). The hyperparameter β represents a scaling factor of the covariance matrices corresponding to the data distributions $\hat{\mathbf{\Sigma}}$, resulting from (19), to those corresponding to the mean distributions $\hat{\mathbf{S}}_{\mathrm{EM}}$, resulting from (23). This hyperparameter is initialized as the average of the eigenvalues of the matrix $\hat{\mathbf{\Sigma}}\hat{\mathbf{S}}^{-1}_{\mathrm{EM}}$. Consequently, it can be calculated as the value of the trace divided by the dimension of the space

$$\hat{\beta}_i(0)=\frac{\sum_{k=1}^{L}Tr\left(\hat{\mathbf{\Sigma}}_{ik}\hat{\mathbf{S}}^{-1}_{ik,EM}\right)}{dL} \tag{24}$$

where L is the number of runs for the initial EM algorithm.

The Wishart distribution $\mathcal{W}(\mathbf{\Sigma}|\nu,\mathbf{S})$ characterizes the inverse of the covariance matrix. The degrees of freedom are initialized for the VEM algorithm as equal to the number of dimensions

$$\hat{\nu}_i(0)=d. \tag{25}$$

For the initialization of $\hat{\mathbf{S}}$, we consider the distribution of $\hat{\mathbf{\Sigma}}$ resulting from (19). We apply a Cholesky factorization onto the matrices $\hat{\mathbf{\Sigma}}_k$, $k=1,\ldots,L$, resulting from successive runs of the EM algorithm. The Cholesky factorization results into an upper triangular matrix $\mathbf{R}_k$ and a lower triangular matrix $\mathbf{R}_k^T$ such that

$$\hat{\mathbf{\Sigma}}^{-1}_{ik}=\mathbf{R}_{ik}\mathbf{R}^T_{ik}. \tag{26}$$

We generate L sub-Gaussian random vectors $\mathbf{B}$, each of dimension d, whose coordinates are independent random variables $\mathcal{N}(0,1)$. The matrix $\hat{\mathbf{S}}$ will be initialized as [11]

$$\hat{\mathbf{S}}_i(0)=\frac{\sum_{k=1}^{L}\mathbf{R}_{ik}\mathbf{B}_k(\mathbf{B}_k\mathbf{R}_{ik})^T}{L}. \tag{27}$$

For the Dirichlet parameters, we use the maximum log-likelihood estimation for the joint Dirichlet distribution (13) [42]. By applying the logarithm in the distribution from (13) and after differentiating the resulting formula with respect to the parameters λ_i, $i = 1, \ldots, N$, we obtain the iterative expression

$$\psi\left(\lambda_i(t)\right) = \psi\left(\sum_{k=1}^{N} \lambda_k(t-1)\right) + E\left[\log(\hat{\alpha}_i)\right] \tag{28}$$

where t is the iteration number, $E[\log(\hat{\alpha}_i)]$ is the expectation of the logarithm of the mixing probability $\hat{\alpha}_i$, which is derived from the distributions obtained from successive runs of (17), and where $\psi(\cdot)$ is the digamma function (the logarithmic derivative of the Gamma function)

$$\psi(\lambda_i) = \frac{\Gamma'(\lambda_i)}{\Gamma(\lambda_i)} \tag{29}$$

where $\Gamma(\cdot)$ is provided in (12) and $\Gamma'(\lambda_i)$ is its derivative with respect to λ_i. We consider the mean of the mixing probability distribution as an appropriate estimate for $\hat{\alpha}_i$. For estimating the parameter λ_i, we first invert $\psi(\lambda_i)$ from (28) and afterward iteratively update using the Newton–Raphson algorithm as

$$\lambda_i(t) = \lambda_i(t-1) - \frac{\psi\left(\lambda_i(t)\right) - \psi\left(\lambda_i(t-1)\right)}{\psi'\left(\lambda_i(t)\right)} \tag{30}$$

where $\psi'(\lambda_i(t))$ is the derivative of $\psi(\lambda_i(t))$ with respect to $\lambda_i(t)$ and t is the iteration number. Just a few iterations are usually necessary to achieve convergence. The values for the Dirichlet hyperparameters achieved at convergence are denoted by $\hat{\lambda}_i(0)$, $i = 1, \ldots, N$, and correspond to the maximum log-likelihood [42].

V. Variational Expectation-Maximization

The direct implementation of (4) would amount to a very heavy computational task involving multidimensional integration. This complex inferring problem is decomposed in a set of simpler calculations, characterized by decoupling the degrees of freedom in the original problem, as provided by (14). An extension of the EM algorithm called VB has been used to provide a simpler calculation of the expressions from (14) [25], [29], [37]. In our approach, we employ a maximum log-likelihood hyperparameter initialization as described in Section IV. The initial hyperparameter values are denoted by $\theta(0) = \{\hat{\mathbf{m}}(0), \hat{\mathbf{S}}(0), \hat{\beta}(0), \hat{\nu}(0), \hat{\lambda}(0)\}$. The algorithm using this initialization is named VEM.

The VEM is an iterative method that consists of two steps at each iteration, namely: 1) variational expectation (VE) and 2) variational maximization (VM). In the VE-step, we compute the expectation of the *a posteriori* probabilities given the hidden variable distributions $q(\mathbf{z}, \mathbf{t}|\mathbf{x}, \theta)$ and their hyperparameters θ. In the VM-step, we find the hyperparameters θ that maximize the variational log-likelihood given the observed data and their *a posteriori* probabilities. For a mixture of Gaussians, the updating equations employed at each step can be derived as follows.

For the VE-step, we calculate the *a posteriori* probabilities for each data sample $\mathbf{x}_j$, depending on the hyperparameters

$$\begin{aligned}\hat{P}(i|\mathbf{x}_j) &= \exp\left[-\frac{1}{2}\log|\hat{\mathbf{S}}_i| + \frac{1}{2}d\log 2 \right. \\ &\quad + \frac{1}{2}\sum_{k=1}^{d}\psi\left(\frac{\hat{\nu}_i + 1 - k}{2}\right) + \psi(\hat{\lambda}_i) - \psi\left(\sum_{k=1}^{N}\hat{\lambda}_k\right) \\ &\quad \left. - \frac{\hat{\nu}_i}{2}(\mathbf{x}_j - \hat{\mathbf{m}}_i)^T\hat{\beta}_i\hat{\mathbf{S}}_i^{-1}(\mathbf{x}_j - \hat{\mathbf{m}}_i) - \frac{d}{2\hat{\beta}_i}\right]\end{aligned} \tag{31}$$

where $i = 1, \ldots, N$ is the mixture component, d is the dimensionality of the data, $j = 1, \ldots, M$ denotes the data index, and $\psi(\cdot)$ is the digamma function provided in (29). Each *a posteriori* probability shows the activation level of the hidden unit i, when the graphical model, whose topology is shown in Fig. 1, is presented with the data sample $\mathbf{x}_j$.

In the VM-step, we perform an intermediary calculation of the mean parameter, as in the EM algorithm, but considering the *a posteriori* probabilities from (31) as

$$\hat{\mu}_{i,\mathrm{VEM}} = \frac{\sum_{j=1}^{M}\mathbf{x}_j\hat{P}(i|\mathbf{x}_j)}{\sum_{j=1}^{M}\hat{P}(i|\mathbf{x}_j)}. \tag{32}$$

The hyperparameters of the distribution of means are updated as in (33) and (34), shown at the bottom of the page, while the hyperparameters corresponding to the scaling factor and the degrees of freedom are updated as

$$\hat{\beta}_i = \hat{\beta}_i(0) + \sum_{j=1}^{M}\hat{P}(i|\mathbf{x}_j) \tag{35}$$

$$\hat{\nu}_i = \hat{\nu}_i(0) + \sum_{j=1}^{M}\hat{P}(i|\mathbf{x}_j). \tag{36}$$

$$\hat{\mathbf{m}}_i = \frac{\hat{\beta}_i(0)\hat{\mathbf{m}}_i(0) + \sum_{j=1}^{M}\mathbf{x}_j\hat{P}(i|\mathbf{x}_j)}{\hat{\beta}_i(0) + \sum_{j=1}^{M}\hat{P}(i|\mathbf{x}_j)} \tag{33}$$

$$\hat{\mathbf{S}}_i = \hat{\mathbf{S}}_i(0) + \sum_{j=1}^{M}\hat{P}(i|\mathbf{x}_j)(\mathbf{x}_j - \hat{\mu}_{i,\mathrm{VEM}})(\mathbf{x}_j - \hat{\mu}_{i,\mathrm{VEM}})^T + \frac{\hat{\beta}_i(0)(\hat{\mu}_{i,\mathrm{VEM}} - \hat{\mathbf{m}}_i(0))(\hat{\mu}_{i,\mathrm{VEM}} - \hat{\mathbf{m}}_i(0))^T\sum_{j=1}^{M}\hat{P}(i|\mathbf{x}_j)}{\hat{\beta}_i(0) + \sum_{j=1}^{M}\hat{P}(i|\mathbf{x}_j)} \tag{34}$$

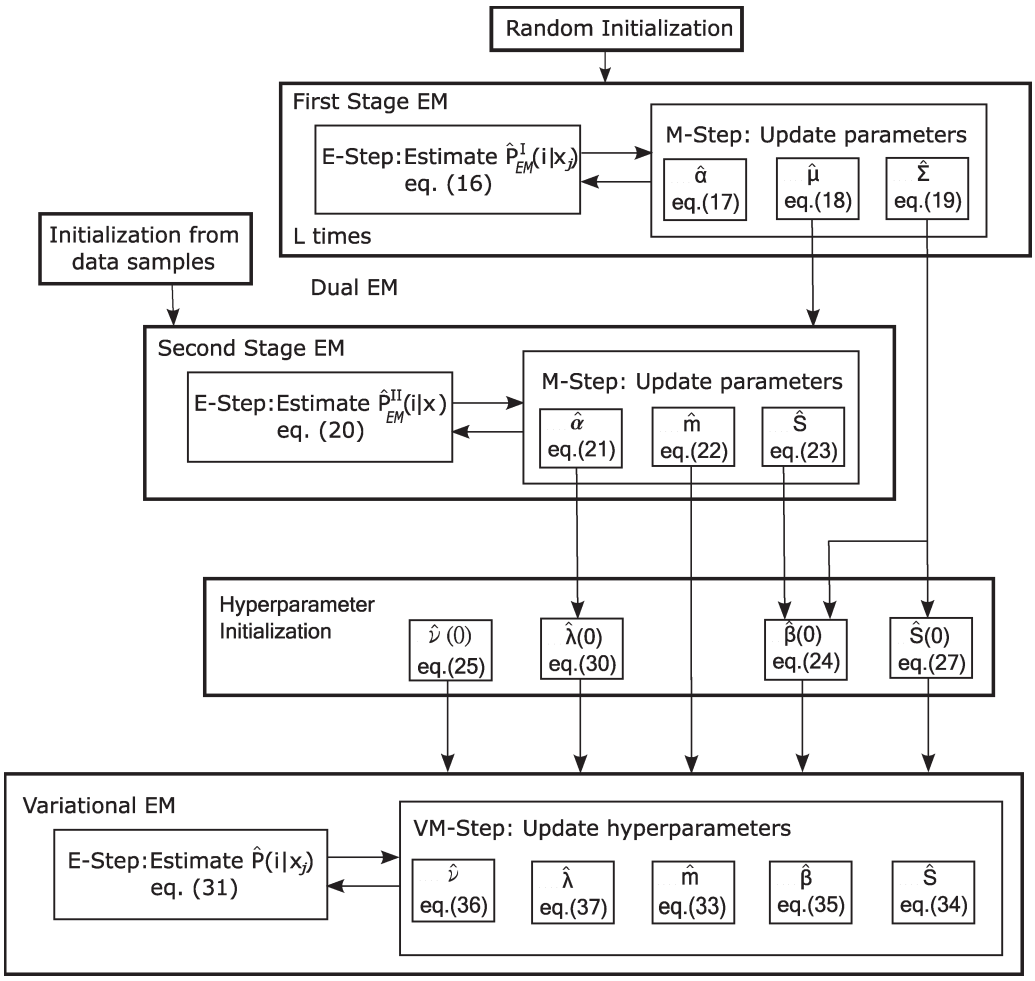


Fig. 2. Flowchart of the VEM training algorithm.

The parameters for Dirichlet distribution are updated as

$$\hat{\lambda}_i = \hat{\lambda}_i(0) + \sum_{j=1}^{M} \hat{P}(i|\mathbf{x}_j). \tag{37}$$

The algorithm is iterative with VE and VM steps alternating at each iteration. The average log-likelihood of the given data set is calculated as

$$L_{\theta,t}(\mathbf{x}) = \frac{1}{M} \sum_{j=1}^{M} \log p_t(\mathbf{x}_j|\theta) \tag{38}$$

where $p_t(\mathbf{x}_j|\theta)$ is the *a posteriori* probability calculated by (7) at iteration t, corresponding to the mixture model resulting from the summation of all component probabilities from (31), calculated for each data sample. The effectiveness of the given data modeling is shown by the increase in the log-likelihood with each iteration. Convergence is achieved when we obtain a small variation in the normalized log-likelihood

$$\frac{L_{\theta,t}(\mathbf{x}) - L_{\theta,t-1}(\mathbf{x})}{L_{\theta,t}(\mathbf{x})} < \varepsilon \tag{39}$$

where ε is a small quantity.

In order to estimate the necessary number of mixture components in the VEM algorithm, corresponding to the number of hidden units from the graphical model, we use the BIC criterion in a similar way as for the EM algorithm in the initialization procedure. This criterion can be rewritten in terms of the KL divergence between the parameter distribution posterior and the parameter distribution prior [33]. The model selection criterion becomes

$$\mathcal{C}_{\mathrm{VEM}}(N) = L_\theta(\mathbf{x}) - \frac{N}{2}\left[3 + d + \frac{d(d+1)}{2}\right] \log M \tag{40}$$

where the first term corresponds to (38) at convergence, N is the number of components, and M of data samples. The number of components N is decided as that corresponding to the largest $\mathcal{C}_{\mathrm{VEM}}(N)$.

The flowchart showing the main steps for the proposed VEM methodology is displayed in Fig. 2. This flowchart shows the initialization of the VEM as done by the dual-EM algorithm. The calculation of each parameter is identified in separate blocks in Fig. 2.

The computational complexity for the EM algorithm is of the order $\mathcal{O}(Md^2N)$ for each iteration [3], where M is the number of data, d is the space dimension, and N is the number of components. This computational complexity per iteration

corresponds to the first initialization stage, where the EM algorithm is run for L iterations. The second EM also requires a similar computational complexity, but for LN data samples. This corresponds to the order of computational complexity of $\mathcal{O}((NL)d^2N) = \mathcal{O}(Ld^2N^2)$ per iteration for a certain number of iterations, until convergence. Usually, it is assumed that $LN \ll M$. The orders of the computational complexity for the VE-step of the VEM algorithm and that for the E-step of the EM algorithm are identical per iteration, as it can be observed from (16) and (31) [25]. The computation required for the VM-step is only marginally higher than that of the M-step of the EM algorithm and has the same order of complexity $\mathcal{O}(Md^2N)$.

VI. Experimental Results

In the following, we describe the experimental results obtained after applying the proposed variational training methodology in two different applications: blind signal detection and color image segmentation.

A. Blind Signal Detection

For this application, we consider artificially generated signals that are used in communication systems. We consider two cases of modulated signals: quadrature amplitude modulated signals (QAM) and phase-shifting keying (PSK)-modulated signals [43]. Each such signal can be represented as a point in the complex number space. For 4-QAM, we have four source signals located at (1, 1), (1, −1), (−1, 1), and (−1, −1). The first component represents the in-phase component, while the second represents the in-quadrature component. Due to various communication channel conditions, such signals are distorted at the receiver forming clusters of data points called constellations. The aim of this application is to identify correctly each one of the four source signals that has been transmitted. In this case, we assume additive Gaussian noise corresponding to a signal-to-noise ratio (SNR) of 8 dB and no interference. The 4-QAM signal constellations are displayed in Fig. 3.

In the second case, we consider 8-PSK signals. In this situation, the eight signal sources are equidistantly located on a circle in the complex signal space. We assume intersymbol and interchannel perturbations together with Gaussian noise. The perturbation channel equations considered in the case of 8-PSK signals are the same with those used in [44], i.e.,

$$\begin{aligned} x_I(t) &= I(t) + 0.2I(t-1) - 0.2Q(t) \\ &\quad - 0.04Q(t-1) + \mathcal{N}(0, 0.11) \\ x_Q(t) &= Q(t) + 0.2Q(t-1) + 0.2I(t) \\ &\quad + 0.04I(t-1) + \mathcal{N}(0, 0.11) \end{aligned} \tag{41}$$

where $(x_I(t),\ x_Q(t))$ forms the in-phase and in-quadrature signal components at time t, and $I(t)$ and $Q(t)$ correspond to the original signal symbols. The Gaussian noise in this case corresponds to an SNR of 22 dB. We consider all possible combinations of interference for $(I,\ Q)$ and generate a total of 64 signals, which are grouped in eight signal constellations corresponding to the distorted signals. For the following experiments, we have generated 960 signals, assuming equal probabilities for all interference cases. These signal constellations are represented in Fig. 4. From this figure, we can observe that signal constellations corresponding to different sources overlap due to interference and noise. Such overlaps among signal constellations cause a challenging problem to a blind source separation algorithm.

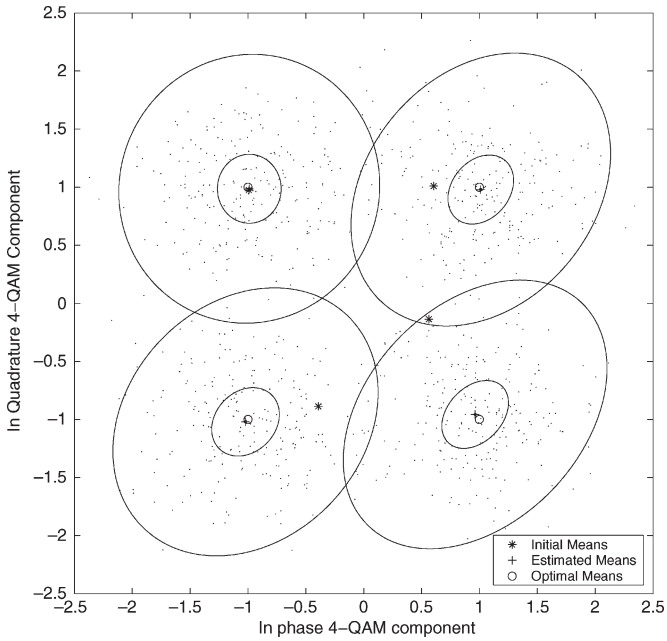


Fig. 3. Blind detection of 4-QAM signals using VEM algorithm.

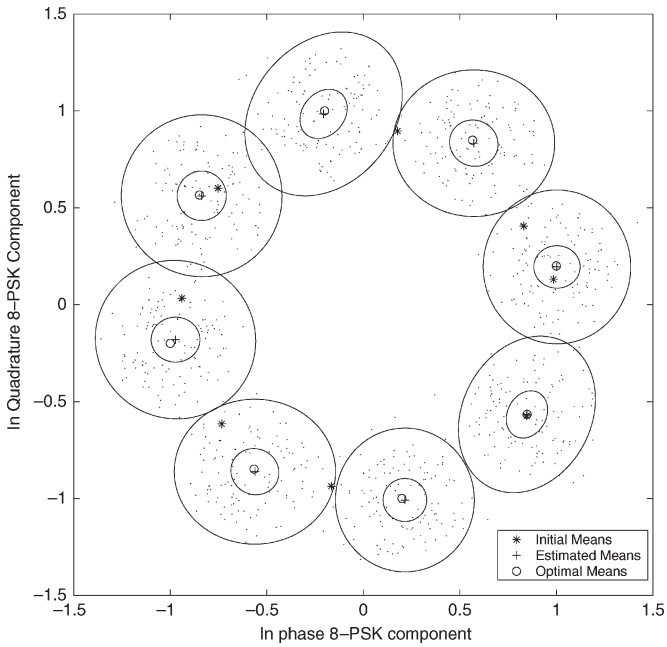


Fig. 4. Blind detection of 8-PSK signals using VEM algorithm.

In this application, blind detection is treated as an unsupervised classification problem. Due to intersymbol, interchannel, and noise interference, each signal constellation is formed from eight clusters, according to (41). We aim to represent these constellations using a variational Gaussian mixture model. The

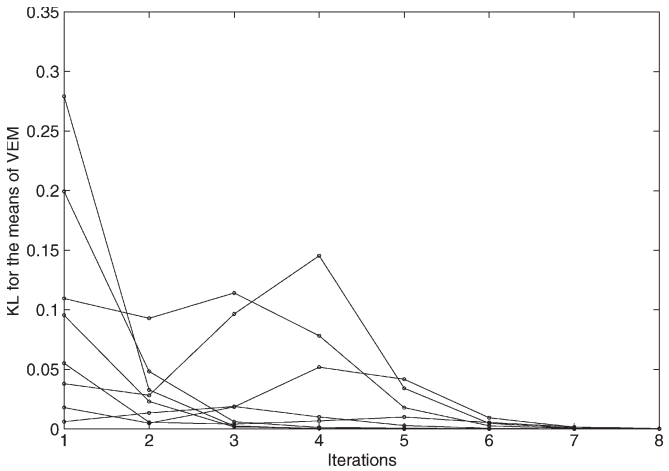


Fig. 5. Convergence illustrated by KL divergence for the mean distributions in 8-PSK data.

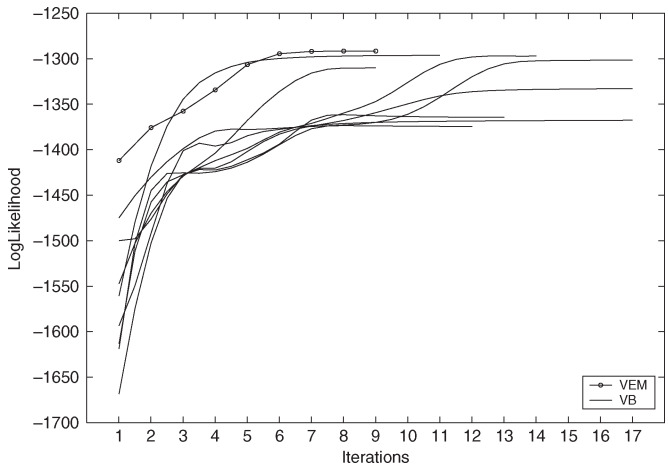


Fig. 6. Convergence of VEM and VB with different initializations.

task in this application is to identify the transmitted symbol for each received signal [44].

We consider a Bayesian statistical model such as that provided in Section III. We apply the VEM algorithm by using the maximum-likelihood estimation initialization described in Section IV. The first stage of the dual EM algorithm was run on the given data samples by using random initialization. Consequently, distributions of parameters corresponding to mixtures of Gaussians were formed. The second stage of the dual EM algorithm was run several times on the distributions of resulting means, while the initialization for the Gaussian means was provided by randomly selected data samples. All the hyperparameters are properly initialized for the VEM algorithm using maximum log-likelihood estimators as presented in Section IV. The location of means provided by the initialization procedure is marked by "$*$", the ideal location for the hypermeans is marked by circles, while the hypermeans estimated by the VEM algorithm are marked by "$+$" in Figs. 3 and 4 for 4-QAM and 8-PSK detection problems, respectively. In these figures, the covariance matrices characterizing the Wishart distribution $\mathbf{S}$ and the distribution of means $\beta\mathbf{S}$ are marked by their corresponding ellipses. Due to the overlap between constellations corresponding to distinct Gaussian components, the initial hypermean estimates provided by the maximum log-likelihood estimation are biased. In Fig. 5, the convergence of KL divergence for the mean distributions is displayed. The KL divergence between the estimated distribution $\hat{\mathcal{N}}_i(\hat{\mathbf{m}}_i, \hat{\mathbf{S}}_i)$ of each mixture component and its corresponding ideal distribution $\mathcal{N}_i(\mu_i, \Sigma_i)$, for $i = 1, \ldots, N$ is calculated as

$$\mathrm{KL}(\hat{\mathcal{N}}_i; \mathcal{N}_i) = \frac{1}{2}\left(\log \frac{|\mathbf{\Sigma}_i|}{|\hat{\mathbf{S}}_i|} + Tr\left(\mathbf{\Sigma}_i^{-1}\hat{\mathbf{S}}_i\right) + (\hat{\mathbf{m}}_i - \mu_i)^T \mathbf{\Sigma}_i^{-1}(\hat{\mathbf{m}}_i - \mu_i) - d \right) \quad (42)$$

where $|\mathbf{\Sigma}_i|$ denotes the determinant of the matrix $\mathbf{\Sigma}_i$ corresponding to the modulated signal and noise. For comparison purposes, we consider the EM algorithm and the VB algorithm. The VB algorithm infers the *a posteriori* probabilities according to the methodology described in Section V but assumes random hyperparameter initialization in an appropriate range of values for each hyperparameter [29], [37]. In Fig. 6, we show the global convergence in terms of the log-likelihood as calculated by (38) for the proposed algorithm and for the VB algorithm. The favorable initialization for the VEM algorithm, achieved by applying maximum log-likelihood techniques on distributions of parameters resulting from several runs of the EM algorithm, is the value corresponding to iteration 1 on the curve marked with circles from Fig. 6. We can observe that the VEM algorithm provides a higher log-likelihood at the convergence than the VB algorithm while eliminating the dependency on the hyperparameter initialization. Table I shows comparative results when applying VEM, VB, and EM algorithms on 4-QAM- and 8-PSK-modulated signals. The errors are measured in terms of global as well as local estimation of individual parameters. For the global estimation, we consider the KL divergence for the posterior distributions and the misclassification error on both training and test sets. The comparison measures consist of the KL of the posterior

$$\mathrm{KL}\left(\hat{P}(i|\mathbf{x}_j); P(i|\mathbf{x}_j)\right) = \frac{1}{MN}\sum_{j=1}^{M}\sum_{i=1}^{N} \hat{P}(i|\mathbf{x}_j) \log \frac{\hat{P}(i|\mathbf{x}_j)}{P(i|\mathbf{x}_j)} \quad (43)$$

where the *a posteriori* probabilities $P(i|\mathbf{x}_j)$ for the given training set can be easily calculated in the case of known signals for 4-QAM and 8-PSK signals. The misclassification error is also calculated for both the training and the test set. For validation purposes, we consider the estimation of individual hyperparameters such as the hypermean bias and the mixing probability bias, both evaluated between the estimated and their ideal values. In Table I, we also show the number of iterations required by each algorithm in order to reach convergence. For estimating the number of components, we have considered the BIC criterion and used the cost functions from (40) for VEM and VB. A similar formula was used for the EM algorithm when assuming its corresponding number of parameters. In Figs. 7 and 8, we display the BIC criterion for 4-QAM and 8-PSK signals, respectively. From these two plots, we can observe that all three algorithms found the right number of

TABLE I
COMPARISON AMONG VEM, VB, AND EM ALGORITHMS IN BLIND SOURCE SEPARATION OF MODULATED SIGNALS

Data	Algorithm	Kullback-Leibler of posterior		Misclassification Error (%)		Bias $\lvert m - \hat{m}\rvert$	Bias Set $\lvert \alpha - \hat{\alpha}\rvert$	No. Iter.
		Train.	Test	Train.	Test			
4 QAM	VEM	0.0255	0.0258	0.63	0.73	0.0319	0.0017	7
	VB	0.1273	0.1330	6.08	6.63	0.2654	0.0294	8
	EM	0.1448	0.1673	11.94	12.18	0.3134	0.0413	24
8 PSK	VEM	0.0257	0.0383	0.73	0.73	0.0147	0.0029	9
	VB	0.0563	0.0735	6.66	6.68	0.1800	0.0169	14
	EM	0.1102	0.1458	13.05	13.28	0.4047	0.0332	34

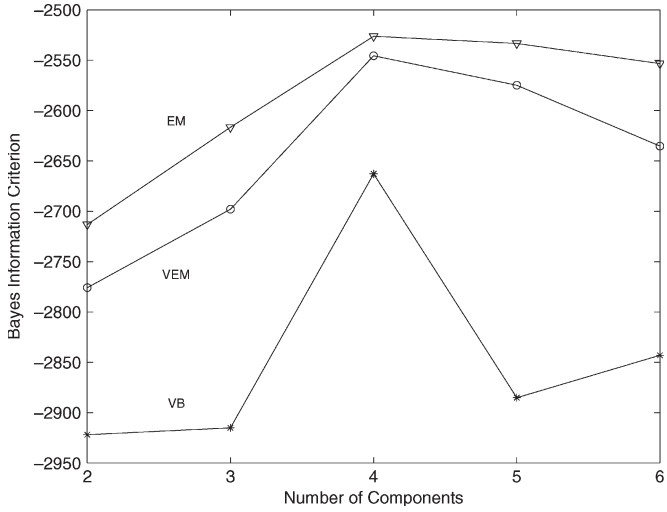


Fig. 7. BIC for 4-QAM.

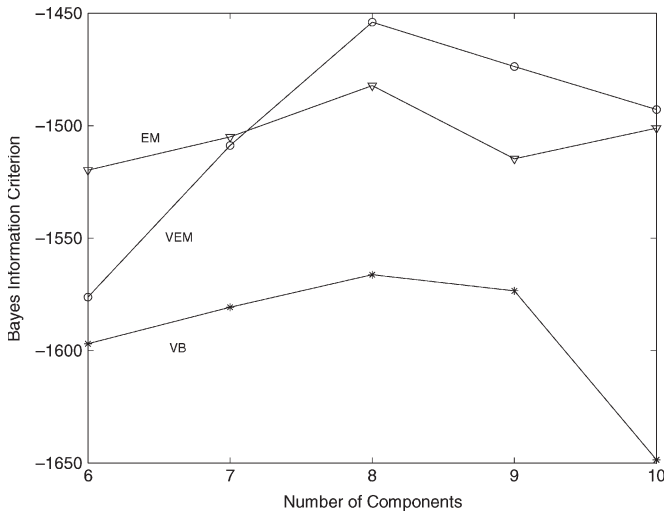


Fig. 8. BIC for 8-PSK.

components as four in 4-QAM and eight in 8-PSK-modulated signals. However, the maxima in the BIC criterion are better defined for the VB algorithm in the case of 4-QAM signals and for the VEM in the case of 8-PSK signals. From the experimental results, we conclude that the VEM algorithm provides a better estimation of the model parameters, eventually achieving better source separation when compared with the other estimation algorithms.

According to the analysis from the previous section, the difference in the order of computational complexity of the VEM algorithm with respect to the EM, as well as with the VB algorithm, if assuming an identical number of iterations, consists only of the computational complexity of the initialization procedure. That would require $\mathcal{O}(Md^2N)$ per iteration, for L iterations in the first EM stage, and $\mathcal{O}(Ld^2N^2)$ per iteration until convergence for the second stage of the dual-EM algorithm used for the initialization of the VEM. However, the number of required iterations until convergence is smaller for the VEM algorithm than is in the case of the other algorithms, as can be seen from Table I.

B. Color Image Segmentation by Color Clustering

In this application, we segment several color images using only color clustering when employing the proposed methodology. Three images called "Sunset," "Lighthouse," and "Forest" are shown in Fig. 9(a)–(c). We can observe that the "Sunset" image displays a rather dark lighting variation in the background, "Lighthouse" contains a mixture of homogeneous color areas and textures in day time lighting conditions, while "Forest" displays natural textures. The first step consists of transforming the color coordinate system from RGB to $\mathrm{L^*u^*v^*}$. $\mathrm{L^*u^*v^*}$ represents an appropriate color coordinate system that has been used for segmenting color images [39], [45], [46].

The input space is three-dimensional (3-D), each pixel representing a vector of three color components. In order to reduce the amount of data, we sample the images by two on each axis. The initialization is performed by employing the dual EM algorithm and by estimating the hyperparameter maximum log-likelihood onto distributions of parameters. The first EM is run by considering ten different random initializations for the parameters, and its output results into a total of $10N$ values for each parameter of the graphical model. The second EM was initialized with data samples from the given data set and is run on the distribution of means resulting from the first EM. We use the maximum log-likelihood methodology provided in Section IV in order to initialize the hyperparameters for the VEM algorithm. After running the VEM algorithm, as described in Section V, we obtain a set of hyperparameters corresponding to the variational Gaussian model. Given these hyperparameters, we calculate the *a posteriori* probabilities (31) for the entire image and consider hard decisions for color image segmentation by taking the MAP probabilities

$$\mathcal{V}_k = \left\{ \mathbf{x}_j | k = \arg \max_{i=1}^{N} \hat{P}(i|\mathbf{x}_j) \right\}. \tag{44}$$

As a consequence, each image is split in a set of regions based on color similarity in the $\mathrm{L^*u^*v^*}$ space by considering a variational Gaussian mixture model for each color image.

The segmented "Sunset" image is shown in Fig. 10(a) when considering seven mixture components, in Fig. 10(b) when considering ten mixture components, and in Fig. 10(c) when using eight mixture components. Each region $\mathcal{V}_k$, $k = 1, \ldots, N$, segmented according to (44), is displayed in the color corresponding to the hypermeans of the respective regions. From

Fig. 9. Original images to be segmented. (a) "Sunset." (b) "Lighthouse." (c) "Forest." (Color version available online at http://ieeexplore.ieee.org.)

Fig. 10. Segmentation of "Sunset" image when using VEM algorithm. (a) Using seven components. (b) Using ten components. (c) Using eight components. (Color version available online at http://ieeexplore.ieee.org.)

Fig. 11. Segmentation of "Lighthouse" image using VEM algorithm. (a) Using eight components. (b) Using ten components. (c) Using nine components. (Color version available online at http://ieeexplore.ieee.org.)

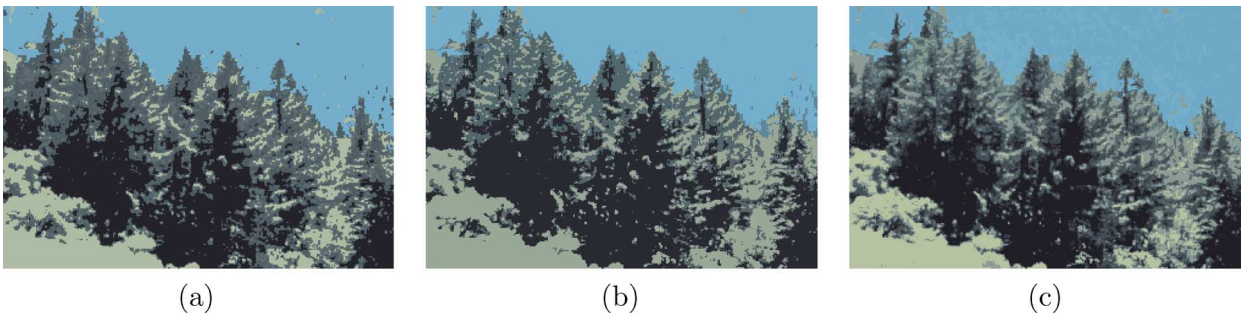


Fig. 12. Segmentation of "Forest" image using VEM algorithm. (a) Using five components. (b) Using six components. (c) Using nine components. (Color version available online at http://ieeexplore.ieee.org.)

these images, we can observe a good separation of the palm tree from the background as well as smooth segmentation of the twilight shadows in the background. A segmented "Lighthouse" image is displayed in Fig. 11(a) when using eight components, in Fig. 11(b) when using ten components, and in Fig. 11(c) when considering nine components. In these segmented images, we can observe a good separation of the sky from the sea and ground, respectively. In Fig. 12(a), we represent the segmented "Forest" when using five components, Fig. 12(b) for six components, and in Fig. 12(c) when considering nine components. In all these images, we can observe a good texture segmentation based exclusively on color information.

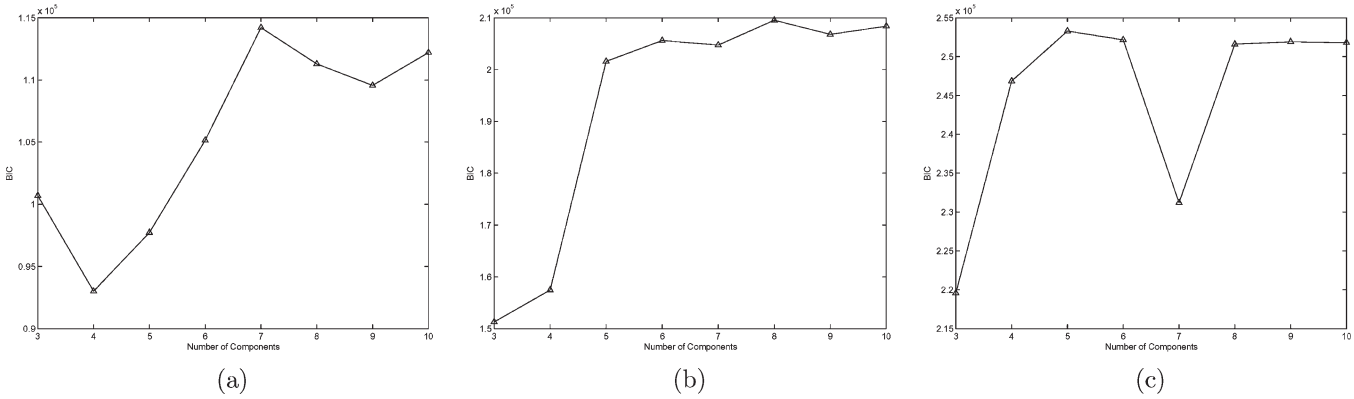


Fig. 13. Estimating the number of Gaussian mixture components using BIC. (a) "Sunset" image. (b) "Lighthouse" image. (c) "Forest" image.

The number of mixture components has been calculated using BIC (40) [40]. The plots displaying the evaluation of $\mathcal{C}_{\mathrm{VEM}}(N)$ for a set of different numbers of components, for each image, are displayed in Fig. 13. From Fig. 13(a), we observe that seven components are needed to segment the "Sunset" image; from Fig. 13(b) we observe that eight components would be more appropriate for the "Lighthouse" image; while from Fig. 13(c) we remark that five components would be sufficient for the "Forest" image. As we observe from these plots, according to the BIC criterion, the images are not properly segmented for a small number of components. When reaching a certain threshold in the number of mixture components we achieve the saturation in the cost function $\mathcal{C}_{\mathrm{VEM}}(N)$. The segmentation results from Figs. 10–12 are considered only for the most appropriate number of hidden units. In Fig. 14, we display the variation of the average log-likelihood $L_\theta(\mathbf{x})$ for the VEM algorithm at each iteration for the best case and when assuming the appropriate number of mixture components.

The EM algorithm has been recently used for color image segmentation [17], [45]. We compare the proposed variational color segmentation algorithm with the EM algorithm. In Fig. 15, we provide comparative segmentation results for the three images when using the EM algorithm for the most appropriate number of components. The numerical comparison criteria consists of the average likelihood and the peak SNR (PSNR), expressed in decibels. The average likelihood is calculated as the average of the log-likelihood for all the image pixels, where the likelihood for each pixel is provided in (38) and $L_\theta(\mathbf{x})$ denotes the log-likelihood obtained at convergence. PSNR is calculated when converting the color image in a grey-level image and after considering the hypermeans as the reference values for the segmented regions, i.e.,

$$\mathrm{PSNR} = 20\log_{10}\left(\frac{255M}{\sqrt{\sum_{j=1}^{M}(x_j-\hat{\mu}_k)^2}}\right) \tag{45}$$

where x_j denotes the grey-level value at pixel j and $\hat{\mu}_k$ is the hypermean estimate that is assigned to that pixel, following the conversion from color to grey level, after the decision given by

$$\mathbf{x}_j \rightarrow \hat{\mu}_k \text{ if } \mathbf{x}_j \in \mathcal{V}_k \tag{46}$$

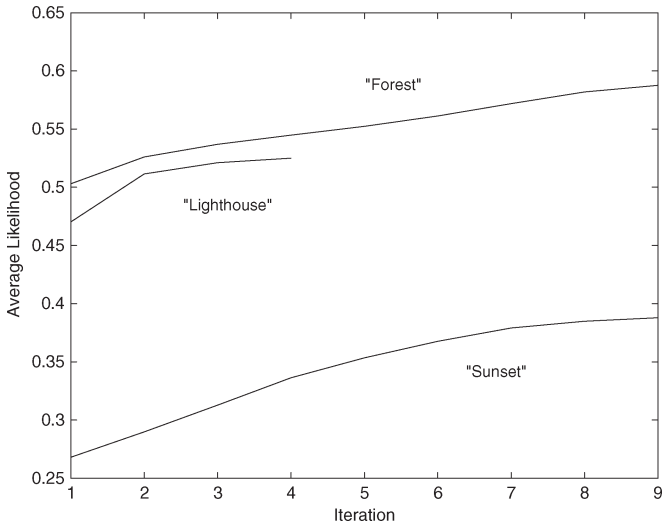


Fig. 14. Convergence in log-likelihood for the VEM algorithm when segmenting the three color images.

for $j = 1, \ldots, M$, according to (44). The convergence condition corresponds to $\varepsilon = 0.01$ in (39). In the first stage of the dual-EM initialization stage for the VEM algorithm, we use the same initialization as for the EM, but considering a prefixed number of iterations. For both EM and VEM algorithms we consider the result as the average of ten different runs. The comparison results are shown in Table II, where we consider the mean result and the standard deviation for the given results.

From Table II, we can observe that the average log-likelihood for the "Forest" image is larger than that for the "Sunset" and "Lighthouse." On the other hand, the PSNR for the same image is smaller than those of the other images. A higher PSNR signifies better image segmentation. In the case of the "Forest" image, the higher proportion of texture in the image causes a higher PSNR when represented by fewer mixture components. In all three color images, we have obtained better segmentation results, according to the average likelihood, PSNR, as well as visually, when using the VEM algorithm instead of the EM.

VII. Conclusion

This paper introduces a new Bayesian algorithm for estimating parameters in Gaussian mixture models. In Bayesian inference approaches, we integrate over distributions of parameters

Fig. 15. Image segmentation using EM algorithm when considering the most appropriate number of components. (a) Using seven components. (b) Using eight components. (c) Using five components. (Color version available online at http://ieeexplore.ieee.org.)

TABLE II
COMPARISON BETWEEN EM AND VEM ALGORITHMS IN COLOR IMAGE SEGMENTATION

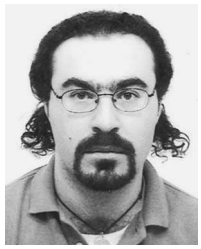
Algorithm	Measure	Color Images		
		"Sunset" (N=7)	"Lighthouse" (N=8)	"Forest" (N=5)
EM	$L_\theta(\mathbf{x})$	0.3676 ± 0.0051	0.5093 ± 0.0228	0.4435 ± 0.0013
	PSNR (dB)	14.76 ± 2.07	12.08 ± 2.49	5.35 ± 0.39
VEM	$L_\theta(\mathbf{x})$	0.3759 ± 0.0063	0.5209 ± 0.0059	0.5587 ± 0.0311
	PSNR (dB)	18.64 ± 0.40	15.88 ± 1.08	10.96 ± 0.59

in order to get better fitting and generalization capabilities. A variational approach is deterministic in nature and was shown to ensure the existence of a lower bound on the approximation error. The proposed algorithm estimating hyperparameters of mixtures of Gaussian distributions is unsupervised. We analyze a new hyperparameter initialization methodology by employing a hierarchical approach for distribution estimation. In the first stage, a dual-EM algorithm is run several times on the given data set. The successive runs of the first EM algorithm provide distributions of parameters. Maximum log-likelihood is employed on these distributions in order to initialize the hyperparameters. The variational algorithm called VEM uses the maximum log-likelihood results as the initial values for hyperparameters. Experimental results have shown that convergence is achieved in fewer iterations, while the final results are improved, by using the proposed initialization. The number of necessary hidden units corresponding to the number of mixture components is estimated using the BIC. The proposed algorithm is applied on both artificial and real data. The first experimental study includes tests on blind signal detection in PSK- and QAM-modulated signals. Low detection errors and reliable model estimation has been achieved when using the VEM algorithm. In the second experiment, the proposed methodology was used to segment several color images, represented in the $L^*u^*v^*$ color space. The results show good color image segmentation when employing the VEM algorithm. Although the computational complexity required by the VEM algorithm is higher than that of the EM or VB, due to the initialization procedure, the number of necessary iterations until convergence is lower.

REFERENCES

[1] D. M. Titterington, A. F. M. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions*. Hoboken, NJ: Wiley, 1985.
[2] A. K. Jain, R. P. W. Duin, and J. C. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.
[3] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Rev.*, vol. 26, no. 2, pp. 195–239, Apr. 1984.
[4] A. G. Bors and M. Gabbouj, "Minimal topology for a radial basis functions neural network for pattern classification," *Digit. Signal Process.: A Rev. J.*, vol. 4, no. 3, pp. 173–188, Jul. 1994.
[5] A. G. Bors and I. Pitas, "Median radial basis function neural network," *IEEE Trans. Neural Netw.*, vol. 7, no. 6, pp. 1351–1364, Nov. 1996.
[6] I. Cha and S. A. Kassam, "RBFN restoration of nonlinearly degraded images," *IEEE Trans. Image Process.*, vol. 5, no. 6, pp. 964–975, Jun. 1996.
[7] M. W. Mak and S. Y. Kung, "Estimation of elliptical basis function parameters by the EM algorithm with application to speaker verification," *IEEE Trans. Neural Netw.*, vol. 11, no. 4, pp. 961–969, Jul. 2000.
[8] M. Lazaro, I. Santamaria, and C. Pantaleon, "A new EM-based training algorithm for RBF networks," *Neural Netw.*, vol. 16, no. 1, pp. 69–77, Jan. 2003.
[9] A. G. Bors and I. Pitas, "Object classification in 3-D images using alpha-trimmed mean radial basis function network," *IEEE Trans. Image Process.*, vol. 8, no. 12, pp. 1730–1744, Dec. 1999.
[10] G. Box and G. Tiao, *Bayesian Inference in Statistical Models*. Reading, MA: Addison-Wesley, 1992.
[11] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*. London, U.K.: Chapman & Hall, 1995.
[12] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via EM algorithm," *J. R. Statist. Soc., B*, vol. 39, no. 1, pp. 1–38, 1977.
[13] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions (Wiley Series in Probability and Statistics)*. New York: Wiley, 1997.
[14] R. S. Blum and J. Yang, "Image fusion using the expectation–maximization algorithm and a Gaussian mixture model," in *Multisensor Surveillance Systems: The Fusion Perspective*, G. L. Foresti, C. S. Regazzoni, and P. Varshney, Eds. Norwell, MA: Kluwer, 2003, pp. 81–96.
[15] 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.
[16] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton, "SMEM algorithm for mixture models," *Neural Comput.*, vol. 12, no. 9, pp. 2109–2118, Sep. 2000.
[17] Z. H. Zhang, C. B. 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, Sep. 2003.
[18] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," in *Learning in Graphical Models*, M. I. Jordan, Ed. Cambridge, MA: MIT Press, 1999, pp. 105–161.

[19] T. S. Jaakkola and M. I. Jordan, "Bayesian parameter estimation via variational methods," *Stat. Comput.*, vol. 10, no. 1, pp. 25–37, Sep. 2000.
[20] ——, "Improving the mean field approximation via the use of mixture distributions," in *Learning in Graphical Models*, M. I. Jordan, Ed. Cambridge, MA: MIT Press, 1999, pp. 163–173.
[21] S. J. Roberts, D. Husmeier, I. Rezek, and W. D. Penny, "Bayesian approaches to Gaussian mixture modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1133–1142, Nov. 1998.
[22] D. Husmeier, "The Bayesian evidence scheme for regularizing probability-density estimating neural networks," *Neural Comput.*, vol. 12, no. 11, pp. 2685–2717, Nov. 2000.
[23] K. Chan, T.-W. Lee, and T. J. Sejnowski, "Variational Bayesian learning of ICA with missing data," *Neural Comput.*, vol. 15, no. 8, pp. 1991–2011, Aug. 2003.
[24] D. J. C. MacKay, "Introduction to Monte Carlo methods," in *Learning in Graphical Models*, M. I. Jordan, Ed. Cambridge, MA: MIT Press, 1999, pp. 175–204.
[25] M. J. Beal and Z. Ghahramani, "The variational Bayesian EM algorithm for incomplete data: With application to scoring graphical model structures," in *Bayesian Statistics 7*, D. Heckerman, A. F. M. Smith, and M. West, Eds. London, U.K.: Oxford Univ. Press, 2003, pp. 453–464.
[26] M. N. Gibbs and D. J. C. MacKay, "Variational Gaussian process classifiers," *IEEE Trans. Neural Netw.*, vol. 11, no. 6, pp. 1458–1464, Nov. 2000.
[27] H. Attias, "A variational Bayesian framework for graphical models," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 12. Cambridge, MA: MIT Press, 2000, pp. 209–215.
[28] T. Minka and J. Lafferty, "Expectation-propagation for the generative aspect model," in *Proc. 18th Conf. Uncertainty Arti. Intell.*, Edmonton, AB, Canada, Aug. 1–4, 2002, pp. 352–359.
[29] Z. Ghahramani and M. J. Beal, "Propagation algorithms for variational Bayesian learning," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 13. Cambridge, MA: MIT Press, 2001, pp. 294–300.
[30] S. Richardson and P. J. Green, "On Bayesian analysis of mixtures with an unknown number of components," *J. Roy. Stat. Soc., Ser. B*, vol. 59, no. 4, pp. 731–792, 1997.
[31] C. Andrieu, N. de Freitas, and A. Doucet, "Robust full Bayesian learning for radial basis networks," *Neural Comput.*, vol. 13, no. 10, pp. 2359–2407, Oct. 2001.
[32] Z. Ghahramani and G. E. Hinton, "Variational learning for switching state-space methods," *Neural Comput.*, vol. 12, no. 4, pp. 831–864, Apr. 2000.
[33] S. J. Roberts and W. D. Penny, "Variational Bayes for generalized autoregressive models," *IEEE Trans. Signal Process.*, vol. 50, no. 9, pp. 2245–2257, Sep. 2002.
[34] N. de Freitas, P. Højen-Sørensen, M. Jordan, and S. Russell, "Variational MCMC," in *Proc. 17th Conf. Uncertainty Artif. Intell.*, Seattle, WA, 2001, pp. 120–127.
[35] Z. Ghahramani and M. J. Beal, "Variational inference for Bayesian mixtures of factor analysers," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 12. Cambridge, MA: MIT Press, 2000, pp. 449–455.
[36] R. A. Choudrey and S. J. Roberts, "Variational mixture of Bayesian independent component analyzers," *Neural Comput.*, vol. 15, no. 1, pp. 213–252, Jan. 2003.
[37] H. Attias, "Inferring parameters and structure of latent variable models by variational Bayes," in *Proc. 15th Conf. Uncertainty Artif. Intell.*, Stockholm, Sweden, 1999, pp. 21–30.
[38] N. Nasios and A. G. Bors, "Blind source separation using variational expectation–maximization algorithm," in *Proc. Int. Conf. Comput. Anal. Images Patterns*, Lecture Notes in Computer Science 2756, Groningen, The Netherlands, 2003, pp. 442–450.
[39] ——, "Variational segmentation of color images," in *Proc. IEEE Int. Conf. Image Process.*, Genova, Italy, 2005, vol. II, pp. 614–617.
[40] G. Schwarz, "Estimating the dimension of a model," *Ann. Stat.*, vol. 7, no. 2, pp. 461–464, 1978.
[41] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*. Singapore: World Scientific, 1989.
[42] G. Ronning, "Maximum-likelihood estimation of Dirichlet distributions," *J. Statist. Comput. Simul.*, vol. 32, no. 4, pp. 215–221, 1989.
[43] M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, *Simulation of Communication Systems*. New York: Plenum, 1992.
[44] A. G. Bors and M. Gabbouj, "Quadrature modulated signal detection based on Gaussian neural networks," in *Proc. IEEE Workshop Vis. Signal Process. Commun.*, Melbourne, Australia, 1993, pp. 113–116.
[45] J. B. Gao, J. Zhang, and M. G. Fleming, "Novel technique for multiresolution color image segmentation," *Opt. Eng.*, vol. 41, no. 3, pp. 608–614, Mar. 2002.
[46] D. A. Forsyth and J. Ponce, *Computer Vision a Modern Approach*. Upper Saddle River, NJ: Prentice-Hall, 2003.



Nikolaos Nasios was born in Thessaloniki, Greece, in 1976. He received the degree from the Department of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece, in 2000. He is currently working toward the Ph.D. degree in the Computer Vision and Pattern Recognition Group, University of York, Heslington, U.K.

His research interests include image processing and pattern recognition with specific focus on statistical methods for data clustering.

Mr. Nasios is a member of the Technical Chamber of Greece.

Adrian G. Bors (M'00–SM'04) received the M.S. degree in electronics engineering from the Polytechnic University of Bucharest, Bucharest, Romania, in 1992 and the Ph.D. degree in informatics from the University of Thessaloniki, Thessaloniki, Greece in 1999.

From September 1992 to August 1993, he was a Research Scientist with the Signal Processing Laboratory, Tampere University of Technology, Finland. From 1993 to 1999, he was a Research Associate, first with the Department of Electrical and Computer Engineering and later with the Department of Informatics at the University of Thessaloniki, Greece. In March 1999, he joined the Department of Computer Science, University of York, U.K., where he is currently a Lecturer. He has authored and coauthored 13 journal papers and more than 50 papers in edited books and international conference proceedings. His research interests include computational intelligence, computer vision, image processing, pattern recognition, digital watermarking, and nonlinear digital signal processing.

Dr. Bors has been an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS since 2001 and a member of technical committees of various international conferences.