Expert Systems with Applications 39 (2012) 5946–5959

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A fully Bayesian model based on reversible jump MCMC and finite Beta mixtures for clustering

Nizar Bouguila *, Tarek Elguebaly

Concordia Institute for Information Systems Engineering, Faculty of Engineering and Computer Science, Concordia University, Montreal, Qc, Canada H3G 2W1

ARTICLE INFO

Keywords:
Beta distribution
Mixture modeling
Bayesian analysis
MCMC
Reversible jump
Gibbs sampling
Metropolis–Hastings
Texture classification
Retrieval

ABSTRACT

The use of mixture models in image and signal processing has proved to be of considerable interest in terms of both theoretical development and in their usefulness in several applications. Researchers have approached the mixture estimation and selection problem, to model complex datasets, with different techniques in the last few years. In theory, it is well-known that full Bayesian approaches, to handle this problem, are fully optimal. The Bayesian learning allows the incorporation of prior knowledge in a formal coherent way that avoids overfitting problems. In this paper, we propose a fully Bayesian approach for finite Beta mixtures learning using a reversible jump Markov chain Monte Carlo (RJMCMC) technique which simultaneously allows cluster assignments, parameters estimation, and the selection of the optimal number of clusters. The adverb "fully" is justified by the fact that all parameters of interest in our model including number of clusters and missing values are considered as random variables for which priors are specified and posteriors are approximated using RJMCMC. Our work is motivated by the fact that Beta mixtures are able to fit any unknown distributional shape and then can be considered as a useful class of flexible models to address several problems and applications involving measurements and features having well-known marked deviation from the Gaussian shape. The usefulness of the proposed approach is confirmed using synthetic mixture data, real data, and through an interesting application namely texture classification and retrieval.

1. Introduction

In recent years Bayesian approaches have found an increased interest in the image and signal processing community (Fitzgerald, Godsill, Kokaram, & Stark, 1999). An increasingly important topic in statistical signal and image processing is the modeling of non-Gaussian signals, features and data. Finite mixture models provide a powerful, flexible and well principled statistical approach and have been commonly used to model complex data in many applications (McLachlan & Peel, 2000). An important problem in mixture modeling is the choice of the components densities. In particular, Gaussian mixture models have received a lot of attention. As a smooth, bell-shaped distribution that can be completely characterized by its mean and its standard deviation, the Gaussian is in general used and justified for asymptotic reasons (i.e. the sample is supposed to be sufficiently large) (Robert, 2007). Although a Gaussian mixture may provide a reasonable approximation to many real-word distribution, it is certainly not always the best approximation especially in image and signal processing application where we often deal with small samples (Meignen & Meignen, 2006). Indeed, there are many phenomena and applications for which the Gaussian model is not realistic (for instance, it is well-known that natural image clutter is generally non-Gaussian). Another important problem is the parameters estimation. This problem is not straightforward and many deterministic and Bayesian approaches have been proposed. In classical deterministic inference, data are taken as random while parameters are taken as fixed and unknown, and the inference is based in general on the likelihood of the data. Although deterministic methods have dominated mixture models estimation, many works have shown that severe problems may arise due to singularities and local maxima in the log-likelihood function. It is well-known, for instance, that the use of maximum likelihood approaches lead to more complex models and then overfitting (Robert, 2007). An alternative approach is the use of pure Bayesian techniques which could give better results. Indeed, the application of Bayesian inference have been found to be useful in many practical applications in different domains thanks to the development of MCMC methods and techniques. The reader interested in the general Bayesian theory

* Corresponding author.
E-mail addresses: bouguila@ciise.concordia.ca (N. Bouguila), t_elgue@encs.concordia.ca (T. Elguebaly).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.11.122

is referred to accessible expositions in Ghosh, Delampady, and Samanta (2006) and Robert (2007).

In Bayesian inference, the parameters themselves are considered random and then follow probability distributions (Robert, 2007). These distributions are called prior distributions and describe our knowledge before considering the data. The likelihood is a part of the model used to update our prior beliefs and this update is summarized by the posterior density. Moreover, Bayesian inference provides consistent learning frameworks for model uncertainty, through the use of posterior model probabilities, which is fundamental in image processing applications (Wilson & Granlund, 1984). Lack of knowledge on the number of clusters is another challenging problem in mixture modeling and considerable efforts already have been made to investigate this important aspect. The majority of the approaches that have been proposed separate the estimation and the selection of the number of components (i.e. a certain criterion should be compared for different number of clusters) (see, for instance, Bouguila & Ziou, 2007; McLachlan & Peel, 2000 for interesting discussions and comparisons between different criteria). Note, however, that both estimation and selection problems are strongly related and depend heavily on the underlying mixture density components choice.

In this paper, we propose to simultaneously estimate and select finite Beta mixture models using the reversible jump samplers introduced by Green (1995) and which have been applied successfully for instance to Gaussian (Dellaportas & Papageorgiou, 2006; Marrs, 1997; Richardson & Green, 1997; Zhang, Chan, Wu, & Chen, 2004), Poisson (Meligkotsidou, 2007; Vrontos, Dellaportas, & Politis, 2000; Viallefont, Richardson, & Green, 2002), exponential (Gruet, Philippe, & Robert, 1999) mixtures, to variable selection (Dellaportas, Forster, & Ntzoufras, 2002), and to model selection in general (Andrieu & Doucet, 1999; Brooks, 2001). The basic idea of RJMCMC approach is that it is possible to move between parameter subspaces corresponding to statistical models, such as mixture models with different number of components, which offers effective model selection (i.e. structure discovery) and produces a good mixing of the Markov chains. Using RJMCMC allows us to explore simultaneously both the parameter and model space by treating the number of clusters as a random variable having a prior distribution[1] and it does not need to be specified in advance, since it can be automatically adjusted during iterations. Our work is motivated by the compactly supported nature of the data generally handled in image and signal processing applications and by the fact that the Beta distribution is able to model any unknown distributional shape generated by this kind of data. Despite these advantages, finite Beta mixtures have been largely ignored and relatively less visited avenue of study compared to finite Gaussian mixtures. Moreover, it is well-know that any continuous density can be well-approximated by a mixture of Beta distributions (see Diaconis & Ylvisaker, 1985, for instance, for formal statement and detailed proof). For this reason the Beta distribution and mixtures of Beta have been widely used to model expert opinion, as a prior, in Bayesian settings (Berkhof, Van Mechelen, & Gelman, 2003; Brooks, 2001; Gelfand, Mallick, & Dey, 1995). In this work, however, Beta is used as a parent distribution to model directly the data.

This paper is structured as follows. After presenting our hierarchical finite mixture Beta Bayesian framework in Section 2, the complete RJMCMC algorithm is discussed and developed in Section 3. Section 4 is devoted to the experimental results. Finally, some conclusions are drawn in Section 5.

[1] It is noteworthy that other works, that we do not investigate in this paper, have put prior distribution on the number of components (see, for instance, Nobile, 2004; Nobile & Fearnside, 2007; Roeder & Wasserman, 1997; Stephens, 2000).

2. Bayesian analysis of Beta mixture model

2.1. Finite general Beta mixture model

2.1.1. General Beta distribution

If the random variable x, where $a < x < b$ and $(a,b) \in \mathbb{R}^2$, follows a general Beta distribution with parameters α and β, then the density function is given by Johnson, Kotz, and Balakrishman (1995):

$$p(x|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{(b-a)^{\alpha+\beta-1}\Gamma(\alpha)\Gamma(\beta)}(x-a)^{\alpha-1}(b-x)^{\beta-1} \quad (1)$$

where $\alpha > 0$ and $\beta > 0$. Note that this distribution is reduced to the well-known Beta when $(a,b) = (0,1)$, which is actually the univariate case of the Dirichlet distribution which has proven high flexibility to model data (Bouguila, Ziou, & Vaillancourt, 2004). The mean and variance of the general Beta distribution are given by:

$$m = E(x) = (b-a)\frac{\alpha}{\alpha+\beta} + a \quad (2)$$

$$v = Var(x) = (b-a)^2\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \quad (3)$$

Using Eq. (2), it is easy to obtain the following location-scale parametrization of the general Beta:

$$p(x|m,s) = \frac{\Gamma(s)}{(b-a)^{s-1}\Gamma\left(\frac{s(m-a)}{b-a}\right)\Gamma\left(s(1-\frac{m-a}{b-a})\right)} \times (x-a)^{\frac{s(m-a)}{b-a}-1}(b-x)^{s(1-\frac{m-a}{b-a})-1} \quad (4)$$

where $s = \alpha + \beta$ and represents the scale of the distribution and m represents the location. Note that this alternative provides interpretable parameters because m and s represent the mean and a measure of the sharpness of the distribution, respectively (Mackay & Peto, 1994). A large value of s produces a sharply peaked distribution around the mean m. And when s decreases, the distribution becomes broader. An additional advantage of this parametrization is that m lies within the bounded space $[a,b]$, leading to an increase in computational efficiency. Therefore, this parametrization will be adopted for learning.

2.1.2. Finite general Beta mixture

A general Beta mixture with M components is defined as:

$$p(x|\Theta) = \sum_{j=1}^{M} p(x|m_j,s_j)p_j \quad (5)$$

where $\{p_j\}$ are the mixing proportions which are constrained to be non-negative and sum to one, and $p(x|m_j,s_j)$ is the general Beta distribution. The symbol $\Theta = (\xi,P)$ refers to the entire set of parameters to be estimated, where $\xi = (m_1,s_1\ldots,m_M,s_M)$, and $P = (p_1,\ldots,p_M)$.

Given a set of data with N observations $\mathcal{X} = \{x_1,\ldots,x_N\}$, the classical approach to estimate the parameters of a mixture model is to maximize the likelihood through the Expectation Maximization (EM) algorithm which theoretical framework was first introduced in the seminal paper by Dempster, Laird, and Rubin (1977). The EM, however, guarantees convergence only to a local maximum which quality depends highly on the initialization step (i.e. as a deterministic approach the EM gets stuck at local maxima that are not globally optimal). Moreover, it is well-known that estimation approaches based on maximizing the likelihood can cause overfitting by preferring complex models. Many researchers have developed modifications and extensions of the EM algorithm (the interested reader is refereed to McLachlan & Krishnan (1997), Meng & van Dyk (1997), Jamshidian & Jennrich (1997)).[2] A detailed

[2] In particular the authors in Jamshidian and Jennrich (1997) classify these extensions into three groups: pure, hybrid and EM-type accelerators.

discussion about the drawbacks of deterministic estimation in the case of finite Beta mixtures can be found in Bouguila, Ziou, and Monga (2006). EM is based on the idea of explicitly representing the mixture components generating each observation via latent allocation variables Z_i, $i = 1,\ldots,N$. Each Z_i is an integer in $\{1,\ldots,M\}$ denoting the unknown component from which x_i is drawn. The unobserved (or missing) vector $Z = (Z_1,\ldots,Z_N)$ is generally called the "membership vector" of the mixture model and its different elements Z_i are supposed to be drawn independently from the distributions

$$p(Z_i = j) = p_j \quad j = 1,\ldots,M. \tag{6}$$

The same idea has an important role when using Bayesian approaches which are now widely applied as an alternative thanks to modern Bayesian computational tools. Bayesian estimation has become feasible due to the development of simulation-based numerical integration techniques such as Markov chain Monte Carlo (MCMC) methods (Robert, 2007). MCMC methods have revolutionized Bayesian statistics by allowing inference for highly complex models which can be treated tractably, albeit numerically, through the simulation of required estimates by running appropriate Markov Chains using specific algorithms such as Gibbs sampler. The Gibbs sampler, however, can be difficult to implement when the conditioning distributions have complicated awkward forms. In this case, solutions include the Metropolis–Hastings algorithm (Robert, 2007) which we will use in this work. Among the important problems that arise in using Bayesian techniques is the choice of priors. In the following, we present our Bayesian model, the priors that we have considered and the resulting posteriors.

2.2. Hierarchical model, priors and posteriors

2.2.1. Hierarchical model

Fully Bayesian analysis considers the number of components M as a parameter in the model for which a conditional distribution should be found. Moreover, the unknowns M, ξ and P, in our mixture model, are regarded as random variables drawn from some prior distributions. The joint distribution of all these variables is

$$p(M,P,Z,\xi,\mathcal{X}) = p(M)p(P|M)P(Z|P,M)p(\xi|Z,P,M)p(\mathcal{X}|\xi,Z,P,M)$$

A common approach is to impose conditional independencies (Richardson & Green, 1997), $p(\xi|Z,P,M) = p(\xi|M)$ and $p(\mathcal{X}|\xi,Z,P,M) = p(\mathcal{X}|\xi,Z)$, which give us the following joint distribution

$$p(M,P,Z,\xi,\mathcal{X}) = p(M)p(P|M)P(Z|P,M)p(\xi|M)p(\mathcal{X}|\xi,Z)$$

It is worth mentioning that if we condition on Z, the distribution of x_i is simply given by the Z_ith component in the mixture, $p(\mathcal{X}|\xi,Z) = \prod_{i=1}^{N} p(x_i|\xi_{Z_i})$. Moreover, an extra layer can be introduced to the hierarchy to represent the model parameters (M,P,ξ) priors, which gives the following final form of the joint distribution

$$p(\lambda,\delta,\eta,M,P,Z,\xi,\mathcal{X}) = p(\lambda)p(\delta)p(\eta)p(M|\lambda)p(P|M,\delta)p(Z|P,M) \\ p(\xi|M,\eta)\prod_{i=1}^{N} p(x_i|\xi_{Z_i}) \tag{7}$$

where λ, δ and η are the hyperparameters on which M, P and ξ depend, respectively.

2.2.2. Priors and posteriors

Let us now define the priors, which we suppose that are all drawn independently[3], of the different parameters in our hierarchical model. We know that each location m_j is defined in the compact support [a,b], then an appealing flexible choice as a prior is a general Beta distribution, with location ε and scale ζ common to all components, which was found flexible in real applications. Thus, m_j for each component is given the following prior:

$$p(m_j|\varepsilon,\zeta) \sim \frac{\Gamma(\zeta)}{(b-a)^{\zeta-1}\Gamma\left(\frac{\zeta(\varepsilon-a)}{b-a}\right)\Gamma\left(\zeta\left(1-\frac{\varepsilon-a}{b-a}\right)\right)} \\ \times (m_j-a)^{\frac{\zeta(\varepsilon-a)}{b-a}-1}(b-m_j)^{\zeta\left(1-\frac{\varepsilon-a}{b-a}\right)-1} \tag{8}$$

Since s_j control the dispersion of the distributions, a common choice as a prior is an inverse gamma with shape ϑ and scale ϖ common to all components (Carlin & Louis, 2000), then

$$p(s_j|\vartheta,\varpi) \sim \frac{\varpi^{\vartheta}\exp(-\varpi/s_j)}{\Gamma(\vartheta)s_j^{\vartheta+1}} \tag{9}$$

using the two previous equations, we have

$$p(\xi|M,\eta) = \prod_{j=1}^{M} p(m_j|\varepsilon,\zeta)p(s_j|\vartheta,\varpi) \\ = \frac{\varpi^{M\vartheta}\Gamma(\zeta)^M \prod_{j=1}^{M} \frac{\exp(-\varpi/s_j)(m_j-a)^{\frac{\zeta(\varepsilon-a)}{b-a}-1}(b-m_j)^{\zeta\left(1-\frac{\varepsilon-a}{b-a}\right)-1}}{s_j^{\vartheta+1}}}{\Gamma(\vartheta)^M(b-a)^{M(\zeta-1)}\left[\Gamma\left(\frac{\zeta(\varepsilon-a)}{b-a}\right)\Gamma\left(\zeta\left(1-\frac{\varepsilon-a}{b-a}\right)\right)\right]^M} \tag{10}$$

Having this priors in hand, the η hyperparameter in Eq. (7) is actually $(\varepsilon,\zeta,\vartheta,\varpi)$. Thus, according to the previous equation and our joint distribution in Eq. (7), the full conditional posterior distributions for m_j and s_j are

$$p(m_j|\ldots) \propto \prod_{j=1}^{M} p(m_j|\varepsilon,\zeta)p(s_j|\vartheta,\varpi)\prod_{i=1}^{N} p(x_i|\xi_{Z_i}) \\ \propto p(m_j|\varepsilon,\zeta)\prod_{i=1}^{N} p(x_i|\xi_{Z_i}) \\ \propto \left[\frac{\Gamma(s_j)}{(b-a)^{s_j-1}\Gamma\left(\frac{s(m_j-a)}{b-a}\right)\Gamma\left(s_j\left(1-\frac{m_j-a}{b-a}\right)\right)}\right]^{n_j} \\ \times \prod_{Z_i=j}\left[(x_i-a)^{\frac{s_j(m_j-a)}{b-a}-1}(b-x_i)^{s_j\left(1-\frac{m_j-a}{b-a}\right)-1}\right] \\ \times \frac{\Gamma(\zeta)}{(b-a)^{\zeta-1}\Gamma\left(\frac{\zeta(\varepsilon-a)}{b-a}\right)\Gamma\left(\zeta\left(1-\frac{\varepsilon-a}{b-a}\right)\right)} \\ \times (m_j-a)^{\frac{\zeta(\varepsilon-a)}{b-a}-1}(b-m_j)^{\zeta\left(1-\frac{\varepsilon-a}{b-a}\right)-1} \tag{11}$$

$$p(s_j|\ldots) \propto \prod_{j=1}^{M} p(m_j|\varepsilon,\zeta)p(s_j|\vartheta,\varpi)\prod_{i=1}^{N} p(x_i|\xi_{Z_i}) \\ \propto p(s_j|\vartheta,\varpi)\prod_{i=1}^{N} p(x_i|\xi_{Z_i}) \\ \propto \left[\frac{\Gamma(s_j)}{(b-a)^{s_j-1}\Gamma\left(\frac{s(m_j-a)}{b-a}\right)\Gamma\left(s_j\left(1-\frac{m_j-a}{b-a}\right)\right)}\right]^{n_j} \\ \times \prod_{Z_i=j}\left[(x_i-a)^{\frac{s_j(m_j-a)}{b-a}-1}(b-x_i)^{s_j\left(1-\frac{m_j-a}{b-a}\right)-1}\right] \\ \times \frac{\varpi^{\vartheta}\exp(-\varpi/s_j)}{\Gamma(\vartheta)s_j^{\vartheta+1}} \tag{12}$$

where $n_j = \sum_{i=1}^{N} \mathbb{I}_{Z_i=j}$ and represents the number of vectors belonging to cluster j.

Moreover, we know that the vector P is defined on the simplex $\left\{(p_1,\ldots,p_M) : \sum_{j=1}^{M-1} p_j < 1\right\}$, then the typical choice, as a prior, for this vector is a Dirichlet distribution with parameters $\delta = ((\delta_1,\ldots,\delta_M)$ (Robert, 2007)

[3] The choice of a simple independence prior structure is a common assumption taken generally when defining Bayesian models.

$$p(P|M,\delta) = \frac{\Gamma\left(\sum_{j=1}^{M}\delta_j\right)}{\prod_{j=1}^{M}\Gamma(\delta_j)}\prod_{j=1}^{M}p_j^{\delta_j-1} \tag{13}$$

According to Eq. (6), we have also

$$p(Z|P,M) = \prod_{j=1}^{M}p_j^{n_j} \tag{14}$$

Using the two previous equations and our joint distribution in Eq. (7), we obtain

$$p(P|\ldots) \propto p(Z|P,M)p(P|M,\delta) \propto \prod_{j=1}^{M}p_j^{n_j}\frac{\Gamma\left(\sum_{j=1}^{M}\delta_j\right)}{\prod_{j=1}^{M}\Gamma(\delta_j)}\prod_{j=1}^{M}p_j^{\delta_j-1} \propto \prod_{j=1}^{M}p_j^{n_j+\delta_j-1} \tag{15}$$

which is actually proportional to a Dirichlet distribution with parameters $(\delta_1+n_1,\ldots,\delta_M+n_M)$. It is noteworthy that the prior and the posterior distributions, $p(P|M,\delta)$ and $\pi(P|\ldots)$, are both Dirichlet. In this case we say that the Dirichlet distribution is a conjugate prior for the mixture proportions. In addition, using Eqs. (6) and (7), we have the following posterior for the membership variables

$$p(Z_i=j|\ldots) \propto \frac{p_j\Gamma(s_j)}{(b-a)^{s_j-1}\Gamma\left(\frac{s_j(m_j-a)}{b-a}\right)\Gamma\left(s_j(1-\frac{m_j-a}{b-a})\right)} \times (x-a)^{\frac{s_j(m_j-a)}{b-a}-1}(b-x)^{s_j(1-\frac{m_j-a}{b-a})-1} \tag{16}$$

In order to have a more flexible model, we introduce an additional hierarchical level by allowing the hyperparameters to follow some selected distributions. The hyperparameters, ε and ζ, associated with the m_j are given uniform (we have started by testing a general Beta prior for ε, and the best experimental results were obtained with location equal to b and scale fixed to 2, which corresponds actually to a uniform distribution) and inverse Gamma priors, respectively:

$$p(\varepsilon) \sim \mathcal{U}_{[a,b]} \tag{17}$$

$$p(\zeta|\varphi,\varrho) \sim \frac{\varrho^{\varphi}\exp(-\varrho/\zeta)}{\Gamma(\varphi)\zeta^{\varphi+1}} \tag{18}$$

Thus, according to these two previous equations and Eqs. (10) and (7), we have

$$p(\varepsilon|\ldots) \propto p(\varepsilon)\prod_{j=1}^{M}p(m_j|\varepsilon,\zeta) \propto \prod_{j=1}^{M}\frac{\Gamma(\zeta)}{(b-a)^{\zeta-1}\Gamma\left(\frac{\zeta(\varepsilon-a)}{b-a}\right)\Gamma\left(\zeta(1-\frac{\varepsilon-a}{b-a})\right)} \times (m_j-a)^{\frac{\zeta(\varepsilon-a)}{b-a}-1}(b-m_j)^{\zeta(1-\frac{\varepsilon-a}{b-a})-1} \tag{19}$$

$$p(\zeta|\ldots) \propto p(\zeta|\varphi,\varrho)\prod_{j=1}^{M}p(m_j|\varepsilon,\zeta) \propto \frac{\varrho^{\varphi}\exp(-\varrho/\zeta)}{\Gamma(\varphi)\zeta^{\varphi+1}}\prod_{j=1}^{M}\frac{\Gamma(\zeta)}{(b-a)^{\zeta-1}\Gamma\left(\frac{\zeta(\varepsilon-a)}{b-a}\right)\Gamma\left(\zeta(1-\frac{\varepsilon-a}{b-a})\right)} \times (m_j-a)^{\frac{\zeta(\varepsilon-a)}{b-a}-1}(b-m_j)^{\zeta(1-\frac{\varepsilon-a}{b-a})-1} \tag{20}$$

The hyperparameters, ϑ and ϖ, associated with the s_j are given inverse Gamma and exponential priors, respectively:

$$p(\vartheta|\lambda,\mu) \sim \frac{\mu^{\lambda}\exp(-\mu/\vartheta)}{\Gamma(\lambda)\vartheta^{\lambda+1}} \tag{21}$$

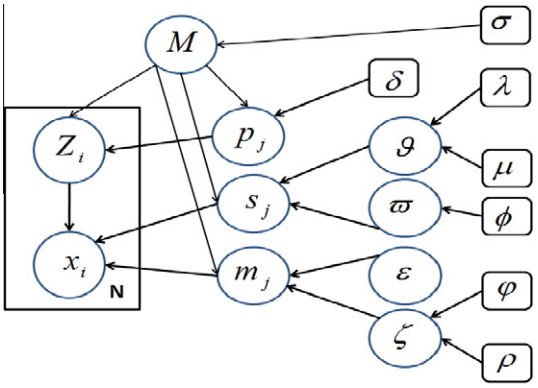

Fig. 1. Graphical model representation of the Bayesian hierarchical finite general Beta mixture model. Nodes in this graph represent random variables, rounded boxes are fixed hyperparameters, boxes indicate repetition (with the number of repetitions in the lower right) and arcs describe conditional dependencies between variables.

$$p(\varpi|\phi) \sim \phi\exp(-\phi\varpi) \tag{22}$$

Thus, according to these two previous equations and Eqs. (10) and (7), we have

$$p(\vartheta|\ldots) \propto p(\vartheta|\lambda,\mu)\prod_{j=1}^{M}p(s_j|\vartheta,\varpi) \propto \frac{\mu^{\lambda}\exp(-\mu/\vartheta)}{\Gamma(\lambda)\vartheta^{\lambda+1}}\prod_{j=1}^{M}\frac{\varpi^{\vartheta}\exp(-\varpi/s_j)}{\Gamma(\vartheta)s_j^{\vartheta+1}} \tag{23}$$

$$p(\varpi|\ldots) \propto p(\varpi|\phi)\prod_{j=1}^{M}p(s_j|\vartheta,\varpi) \propto \phi\exp(-\phi\varpi)\prod_{j=1}^{M}\frac{\varpi^{\vartheta}\exp(-\varpi/s_j)}{\Gamma(\vartheta)s_j^{\vartheta+1}} \tag{24}$$

For the number of components M, which has no particular reason to be fixed in advance, we take as a prior a common choice which is a Uniform $\{1,\ldots,\sigma\}$ distribution, where σ is a constant representing the maximum value allowed for M. Our hierarchical model can be displayed as a directed acyclic graph (DAG) as shown in Fig. 1.

3. Reversible jump MCMC algorithm

3.1. RJMCMC move types

Let Δ_M denotes the complete set of unknown variables (i.e. the sate variable), $\Delta_M = ((Z,P,M,\xi,\vartheta,\varpi,\varepsilon,\zeta)$. We consider also a countable family of move types, indexed by $t = 1,2,\ldots$. In our case, and following (Richardson & Green, 1997), the moves consist of: (1) updating the mixing parameters, (2) updating the parameters s and m, (3) updating Z, (4) updating the hyperparameters $\vartheta,\varpi,\varepsilon,\zeta$, (5) splitting one component into two, or merging two into one, (6) the birth or death of an empty component. In Richardson & Green (1997) *a sweep* is defined as a complete pass over the six moves and is considered as the basis time step of the complete learning algorithm. The first four moves do not change the dimensionality of the parameter vector and are actually classic Gibbs sampling moves. Note, however, that moves (5) and (6) necessitate changing (ξ,P,Z) and changing M by 1. The MCMC step representing move (5) takes the form of a Metropolis–Hastings step by proposing a move from a state Δ_M to Δ'_M with a target probability distribution (posterior distribution) $p(\Delta_M|\mathcal{X})$ and proposal distribution $q_t(\Delta_M,\Delta'_M)$ for the move t. When the current state is Δ_M, a given move t to destination Δ'_M is accepted with probability

$$\pi_t(\Delta_M,\Delta'_M)=\min\left\{1,\frac{p(\Delta'_M|\mathcal{X})q_t(\Delta'_M,\Delta_M)}{p(\Delta_M|\mathcal{X})q_t(\Delta_M,\Delta'_M)}\right\} \tag{25}$$

When we have a move, lying in a higher dimensional space, from a state Δ_M to another state Δ'_M, it is possible to implement this move by drawing a vector of continuous random variables u, independent of Δ_M (Richardson & Green, 1997). And the new Δ'_M state is set through an invertible deterministic function of Δ_M and u: $f(\Delta_M,u)$. Thus, the move acceptance probability is given by

$$\pi_t(\Delta_M,\Delta'_M)=\min\left\{1,\frac{p(\Delta'_M|\mathcal{X})r_t(\Delta'_M)}{p(\Delta_M|\mathcal{X})r_t(\Delta_M)q(u)}\left|\frac{\partial\Delta'_M}{\partial(\Delta_M,u)}\right|\right\} \tag{26}$$

where $r_t(\Delta_M)$ is the probability of choosing move type t when in state Δ_M, $q(u)$ is the density function of u and $\left|\frac{\partial\Delta'_M}{\partial(\Delta_M,u)}\right|$ is the Jacobian function arising from the variable change from (Δ_M,u) to Δ'_M.

3.2. Implementation of the moves

3.2.1. Gibbs sampling moves

As we mentioned above the first four moves are classic Gibbs sampling moves. For the first move the mixing parameters are generated from Eq. (15). The second move is based on the generation of m_j and s_j. It is noteworthy that the sampling of m_j and s_j is more complex, since the conditional posteriors given by Eqs. (11) and (12) do not have known forms. Thus, we have used the Metropolis–Hastings algorithm (M–H) (see, for instance Chib & Greenberg, 1995, for a detailed introductory exposition and discussions). At iteration t, the steps of the M–H algorithm, to generate s_j, can be described as follows.

(1) Generate $\tilde{s}_j\sim q\left(s_j|s_j^{(t-1)}\right)$ and $u\sim\mathcal{U}_{[0,1]}$

(2) Compute $r=\frac{p(\tilde{s}_j|\ldots)q\left(s_j^{(t-1)}|\tilde{s}_j\right)}{p\left(s_j^{(t-1)}|\ldots\right)q\left(\tilde{s}_j|s_j^{(t-1)}\right)}$

(3) If $r<u$ then $s_j^{(t)}=\tilde{s}_j$ else $s_j^{(t)}=s_j^{(t-1)}$

The major problem in this algorithm is the need to choose the proposal distribution q. The most generic proposal is the random walk Metropolis–Hastings algorithm where each unconstrained parameter is the mean of the proposal distribution for the new value. As $\tilde{s}_j>0$, we have chosen the following proposal $\tilde{s}_j\sim\mathcal{LN}\left(\log\left(s_j^{(t-1)}\right),e^2\right)$, where $\mathcal{LN}\left(\log\left(s_j^{(t-1)}\right),e^2\right)$ refers to the log-normal distribution with mean $\log\left(s_j^{(t-1)}\right)$ and variance e^2. Note that this is actually equivalent to $\log(\tilde{s}_j)=\log\left(s_j^{(t-1)}\right)+\epsilon_j$, where $\epsilon_j\sim\mathcal{N}(0,e^2)$. In the case of m_j we have opted for general Beta proposals, centered at the current values, to assure that $m_j\in[a,b]$. With these proposals the M–H algorithm, to generate m_j, is composed of the following steps:

(1) Generate $\tilde{m}_j\sim\mathcal{B}\left(m_j^{(t-1)},S\right)$ and $u\sim\mathcal{U}_{[0,1]}$.

(2) Compute $r=\frac{p(\tilde{m}_j|\ldots)\mathcal{B}\left(m_j^{(t-1)}|\tilde{m}_j,S\right)}{p\left(m_j^{(t-1)}|\ldots\right)\mathcal{B}\left(\tilde{m}_j|m_j^{(t-1)},S\right)}$

(3) If $r<u$ then $m_j^{(t)}=\tilde{m}_j$ else $m_j^{(t)}=m_j^{(t-1)}$

where $\mathcal{B}\left(m_{jl}^{(t-1)},S\right)$ is a general Beta distribution with location $m_j^{(t-1)}$ and scale S. The third move is based on the generation of the missing data Z_i, $i=1,\ldots,N$ from standard uniform random variables r_n, where $Z_i=j$ if $p(Z_i=1|\ldots)+\ldots,p(Z_i=j-1|\ldots)<r_n\leqslant p(Z_i=1|\ldots)+\ldots+p(Z_i=j|\ldots)$ (see Eq. (16)) (Zhang et al., 2004). The fourth move consists of updating the hyperparameters ϑ, ϖ, ε, ζ. The posterior distribution of ε, ζ, ϑ and ϖ, given by Eqs. 19, 20, 23 and 24, respectively, are not of standard forms. However, it is possible to show that they are log-concave (Applegate & Kannan, 1991) (i.e. it is straightforward to show that the second derivatives of the logarithms of these functions are negative), then the samples generation is based on the adaptive rejection sampling (ARS) (Gilks & Wild, 1993).

3.2.2. Split and merge moves

In move (5), we have to choose between splitting or merging a given component with probabilities a_M and $b_M=1-a_M$, respectively, depending on M. Note that $b_1=0$ and $a_\sigma=0$ (recall that σ is a constant representing the maximum value allowed for M), otherwise $a_M=b_M=0.5$. The merging proposal works as follows: choose two components j_1 and j_2, where $m_{j_1}<m_{j_2}$ with no other $m_j\in[m_{j_1},m_{j_2}]$ (i.e. adjacency condition). If these components are merged, we reduce M by 1, which forms a new components j^* containing all the observation previously allocated to j_1 and j_2 and then creates values for p_{j^*}, s_{j^*}, m_{j^*}, by preserving the first two moments, as follows (see Appendix E)

$$p_{j^*}=p_{j_1}+p_{j_2} \tag{27}$$

$$m_{j^*}=\frac{p_{j_1}m_{j_1}+p_{j_2}m_{j_2}}{p_{j_1}+p_{j_2}} \tag{28}$$

$$s_{j^*}=\frac{p_{j^*}(m_{j^*}-a)(b-m_{j^*})}{p_{j_1}\left(m_{j_1}^2+\frac{(m_{j_1}-a)(b-m_{j_1})}{s_{j_1}+1}\right)+p_{j_2}\left(m_{j_2}^2+\frac{(m_{j_2}-a)(b-m_{j_2})}{s_{j_2}+1}\right)-p_{j^*}m_{j^*}^2}-1 \tag{29}$$

When the decision is to split, we choose a component j^* at random to define two new components j_1 and j_2 having weights and parameters $(p_{j_1},m_{j_1},s_{j_1})$ and $(p_{j_2},m_{j_2},s_{j_2})$, respectively, conforming to Eqs. (27)–(29). According to this transformation, there are 3 degrees of freedom, thus we need to generate 3 random numbers $u=(u_1,u_2,u_3)$ drawn from Beta distributions with parameters (2,2) (2,2) and (1,1), respectively (Richardson & Green, 1997). The split transformations are thus defined as following (see Appendix E)

$$p_{j_1}=u_1p_{j^*}\qquad p_{j_2}=(1-u_1)p_{j^*} \tag{30}$$

$$\begin{aligned}m_{j_1}&=m_{j^*}-u_2\sqrt{\frac{(m_{j^*}-a)(b-m_{j^*})p_{j_2}}{(s_{j^*}+1)p_{j_1}}}\\ m_{j_2}&=m_{j^*}+u_2\sqrt{\frac{(m_{j^*}-a)(b-m_{j^*})p_{j_1}}{(s_{j^*}+1)p_{j_2}}}\end{aligned} \tag{31}$$

$$\begin{aligned}s_{j_1}&=\frac{(m_{j_1}-a)(b-m_{j_1})}{u_3(1-u_2^2)\frac{(m_{j^*}-a)(b-m_{j^*})}{s_{j^*}+1}\frac{p_{j^*}}{p_{j_1}}}-1\\ s_{j_2}&=\frac{(m_{j_2}-a)(b-m_{j_2})}{(1-u_3)(1-u_2^2)\frac{(m_{j^*}-a)(b-m_{j^*})}{s_{j^*}+1}\frac{p_{j^*}}{p_{j_2}}}-1\end{aligned} \tag{32}$$

Note that we have also to check the adjacency condition previously defined for the split move. If this condition passed, then we assign the different x_i previously in j^* in j_1 or j_2 using Eq. (16) (i.e. Bayes rule). If the condition is not satisfied, we reject the move in order to preserve the reversibility of split/combine moves.

Now, we calculate the acceptance probabilities of split and combine moves: $\min\{1,A\}$ and $\min\{1,A^{-1}\}$, where we have the following according to Eq. (26):

$$A=\frac{p(Z,P,M+1,\xi,\vartheta,\varpi,\varepsilon,\zeta|\mathcal{X})b_{M+1}}{p(Z,P,M,\xi,\vartheta,\varpi,\varepsilon,\zeta|\mathcal{X})a_MP_{alloc}q(u)}\left|\frac{\partial\Delta'_M}{\partial(\Delta_M,u)}\right| \tag{33}$$

where P_{alloc} is the probability of making this particular current allocation of data to components j_1 and j_2:

Table 1
Summary of the results for the 100 generated data sets. $\widehat{M}$ denotes the obtained number of clusters.

Number of clusters M	Number of datasets	$\widehat{M}=2$ (%)	$\widehat{M}=3$ (%)	$\widehat{M}=4$ (%)	$\widehat{M}=5$ (%)	$\widehat{M}=6$ (%)	$\widehat{M}=7$ (%)
$M=2$	25	100	0	0	0	0	0
$M=3$	20	0	100	0	0	0	0
$M=4$	15	0	6.67	92.23	0	0	0
$M=5$	15	0	0	0	100	0	0
$M=6$	15	0	0	6.67	6.67	86.66	0
$M=7$	10	0	0	0	0	10	90

$$P_{alloc}=\prod_{Z_i=j_1}\frac{p_{j_1}p(x_i|m_{j_1},s_{j_1})}{p_{j_1}p(x_i|m_{j_1},s_{j_1})+p_{j_2}p(x_i|m_{j_2},s_{j_2})}\prod_{Z_i=j_2}\times\frac{p_{j_2}p(x_i|m_{j_2},s_{j_2})}{p_{j_1}p(x_i|m_{j_1},s_{j_1})+p_{j_2}p(x_i|m_{j_2},s_{j_2})}=\frac{\prod_{Z_i=j_1}p_{j_1}p(x_i|m_{j_1},s_{j_1})\prod_{Z_i=j_2}p_{j_2}p(x_i|m_{j_2},s_{j_2})}{\prod_{Z_i=j^*}p_{j_1}p(x_i|m_{j_1},s_{j_1})+p_{j_2}p(x_i|m_{j_2},s_{j_2})} \quad (34)$$

$$q(u)=p(u_1)p(u_2)p(u_3) \quad (35)$$

$\frac{p(Z,P,M+1,\xi,\vartheta,\varpi,\varepsilon,\zeta|\mathcal{X})}{p(Z,P,M,\xi,\vartheta,\varpi,\varepsilon,\zeta|\mathcal{X})}$ is developed in Appendix E and it is straightforward to show that $\left|\frac{\partial\Delta'_M}{\partial(\Delta_M,u)}\right|$ is given by

$$\left|\frac{\partial\Delta'_M}{\partial(\Delta_M,u)}\right|=\left|\frac{\partial(p_{j_1},p_{j_2},m_{j_1},m_{j_2},s_{j_1},s_{j_2})}{\partial(u_1,p_{j^*},m_{j^*},u_2,s_{j^*},u_3)}\right|=p_{j^*}\frac{(m_{j_2}-m_{j_1})(s_{j_1}+1)(s_{j_2}+1)}{u_2(1-u_2^2)u_3(1-u_3)(s_{j^*}+1)} \quad (36)$$

which is the Jacobian that arises from transforming $(u_1,p_{j^*},m_{j^*},u_2,s_{j^*},u_3)$ to $(p_{j_1},p_{j_2},m_{j_1},m_{j_2},s_{j_1},s_{j_2})$.

3.2.3. Birth and death moves

In move (6), birth-and-death, the first step is to choose randomly between birth and death with probabilities a_M and b_M as above. The birth step consists in adding a new general Beta component in the mixture by generating its parameters, m_{j^*} and s_{j^*}, from the associated prior distributions given by Eqs. (8) and (9), respectively. The weight of the new component, p_{j^*}, is generated from the marginal distribution of p_{j^*} derived from the distribution of $P=((p_1,\ldots,p_M,p_{j^*})$. The vector P follows a Dirichlet with parameters $(\delta_1,\ldots,\delta_M,\delta_{j^*})$ (see Eq. (13)), thus the marginal of p_{j^*} is a Beta distribution with parameters $(\delta_{j^*},\sum_{j=1}^M\delta_j)$ (Johnson et al., 1995). Note that in order to keep the mixture constraint $\sum_{j=1}^M p_j+p_{j^*}=1$, the previous weights $p_j, j=1,\ldots,M$ have to be rescaled and then all multiplied by $(1-p_{j^*})$. The Jacobian corresponding to the birth move is then $(1-p_{j^*})^M$. For the opposite move, we choose randomly an existing empty component to delete, then of course the remaining weights have to be rescaled to keep the unit-sum constraint. The acceptance probabilities of birth and death moves: $\min\{1,A\}$ and $\min\{1,A^{-1}\}$, are calculated according to Eq. (26) (see Appendix E):

$$A=\frac{p(M+1)}{p(M)}\frac{\Gamma\left(\delta_{j^*}+\sum_{j=1}^M\delta_j\right)}{\Gamma(\delta_{j^*})\Gamma\left(\sum_{j=1}^M\delta_j\right)}p_{j^*}^{\delta_{j^*}-1}(1-p_{j^*})^{N+\sum_{j=1}^M\delta_j-M}(M+1)\times\frac{b_{M+1}}{a_M(M_0+1)}\frac{1}{p(p_{j^*})}(1-p_{j^*})^M \quad (37)$$

where M_0 is the number of empty components before the birth.

4. Experimental results

In this section we report results on different interesting applications. In the first application, we briefly discuss the results obtained with some artificially generated data sets. The discussion will not be very detailed since, the experiments with generated data are not as significant as those with real data. We investigate the effectiveness of our algorithm by applying it on four well-known real data sets, while comparing it to the RJMCMC in the case of Gaussian mixture model (Richardson & Green, 1997) in the second application. Last but not least, we demonstrate the usefulness of our algorithm for texture image classification and retrieval. In these applications our specific choices for the hyperparameters were $\eta_1=\ldots,\eta_M=1$, $(\varphi,\varrho,\lambda,\mu,\phi)=(2,5,0.2,2,1)$,$S$ and e^2 (in the M–H algorithms) were set to 2 and 0.01, respectively, and σ (the maximum value allowed for M) was set to 30.

4.1. Synthetic data sets

We dedicate this section for the analysis of generated data. The goal of this section is to investigate if our algorithm is able to: estimate the mixture parameters and select the number of clusters effectively. We generated 100 data sets using different parameters and number of clusters, in order to investigate the method on a wide range of data. Applying our methods on these data sets, we found that it was able to identify the right number of clusters in 96% of the cases as shown in Table 1. For the 4% wrongly modeled data sets, we can notice that they were fitted with a smaller number of components. It is noteworthy that in each of these four cases the probabilities for the right number of components were quite close to the ones chosen. We choose four data sets for more in details inspection. The real and estimated parameters for these data sets are given in Table 2, and the real and estimated histograms are drawn in Fig. 2. According to these results it is clear that our algorithm has very good learning capabilities.

4.2. Real data sets

We devote this section for real data modeling and analysis. We apply our algorithm on four standard widely used data sets:

Table 2
Parameters of four different generated data sets. N represents the number of elements in each data set. m_j, s_j, and p_j are the real parameters. $\hat{m}_j,\hat{s}_j$, and $\hat{p}_j$ are the estimated parameters.

	j	m_j	s_j	p_j	$\hat{m}_j$	$\hat{s}_j$	$\hat{p}_j$
Data 1 ($N=3365$)	1	2.00	10.00	0.60	1.94	12.31	0.57
	2	5.00	19.00	0.40	4.90	17.71	0.43
Data 2 ($N=3647$)	1	1.00	12.00	0.35	0.90	15.11	0.34
	2	3.50	22.00	0.35	3.35	26.00	0.35
	3	6.00	13.00	0.30	5.90	16.57	0.31
Data 3 ($N=3703$)	1	1.00	15.00	0.10	1.17	18.90	0.11
	2	3.00	14.00	0.30	3.01	12.70	0.28
	3	5.00	19.00	0.20	4.85	18.79	0.20
	4	6.50	17.00	0.40	6.51	23.40	0.41
Data 4 ($N=3706$)	1	1.00	15.00	0.10	1.11	19.22	0.12
	2	2.00	25.00	0.20	2.03	23.91	0.19
	3	4.00	14.00	0.30	3.82	15.83	0.28
	4	5.00	19.00	0.20	5.14	19.05	0.21
	5	6.50	17.00	0.20	6.53	16.69	0.20

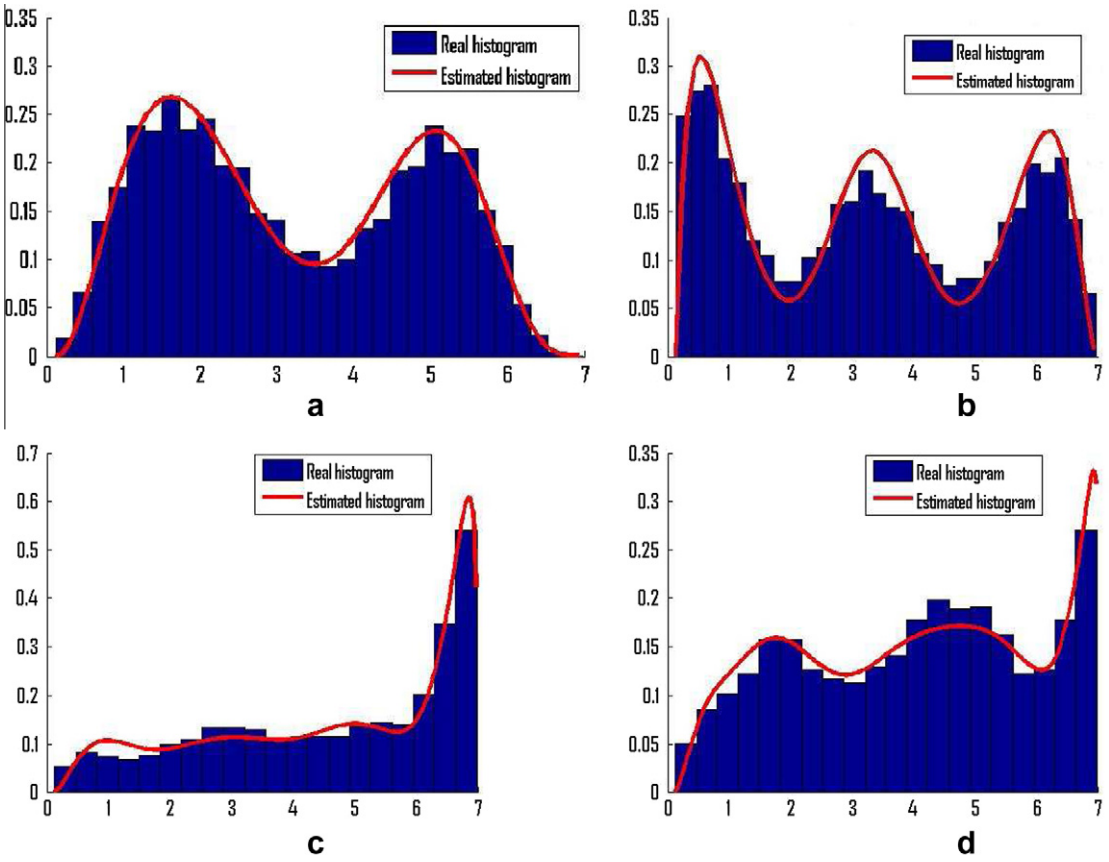

Fig. 2. Real and estimated histograms for four generated data sets. (a) a 2 components mixture, (b) a 3 components mixture, (c) a 4 components mixture, (d) a 5 components mixture.

Table 3
Estimated posterior probabilities of the number of components given the data for the four data sets, with percentage of accepted split–combine, and birth–death moves.

Data set	N	$p(k\|\mathcal{X})$	Proportion (%) of split–combine moves accepted	Proportion (%) of birth–death moves accepted
Enzyme	245	$p(1\|\mathcal{X}) = 0.0000$ $p(2\|\mathcal{X}) = 0.1712$ $p(3\|\mathcal{X}) = 0.4152$ $p(4\|\mathcal{X}) = 0.3782$ $p(5\|\mathcal{X}) = 0.0341$ $\sum_{k\geqslant 6}^{\sigma} p(k\|\mathcal{X}) = 0.0013$	6.68	3.02
Acidity	155	$p(1\|\mathcal{X}) = 0.0000$ $p(2\|\mathcal{X}) = 0.4307$ $p(3\|\mathcal{X}) = 0.3354$ $p(4\|\mathcal{X}) = 0.1942$ $p(5\|\mathcal{X}) = 0.0231 p(6\|\mathcal{X}) = 0.0154$ $\sum_{k\geqslant 7}^{\sigma} p(k\|\mathcal{X}) = 0.0012$	10.17	7.52
Galaxy	82	$p(1\|\mathcal{X}) = 0.0000$ $p(2\|\mathcal{X}) = 0.0010$ $p(3\|\mathcal{X}) = 0.0210$ $p(4\|\mathcal{X}) = 0.2031$ $p(5\|\mathcal{X}) = 0.3671 p(6\|\mathcal{X}) = 0.0798$ $p(7\|\mathcal{X}) = 0.3167$ $p(8\|\mathcal{X}) = 0.0094$ $\sum_{k\geqslant 9}^{\sigma} p(k) = 0.0019$	9.32	16.71
Stamp	485	$p(1\|\mathcal{X}) = 0.0000$ $p(2\|\mathcal{X}) = 0.0000$ $p(3\|\mathcal{X}) = 0.0001$ $p(4\|\mathcal{X}) = 0.5612$ $p(5\|\mathcal{X}) = 0.3574 p(6\|\mathcal{X}) = 0.0231$ $p(7\|\mathcal{X}) = 0.0556$ $p(8\|\mathcal{X}) = 0.0012$ $\sum_{k\geqslant 9}^{\sigma} p(k\|\mathcal{X}) = 0.0014$	4.87	2.16

enzyme, acidity, galaxy, and stamp [4]. The first data describes an enzymatic activity in the blood among a group of 245 unrelated individuals, and the second one is an acidity index measured in a sample of 155 lakes in the northeastern United States. The third one consists of the velocities of 82 distant galaxies, diverging from our own galaxy, as for the last data set it consists of thickness of 485 postage stamps produced in Mexico. The enzyme data set was analyzed in several research papers such as in Bechtel, Bonaiti-Pellie, Poisson, Magnette, & Bechtel (1993) where it was modeled by two skewed distributions, and in Richardson & Green (1997) where the use of three to five Gaussian components was favored. For the acidity data and Galaxy data sets, three to five components were generally identified (Richardson & Green, 1997). The 1872 Hidalgo postage stamps of Mexico data set was introduced in Izenman & Sommer (1988) and has been used in several research papers (see, for instance, Basford, McLachlan, & York, 1997; Yang & Liu, 2002) which identified seven and three components Gaussian model with equal and unequal variances, respectively. Using these four datasets we compared our model to the one in Richardson & Green (1997). In all the runs the number of components has never exceeded fifteen. Estimated posterior probabilities of the number of components given the data for the four data are given in Table 3. For the enzyme data our algorithm favors 3–5 components with maximum posterior probability

[4] http://www.maths.uq.edu.au/~gjm/DATA/mmdata.html.

for three components, same as the GMM, this is due to the fact that the enzyme data are not skewed (see Fig. 3). For the acidity data set a mixture of two components was chosen as shown in Fig. 4. For the galaxy and stamp data sets the data are highly skewed and spread which force the algorithm to use a higher number of components, for this reason our algorithm supports the use of five components for both data sets (see Figs. 5 and 6). In each case, we can relate the number of components to the skewness to the data. Also note that the general Beta can model skewed data which is not the case for the Gaussian, and this advantage is demonstrated for the acidity and galaxy datasets where our algorithm favors the use of two and five components, respectively, compared to three and six for the GMM (see Table 4). According to the experiments presented here it is clear that the general Beta mixture model outperforms the Gaussian one, by representing the data effectively with less number of components. This result was already expected due to the fact that the general Beta mixture model is more flexible which helps it to represent highly spread and hard to model data.

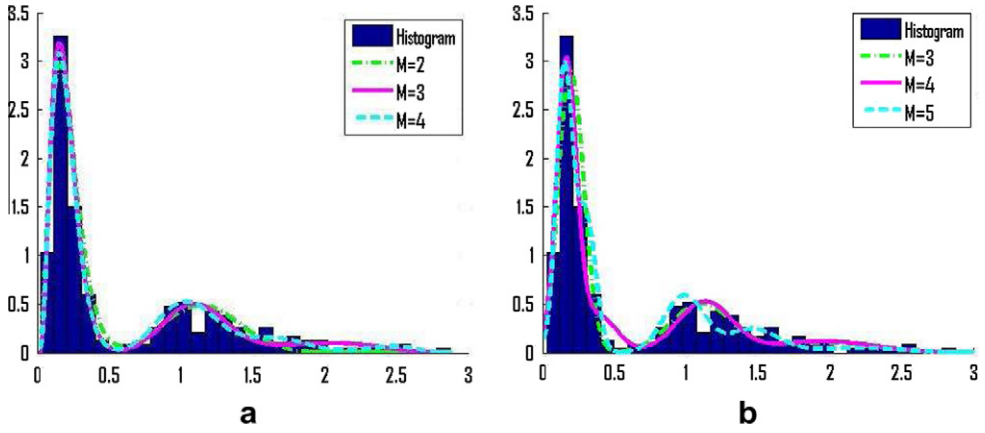

Fig. 3. Enzyme data modeling when considering the mixtures with the highest probabilities. (a) Beta mixture models. (b) Gaussian mixture models.

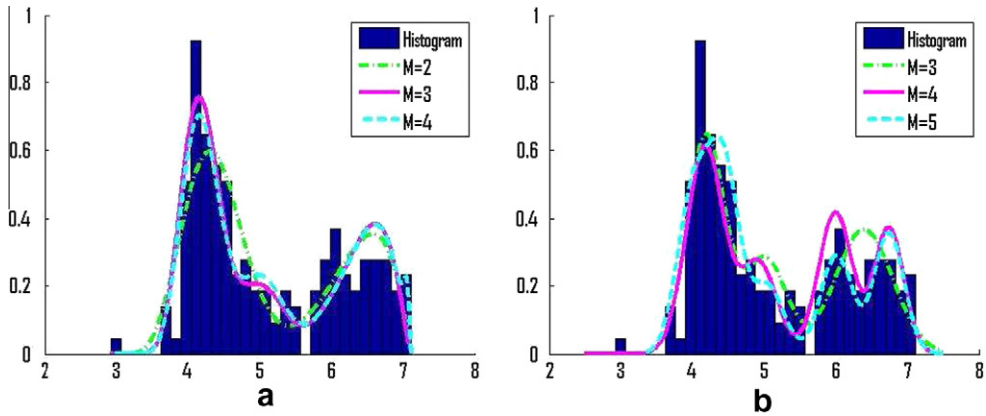

Fig. 4. Acidity data modeling when considering the mixtures with the highest probabilities. (a) Beta mixture models. (b) Gaussian mixture models.

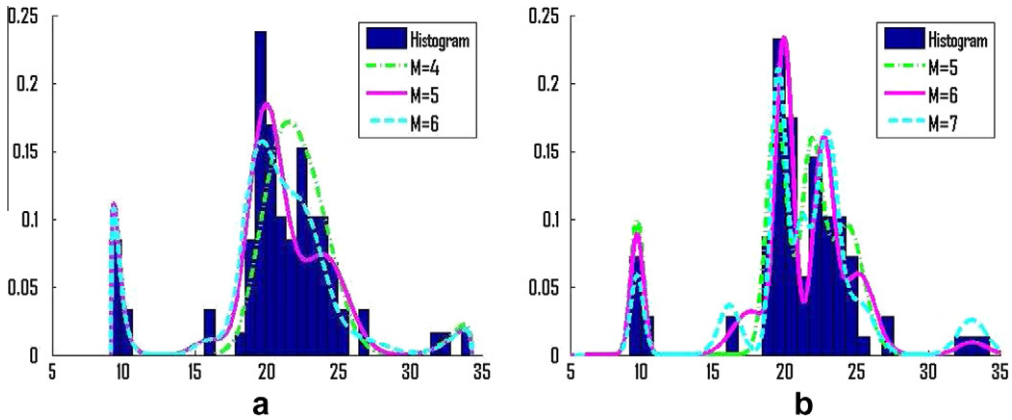

Fig. 5. Galaxy data modeling when considering the mixtures with the highest probabilities. (a) Beta mixture models. (b) Gaussian mixture models.

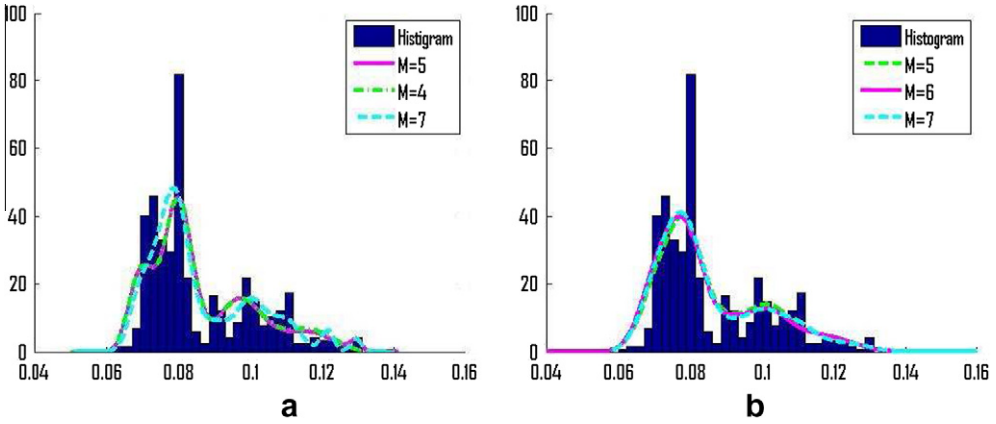

Fig. 6. Stamp data modeling when considering the mixtures with the highest probabilities. (a) Beta mixture models. (b) Gaussian mixture models.

Table 4
Parameters of the mixture models representing the different tested real data sets. j component number. m_j, s_j, and p_j are the real parameters. $\hat{m}_j, \hat{s}_j$, and $\hat{p}_j^{\beta}$ are the Beta mixture estimated parameters. $\hat{\mu}_j, \hat{\sigma}_j^2$, and $\hat{p}_j^{g}$ are the Gaussian mixture estimated parameters.

	j	$\hat{m}_j$	$\hat{s}_j$	$\hat{p}_j^{\beta}$	$\hat{\mu}_j$	$\hat{\sigma}_j^2$	$\hat{p}_j^{g}$
Enzyme	1	0.1960	64.1054	0.6450	0.1962	0.0078	0.6431
	2	1.1008	42.9680	0.2630	1.1006	0.0448	0.2637
	3	1.9492	13.4360	0.0920	1.9492	0.1271	0.0932
Acidity	1	4.3615	23.7136	0.3500	4.1865	0.0691	0.4240
	2	6.3149	11.8544	0.3854	5.0168	0.0868	0.2086
	3				6.3926	0.1593	0.3674
Galaxy	1	9.7101	69.0788	0.0854	9.7101	0.1931	0.0988
	2	16.1737	106.9899	0.0320	17.5649	1.5392	0.1006
	3	20.0898	109.0967	0.5366	19.9782	0.3540	0.3416
	4	24.1064	54.3933	0.3095	22.7039	0.4731	0.2692
	5	32.9518	28.1897	0.0366	25.1517	1.2112	0.1655
	6				33.0427	1.0256	0.0243
Stamp	1	0.0705	103.0930	0.2202	0.0710	0.2320×10^{-4}	0.2133
	2	0.0803	129.4502	0.4270	0.0789	0.2392×10^{-4}	0.4152
	3	0.0971	57.5257	0.2094	0.0907	0.2709×10^{-4}	0.1015
	4	0.1140	28.4719	0.1168	0.1016	0.3080×10^{-4}	0.1666
	5	0.1284	580.7500	0.0265	0.1157	0.7166×10^{-4}	0.1035

4.3. Texture images classification and retrieval

4.3.1. Approach

An interesting difficult problem in image processing is texture analysis. Indeed, texture provides important characteristics for surface and object identification (depth and orientation, for instance) in many types of images (satellite, medical, etc.) and plays an important role in several applications such as content-based image categorization, browsing and retrieval (Manjunath & Ma, 1996). Methods for texture analysis can be grouped into four major categories: statistical, geometrical, model based and signal processing approaches (Tuceryan & Jain, 1998). An efficient technique for analyzing image textures is the multichannel decomposition approach, using the assumption that the energy distribution in the frequency domain identifies texture, based on Gabor filters (Bovik, Clark, & Geisler, 1990; Dunn, Higgins, & Wakeley, 1994; Dunn & Higgins, 1995; Grigorescu, Petkov, & Kruizinga, 2002), wavelet transforms (Chang & Kuo, 1993; Laine & Fan, 1993; Unser, 1995; Wouwer, Scheunders, & Dyck, 1999) and steerable pyramids (Freeman & Adelson, 1991; Simoncelli & Freeman, 1995; Wu, Chan, & Huang, 2003). A detailed survey and intersting discussions can be found in Randen & Husøy (1999), also. The texture can then be modeled by the marginal densities of the coefficients of the resulted filtered subband images which allows a more compact representation than histograms which necessitate many parameters (hundreds). This approach is also justified by some psychological researches on human texture perception which have shown that textures producing similar marginal densities are very difficult to discriminate (Do & Vetterli, 2002). It is noteworthy that the sub-

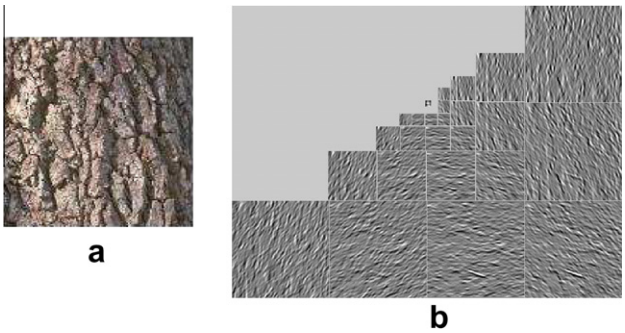

Fig. 7. Original image and its steerable pyramid decomposition. (a) Original image from Bark group in Vistex. (b) Sub-images output using five level steerable pyramid.

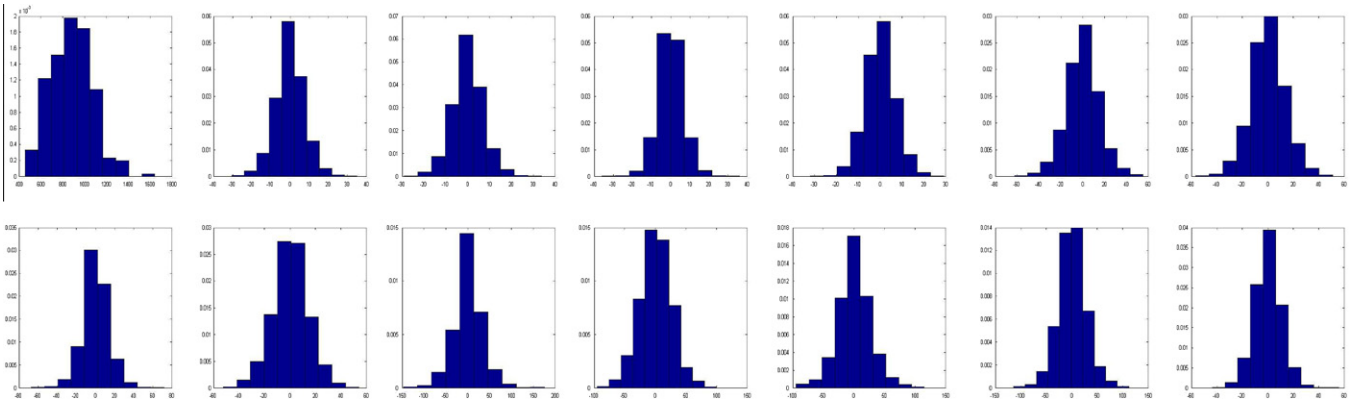

Fig. 8. Histograms of the 14 sub-images of the steerable pyramid.

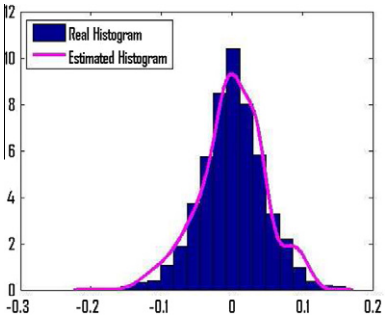

Fig. 9. A sub-image histogram fitted by a Beta mixture model.

band marginal densities are generally non-Gaussian especially for natural images texture (Wainwright & Simoncelli, 2000). In this section we propose an approach for texture images classification and retrieval based on our finite general Beta mixture model. In our classification framework, an image texture is first decomposed into sub-bands using steerable filters (Freeman & Adelson, 1991; Simoncelli & Freeman, 1995). Fig. 7 shows a texture image and its multiscale version in a pyramid hierarchy. The histograms of the resulted filtered images are show in Fig. 8 which shows clearly that the Gaussian assumption would be inappropriate. Then, each sub-band's marginal density is approximated by a finite general Beta mixture model using our Bayesian learning algorithm (see Fig. 9). As a result each texture image will be represented by a set of finite general Beta mixture models which can be viewed as the signatures of the image. Finally the Earth Mover's Distance (EMD) (Rubner, Tomasi, & Guibas, 2000) is used to measure the distribution similarity between a set of components representing an input image texture (i.e. test image) and sets of components representing texture classes (i.e. training images). In our case, EMD can be viewed as the minimum cost of changing one mixture into another, when the cost of moving probability mass from components in the first mixture to components in the second mixture is calculated using Kullback–Leibler (KL) divergence given by:

$$D(f_i||g_j) = \int f_i(x) \log\left(\frac{f_i(x)}{g_j(x)}\right) dx \tag{38}$$

where f_i is the component i of the input sub-image mixture which we suppose that it has m components with weights p_{fi}, and g_j is the component j of the class sub-image mixture which we suppose that it has n components with weights p_{gj}. For two general Beta distributions f_i and g_j the KL divergence has a closed form expression and we can show that is given by (see Appendix E)

$$\begin{aligned} D(f_i||g_j) = \log & \left[\frac{\Gamma(\alpha_i+\beta_i)\Gamma(\alpha_j)\Gamma(\beta_j)}{(b-a)^{\alpha_i+\beta_i-\alpha_j-\beta_j}\Gamma(\alpha_j+\beta_j)\Gamma(\alpha_i)\Gamma(\beta_i)}\right] \\ & -(\beta_j-\beta_i+\alpha_j-\alpha_i)\log(b-a) \\ & +(\alpha_j-\alpha_i+\beta_j-\beta_i)\Psi(\alpha_i+\beta_i) \\ & -(\alpha_j-\alpha_i)\Psi(\alpha_i)-(\beta_j-\beta_i)\Psi(\beta_i) \end{aligned} \tag{39}$$

where (α_i, β_i) are the parameters of f_i, (α_j, β_j) are the parameters of g_j, and Ψ is the digamma function. With the KL divergence in hand we have to start the minimization problem in which we need to get the $m \times n$ matrix F, where f_{ij} is the amount of weight p_{fi} matched to p_{gj}, that will minimize the following equation

$$EMD_{sub} = \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} D(f_i||g_j) \tag{40}$$

and subjected to the following constraints: (1) $f_{ij} \geqslant 0$, where $1 \leqslant i \leqslant m$ and $1 \leqslant j \leqslant n$, (2) $\sum_{i=1}^{m} f_{ij} = p_{gj}$, where $1 \leqslant j \leqslant n$, (3) $\sum_{j=1}^{n} f_{ij} = p_{fi}$, where $1 \leqslant i \leqslant m$, (4) $\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = min(\sum_{i=1}^{m} p_{fi}, \sum_{j=1}^{n} p_{gj}) = 1$. Note that when the image texture is decomposed of L sub-bands, then the total EMD is the sum of that of each sub-band, $EMD = \sum_{l=1}^{L} EMD_{sub_l}$. By computing the EMD between the input texture image and each texture class, each image is affected to the class for which the EMD is the smallest.

4.3.2. Results

We performed our classifications experiments using the Vistex data set [5]. Six homogeneous texture groups (Bark, Fabric, Food, Metal, Water, and Sand) were considered (see Fig. 10). We used four 512×512 images from each of the Bark, Fabric, and Metal texture groups, and six 512×512 from each of the Food, Water, and Sand texture groups, then we divided each image into sixty-four 64×64 subimages. Thus, we obtained a total of 256 subimages for each class in the first three groups, and 384 subimages for each class in the second three groups. We then applied our classification approach 10 times, each time using 24 subimages of each original texture image for training and the remaining 40 for testing. This brought us to a total of 720 images from all six groups as training samples for our algorithm, and 1200 as testing samples. Moreover, we applied our algorithm by using first three levels pyramid and second by using five levels pyramid. The classification results, when using both finite general Beta and Gaussian mixture models, are given in Table 5. From these results we can observe that our algorithm has a higher accuracy then the Gaussian which is a further endorsement of our model. In addition, and as expected, the five levels pyramid improves the performance over the three levels

[5] MIT Vision and Modeling Group (http://vismod.www.media.mit.edu).

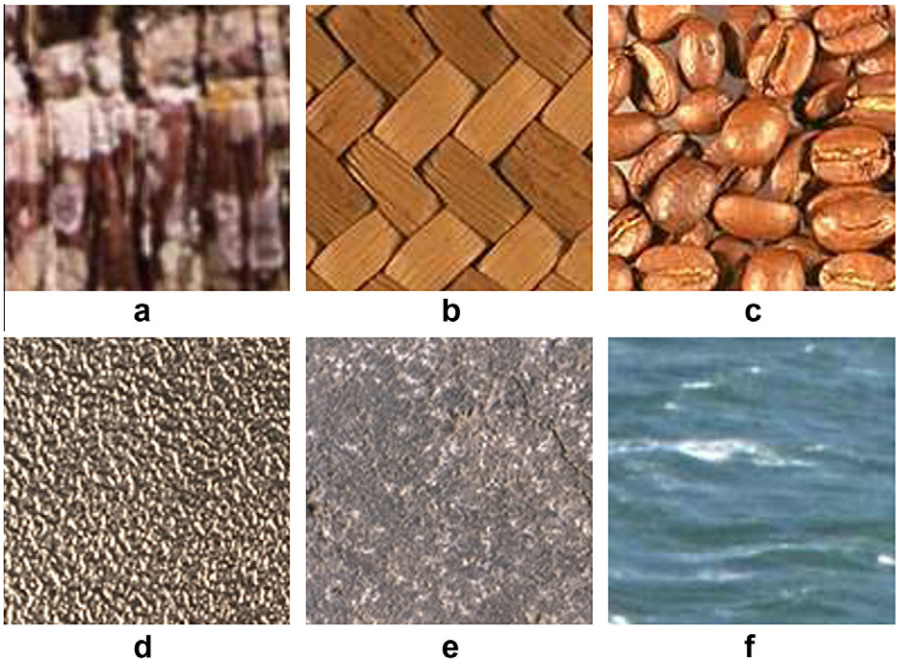

Fig. 10. Sample images from each group. (a) Bark. (b) Fabric. (c) Food. (d) Metal. (e) Sand. (f) Water.

Table 5
The average classification accuracy (%) of the two different methods.

Method	Using 3 levels pyramid (%)	Using 5 levels pyramid (%)
General Beta mixture models	92.50 ± 1.41	93.58 ± 1.33
Gaussian mixture models	91.67 ± 1.66	92.58 ± 1.08

Table 6
Average precision rate (%) of the two different methods.

Method	Using 3 levels pyramid (%)	Using 5 levels pyramid (%)
General Beta mixture models	75.32	78.12
Gaussian mixture models	71.56	73.32

pyramids, yet this improvement is very small compared to the enormous difference in computational time.

We conducted another experiment designed to retrieve images similar to a given query. Our retrieval approach can be divided into two steps. First task, is the same as in the classification approach, since we have to choose the nearest texture group to the query. For the second step, we compared the input image (i.e., query) with the other images in the same group and retrieve the closest images to our query using the EMD. We applied our retrieval process twice former using three levels pyramid and latter using five levels pyramid. To measure the retrieval rates (precision and recall), each image was used as a query and the number of relevant images among those that were retrieved was noted. Table 6 presents the retrieval rates obtained in terms of precision when 64 images are retrieved each time in response to a query. Note that in this case the precision and recall are the same because for a given image

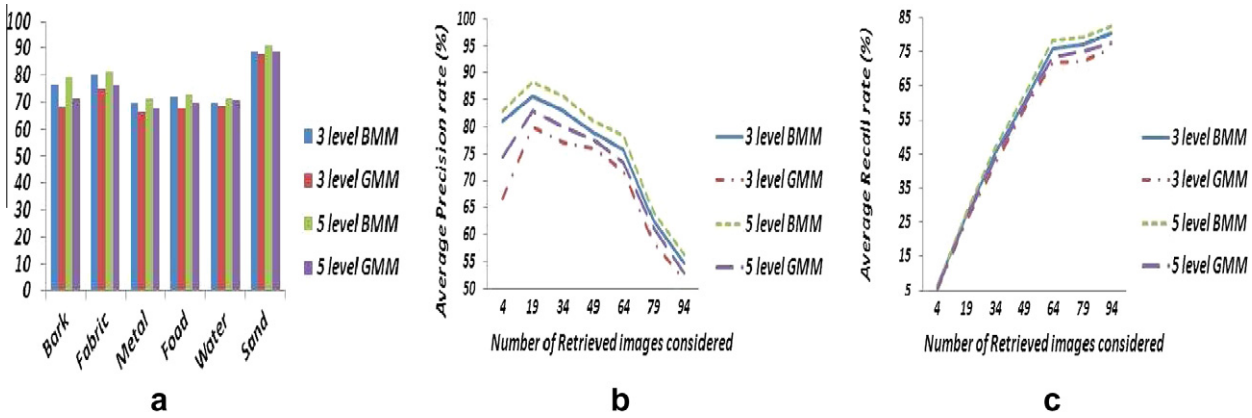

Fig. 11. Precision and recall. (a) Average precision when 64 retrieved images are considered for each class. (b) Average precision when varying the number of retrieved images. (c) Average recall when varying the number of retrieved images.

we have at most 64 images which are similar to it. Fig. 11(a) represents the average (averaged over all the queries) precision rate for different texture classes when we consider only the first 64 images retrieved. Fig. 11(b) and (c) show two graphs illustrating the overall precision and recall, respectively, of our retrieval method when varying the total number of images retrieved taken into consideration.

5. Conclusion

We presented a fully Bayesian analysis, coupled with MCMC techniques, of finite Beta mixtures with unknown number of components. The proposed algorithm automatically handles the problem of the specification of the number of clusters on the basis of the RJMCMC approach, which allows varying the dimension of the mixture, by constructing split and merge moves that rely on moment matching. The results from applying the proposed model to different applications have been presented and justify further the recent interest on the use of Bayesian machinery in image processing. The finite Beta mixture has many appealing advantages that make it useful for a variety of image processing applications which require the modeling of non Gaussian data.

Acknowledgment

The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC) and a NATEQ Nouveaux Chercheurs Grant.

Appendix A. Proof of Eq. (29)

We can show that the new variance v_{j^*} for component j^* satisfies (Richardson & Green, 1997)

$$p_{j^*}(m_{j^*}^2 + v_{j^*}) = p_{j_1}\left(m_{j_1}^2 + v_{j_1}\right) + p_{j_2}\left(m_{j_2}^2 + v_{j_2}\right) \tag{41}$$

Besides, according to Eq. (2) we have

$$\alpha_j = \frac{m_j - a}{b-a}s_j \quad \beta_j = s_j - \frac{m_j - a}{b-a}s_j = s_j\left(1 - \frac{m_j - a}{b-a}\right) = \frac{b-m_j}{b-a}s_j$$

Using the two previous equations, Eq. (3) becomes

$$\begin{aligned} v_j &= (b-a)^2\frac{\alpha_j\beta_j}{(\alpha_j+\beta_j)^2(\alpha_j+\beta_j+1)} = (b-a)^2\frac{\alpha_j\beta_j}{s_j^2(s_j+1)} \\ &= \frac{(m_j-a)(b-m_j)}{s_j+1} \end{aligned} \tag{42}$$

substituting the previous equation into Eq. (41), we obtain

$$\begin{aligned} p_{j^*}\left(m_{j^*}^2 + \frac{(m_{j^*}-a)(b-m_{j^*})}{s_{j^*}+1}\right) &= p_{j_1}\left(m_{j_1}^2 + \frac{(m_{j_1}-a)(b-m_{j_1})}{s_{j_1}+1}\right) \\ &\quad + p_{j_2}\left(m_{j_2}^2 + \frac{(m_{j_2}-a)(b-m_{j_2})}{s_{j_2}+1}\right) \end{aligned}$$

Thus,

$$\frac{(m_{j^*}-a)(b-m_{j^*})}{s_{j^*}+1} = \frac{p_{j_1}\left(m_{j_1}^2 + \frac{(m_{j_1}-a)(b-m_{j_1})}{s_{j_1}+1}\right) + p_{j_2}\left(m_{j_2}^2 + \frac{(m_{j_2}-a)(b-m_{j_2})}{s_{j_2}+1}\right)}{p_{j^*}} - m_{j^*}^2,$$

and

$$s_{j^*} = \frac{p_{j^*}(m_{j^*}-a)(b-m_{j^*})}{p_{j_1}\left(m_{j_1}^2 + \frac{(m_{j_1}-a)(b-m_{j_1})}{s_{j_1}+1}\right) + p_{j_2}\left(m_{j_2}^2 + \frac{(m_{j_2}-a)(b-m_{j_2})}{s_{j_2}+1}\right) - p_{j^*}m_{j^*}^2} - 1$$

Appendix B. Proof of Eq. (32)

When we split a component j^* to define two new components j_1 and j_2 having weights and parameters $(p_{j_1}, m_{j_1}, s_{j_1})$ and $(p_{j_2}, m_{j_2}, s_{j_2})$, respectively, we can set the following (Richardson & Green, 1997)

$$v_{j_1} = u_3(1-u_2^2)v_{j^*}\frac{p_{j^*}}{p_{j_1}} \tag{43}$$

$$v_{j_2} = (1-u_3)(1-u_2^2)v_{j^*}\frac{p_{j^*}}{p_{j_2}} \tag{44}$$

By substituting Eq. (42) into Eq. (43), we obtain $\frac{(m_{j_1}-a)(b-m_{j_1})}{s_{j_1}+1} = u_3(1-u_2^2)\frac{(m_{j^*}-a)(b-m_{j^*})}{s_{j^*}+1}\frac{p_{j^*}}{p_{j_1}}$, thus

$$s_{j_1} = \frac{(m_{j_1}-a)(b-m_{j_1})}{u_3(1-u_2^2)\frac{(m_{j^*}-a)(b-m_{j^*})}{s_{j^*}+1}\frac{p_{j^*}}{p_{j_1}}} - 1$$

By substituting Eq. (42) into Eq. (44), we obtain $\frac{(m_{j_2}-a)(b-m_{j_2})}{s_{j_2}+1} = (1-u_3)(1-u_2^2)\frac{(m_{j^*}-a)(b-m_{j^*})}{s_{j^*}+1}\frac{p_{j^*}}{p_{j_1}}$, thus

$$s_{j_2} = \frac{(m_{j_2}-a)(b-m_{j_2})}{(1-u_3)(1-u_2^2)\frac{(m_{j^*}-a)(b-m_{j^*})}{s_{j^*}+1}\frac{p_{j^*}}{p_{j_1}}} - 1$$

Appendix C

$$\begin{aligned} &\frac{p(Z,P,M+1,\xi,\vartheta,\varpi,\varepsilon,\zeta|\mathcal{X})}{p(Z,P,M,\xi,\vartheta,\varpi,\varepsilon,\zeta|\mathcal{X})} \\ &= (\text{likelihood ratio}) \\ &\quad\times\frac{(M+1)!p(M+1)p(P|M+1,\delta)p(Z|P,M+1)p(\xi|M+1,\eta)}{M!p(M)p(P|M,\delta)p(Z|P,M)p(\xi|M,\eta)} \\ &\quad\times p(\varepsilon)p(\zeta|\varphi,\varrho)p(\vartheta|\lambda,\mu)p(\varpi|\phi) \end{aligned} \tag{45}$$

where "likelihood" ratio is the ratio of the likelihood using the new parameter set, corresponding to $M+1$ components, to that for the old one corresponding to M components:

$$\text{likelihood ratio} = \frac{\prod_{i=1,Z_i=j_1}^{N}p(x_i|\Theta)\prod_{i=1,Z_i=j_2}^{N}p(x_i|\Theta)}{\prod_{i=1,Z_i=j^*}^{N}p(x_i|\Theta)} \tag{46}$$

$$\begin{aligned} \frac{p(P|M+1,\delta)}{p(P|M,\delta)} &= \frac{\frac{\Gamma\left(\sum_{j=1}^{M+1}\delta_j\right)}{\prod_{j=1}^{M+1}\Gamma(\delta_j)}\prod_{j=1}^{M+1}p_j^{\delta_j-1}}{\frac{\Gamma\left(\sum_{j=1}^{M}\delta_j\right)}{\prod_{j=1}^{M}\Gamma(\delta_j)}\prod_{j=1}^{M}p_j^{\delta_j-1}} \\ &= \frac{\frac{\Gamma\left(\sum_{j=1}^{M-1}\delta_j+\delta_{j_1}+\delta_{j_2}\right)}{\prod_{j=1}^{M-1}\Gamma(\delta_j)\Gamma(\delta_{j_1})\Gamma(\delta_{j_2})}\prod_{j=1}^{M-1}p_j^{\delta_j-1}p_{j_1}^{\delta_{j_1}-1}p_{j_2}^{\delta_{j_1}-1}}{\frac{\Gamma\left(\sum_{j=1}^{M-1}\delta_j+\delta_{j^*}\right)}{\prod_{j=1}^{M-1}\Gamma(\delta_j)\Gamma(\delta_{j^*})}\prod_{j=1}^{M-1}p_j^{\delta_j-1}p_{j^*}^{\delta_{j^*}-1}} \\ &= \frac{\frac{\Gamma\left(\sum_{j=1}^{M-1}\delta_j+\delta_{j_1}+\delta_{j_2}\right)}{\Gamma(\delta_{j_1})\Gamma(\delta_{j_2})}p_{j_1}^{\delta_{j_1}-1}p_{j_2}^{\delta_{j_1}-1}}{\frac{\Gamma\left(\sum_{j=1}^{M-1}\delta_j+\delta_{j^*}\right)}{\Gamma(\delta_{j^*})}p_{j^*}^{\delta_{j^*}-1}} \\ &= \frac{\Gamma\left(\sum_{j=1}^{M-1}\delta_j+\delta_{j_1}+\delta_{j_2}\right)\Gamma(\delta_{j^*})p_{j_1}^{\delta_{j_1}-1}p_{j_2}^{\delta_{j_1}-1}}{\Gamma(\delta_{j_1})\Gamma(\delta_{j_2})\Gamma\left(\sum_{j=1}^{M-1}\delta_j+\delta_{j^*}\right)p_{j^*}^{\delta_{j^*}-1}} \end{aligned} \tag{47}$$

$$\frac{p(Z|P,M+1)}{p(Z|P,M)}=\frac{p_{j_1}^{n_{j_1}}p_{j_2}^{n_{j_2}}\prod_{j=1}^{M-1}p_j^{n_j}}{p_{j^*}^{n_{j^*}}\prod_{j=1}^{M-1}p_j^{n_j}}=\frac{p_{j_1}^{n_{j_1}}p_{j_2}^{n_{j_2}}}{p_{j^*}^{n_{j^*}}} \tag{48}$$

where n_{j_1} and n_{j_2} are the numbers of observations to be assigned to components j_1 and j_2.

$$\frac{p(\xi|M+1,\eta)}{p(\xi|M,\eta)}=\frac{\varpi^{\vartheta}\Gamma(\zeta)\exp\left(-\varpi\left(\frac{s_{j_1}+s_{j_2}}{s_{j_1}s_{j_2}}-\frac{1}{s_{j^*}}\right)\right)\left(\frac{(m_{j_1}-a)(m_{j_2}-a)}{m_{j^*}-a}\right)^{\frac{\zeta(\varepsilon-a)}{b-a}-1}\left(\frac{(b-m_{j_1})(b-m_{j_2})}{b-m_{j^*}}\right)^{\zeta(1-\frac{\varepsilon-a}{b-a})-1}}{\left(\frac{s_{j_1}s_{j_2}}{s_{j^*}}\right)^{\vartheta+1}\Gamma(\vartheta)(b-a)^{(\zeta-1)}\left[\Gamma\left(\frac{\zeta(\varepsilon-a)}{b-a}-1\right)\Gamma\left(\zeta(1-\frac{\varepsilon-a}{b-a})\right)\right]} \tag{49}$$

and where the term $M!$ arises due to the exchangeability of the priors of the ξ parameters. Indeed, it is known that label-switching is of important concern, and numerous papers have discussed this subject. In our case we have adopted a simple approach that has been found effective in practice and according to our experimental results. Indeed, we impose an identifiability constraint on the parameter space which is $m_1 \leqslant m_2 \leqslant \cdots \leqslant m_M$. It is noteworthy that using this constraint results in $M!$ ways of labeling the mixture components.

Appendix D

According to Eq. (26), we have the following in the case of the birth of an empty component, where now $(p_{j^*}, m_{j^*}, s_{j^*})$ play the role of u:

$$A=\frac{p(Z,P,M+1,\xi,\vartheta,\varpi,\varepsilon,\zeta|\mathcal{X})b_{M+1}}{p(Z,P,M,\xi,\vartheta,\varpi,\varepsilon,\zeta|\mathcal{X})a_M p(p_{j^*})p(m_{j^*})p(s_{j^*})}\left|\frac{\partial\Delta'_M}{\partial(\Delta_M,u)}\right| \tag{50}$$

$\frac{p(Z,P,M+1,\xi,\vartheta,\varpi,\varepsilon,\zeta|\mathcal{X})}{p(Z,P,M,\xi,\vartheta,\varpi,\varepsilon,\zeta|\mathcal{X})}$ is developed in Appendix E. In the birth case, however, the likelihood ratio is 1 and we have also

$$\frac{p(P|M+1,\delta)}{p(P|M,\delta)}=\frac{\frac{\Gamma\left(\sum_{j=1}^{M}\delta_j\right)\Gamma(\delta_{j^*}+\sum_{j=1}^{M}\delta_j)}{\Gamma(\delta_{j^*})\Gamma\left(\sum_{j=1}^{M}\delta_j\right)\prod_{j=1}^{M}\Gamma(\delta_j)}\prod_{j=1}^{M}(p_j(1-p_{j^*}))^{\delta_j-1}p_{j^*}^{\delta_{j^*}-1}}{\frac{\Gamma\left(\sum_{j=1}^{M}\delta_j\right)}{\prod_{j=1}^{M}\Gamma(\delta_j)}\prod_{j=1}^{M}p_j^{\delta_j-1}}$$

$$=\frac{\Gamma(\delta_{j^*}+\sum_{j=1}^{M}\delta_j)}{\Gamma(\delta_{j^*})\Gamma(\sum_{j=1}^{M}\delta_j)}p_{j^*}^{\delta_{j^*}-1}(1-p_{j^*})^{\sum_{j=1}^{M}\delta_j-M} \tag{51}$$

$$\frac{p(Z|P,M+1)}{p(Z|P,M)}=\frac{p_{j^*}^0\prod_{j=1}^{M}(p_j(1-p_{j^*}))^{n_j}}{\prod_{j=1}^{M}p_j^{n_j}}=(1-p_{j^*})^{\sum_{j=1}^{M}n_j}=(1-p_{j^*})^N \tag{52}$$

Note also that our specific choice of generating m_{j^*} and s_{j^*}, from the associated prior distributions given by Eqs. (8) and (9), respectively, simplify the calculations, since $\frac{p(\xi|M+1,\eta)}{p(\xi|M,\eta)q(u)}=\frac{1}{p(p_{j^*})}$. Recall that the Jacobian is $(1-p_{j^*})^M$, thus

$$A=\frac{p(M+1)}{p(M)}\frac{\Gamma(\delta_{j^*}+\sum_{j=1}^{M}\delta_j)}{\Gamma(\delta_{j^*})\Gamma\left(\sum_{j=1}^{M}\delta_j\right)}p_{j^*}^{\delta_{j^*}-1}(1-p_{j^*})^{N+\sum_{j=1}^{M}\delta_j-M}(M+1)$$
$$\times\frac{b_{M+1}}{a_M(M_0+1)}\frac{1}{p(p_{j^*})}(1-p_{j^*})^M \tag{53}$$

where M_0 is the number of empty components before the birth and $p(p_{j^*})$ is a Beta distribution with parameters $(\delta_{j^*}, \sum_{j=1}^{M}\delta_j)$.

Appendix E. Proof of Eq. (39)

If a 2-parameter density p belongs to the exponential family, then we can write it as the following (Brown, 1986)

$$p(x|\theta)=H(x)\exp(G(\theta)^{tr}T(x)+\Phi(\theta)) \tag{54}$$

where $G(\theta)=(G_1(\theta),G_2(\theta)), T(x)=(T_1(x),T_2(x))$ and tr denotes the transpose. The K-L divergence between two exponential distributions is given by Brown (1986)

$$D(p(x|\theta)\|p'(x|\theta'))=\Phi(\theta)-\Phi(\theta')+[G(\theta)-G(\theta')]^{tr}E_\theta[T(x)] \tag{55}$$

where E_θ is the expectation with respect to $p(x|\theta)$. Moreover, we have the following (Brown, 1986): $E_\theta[T(x)]=-\Phi'(\theta)$. The general Beta distribution can be written as an exponential density. In fact, we can easily show that

$$p(x|\alpha,\beta)=\frac{\Gamma(\alpha+\beta)}{(b-a)^{\alpha+\beta-1}\Gamma(\alpha)\Gamma(\beta)}(x-a)^{\alpha-1}(b-x)^{\beta-1}$$
$$=\exp[\log(\Gamma(\alpha+\beta))-(\alpha+\beta-1)\log(b-a)-\log(\Gamma(\alpha))$$
$$-\log(\Gamma(\beta))+(\alpha-1)\log(x-a)+(\beta-1)\log(b-x)]$$

Then by letting: $\Phi(\alpha,\beta)=\log(\Gamma(\alpha+\beta))-\log(\Gamma(\alpha))-\log(\Gamma(\beta))-(\alpha+\beta)\log(b-a)$, $G_1(\alpha,\beta)=\alpha$, $G_2(\alpha,\beta)=\beta$, $T_1(x)=\log(x-a)$, $T_2(x)=\log(b-x)$, and $H(x)=\exp(-\log(x-a)-\log(b-x)+\log(b-a))$, we obtain $E_\theta[\log(x-a)]=-\Psi(\alpha+\beta)+\Psi(\alpha)+\log(b-a)$, $E_\theta[\log(b-x)]=-\Psi(\alpha+\beta)+\Psi(\beta)+\log(b-a)$,

$$\begin{aligned}D(p(x|\theta)\|p'(x|\theta'))&=\log(\Gamma(\alpha+\beta))-\log(\Gamma(\alpha'+\beta'))\\&+\log(\Gamma(\alpha'))-\log(\Gamma(\alpha))+\log(\Gamma(\beta'))\\&-\log(\Gamma(\beta))-(\alpha+\beta-\alpha'-\beta')\log(b-a)\\&+(\alpha-\alpha')[\Psi(\alpha+\beta)-\Psi(\alpha)-\log(b-a)]\\&+(\beta-\beta')[\Psi(\alpha+\beta)-\Psi(\beta)-\log(b-a)]\\&=\log\left[\frac{\Gamma(\alpha+\beta)\Gamma(\alpha')\Gamma(\beta')}{(b-a)^{\alpha+\beta-\alpha'-\beta'}\Gamma(\alpha'+\beta')\Gamma(\alpha)\Gamma(\beta)}\right]\\&+(\alpha'-\alpha)[\Psi(\alpha+\beta)-\Psi(\alpha)-\log(b-a)]\\&+(\beta'-\beta)[\Psi(\alpha+\beta)-\Psi(\beta)-\log(b-a)]\end{aligned}$$

References

Andrieu, C., & Doucet, A. (1999). Joint Bayesian model selection and estimation of noisy sinusoids via reversible jump MCMC. *IEEE Transactions on Signal Processing, 47*(10), 2667–2676.

Applegate, D., & Kannan, R. 1991. Sampling and integration of near log-concave functions. In *Proceedings of the Twenty-third Annual ACM Symposium on Theory of Computing*, pp. 156–163.

Basford, K. E., McLachlan, G. J., & York, M. G. (1997). Modelling the distribution of stamp paper thickness via finite normal mixtures: The 1872 Hidalgo stamp issue of Mexico revisited. *Journal of Applied Statistics, 24*(2), 169–179.

Bechtel, Y. C., Bonaiti-Pellie, C., Poisson, N., Magnette, J., & Bechtel, P. R. (1993). A population and family study of N-acetyltransferase using caffeine urinary metabolites. *Clinical Pharmacology and Therapeutics, 54*(2), 134–141.

Berkhof, J., Van Mechelen, I., & Gelman, A. (2003). A Bayesian approach to the selection and testing of mixture models. *Statistica Sinica, 13*, 423–442.

Bouguila, N., & Ziou, D. (2007). High-dimensional unsupervised selection and estimation of a finite generalized Dirichlet mixture model based on minimum message length. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(10), 1716–1731.

Bouguila, N., Ziou, D., & Monga, E. (2006). Practical Bayesian estimation of a finite beta mixture through Gibbs sampling and its applications. *Statistics and Computing, 16*(2), 215–225.

Bouguila, N., Ziou, D., & Vaillancourt, J. (2004). Unsupervised learning of a finite mixture model based on the Dirichlet distribution and its application. *IEEE Transactions on Image Processing, 13*(11), 1533–1543.

Bovik, A. C., Clark, M., & Geisler, W. S. (1990). Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*(1), 55–73.

Brooks, S. P. (2001). On Bayesian analyses and finite mixtures for proportions. *Statistics and Computing, 11*(2), 179–190.

Brown, L. D. (1986). *Fundamentals of statistical exponential families with applications in statistical decision theory*. Hayward, CA: Institute of Mathematical Statistics.
Carlin, B. P., & Louis, T. A. (2000). *Bayes and empirical Bayes methods for data analysis* (second ed.). Chapman & Hall/CRC.
Chang, T., & Kuo, C.-C. J. (1993). Texture analysis and classification with tree-structured wavelet transform. *IEEE Transactions on Image Processing, 2*(4), 429–441.
Chib, S., & Greenberg, E. (1995). Understanding the Metropolis–Hastings algorithm. *The American Statistician, 49*(4), 327–335.
Dellaportas, P., Forster, J. J., & Ntzoufras, I. (2002). On Bayesian model and variable selection using MCMC. *Statistics and Computing, 12*(1), 27–36.
Dellaportas, P., & Papageorgiou, I. (2006). Multivariate mixtures of normals with unknown number of components. *Statistics and Computing, 16*(1), 57–68.
Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, B, 39*, 1–38.
Diaconis, P., & Ylvisaker, D. 1985. Quantifying prior opinion. In: Lindley, D. V., Bernardo, J. M., DeGroot, M. H., Smith, A. F. M. (Eds.), *Bayesian Statistics*, vol. 2, pp. 133–156.
Do, M. N., & Vetterli, M. (2002). Wavelet-based texture retrieval using generalized Gaussian density and Kullback–Leibler distance. *IEEE Transactions on Image Processing, 11*(2), 146–158.
Dunn, D., & Higgins, W. E. (1995). Optimal Gabor filters for texture segmentation. *IEEE Transactions on Image Processing, 4*(7), 947–964.
Dunn, D., Higgins, W. E., & Wakeley, J. (1994). Texture segmentation using 2-D Gabor elementary functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 16*(2), 130–149.
Fitzgerald, W. J., Godsill, S. J., Kokaram, A. C., & Stark, J. A. (1999). Bayesian methods in signal and image processing. In A. P. Dawid, J. M. Bernardo, J. O. Berger, & A. F. M. Smith (Eds.). *Bayesian Statistics* (vol. 6, pp. 239–254). Oxford University Press.
Freeman, W. T., & Adelson, E. H. (1991). The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*(9), 891–906.
Gelfand, A. E., Mallick, B. K., & Dey, D. K. (1995). Modeling expert opinion arising as a partial probabilistic specification. *Journal of the American Statistical Association, 90*(430), 598–604.
Ghosh, J. K., Delampady, M., & Samanta, T. (2006). *An introduction to Bayesian analysis theory and methods*. Springer.
Gilks, W. R., & Wild, P. (1993). Algorithm AS 287: Adaptive rejection sampling from log-concave density functions. *Applied Statistics, 42*(4), 701–709.
Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika, 82*(4), 711–732.
Grigorescu, S. E., Petkov, N., & Kruizinga, P. (2002). Comparison of texture features based on Gabor filters. *IEEE Transactions on Image Processing, 11*(10), 1160–1167.
Gruet, M-A., Philippe, A., & Robert, C. P. (1999). MCMC control spreadsheets for exponential mixture estimation. *Journal of Computational and Graphical Statistics, 8*(2), 298–317.
Izenman, A. J., & Sommer, C. J. (1988). Philatelic mixtures and multimodal densities. *Journal of the American Statistical Association, 83*(404), 941–953.
Jamshidian, M., & Jennrich, R. I. (1997). Acceleration of the EM Algorithm by using quasi-Newton methods. *Journal of the Royal Statistical Society: Series B, 59*(3), 569–587.
Johnson, N. L., Kotz, S., & Balakrishman, N. (1995). *Continous univariate distributions* (vol. 2). New York: John Wiley and Sons.
Laine, A., & Fan, J. (1993). Texture classification by wavelet packet signatures. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 15*(11), 1186–1191.
Mackay, D. J. C., & Peto, L. (1994). A hierarchical Dirichlet language model. *Natural Language Engineering, 1*(3), 1–19.
Manjunath, B. S., & Ma, W. Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*(8), 837–842.
Marrs, A. D. 1997. An application of reversible-jump MCMC to multivariate spherical Gaussian mixtures. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 577–583.
McLachlan, G. J., & Krishnan, T. (1997). *The EM algorithm and extensions*. New York: Wiley.
McLachlan, G. J., & Peel, D. (2000). *Finite mixture models*. New York: Wiley.
Meignen, S., & Meignen, H. (2006). On the modeling of small sample distributions with generalized Gaussian density in a maximum likelihood framework. *IEEE Transactions on Image Processing, 15*(6), 1647–1652.
Meligkotsidou, L. (2007). Bayesian multivariate Poisson mixtures with an unknown number of components. *Statistics and Computing, 17*(2), 93–107.
Meng, X., & van Dyk, D. (1997). The EM algorithm – An old folk song sung to a fast new tune (with discussion). *Journal of the Royal Statistical Society, Series B, 59*(3), 511–567.
Nobile, A. (2004). On the posterior distribution of the number of components in a finite mixture. *Annals of Statistics, 32*(5), 2044–2073.
Nobile, A., & Fearnside, A. T. (2007). Bayesian finite mixtures with an unknown number of components: The allocation sampler. *Statistics and Computing, 17*(2), 147–162.
Randen, T., & Husøy, J. H. (1999). Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 21*(4), 291–310.
Richardson, S., & Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: Series B, 59*(4), 731–792.
Robert, C. P. (2007). *The Bayesian choice from decision-theoretic foundations to computational implementation*. Springer.
Roeder, K., & Wasserman, L. (1997). Practical Bayesian density estimation using mixture of normals. *Journal of the American Statistical Association, 92*, 894–902.
Rubner, Y., Tomasi, C., & Guibas, L. (2000). The Earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision, 40*(2), 99–121.
Simoncelli, E. P., & Freeman, W. T. 1995. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 444–447.
Stephens, M. (2000). Bayesian analysis of mixture models with an unknown number of components: An alternative to reversible jump methods. *Annals of Statistics, 28*, 40–74.
Tuceryan, M., & Jain, A. K. 1998. Texture analysis. In *The Handbook of Pattern Recognition and Computer Vision*, pp. 207–248.
Unser, M. (1995). Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing, 4*(11), 1549–1560.
Viallefont, V., Richardson, S., & Green, P. J. (2002). Bayesian analysis of Poisson mixtures. *Journal of Nonparametric Statistics, 14*(1–2), 181–202.
Wainwright, M. J., & Simoncelli, E. P. (2000). Scale mixtures of Gaussians and the statistics of natural images. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 855–861.
Wilson, R., & Granlund, G. H. (1984). The uncertainty principle in image processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 6*(6), 758–767.
Wouwer, G. V., Scheunders, P., & Dyck, D. V. (1999). Statitical texture characterization from discrete wavelet representations. *IEEE Transactions on Image Processing, 8*(4), 592–598.
Wu, Y., Chan, K. L., & Huang, Y. 2003. Image Texture Classification Based on Finite Gaussian Mixture Models. In *Proceedings of the third international workshop on texture analysis and synthesis, ninth International Conference on Computer Vision (ICCV)*, pp. 107–112.
Yang, X., & Liu, J. (2002). Mixture density estimation with group membership functions. *Pattern Recognition Letters, 23*(5), 501–512.
Zhang, Z., Chan, K. L., Wu, Y., & Chen, C. (2004). Learning a multivariate Gaussian mixture model with the reversible jump MCMC algorithm. *Statistics and Computing, 14*(4), 343–355.