Pattern Anal Applic (2009) 12:151–166
DOI 10.1007/s10044-008-0111-4

On Bayesian analysis of a finite generalized Dirichlet mixture via a Metropolis-within-Gibbs sampling

Nizar Bouguila · Djemel Ziou · Riad I. Hammoud

Received: 3 April 2007 / Accepted: 23 January 2008 / Published online: 14 March 2008

Abstract In this paper, we present a fully Bayesian approach for generalized Dirichlet mixtures estimation and selection. The estimation of the parameters is based on the Monte Carlo simulation technique of Gibbs sampling mixed with a Metropolis-Hastings step. Also, we obtain a posterior distribution which is conjugate to a generalized Dirichlet likelihood. For the selection of the number of clusters, we used the integrated likelihood. The performance of our Bayesian algorithm is tested and compared with the maximum likelihood approach by the classification of several synthetic and real data sets. The generalized Dirichlet mixture is also applied to the problems of IR eye modeling and introduced as a probabilistic kernel for Support Vector Machines.

Keywords Generalized Dirichlet mixture · EM · Bayesian analysis · Gibbs sampling · Metropolis–Hastings · SVM · Images classification

1 Introduction

Finite generalized Dirichlet mixtures [1] have been shown as a robust alternative of normal mixtures [2–4] which have been widely used in the last few years [5, 6]. Tremendous improvements and applications have been made in across many research fields which involves the statistical modeling of data such as astronomy, ecology, bioinformatics, pattern recognition, computer vision and machine learning [7]. Mixture distributions are typically used to model data in which each observation is assumed to have arisen from one of a number of different groups. The Gaussian probability density function (pdf) is widely accepted and used in mixture modeling to analyze both one-dimensional and multidimensional data. At the same time, other models have not received much attention. In a previous work, we proposed the Dirichlet distribution, which has high flexibility to model data, to overcome some problems of the Gaussian [8–11]. In contrast with other distributions, the Dirichlet permits multiple symmetric and asymmetric modes which makes it very useful in many applications. Despite its flexibility, the Dirichlet distribution has a restrictive negative covariance matrix as shown is [1] where we proposed the use of the generalized Dirichlet distribution to overcome this problem.

Two main problems in the case of finite mixture models are the estimation of the parameters and the selection of the number of clusters. The estimation of the parameters of finite mixture distribution has recently come under close scrutiny [2]. Computational methods, have also appeared, including the EM algorithm proposed by Dempster et al. [12]. However, the EM algorithm for finite mixture has several drawbacks [13]. These drawbacks are mainly optimization problems. For example, the occurrence of local maxima and singularities in the likelihood function will often cause problems for a deterministic gradient method [14]. Moreover, another important issue is the estimation in higher dimensions [15]. Indeed, it can be hard to to obtain reliable estimates when the dimensionality of input data is high. By reliable we mean estimates that

N. Bouguila (✉)
Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montreal, QC H3G 1T7, Canada
e-mail: bouguila@ciise.concordia.ca

D. Ziou
Département d'Informatique, Université de Sherbrooke, Sherbrooke, QC J1K 2R1, Canada
e-mail: djemel.ziou@usherbrooke.ca

R. I. Hammoud
Delphi Corporation, Delphi Electronics and Safety, Kokomo, IN, USA
e-mail: riad.hammoud@delphi.com

possess generalization capabilities to predict the densities at new data points. In recent developments of computational methods, there is a growing interest in Bayesian methods which are considered as an alternative way to deal with mixture models. Given a proper prior, a Bayesian approach to the mixture estimation problem always provides estimators which can be written explicitly for conjugate priors [16]. Besides, Bayesian approaches are based on simulation methods, such as Gibbs sampling, which explore high-density regions. The stochastic aspect of these simulation methods ensures the escape from local maxima [17]. Diebolt and Robert [16] used data augmentation and Gibbs sampling as approximation methods for evaluating the posterior distribution and Bayes estimators of univariate Gaussian mixtures. Roeder and Wasserman [18] proposed another Bayesian approach to deal with the problem of improper posteriors. Bensmail et al. [19] introduced a fully Bayesian analysis to clustering based on parsimonious geometric modeling of the within-group covariance matrices in mixture of multivariate normal distributions. Tsung et al. [20] used Bayesian inference to analyze finite mixtures of multivariate t distributions. Tsionas [21] considers the estimation of the parameters of the multivariate Gamma distribution using Gibbs sampling with data augmentation. Bouguila et al. proposed a Bayesian algorithm to estimate finite Beta mixture's parameters [22]. Brooks adopted MCMC simulation techniques for finite Beta–Binomial mixtures to model proportions [23]. Another important problem in the case of finite mixture models is the selection of the number of components. For this problem, there is a number of possible solutions such as Bayes factors [24, 25], entropy distance or K–L divergence [26, 27], reversible jump MCMC [25, 28], and birth-and-death processes [29, 30].

In this article, we extend the unsupervised selection and estimation approach, we proposed in [1, 31–34] to deal with a mixture of generalized Dirichlet distributions from Bayesian viewpoints using the Gibbs sampling for the estimation and the integrated likelihood for the selection of the number of components. The choice of the generalized Dirichlet distribution is motivated by its interesting properties. Comparing with our previous works [1, 31–34], this paper involves several specific contributions including: the determination of a conjugate prior to the generalized Dirichlet distribution by taking into account the fact that this distribution is exponential, proposing an MCMC algorithm based on both Gibbs sampling and Metropolis-Hastings for the estimation of the parameters and Bayes factors for the selection of the number of clusters, introducing generalized Dirichlet mixtures as SVMs probabilistic kernels which clearly improve the classification accuracy in the case of non-Gaussian data and applying generalized Dirichlet mixtures for new challenging problems such as IR eye modeling. In Sect. 2, we present the generalized Dirichlet mixture, we develop a natural conjugate distribution for the generalized Dirichlet distribution, and we describe Bayesian estimation for this mixture using Gibbs sampling. In Sect. 3, we calculate the integrated likelihood from the Gibbs sampling outputs and use it to determine the number of components. Section 4 is devoted to the experimental results. Some concluding remarks are given in Sect. 5.

2 Bayesian learning of a finite generalized Dirichlet mixture

2.1 The generalized Dirichlet mixture

If the random vector $\mathbf{X} = (X_1,...,X_d)$ follows a generalized Dirichlet distribution, the joint density function is given by [35]:

$$p(X_1,\ldots,X_d) = \prod_{i=1}^{d} \frac{\Gamma(\alpha_i+\beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} X_i^{\alpha_i-1}\left(1-\sum_{j=1}^{i} X_j\right)^{\gamma_i} \tag{1}$$

for $\sum_{i=1}^{d} X_i < 1$ and $0 < X_i < 1$ for $i = 1, ..., d$, where $(\alpha_1, \beta_1, \ldots, \alpha_d, \beta_d) \in \mathbb{R}^{+d}$ are the parameters of the distribution, $\gamma_i = \beta_i - \alpha_{i+1} - \beta_{i+1}$ for $i = 1, ..., d-1$ and $\gamma_d = \beta_d - 1$. Note that the generalized Dirichlet distribution is reduced to a Dirichlet distribution when $\beta_i = \alpha_{i+1} + \beta_{i+1}$. We note that the generalized Dirichlet distribution is defined in the compact support [0,1] in contrast of the Gaussian, for example, which is defined in $\mathbb{R}$. However, we can generalize it easily to be defined in a compact support of the form $[A, B]$, where $(A, B) \in \mathbb{R}^2$ [34] (see, for example, [36] in the case of the Beta distribution and [37] in the case of the Dirichlet). Having a compact support is an interesting property for a given density because of the nature of data in general. Generally, we model data which are compactly supported, such as data originating from videos, images or text. A finite generalized Dirichlet mixture with M components is defined as:

$$p(\mathbf{X}|\Theta) = \sum_{j=1}^{M} p(\mathbf{X}|\xi_j)P_j \tag{2}$$

where the P_j are the mixing probabilities and $p(\mathbf{X}|\xi_j)$ is the generalized Dirichlet distribution.

The symbol $\Theta = (\xi, P)$ refers to the entire set of parameters to be estimated, where $\xi = (\xi_1, \ldots, \xi_M)$, $\xi_j = (\alpha_{j1}, \beta_{j1}, \ldots, \alpha_{jd}, \beta_{jd})$ is the set of parameters defining the j-th component, and $P = (P_1,...,P_M)$. Of course, being probabilities, the P_j must satisfy

$$0 < P_j \le 1, \quad j = 1, \ldots, M \tag{3}$$

$$\sum_{j=1}^{M} P_j = 1 \tag{4}$$

2.2 Bayesian learning

Given a set of N independent vectors $\mathcal{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$ described by a generalized Dirichlet mixture, an important problem is the estimation of the mixture parameters. As we mentioned in the previous section, Bayesian techniques are now widely used to resolve this problem.

Let $\mathbf{Z}_i = (Z_{i1}, \ldots, Z_{iM})$, with

$$Z_{ij} = \begin{cases} 1 & \text{if } \mathbf{X}_i \quad \text{belongs to class } j \\ 0 & \quad\quad\;\; \text{otherwise} \end{cases} \tag{5}$$

constituting the "missing" data and $\mathcal{Z} = (\mathbf{Z}_1, \ldots, \mathbf{Z}_N)$. In the Bayesian paradigm information brought by the complete data $(\mathcal{X}, \mathcal{Z})$, a realization of $(\mathcal{X}, \mathcal{Z}) \sim p(\mathcal{X}, \mathcal{Z}|\Theta)$, is combined with prior information about the parameters Θ that is specified in a prior distribution with density $\pi(\Theta)$ and summarized in a probability distribution $\pi(\Theta|\mathcal{X}, \mathcal{Z})$, called the posterior distribution [14]. This is derived from the joint distribution $p(\mathcal{X}, \mathcal{Z}|\Theta)\pi(\Theta)$, according to Bayes formula

$$\pi(\Theta|\mathcal{X}, \mathcal{Z}) = \frac{p(\mathcal{X}, \mathcal{Z}|\Theta)\pi(\Theta)}{\int p(\mathcal{X}, \mathcal{Z}|\Theta)\pi(\Theta)} \propto p(\mathcal{X}, \mathcal{Z}|\Theta)\pi(\Theta) \tag{6}$$

where $\int p(\mathcal{X}, \mathcal{Z}|\Theta)\pi(\Theta)$ is the marginal density of the complete data $(\mathcal{X}, \mathcal{Z})$. Having this posterior distribution in hand we can simulate $\Theta \sim \pi(\Theta|\mathcal{X}, \mathcal{Z})$ rather than computing them. This simulation technique is now well known as Gibbs sampling. The Gibbs sampler is the most commonly used approach in Bayesian mixture estimation. In fact, a solution to the computational problem is to take advantage of the missing data, that is to associate with each observation $\mathbf{X}_i$ a missing multinomial variable $\mathbf{Z}_i \sim \mathcal{M}(1; \hat{Z}_{i1}, \ldots, \hat{Z}_{iM})$, where

$$\hat{Z}_{ij} = \frac{p(\mathbf{X}_i|\xi_j)P_j}{\sum_{j=1}^{M} p(\mathbf{X}_i|\xi_j)P_j} \tag{7}$$

Denote by $\pi(P|\mathcal{Z}, \mathcal{X})$ the density of the distribution of P given $\mathcal{Z}$ and $\mathcal{X}$. This distribution is in fact independent of $\mathcal{X}$, $\pi(P|\mathcal{Z}, \mathcal{X}) = \pi(P|\mathcal{Z})$. The standard Gibbs sampler for mixture models is based on the successive simulation of $\mathcal{Z}, P$ and ξ. Then, the general Gibbs sampling for mixture models is as follows [38]

1. Initialization
2. Step t: For $t = 1, \ldots$
   - (a) Generate $\mathbf{Z}_i^{(t)} \sim \mathcal{M}\left(1; \hat{Z}_{i1}^{(t-1)}, \ldots, \hat{Z}_{iM}^{(t-1)}\right)$
   - (b) Generate $P^{(t)}$ from $\pi(P|\mathcal{Z}^{(t)})$
   - (c) Generate $\xi^{(t)}$ from $\pi(\xi|\mathcal{Z}^{(t)}, \mathcal{X})$

We start by the distribution $\pi(P|\mathcal{Z})$ and we have

$$\pi(P|\mathcal{Z}) \propto \pi(P)\pi(\mathcal{Z}|P) \tag{8}$$

We determine now $\pi(P)$ and $\pi(\mathcal{Z}|P)$. We know that the vector P is defined on the simplex $\{(P_1, \ldots, P_M) : \sum_{j=1}^{M-1} P_j < 1\}$, then a natural choice, as a prior, for this vector is the Dirichlet distribution [38]

$$\pi(P) = \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} P_j^{\eta_j - 1} \tag{9}$$

where $\eta = (\eta_1, \ldots, \eta_M)$ is the parameter vector of the Dirichlet distribution. Moreover, we have

$$\begin{aligned} \pi(\mathcal{Z}|P) &= \prod_{i=1}^{N} \pi(\mathbf{Z}_i|P) = \prod_{i=1}^{N} P_1^{Z_{i1}} \ldots P_M^{Z_{iM}} \\ &= \prod_{i=1}^{N} \prod_{j=1}^{M} P_j^{Z_{ij}} = \prod_{j=1}^{M} P_j^{n_j} \end{aligned} \tag{10}$$

where $n_j = \sum_{i=1}^{N} \mathbb{I}_{Z_{ij}=j}$. Then

$$\begin{aligned} \pi(P|\mathcal{Z}) &= \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} P_j^{\eta_j - 1} \prod_{j=1}^{M} P_j^{n_j} \\ &= \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} P_j^{\eta_j + n_j - 1} \\ &\propto \mathcal{D}(\eta_1 + n_1, \ldots, \eta_M + n_M) \end{aligned} \tag{11}$$

where $\mathcal{D}$ is a Dirichlet distribution with parameters $(\eta_1 + n_1, \ldots, \eta_M + n_M)$. We note that the prior and the posterior distributions, $\pi(P)$ and $\pi(P|\mathcal{Z})$, are both Dirichlet. In this case we say that the Dirichlet distribution is a conjugate prior for the mixture proportions.

For a mixture of generalized Dirichlet distributions, it is therefore possible to associate with each ξ_j a prior $\pi_j(\xi_j)$. For this we use the fact that the generalized Dirichlet distribution belongs to the exponential family. In fact, if a S-parameter density p belongs to the exponential family, then we can write it as the following [39, 40]

$$p(\mathbf{X}|\theta) = H(\mathbf{X}) \exp\left(\sum_{l=1}^{S} G_l(\theta) T_l(\mathbf{X}) + \Phi(\theta)\right) \tag{12}$$

In this case a conjugate prior on θ is given by [39, 40]

$$\pi(\theta) \propto \exp\left(\sum_{l=1}^{S} \rho_l G_l(\theta) + \kappa \Phi(\theta)\right) \tag{13}$$

where $\rho = (\rho_1, \ldots, \rho_S) \in \mathbb{R}^S$ and $\kappa > 0$ are referred as hyperparameters. The generalized Dirichlet distribution can be written as an exponential density. In fact, we have (see Appendix 1)

$$\begin{aligned} p(\mathbf{X}|\xi_j) &= \exp\Bigg[\sum_{l=1}^{d} \Big(\log\big(\Gamma(\alpha_{jl} + \beta_{jl})\big) - \mathit{log}\big(\Gamma(\alpha_{jl})\big) - \log\big(\Gamma(\beta_{jl})\big)\Big) \\ &\quad + \sum_{l=1}^{d} \big((\alpha_{jl} - 1)\log(X_l)\big) + (\beta_{jd} - 1)\log\left(1 - \sum_{t=1}^{2d} X_t\right) \\ &\quad + \sum_{l=d+1}^{2d-1} \left((\beta_{jl-d} - \alpha_{jl-d+1} - \beta_{jl-d+1})\log\left(1 - \sum_{t=1}^{l-d} X_t\right)\right)\Bigg] \end{aligned}$$

Then by letting

$$
\begin{aligned}
S &= 2d\\
\Phi(\xi_j) &= \sum_{l=1}^{d}\big(\log\big(\Gamma(\alpha_{jl}+\beta_{jl})\big)-\log\big(\Gamma(\alpha_{jl})\big)-\log\big(\Gamma(\beta_{jl})\big)\big)\\
G_l(\xi_j) &= \alpha_{jl},\ l=1,\ldots,d\\
G_l(\xi_j) &= \beta_{jl-d}-\alpha_{jl-d+1}-\beta_{jl-d+1},\ l=d+1,\ldots,2d-1\\
G_{2d}(\xi_j) &= \beta_{jd}\\
T_l(\mathbf{X}) &= \log(X_l),\ l=1,\ldots,d\\
T_l(\mathbf{X}) &= \log\left(1-\sum_{t=1}^{l-d}X_t\right),\ l=d+1,\ldots,2d\\
H(\mathbf{X}) &= \exp\left(-\sum_{l=1}^{d}\log(X_l)-\log\left(1-\sum_{t=1}^{d}X_t\right)\right)
\end{aligned}
$$

The prior is

$$
\begin{aligned}
\pi(\xi_j) \propto \exp&\Bigg[\sum_{l=1}^{d}\rho_l\alpha_{jl}+\sum_{l=d+1}^{2d-1}\rho_l\big(\beta_{jl-d}-\alpha_{jl-d+1}-\beta_{jl-d+1}\big)\\
&+\kappa\sum_{l=1}^{d}\big(\log\big(\Gamma(\alpha_{jl}+\beta_{jl})\big)-\log\big(\Gamma(\alpha_{jl})\big)-\log\big(\Gamma(\beta_{jl})\big)\big)\\
&+\rho_{2d}\beta_{jd}\Bigg]\\
&\propto \exp\Bigg[\kappa\sum_{l=1}^{d}\big(\log\big(\Gamma(\alpha_{jl}+\beta_{jl})\big)-\log\big(\Gamma(\alpha_{jl})\big)\\
&-\log\big(\Gamma(\beta_{jl})\big)\big)+\sum_{l=1}^{d}\rho_l\alpha_{jl}+\sum_{l=1}^{d}\rho_{l+d}\gamma_{jl}\Bigg] \qquad (14)
\end{aligned}
$$

The prior hyperparameters are: $(\rho_1,\ldots,\rho_{2d},\kappa)$. Having this prior, $\pi(\xi_j)$, the posterior distribution is then (see Appendix 2)

$$
\begin{aligned}
\pi(\xi_j|\mathcal{Z},\mathcal{X}) &\propto \pi(\xi_j)\prod_{Z_{ij}=1}p(\mathbf{X}_i|\xi_j)\\
&\propto \exp\Bigg[\sum_{l=1}^{d}\alpha_{jl}\left(\rho_l+\sum_{Z_{ij}=1}\log(X_{il})\right)\\
&+\sum_{l=1}^{d}\gamma_{jl}\left(\rho_{l+d}+\sum_{Z_{ij}=1}\log(1-\sum_{t=1}^{l}X_{it})\right)\\
&+(\kappa+n_j)\sum_{l=1}^{d}\big(\log\big(\Gamma(\alpha_{jl}+\beta_{jl})\big)-\log\big(\Gamma(\alpha_{jl})\big)\\
&-\log\big(\Gamma(\beta_{jl})\big)\big)\Bigg] \qquad (15)
\end{aligned}
$$

We can see clearly that the posterior and the prior distributions have the same form, then $\pi(\xi_j)$ is really a conjugate prior on ξ_j. According to the posterior hyperparameters

$$
\begin{aligned}
&\Bigg(\rho_1+\sum_{Z_{ij}=1}\log(X_{i1}),\ldots,\rho_d+\sum_{Z_{ij}=1}\log(X_{id}),\\
&\rho_{d+1}+\sum_{Z_{ij}=1}\log\left(1-\sum_{t=1}^{1}X_{it}\right),\ldots,\\
&\rho_{2d}+\sum_{Z_{ij}=1}\log\left(1-\sum_{t=1}^{d}X_{it}\right),\kappa+n_j\Bigg)
\end{aligned}
$$

a samples modifies the prior hyperparameters by adding $T_l(\mathbf{X})$ or n_j to the previous values. This information, could be used to get the prior hyperparameters. Indeed, following [41, 42], once the sample $\mathcal{X}$ is known, we can use it to get the prior hyperparameters [19]. Then, we held $(\eta_1,\ldots,\eta_M)$ and $(\rho_1,\ldots,\rho_{2d},\kappa)$ fixed at: $\eta_j=1, j=1,\ldots,M,\quad \rho_l=\sum_{i=1}^{N}\log(X_{il}),\quad \rho_{l+d}=\sum_{i=1}^{N}\log(1-\sum_{t=1}^{l}X_{it}),\ l=1,\ldots,d,\kappa=n_j$. Having all the posterior probabilities in hand, the steps of the Gibbs sampler are

1. Initialization
2. Step t: For t = 1,...
   (a) Generate $\vec{Z}_i^{(t)}\sim\mathcal{M}(1;\hat{Z}_{i1}^{(t-1)},\ldots,\hat{Z}_{iM}^{(t-1)})$
   (b) Compute $n_j^{(t)}=\sum_{i=1}^{N}\mathbb{I}_{Z_{ij}^{(t)}=j}$
   (c) Generate $P^{(t)}$ from Eq. 11
   (d) Generate $\xi_j^{(t)}$ (j = 1,..., M) from Eq. 15 using the Metropolis–Hastings (M–H) algorithm.

Note that we are using a hybrid MCMC algorithm based on both Gibbs and M-H sampling. This approach is better-known as Metropolis-within-Gibbs sampling [14]. It is used when one of the parameters is hard to sample and then an M–H step is needed. The M–H algorithm offers a solution to the problem of simulating from the posterior distribution [14]. Some extensions of the M–H and other approaches to simulate from mixture posterior distributions can be found in [43]. Starting from point $\xi_j^{(0)}$, the corresponding Markov chain explores the surface of the posterior distribution. At iteration t, the steps of the M-H algorithm can be described as follows:

1. Generate $\tilde{\xi}_j\sim q(\xi_j|\xi_j^{(t-1)})$ and $U\sim\mathcal{U}_{[0,1]}$
2. Compute $r=\frac{\pi(\tilde{\xi}_j|\mathcal{Z},\mathcal{X})q(\xi_j^{(t-1)}|\tilde{\xi}_j)}{\pi(\xi_j^{(t-1)}|\mathcal{Z},\mathcal{X})q(\tilde{\xi}_j|\xi_j)^{(t-1)}}$
3. If $r<u$ then $\xi_j^{(t)}=\tilde{\xi}_j$ else $\xi_j^{(t)}=\xi_j^{(t-1)}$

Where $\tilde{\xi}_j=(\tilde{\alpha}_{j1},\tilde{\beta}_{j1},\ldots,\tilde{\alpha}_{jd},\tilde{\beta}_{jd})$. The major problem in this algorithm is the need to choose the proposal distribution q. The most generic proposal is the random walk M–H algorithm where each unconstrained parameter is the mean of the proposal distribution for the new value. As all the $\tilde{\alpha}_{jl}>0,\tilde{\beta}_{jl}>0,\quad l=1,\ldots,d$, we have chosen the following proposals

$$\tilde{\alpha}_{jl} \sim \mathcal{LN}(\log(\alpha_{jl}^{(t-1)}), \sigma_1^2) \tag{16}$$

$$\tilde{\beta}_{jl} \sim \mathcal{LN}(\log(\beta_{jl}^{(t-1)}), \sigma_2^2) \tag{17}$$

where $\mathcal{LN}(\log(\alpha_j^{(t-1)}), \sigma_1^2)$ and $\mathcal{LN}(\log(\beta_j^{(t-1)}), \sigma_2^2)$ refer to the log-normal distributions with mean $\log(\alpha_j^{(t-1)})$ and variance σ_1^2 and mean $\log(\beta_j^{(t-1)})$ and variance σ_2^2, respectively. Note that Eqs. 16 and 17 are equivalent to

$$\log(\tilde{\alpha}_{jl}) = \log(\alpha_{jl}^{(t-1)}) + \epsilon_1 \tag{18}$$

$$\log(\tilde{\beta}_{jl}) = \log(\beta_{jl}^{(t-1)}) + \epsilon_2 \tag{19}$$

where $\epsilon_1 \sim \mathcal{N}(0, \sigma_1^2)$ and $\epsilon_2 \sim \mathcal{N}(0, \sigma_2^2)$. with these proposals the random walk M–H algorithm is composed of the following steps

1. Generate $\tilde{\alpha}_{jl} \sim \mathcal{LN}(\log(\alpha_{jl}^{(t-1)}), \sigma_1^2), \tilde{\beta}_{jl} \sim \mathcal{LN}(\log(\beta_{jl}^{(t-1)}), \sigma_2^2), \quad l = 1, \ldots, d$ and $U \sim \mathcal{U}_{[0,1]}$.
2. Compute
$$r = \frac{\pi(\tilde{\xi}_j|\mathcal{Z},\mathcal{X}) \prod_{l=1}^{d} \mathcal{LN}(\alpha_{jl}^{(t-1)}|\log(\tilde{\alpha}_{jl}),\sigma_1^2)\mathcal{LN}(\beta_{jl}^{(t-1)}|\log(\tilde{\beta}_{jl}),\sigma_2^2)}{\pi(\xi_j^{(t-1)}|\mathcal{Z},\mathcal{X}) \prod_{l=1}^{d} \mathcal{LN}(\tilde{\alpha}_{jl}|\log(\alpha_{jl}^{(t-1)}),\sigma_1^2)\mathcal{LN}(\tilde{\beta}_{jl}|\log(\beta_{jl}^{(t-1)}),\sigma_2^2)}$$
$$= \frac{\pi(\tilde{\xi}_j|\mathcal{Z},\mathcal{X}) \prod_{l=1}^{d} \tilde{\alpha}_{jl}\tilde{\beta}_{jl}}{\pi(\xi_j^{(t-1)}|\mathcal{Z},\mathcal{X}) \prod_{l=1}^{d} \alpha_{jl}^{(t-1)}\beta_{jl}^{(t-1)}}$$
3. If $r < u$ then $\xi_j^{(t)} = \tilde{\xi}_j$ else $\xi_j^{(t)} = \xi_j^{(t-1)}$

3 Estimating the number of components

The determination of the number of components M is problematical. The multi-dimensional nature of our data means that reversible jump MCMC techniques [25, 28] would be extremely complicated to operate efficiently, as would the Markov birth–death process approach [29, 30]. In order to determine the number of clusters M, we will use the integrated likelihood [24] defined by

$$p(\mathcal{X}|M) = \int \pi(\Theta|\mathcal{X}, M)\mathrm{d}\Theta = \int p(\mathcal{X}|\Theta, M)\pi(\Theta|M)\mathrm{d}\Theta \tag{20}$$

where Θ is the vector of parameters of a finite mixture model, $\pi(\Theta|M)$ is its prior density, and $p(\mathcal{X}|\Theta, M)$ is the likelihood function taking into account that the number of clusters is M. The main problem now is how to compute the integrated likelihood. In order to resolve this problem, let $\hat{\Theta}$ denotes the posterior mode, satisfying

$$\frac{\partial \log(\pi(\hat{\Theta}|\mathcal{X}, M))}{\partial \Theta} = 0 \tag{21}$$

where $\frac{\partial \log(\pi(\hat{\Theta}|\mathcal{X},M))}{\partial \Theta}$ denotes the gradient of $\log(\pi(\hat{\Theta}|\mathcal{X}, M))$ evaluated at $\Theta = \hat{\Theta}$. The Hessian matrix of minus the $\log(\pi(\hat{\Theta}|\mathcal{X}, M))$ evaluated at $\Theta = \hat{\Theta}$ is denoted by $H(\hat{\Theta})$. To approximate the integral given by Eq. 20, the integrand is expanded in a second-order Taylor series about the point $\Theta = \hat{\Theta}$, and the Laplace approximation gives

$$\begin{aligned} p(\mathcal{X}|M) &= \int \exp\left(\log(\pi(\hat{\Theta}|\mathcal{X}, M) - \frac{1}{2}(\Theta - \hat{\Theta})^T H(\hat{\Theta})(\Theta - \hat{\Theta})\right)\mathrm{d}\Theta \\ &= \pi(\hat{\Theta}|\mathcal{X}, M) \int \exp\left(-\frac{1}{2}(\Theta - \hat{\Theta})^T H(\hat{\Theta})(\Theta - \hat{\Theta})\right)\mathrm{d}\Theta \\ &= \pi(\hat{\Theta}|\mathcal{X}, M)(2\pi)^{\frac{N_p}{2}}\sqrt{|H(\hat{\Theta})|} \\ &= p(\mathcal{X}|\hat{\Theta}, M)\pi(\hat{\Theta}|M)(2\pi)^{\frac{N_p}{2}}\sqrt{|H(\hat{\Theta})|} \end{aligned} \tag{22}$$

where N_p is the number of parameters to be estimated and is equal to $(2d + 1)M$ in our case, and $|H(\hat{\Theta})|$ is the determinant of the Hessian matrix. For numerical reasons, it is better to work with the Laplace approximation on the logarithm scale. Taking logarithms, we can rewrite Eq. 22 as

$$\begin{aligned} \log(p(\mathcal{X}|M)) &= \log(p(\mathcal{X}|\hat{\Theta}, M)) + \log(\pi(\hat{\Theta}|M)) \\ &\quad + \frac{N_p}{2}\log(2\pi) + \frac{1}{2}\log(|H(\hat{\Theta})|) \end{aligned} \tag{23}$$

In our case an analytic solution for $\hat{\Theta}$ and $H(\hat{\Theta})$ is not available. Thus, we can use the Laplace–Metropolis estimator which consists of estimating $\hat{\Theta}$ and $H(\hat{\Theta})$ from the Gibbs sampler outputs [44]. The estimation of $\hat{\Theta}$ can be based on different ways such as that Θ in the sample at which $p(\mathcal{X}|\hat{\Theta}, M)$ achieves its maximum, finding the componentwise posterior means, finding the componentwise posterior medians and finding the multivariate medians (see [44] for more details and interesting discussions). In our case, we have chosen the first approach (i.e. we estimate $\hat{\Theta}$ as that the Θ in the sample at which the likelihood $p(\mathcal{X}|\hat{\Theta}, M)$ achieves its maximum) which is the simplest and the most accurate when the likelihood is easy to calculate as in our model [44]. The other quantity $H(\hat{\Theta})$ is asymptotically equal to the posterior variance matrix. Then, we could estimate it by the sample covariance matrix of the posterior simulation output [44]. Note that Eq. 22 could be also approximated by $N^{N_p/2}p(\mathcal{X}|\hat{\Theta}, M)$ [18], which gives us the schwarz criterion [45]:

$$\log(p(\mathcal{X}|M)) = \log(p(\mathcal{X}|\hat{\Theta}, M)) - \frac{N_p}{2}\log(N) \tag{24}$$

The schwarz criterion is also know as the Bayes information criterion (BIC) and coincides formally (but not conceptually) with the first version of the minimum description length criterion (MDL) [46]. Note, however, that the approximation given by Eq. 24 is theoretically motivated just under some regularity conditions as discussed by Kass and Wasserman [47]. Unfortunately, these conditions do not hold in the case of mixture models [18]. Having Eq. 23 in hand, the number of components in the mixture model is taken to be $\{M/\log(p(\mathcal{X}|M)) = \max_M \log(p(\mathcal{X}|M)), M = M_{min}, \ldots, M_{max}\}$.

4 Experimental results

This section has two main goals: comparing the maximum likelihood approach that we have previously proposed, through a hybrid stochastic expectation maximization (HSEM) algorithm [1], with our Bayesian algorithm and comparing the performance of the generalized Dirichlet mixture and the Gaussian mixture in several applications.

4.1 Simulated and real data sets

In the first application we investigate the properties of our Bayesian algorithm, in terms of estimation and selection, on six two-dimensional synthetic data sets. We choose $d = 2$ purely for ease of representation. In fact, we tested the effectiveness of the algorithm for estimating the mixture's parameters and selecting the number of clusters by generating data sets using different parameters. The number of elements N generated, and the real and estimated parameters of these generated data sets are given in Table 1. From Fig. 1, which represents the resultant mixtures, we see that we obtain different shapes (symmetric and asymmetric modes). Figure 2 gives the number of clusters determined for the generated data sets when $\log(p(\mathcal{X}|M))$ is approximated by Eq. 23. We can see clearly that the correct number of clusters is favored for all data sets. Figure 3 shows the number of clusters that we have obtained when we used Bayes information criterion. From this Figure, we can see that we obtained the exact number of clusters just for data set 1. Note that in Figs. 2 and 3 the values of $\log(p(\mathcal{X}|M))$ were averaged over 200 simulations draws. The last 2000 iterations from the Metropolis-within-Gibbs sampler output (we used 5,000 iterations in all and we discarded the first

Table 1 Parameters of the different generated data sets. n_j represents the number of the elements in cluster j. α_{j1}, β_{j1}, α_{j2}, β_{j2} and p_j are the real parameters. $\hat{\alpha}_{j1}, \hat{\beta}_{j1}, \hat{\alpha}_{j2}, \hat{\beta}_{j2}$ and $\hat{p}_j$ are the estimated parameters

	j	α_{j1}	β_{j1}	α_{j2}	β_{j2}	p_j	$\hat{\alpha}_{j1}$	$\hat{\beta}_{j1}$	$\hat{\alpha}_{j2}$	$\hat{\beta}_{j2}$	$\hat{p}_j$
Data set 1 ($N = 200$)	1	12	50	35	20	0.5	11.23	48.97	35.14	34.27	0.49
	2	32	60	13	20	0.5	31.42	58.98	13.11	19.75	0.51
Data set 2 ($N = 400$)	1	12	50	35	20	0.3	12.09	49.13	35.38	19.77	0.29
	2	32	60	13	20	0.3	31.71	60.22	12.67	19.91	0.29
	3	20	60	20	60	0.4	20.15	59.89	20.09	59.63	0.42
Data set 3 ($N = 400$)	1	3	43	32	100	0.3	2.98	42.88	32.15	99.51	0.31
	2	70	100	5	55	0.3	69.85	99.74	5.19	54.89	0.29
	3	40	80	26	20	0.2	40.01	79.94	26.08	19.90	0.2
	4	15	90	50	50	0.2	14.96	90.14	49.83	50.13	0.2
Data set 4 ($N = 500$)	1	3	43	32	100	0.2	2.87	43.12	31.91	99.90	0.19
	2	70	100	5	55	0.2	69.98	100.21	4.90	54.92	0.19
	3	40	80	26	20	0.2	39.86	79.93	27.85	18.11	0.21
	4	15	90	50	50	0.2	14.12	91.01	50.44	49.19	0.2
	5	20	60	20	60	0.2	19.76	60.05	19.44	59.92	0.21
Data set 5 ($N = 1{,}000$)	1	3	43	32	100	0.2	3.33	42.80	31.91	99.09	0.2
	2	70	100	5	55	0.2	96.79	99.16	5.21	54.19	0.18
	3	40	80	26	20	0.2	38.99	80.86	25.95	20.02	0.21
	4	15	90	50	50	0.2	14.77	90.08	50.11	49.04	0.2
	5	20	60	20	60	0.1	19.88	59.97	20.15	59.89	0.11
	6	31	141	295	430	0.1	30.88	140.93	294.47	428.02	0.1
Data set 6 ($N = 1{,}000$)	1	3	43	32	100	0.2	3.14	42.85	32.31	99.07	0.19
	2	70	100	5	55	0.2	71.02	99.05	4.52	54.84	0.19
	3	40	80	26	20	0.2	39.91	80.13	25.76	19.77	0.19
	4	15	90	50	50	0.1	14.93	90.09	49.77	49.77	0.11
	5	20	60	20	60	0.1	19.88	60.04	19.87	59.90	0.1
	6	31	141	295	430	0.1	30.87	138.98	295.53	428.89	0.11
	7	118	275	41	63	0.1	117.09	275.93	40.87	62.69	0.11

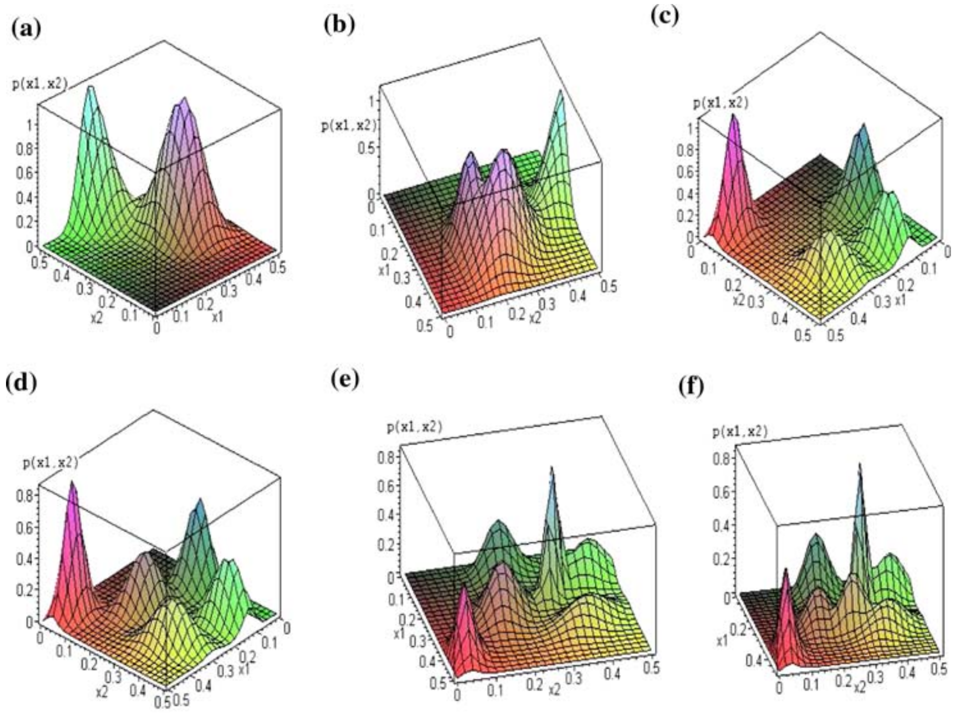


Fig. 1 Mixture densities for the generated data sets

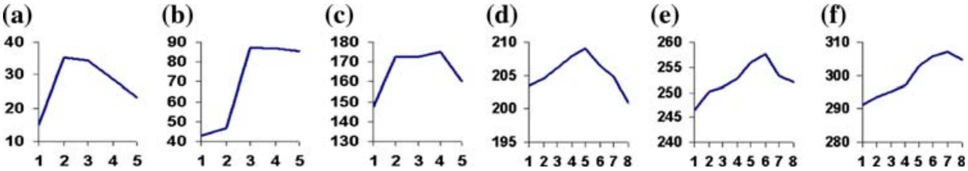


Fig. 2 Number of clusters found for the different generated data sets when Laplace approximation is used. **a** Data set 1, **b** Data set 2, **c** Data set 3, **d** Data set 4, **e** Data set 5, **f** Data set 6

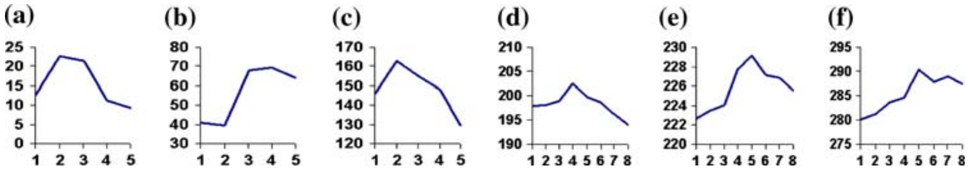


Fig. 3 Number of clusters found for the different generated data sets using the BIC criterion. **a** Data set 1, **b** Data set 2, **c** Data set 3, **d** Data set 4, **e** Data set 5, **f** Data set 6

800 as "burn-in") for data set 1 are shown in Fig. 4. Note that we obtained similar results for 200 different simulations. We also applied our algorithm to two well-known real data sets: Anderson's iris and wine. Iris consists of 50 samples for each of the three classes presented in the data, *Iris Versicolor*, *Iris Verginica* and *Iris Setosa*; each datum is four-dimensional and consists of measures of the plants' morphology. The wine data set consists of 178 13-dimensional vectors which are a set of chemical analyzes of three types of wine. Figure 5 shows that we obtain the correct number of clusters when using both the Laplace approximation and the BIC. Using our Bayesian algorithm, the accuracy was 98.66% (2 errors in 150 data samples) for the Iris data set and 98.31% (3 errors in 178 data samples), which are the same results found when we used the HSEM algorithm [1]. To show the advantages of the Bayesian approach, Table 2 lists the misclassified data samples in both the iris and wine data sets with their posterior probabilities when using both the pure Bayesian approach and the HSEM algorithm. From this table, we can see clearly that the pure Bayesian approach has increased estimated posterior probabilities, associated with the misclassified data samples, of belonging to the correct cluster. In the case of iris data set, for example, the improved percentages $\left(\frac{\text{Bayesian}-\text{HSEM}}{\text{HSEM}} \times 100\%\right)$ of belonging to the right clusters of the two misclassified samples were 17.07 and 9.32%. The posterior probabilities, in the case of the wine data set, were improved by: 4.54, 16.66 and 4%.

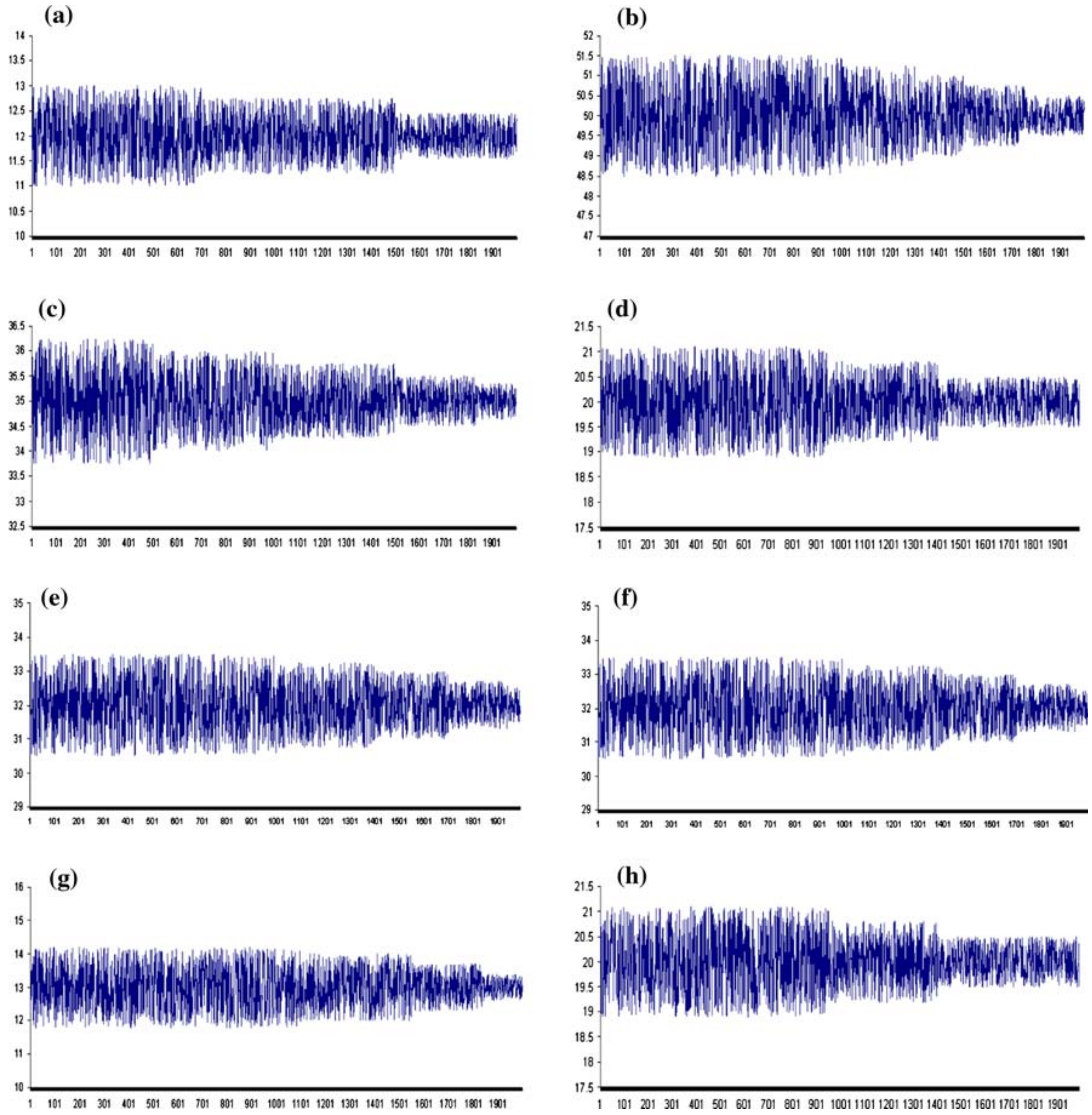


Fig. 4 Time series plot of the Gibbs-within-Metropolis iterations. **a** Iterations for $\hat{\alpha}_{11}$, **b** iterations for $\hat{\beta}_{11}$, **c** iterations for $\hat{\alpha}_{12}$, **d** iterations for $\hat{\beta}_{12}$, **e** iterations for $\hat{\alpha}_{21}$, **f** iterations for $\hat{\beta}_{21}$, **g** iterations for $\hat{\alpha}_{22}$, **h** iterations for $\hat{\beta}_{22}$

4.2 Human eyes modeling

Human eyes are one of the most important and salient features of the human face. Of all the features of the human face, the eyes are the most telling about a person's state of mind and emotion. Constructing accurate eye models is a challenging problem which provide excellent cues for different applications such as eyes tracking for driver fatigue and behavior, face detection and recognition, facial expression understanding, eye typing, drowsiness detection, and human computer interaction [48–54]. This problem is challenging because of many facts such as the difference of the eye shape from one subject to another and the drastic occlusions, changes in brightness-level and eye closure due to blinking [55, 56]. In addition, the eye appearance changes significantly with glasses [57]. The glares on the glasses caused by light reflections represent a real challenge to modeling. Two approaches can be used

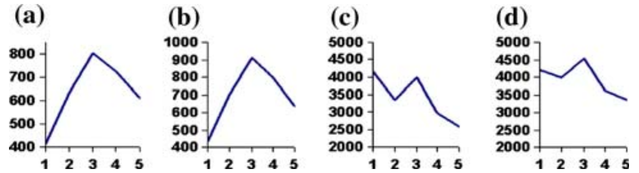


Fig. 5 Number of clusters found for the two real data sets using BIC and Laplace approximation. **a** Iris with BIC, **b** Iris with Laplace approximation, **c** wine with BIC, **d** wine with Laplace approximation

Table 2 Comprarison of posterior probabilities obtained for the missclassified data samples when using both the HSEM and the Bayesian approach

Data set	Misclassified data sample (correct class)	Class	HSEM	Bayesian
Iris	1 (**1**)	1	**0.41**	**0.48**
		2	0.49	0.49
		3	0.10	0.03
	2(**2**)	1	0.51	0.50
		2	**0.43**	**0.47**
		3	0.06	0.03
Wine	1 (**1**)	1	**0.44**	**0.46**
		2	0.09	0.07
		3	0.47	0.47
	2 (**2**)	1	0.07	0.01
		2	**0.42**	**0.49**
		3	0.51	0.50
	3 (**3**)	1	0.11	0.08
		2	0.48	0.47
		3	**0.41**	**0.45**

for eye modeling: image-based passive approaches and active IR-based approaches [58]. The model used in our application falls into the second category which is based on the unique intensity distribution and/or shape of the eyes. Indeed, a basic step toward building efficient eye models is choosing an adequate representation of the eye. In our approach we have employed both the edge-orientation histograms which provide spatial information [59] and the co-occurrence matrices which capture the local spatial relationships between gray levels [60].

In our experiments, we have used a database containing 9634 image patches of six classes (closed-eye, closed non-eye, eye with glasses, non-eye with glasses, open-eye, open non-eye). Images are gray-scale infrared of 120 × 64 pixels each and represents the eyes and other parts of the face of different subjects, under different face orientation, and under different illumination conditions. Figure 6 shows examples of images from the different classes. The image database was divided into two sets: training set and test set. The repartition of the different classes in the training and test sets is given in Table 3. The images in the

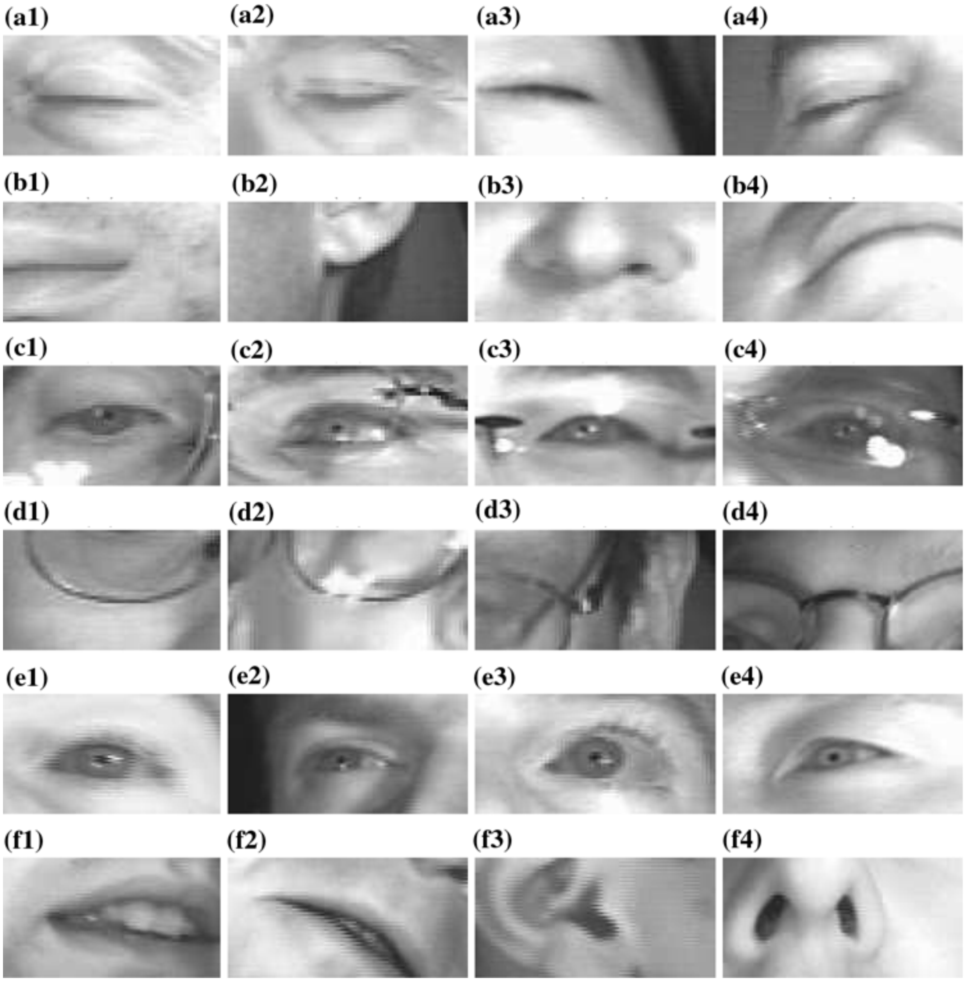


Fig. 6 Sample images from each set. **a1**–**a4** Closed eye, **b1**–**b4** closed non-eye, **c1**–**c4** eye with glasses, **d1**–**d4** non-eye with glasses, **e1**–**e4** open eye, **f1**–**f4** open non-eye

Table 3 Repartition of the different classes in the training and test sets

class	Training set	Testing set
Closed-eye	289	289
Closed non-eye	680	679
Eyewith glasses	695	695
Non-eye with glasses	1,619	1,619
Open-eye	1,082	1,083
Open non-eye	452	452

training set were used to train a classifier. In order to determine the vector of characteristics for each image, we have computed a set a features derived from the co-occurrence matrices. It has been noted that to obtain good results, many co-occurrence matrices should be computed, each one considering a given neighborhood and direction. In our experiments, we have considered the following four neighborhoods: $(1;0),(1;\frac{\pi}{4}),(1;\frac{\pi}{2})$, and $(1;\frac{3\pi}{4})$ [61]. For each of these neighborhoods, we calculated the corresponding co-occurrence, then derived from it the following features: mean, variance, energy, correlation, entropy, contrast, homogeneity, and cluster prominence [62]. Besides, a histogram of edge directions is used. The edge information contained in the images is extracted using the Canny edge operator [63]. The corresponding edge directions are quantized into 72 bins of 5° each. Using the co-occurrence matrices and the histogram of edge directions each image was represented by a 110-dimensional vector. Our Bayesian approach was then used to model each class in the training set by a finite mixture of generalized Dirichlet. Figure 7 shows the number of clusters obtained for the different mixtures representing the different classes in the training set.

In order to perform the assignments of the images in the test set to the different classes, we have used the following rule: $\mathbf{X} \longmapsto \arg\max_k p(\mathbf{X}|\Theta_k)$, where $\mathbf{X}$ is a 110-dimensional vector of features representing an input test image to be assigned to a class and $p(\mathbf{X}|\Theta_k)$ is a mixture of distributions representing class k, $k = 1,\ldots,6$. The confusion matrix for our application is given in Table 4. In this confusion matrix, the cell (class i, class j) represents the number of images from class i which are classified as class j. The number of images misclassified was 183 in all, which represents an accuracy of 96.20%. Table 5 shows the confusion matrix for the Gaussian mixture (an accuracy of 93.06).

4.3 Application to SVM

Support vector machine (SVM) is a two-class classification method that have been used successfully in many applications dealing with data and images classification [64]. An important issue in the case of SVM is the choice of the kernel function $K(\mathbf{X}_i, \mathbf{X}) = \Phi(\mathbf{X}_i)\Phi^{\mathrm{T}}(\mathbf{X})$ which measures the similarity between data vectors $\mathbf{X}_i$ and $\mathbf{X}$ and which is used to overcome the problem of nonlinear separability of the data, by mapping the data into another dot product space F using a nonlinear map

$$\Phi : \mathbb{R}^d \rightarrow F \tag{25}$$

In this new space F, the classes will be linearly separable. However, in most of the applications, the intrinsic structure of the data has been ignored by standard kernels such as Gaussian, radial basis function (RBF), polynomial and Fisher kernels. An interesting approach to this problem is the generation of Mercer Kernels directly from data, using finite mixture models [65], as follows:

$$\begin{aligned} K(\mathbf{X}_i, \mathbf{X}) = \Phi(\mathbf{X}_i)\Phi^T(\mathbf{X}) &= \frac{1}{f(\mathbf{X}_i, \mathbf{X})}\sum_{j=1}^{M} p(j|\mathbf{X}_i)p(j|\mathbf{X}) \\ &= \frac{1}{f(\mathbf{X}_i, \mathbf{X})}\sum_{j=1}^{M} \hat{Z}_{ij}\hat{Z}_{xj} \end{aligned} \tag{26}$$

where $\hat{Z}_{xj}$ is the posterior probability that $\mathbf{X}$ belongs to cluster j, $f(\mathbf{X}_i, \mathbf{X})$ is a normalization factor so that $f(\mathbf{X}_i, \mathbf{X}_i) = 1$, $i = 1,\ldots,l$. With this approach, we have: $\Phi(\mathbf{X}_i) \propto (\hat{Z}_{i1}, \hat{Z}_{i2}, \ldots, \hat{Z}_{iM})$. Note that this Kernel verify Mercer's condition, since it is clearly symmetric and positive semi-definite (the proof is straightforward, see [65] for more details), and then can be applied for SVM [65]. Intuitively, two vectors are similar if they have similar posterior probabilities. Indeed, from a probabilistic point of view two data vectors will have a larger similarity if they are placed in the same cluster of a mixture distribution. In

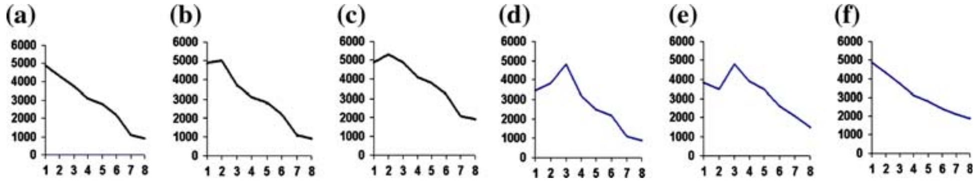


Fig. 7 Number of clusters determined to represent each of the six classes in the training set **a** closed-eye (C1), **b** closed non-eye (C2), **c** eye with glasses (C3), **d** non-eye with glasses (C4), **e** open-eye (C5), **f** open non-eye (C6)

Table 4 Confusion matrix for generalized Dirichlet mixture

	C1	C2	C3	C4	C5	C6
C1	278	5	4	1	1	0
C2	11	659	5	2	1	1
C3	9	7	672	3	2	2
C4	8	13	25	1,567	0	6
C5	4	7	39	1	1,019	13
C6	1	1	1	2	8	439

Table 5 Confusion matrix for Gaussian mixture

	C1	C2	C3	C4	C5	C6
C1	261	18	7	2	1	0
C2	24	643	7	3	1	1
C3	14	9	662	6	2	2
C4	17	31	59	1,497	7	8
C5	7	9	54	4	991	18
C6	4	3	3	4	9	429

Table 6 Classification accuracies using different kernels

Method	Classification accuracy %
PCA	83.21
LDA	68.23
Linear	86.98
Polynomial (degree 2)	86.98
Polynomial (degree 3)	87.19
Polynomial (degree 4)	85.11
Gaussian ($\sigma = 1$)	53.97
Gaussian ($\sigma = 2$)	88.30
Gaussian ($\sigma = 3$)	89.26
Gaussian ($\sigma = 4$)	88.64
Gaussian ($\sigma = 5$)	88.64
Gaussian mixture	93.41
Generalized Dirichlet mixture	96.35

[65], the author proposed the use of Gaussian mixture kernels. The shapes of the mixture distributions can; however, significantly affect the similarity measurement of the two vectors. In this section, we propose the use of the generalized Dirichlet mixture to generate the kernels, since it offers more shapes than the Gaussian. In the following, we will validate the generalized Dirichlet mixture kernel by two applications. In the first one, our model is applied to the eye images database used in the previous section to distinguish automatically eye images from non-eye images. In the second application, we develop a classifier system able to automatically differentiate photographs of real scenes from graphics.

4.3.1 Distinguishing eye from non-eye images

Our goal in this experiment is to use a set of images labeled as eye or non-eye in order to build a classifier that can determine to which class a novel test image belongs. The SVM is trained for on eye patches and non-eye patches in the feature space. The ultimate goal of this learning process is to find the vectors representing the optimal boundary between the two classes. A generative eye and non-eye models are therefore representing the eye and non-eye distributions, in a compact and efficient way. This application may be of interest in the case of driver behavior analysis. The car may want to know if the driver's eyes are kept closed. This application is also motivated by the need to have a model for closed eyes (in the case of blinking, for example), since the majority of eye tackers only work well for open eyes by tracking the eyes locations [66]. As shown in Table 3, our training set has 2,066 positive images (eye) and 2,751 negative images (non-eye) and the testing set contains 2,067 eye images and 2,750 non-eye images. Table 6 shows the classification results when we used SVM with different kernels (linear, polynomial with different degrees, Gaussian with different σ, gaussian mixture, generalized Dirichlet mixture). We can see clearly that we have achieved the best accuracy result by using Gaussian mixture (93.41%) and generalized Dirichlet mixture kernels (96.35%). This can be explained by the fact that these kernels are learned directly from the data and are not predefined like the linear, polynomial or Gaussian kernels.

4.3.2 Distinguishing graphics from real scenes images

With the explosive growth of the World Wide Web, a large amount of multimedia content is now available [67]. This content is mainly composed of images which can be realistic (real scenes images) or photorealistic (generated by computer graphics rendering software). Differentiating graphics from real scenes images is a crucial task in different domains such as digital forensics where it is important to distinguish real nude scenes from computer generated ones [68]. In this application, Generalized Dirichlet mixture kernels are introduced in SVM and applied to develop a classifier system able to automatically differentiate photographs of real scenes from graphics. Shown in Figs. 8 and 9 are images taken from a database of 20,000 real scenes images and 10,000 graphics.

From this database, different features were extracted. These features consist of the image metrics described in [69] to them we have added a set a features (mean, variance, energy, correlation, entropy, contrast, homogeneity, and cluster prominence) derived from the correlogram [70]

Fig. 8 Examples from a database of 20,000 real scenes images

Fig. 9 Examples from a database of 10,000 graphics

Table 7 Classification accuracy average over 20 random training/testing split

Kernel	Training		Testing	
	Images (%)	Graphics (%)	Images (%)	Graphics (%)
Generalized Dirichlet mixture	71.09	98.32	68.32	95.87
Gaussian mixture	68.02	85.34	65.32	82.55
RBF	61.14	87.41	57.08	74.11

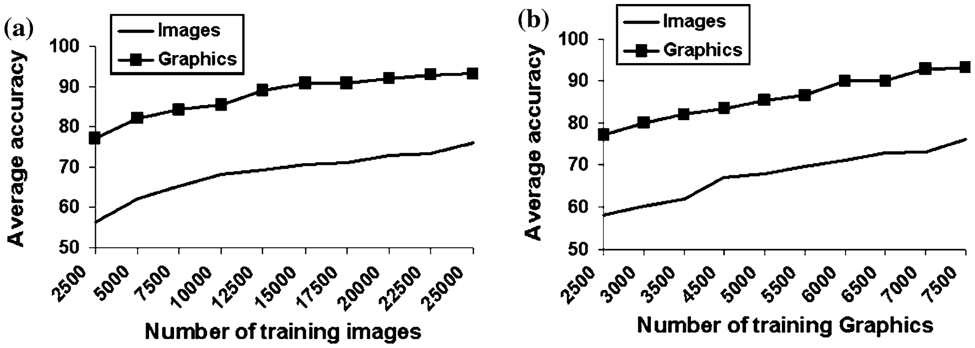


Fig. 10 Average of the classification accuracy as a function of the number of training features vectors. **a** Influence of the number of the training images when the number of graphics in the training set is fixed to 5,000. **b** Influence of the number of the training graphics when the number of images in the training set is fixed to 10,000

by considering 4 different neighborhoods as in the previous section. From the 30,000 features vectors, 10,000 images and 5,000 graphics were used to train the SVM. The remaining vectors were used to test the classifier. Table 7 shows the classification accuracy average when we randomly split, 20 times, the features vectors into training and test sets and by using two other different kernels (Gaussian mixture kernels and RBF kernels). We can see clearly that the generalized Dirichlet mixture kernel gives the best results. Figure 10 shows the influence of the number of features vectors, in the training sets, on the classification

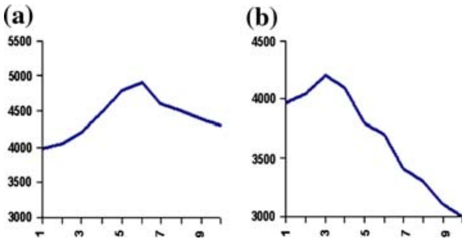


Fig. 11 Number of clusters determined to represent each of the training sets. **a** Images training set, **b** graphics training set

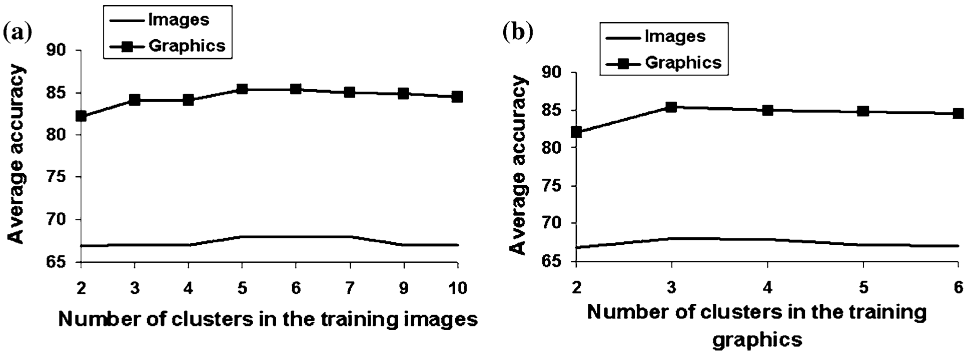


Fig. 12 Classification accuracy as a function of the number of clusters representing the training features vectors. **a** Influence of the number of clusters representing the training images when the number of graphics in the training set is fixed to 5,000 represented by 3 clusters. **b** Influence of the number of clusters representing the training graphics when the number of images in the training set is fixed to 10,000 represented by 6 clusters

accuracy for the test data set when we used generalized Dirichlet mixture kernels. We can note that the accuracy increases as the number of training vectors increases.

We further tested our Generalized Dirichlet mixture kernel by experimenting the influence of the number of components in the mixture model on the accuracy. Figure 11 shows the number of clusters determined by our algorithm to represent the images and graphics training sets. The numbers of clusters determined by our algorithm were six and three for the images and graphics data sets, respectively. Shown in Fig. 12 the classification accuracy as a function of the number of clusters in the training sets. From this Figure, we can see clearly that we reach the best classification accuracy when the number of clusters is equal to the number determined by our algorithm.

5 Conclusion

We have presented a fully Bayesian analysis of finite generalized Dirichlet mixtures. Our Bayesian learning algorithm is based on the development of conjugate prior–posterior distributions and on the Monte Carlo simulation technique of Gibbs sampling mixed with a M–H step. For the estimation of the number of clusters describing the mixture model, we used the marginal likelihood with Laplace approximation. We have shown different results demonstrating clearly that Bayesian estimation and selection gives reliable estimates. We exhibited our Bayesian algorithm in different clustering tasks involving synthetic and real data sets, IR eye modeling and distinguishing real scenes images and graphics. We hope that many other image processing, computer vision and pattern recognition applications will benefit from using this Bayesian model.

Acknowledgments The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC), a NATEQ Nouveaux Chercheurs Grant, and a start-up grant from Concordia University. The author would like to thank the anonymous referees and the associate editor for their helpful comments.

Appendix 1: Proof of Eq. 14

$$
\begin{aligned}
p(\mathbf{X}|\xi_j) &= \prod_{l=1}^{d} \frac{\Gamma(\alpha_{jl}+\beta_{jl})}{\Gamma(\alpha_{jl})\Gamma(\beta_{jl})} X_l^{\alpha_{jl}-1} \left(1-\sum_{t=1}^{l} X_t\right)^{\gamma_{jl}} \\
&= \exp\Bigg[\sum_{l=1}^{d} \log\big(\Gamma(\alpha_{jl}+\beta_{jl})\big) - \log\big(\Gamma(\alpha_{jl})\big) \\
&\quad - \log\big(\Gamma(\beta_{jl})\big) + (\alpha_{jl}-1)\log(X_l) \\
&\quad + \gamma_{jl}\log\left(1-\sum_{t=1}^{l} X_t\right)\Bigg] \\
&= \exp\Bigg[\sum_{l=1}^{d} \Big(\log\big(\Gamma(\alpha_{jl}+\beta_{jl})\big) - \log\big(\Gamma(\alpha_{jl})\big) \\
&\quad - \log\big(\Gamma(\beta_{jl})\big)\Big) \\
&\quad + \sum_{l=1}^{d} \left((\alpha_{jl}-1)\log(X_l) + \gamma_{jl}\log\left(1-\sum_{t=1}^{l} X_t\right)\right)\Bigg] \\
&= \exp\Bigg[\sum_{l=1}^{d} \Big(\log\big(\Gamma(\alpha_{jl}+\beta_{jl})\big) - \log\big(\Gamma(\alpha_{jl})\big) \\
&\quad - \log\big(\Gamma(\beta_{jl})\big)\Big) + \sum_{l=1}^{d} \big((\alpha_{jl}-1)\log(X_l)\big) \\
&\quad + \sum_{l=d+1}^{2d-1} \left((\beta_{jl-d} - \alpha_{jl-d+1} - \beta_{jl-d+1})\log\left(1-\sum_{t=1}^{l-d} X_t\right)\right) \\
&\quad + (\beta_{jd}-1)\log\left(1-\sum_{t=1}^{2d} X_t\right)\Bigg]
\end{aligned}
\tag{27}
$$

Appendix 2: Proof of Eq. 15

$$
\begin{aligned}
&\pi(\xi_j|\mathcal{Z},\mathcal{X}) \propto \pi(\xi_j)\prod_{Z_{ij}=1} p(\mathbf{X}_i|\xi_j)\\
&\propto \exp\Bigg[\sum_{l=1}^{d}\rho_l\alpha_{jl} + \sum_{l=1}^{d}\rho_{l+d}\gamma_{jl}\\
&\quad +\kappa\sum_{l=1}^{d}\big(\log\big(\Gamma(\alpha_{jl}+\beta_{jl})\big) - \log\big(\Gamma(\alpha_{jl})\big) - \log\big(\Gamma(\beta_{jl})\big)\big)\Bigg]\\
&\quad \times\left(\prod_{l=1}^{d}\frac{\Gamma(\alpha_{jl}+\beta_{jl})}{\Gamma(\alpha_{jl})\Gamma(\beta_{jl})}\right)^{n_j}\prod_{Z_{ij}=1}\left(\prod_{l=1}^{d}X_{il}^{\alpha_{jl}-1}\left(1-\sum_{t=1}^{l}X_{it}\right)^{\gamma_{jl}}\right)\\
&\propto \exp\Bigg[\sum_{l=1}^{d}\rho_l\alpha_{jl} + \sum_{l=1}^{d}\rho_{l+d}\gamma_{jl}\\
&\quad +\kappa\sum_{l=1}^{d}\big(\log\big(\Gamma(\alpha_{jl}+\beta_{jl})\big) - \log\big(\Gamma(\alpha_{jl})\big) - \log\big(\Gamma(\beta_{jl})\big)\big)\Bigg]\\
&\quad \times\exp\Bigg[n_j\sum_{l=1}^{d}\big(\log\big(\Gamma(\alpha_{jl}+\beta_{jl})\big) - \log\big(\Gamma(\alpha_{jl})\big)\\
&\quad - \log\big(\Gamma(\beta_{jl})\big)\big) + \sum_{Z_{ij}=1}\Bigg[\sum_{l=1}^{d}\big((\alpha_{jl}-1)\log(X_{il})\big)\\
&\quad +\sum_{l=1}^{d}\left(\gamma_{jl}\log\left(1-\sum_{t=1}^{l}X_{it}\right)\right)\Bigg]\Bigg]\\
&\propto \exp\Bigg[\sum_{l=1}^{d}\alpha_{jl}\left(\rho_l + \sum_{Z_{ij}=1}\log(X_{il})\right)\\
&\quad +\sum_{l=1}^{d}\gamma_{jl}\left(\rho_{l+d} + \sum_{Z_{ij}=1}\log\left(1-\sum_{t=1}^{l}X_{it}\right)\right)\\
&\quad +(\kappa+n_j)\sum_{l=1}^{d}\big(\log\big(\Gamma(\alpha_{jl}+\beta_{jl})\big) - \log\big(\Gamma(\alpha_{jl})\big)\\
&\quad - \log\big(\Gamma(\beta_{jl})\big)\big)\Big]
\end{aligned}
\tag{28}
$$

References

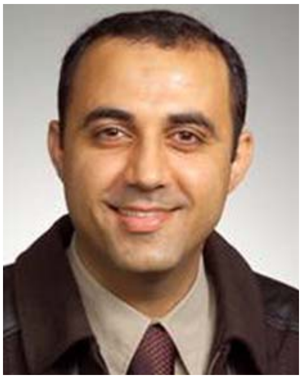
1. Bouguila N, Ziou D (2006) A hybrid SEM algorithm for high-dimensional unsupervised learning using a finite generalized Dirichlet mixture. IEEE Trans Image Process 15(9):2657–2668
2. McLachlan GJ, Peel D (2000) Finite mixture models. Wiley, New York
3. Titterington DM, Smith AFM, Markov UE (1985) Statistical analysis of finite mixture distributions. Wiley, New York
4. Everitt BS, Hand DJ (1981) Finite mixture distributions. Chapman and Hall, London
5. Hammoud RI, Mohr R (2000) Mixture densities for video objects recognition. In: Proceedings of the international conference on pattern recognition, ICPR2000, pp 2071–2075
6. Funaro M, Marinaro M, Petrosino A, Scarpetta S (2002) Finding hidden events in astrophysical data using PCA and mixture of Gaussians clustering. Pattern Anal Appl 5:15–22
7. Jain AK, Duin RPW, Mao J (2000) Statistical pattern recognition: a review. IEEE Trans Pattern Anal Mach Intell 22(1):4–37
8. Bouguila N, Ziou D, Vaillancourt J (2004) Unsupervised learning of a finite mixture model based on the Dirichlet distribution and its application. IEEE Trans Image Process 13(11):1533–1543
9. Bouguila N, Ziou D (2006) Unsupervised selection of a finite Dirichlet mixture model: an MML-based approach. IEEE Trans Knowl Data Eng 18(8):993–1009
10. Bouguila N, Ziou D (2006) Online clustering via finite mixtures of Dirichlet and minimum message length. Eng Appl Artif Intell 19(4):371–379
11. Bouguila N, Ziou D (2005) On fitting finite Dirichlet mixture using ECM and MML. In: Singh S, Singh M, Apté C, Perner P (eds) Pattern recognition and data mining, third international conference on advances in pattern recognition, ICAPR (1). Springer, LNCS, vol 3686. Springer, Heidelberg, pp 172–182
12. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J Roy Stat Soc B 39:1–38
13. McLachlan GJ, Krishnan T (1997) The EM algorithm and extensions. Wiley, New York
14. Robert CP, Casella G (1999) Monte Carlo statistical methods. Springer, Heidelberg
15. Scott DW, Thompson JR (1983) Probability density estimation in higher dimensions. Computer Science and Statistics, pp 173–179
16. Diebolt J, Robert CP (1994) Estimation of finite mixture distributions through Bayesian sampling. J Roy Stat Soc B 56(2):363–375
17. Neal RM (1991) Bayesian mixture modeling. In: Erickson GJ, Smith R, Neudorfer PO (eds) Maximum entropy and Bayesian methods: proceedings of the 11th international workshop on maximum entropy and Bayesian methods of statistical analysis. Kluwer, Dordrecht, pp 197–211
18. Roeder K, Wasserman L (1997) Practical Bayesian density estimation using mixtures of normals. J Am Stat Assoc 92:894–902
19. Bensmail H, Celeux G, Raftery A, Robert CP (1997) Inference in model-based cluster analysis. Stat Comput 7:1–10
20. Tsung IL, Jack CL, Huey FN (2004) Bayesian analysis of mixture modeling using the multivariate t distribution. Stat Comput 14:119–130
21. Tsionas EG (2004) Bayesian inference for multivariate gamma distributions. Stat Comput 14:223–233
22. Bouguila N, Ziou D, Monga E (2006) Practical Bayesian estimation of a finite beta mixture through Gibbs sampling and its applications. Stat Comput 16(2):215–225
23. Brooks SP (2001) On Bayesian analyses and finite mixtures for proportions. Stat Comput 11:179–190
24. Kass RE, Raftery AE (1995) Bayes factors. J Am Stat Assoc 90:773–795
25. Richardson S, Green PJ (1997) On Bayesian analysis of mixtures with an unknown number of components (with discussion). J Roy Stat Soc B 59:731–792
26. Mengersen K, Robert CP (1996) Testing for mixtures: a Bayesian entropic approach (with discussion). In: Dawid A, Lindley D, Berger J, Bernardo J, Smith A (eds) Bayesian statistics, vol 5. Oxford University Press, Oxford, pp 255–276
27. Sahu S, Cheng R (2003) A fast distance based approach for determining the number of components in mixtures. Can J Stat 31:3–22
28. Gruet M, Philippe A, Robert CP (1999) MCMC control spreadsheets for exponential mixture estimation. J Comput Graph Stat 8:298–317
29. Stephens M (2000) Bayesian analysis of mixture models with an unknown number of components: an alternative to reversible jump methods. Ann Stat 28:40–74

30. Cappé O, Robert CP, Rydén T (2002) Reversible jump MCMC converging to birth-and-death MCMC and more general continuous time samplers. J Roy Stat Soc B 65:679–700
31. Bouguila N, Ziou D (2004) A powerful finite mixture model based on the generalized Dirichlet distribution: unsupervised learning and applications. In: Proceedings of the 17th international conference on pattern recognition, ICPR2004, pp 280–283
32. Bouguila N, Ziou D (2004) Dirichlet-based probability model applied to human skin detection. In: IEEE international conference on acoustics, speech, and signal processing, ICASSP2004, pp 521–524
33. Bouguila N, Ziou D (2005) MML-based approach for high-dimensional learning using the generalized Dirichlet mixture. In: Proceedings of the 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)—workshops, vol 03, p 53
34. Bouguila N, Ziou D (2007) High-dimensional unsupervised selection and estimation of a finite generalized Dirichlet mixture model based on minimum message length. IEEE Trans Pattern Anal Mach Intell 29(10):1716–1731
35. Kotz S, Balakrishnan N, Johnson NL (2000) Continous multivariate distributions, vol 1. Wiley-Interscience, New York
36. Beckman RJ, Tietjen GL (1978) Maximum likelihood estimation for the beta distribution. J Stat Comput Simulat 7:253–258
37. Bouguila N, Ziou D, Vaillancourt J (2003) Novel mixtures based on the Dirichlet distribution: application to data and image classification. In: machine learning and data mining in pattern recognition (MLDM2003). LNAI2734, pp 172–181
38. Marin JM, Mengersen K, Robert CP (2004) Bayesian modeling and inference on mixtures of distributions. In: Dey D, Rao CR (eds) Handbook of statistics, vol 25. Elsevier, Amsterdam
39. Robert CP (2001) The Bayesian choice. Springer, Heidelberg
40. Lee PM (1997) Bayesian statistics: an introduction. Arnold
41. Kleiter GD (1992) Bayesian diagnosis in expert systems. Artif Intell 54(1–2):1–32
42. Castillo E, Hadi AS, Solares C (1997) Learning and updating of uncertainty in Dirichlet models. Mach Learn 26(1):43–63
43. Celeux G, Hurn M, Robert CP (2000) Computational and inferential difficulties with mixture posterior distributions. J Am Stat Assoc 95:957–970
44. Lewis SM, Raftery AE (1997) Estimating Bayes factors via posterior simulation with the Laplace–Metropolis estimator. J Am Stat Assoc 92:648–655
45. Schwarz G (1978) Estimating the dimension of a model. Ann Stat 6:461–464
46. Rissanen J (1978) Modeling by shortest data description. Automatica 14:465–471
47. Kass RE, Wasserman L (1995) A reference Bayesian test for nested hypothesis and its relationship to the Schwarz Criterion. J Am Stat Assoc 90:928–934
48. Grauman K, Betke M, Gips J, Bradski GR (2001) Communication via eye blinks: detection and duration analysis in real time. In: IEEE conference on computer vision and pattern recognition, CVPR, pp 1010–1017
49. Haro A, Flickner M, Essa I (2000) Detecting and tracking eyes by using their physiological properties. In: IEEE conference on computer vision and pattern recognition, CVPR, pp 163–168
50. Hansen DW, Hammoud RI (2005) Boosting particle filter-based eye tracker performance through adapted likelihood function to reflexions and light changes. In: IEEE conference on advanced video and signal based surveillance, pp 111–116
51. Hansen DW, Hansen JP, Nielsen M, Johansen AS (2003) Eye typing using Markov and active appearance models. In: IEEE workshop on applications on computer vision, pp 132–136
52. Edenborough N, Hammoud RI, Harbach A et al (2004) Drowsy driver monitor from Delphi. In: Demo session, IEEE conference on computer vision and pattern recognition, CVPR
53. Hansen DW, Hammoud RI (2007) An improved likelihood model for eye tracking. Comp Vis Image Understanding 106:2–3
54. Al-Zubi RT, Abu-Al-Nadi DI (2007) Automated personal identification system based on human Iris analysis. Pattern Anal Appl 10:147–164
55. Hammoud RI (2005) A Robust eye position tracker based on invariant local features, eye motion, and infrared-eye responses. In: SPIE automatic target recognition XV, vol 5807, pp 35–43
56. Bouguila N, Ziou D, Hammoud RI (2007) A Bayesian non-Gaussian mixture analysis: application to eye modeling. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition (CVPR)
57. Jiang X, Binkert M, Achermann B, Bunke H (2000) Towards detection of glasses in facial images. Pattern Anal Appl 3:9–18
58. Zhu Z, Ji Q (2005) Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. Comp Vis Image Understanding 98:124–154
59. Jain AK, Vailaya A (1996) Image retrieval using color and shape. Pattern Recogn 29(8):1233–1244
60. Haralick RM, Shanmugan K, Dinstein I (1973) Texture Features for Image Classification. IEEE Trans Syst Man Cybern 8:610–621
61. Randen T, Husoy JH (1999) Filtering for texture classication: a comparative study. IEEE Trans Pattern Anal Mach Intell 21(4):291–310
62. Unser M (1986) Sum and difference histograms for texture classification. IEEE Trans Pattern Anal Mach Intell 8(1):118–125
63. Canny J (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell 8:679–698
64. Vapnik VN (1998) Statistical learning theory. Wiley, New York
65. Srivastava AN (2004) Mixture density Mercer Kernels: a method to learn kernels directly from data. In: Proceedings of the fourth SIAM international conference on data mining
66. Tian Y, Kanade T, Cohn JF (2000) Dual-state parametric eye tracking. In: Proceedings of IEEE international conference on automatic face and gesture recognition (FGR), pp 110–115
67. Bouguila N, Ziou D (2007) Unsupervised learning of a finite discrete mixture: applications to texture modeling and image databases summarization. J Vis Commun Image Representation 18(4):295–309
68. Lyu S, Farid H (2005) How realistic is photorealistic?. IEEE Trans Signal Process 53:845–850
69. Athitsos V, Swain MJ, Frankel C (1997) Distinguishing photographs and graphics on the World Wide Web. In: IEEE workshop on content-based access of image and video libraries, pp 10–17
70. Huang J, Kumar SR, Mitra M, Zhu W, Zabih R (1999) Spatial color indexing and applications. Int J Comp Vis 35(3):245–268

Author Biographies

Nizar Bouguila received the engineer degree from the University of Tunis in 2000, the M.Sc. and Ph.D degrees from Sherbrooke University in 2002 and 2006, respectively, all in computer science. He is currently an Assistant Professor with the Concordia Institute for Information Systems Engineering (CIISE) at Concordia University, Montreal, Qc, Canada. His research interests include image processing, machine learning, 3D graphics, computer vision, and pattern recognition. In 2007, Dr. Nizar Bouguila received the best Ph.D thesis award in engineering and natural sciences from Sherbrooke University, was awarded the prestigious Prix d'excellence de l'association des doyens des etudes superieures au Quebec (best Ph.D thesis award in engineering and natural sciences in Quebec) and was a runner-up for the prestigious NSERC doctoral prize.

Djemel Ziou received the B.Eng. degree in Computer Science from the University of Annaba (Algeria) in 1984, and Ph.D degree in Computer Science from the Institut National Polytechnique de Lorraine (INPL), France in 1991. From 1987 to 1993 he served as lecturer in several universities in France. During the same period, he was a researcher in the Centre de Recherche en Informatique de Nancy (CRIN) and the Institut National de Recherche en Informatique et Automatique (INRIA) in France. Presently, he is full Professor at the department of computer science, Sherbrooke University, QC, Canada. He is holder of the NSERC/Bell Canada Research Chair in personal imaging. He has served on numerous conference committees as member or chair. He heads the laboratory MOIVRE and the consortium CoRIMedia which he founded. His research interests include image processing, information retrieval, computer vision and pattern recognition.



Dr. Riad I. Hammoud is a research scientist working on safety and security applications at Delphi Electronics & Safety. He holds a PhD degree in "Computer Vision and Robotics" from INRIA Rhone-Alpes (France) since February 2001. He authored several Springer books including "Face Biometrics for Personal Identification" and "Passive Eye Monitoring". He was appointed in 2004 and 2005 as guest editor of two special issues of IJCV and CVIU. He is the founder of the IEEE workshop series on perception Beyond the Visible Spectrum (OTCBVS). He is the architect of the core algorithms of several vision-based safety and security products of Delphi including driver fatigue and distraction alert, and driver identification. Dr. Riad I. Hammoud holds numerous patents and was nominated by US government as an outstanding researcher/professor in 2005.