A Hybrid SEM Algorithm for High-Dimensional Unsupervised Learning Using a Finite Generalized Dirichlet Mixture

Nizar Bouguila, *Member, IEEE*, and Djemel Ziou

***Abstract*—This paper applies a robust statistical scheme to the problem of unsupervised learning of high-dimensional data. We develop, analyze, and apply a new finite mixture model based on a generalization of the Dirichlet distribution. The generalized Dirichlet distribution has a more general covariance structure than the Dirichlet distribution and offers high flexibility and ease of use for the approximation of both symmetric and asymmetric distributions. We show that the mathematical properties of this distribution allow high-dimensional modeling without requiring dimensionality reduction and, thus, without a loss of information. This makes the generalized Dirichlet distribution more practical and useful. We propose a hybrid stochastic expectation maximization algorithm (HSEM) to estimate the parameters of the generalized Dirichlet mixture. The algorithm is called stochastic because it contains a step in which the data elements are assigned randomly to components in order to avoid convergence to a saddle point. The adjective "hybrid" is justified by the introduction of a Newton–Raphson step. Moreover, the HSEM algorithm autonomously selects the number of components by the introduction of an agglomerative term. The performance of our method is tested by the classification of several pattern-recognition data sets. The generalized Dirichlet mixture is also applied to the problems of image restoration, image object recognition and texture image database summarization for efficient retrieval. For the texture image summarization problem, results are reported for the *Vistex* texture image database from the MIT Media Lab.**

***Index Terms*—Clustering, correlogram, expectation maximization (EM), finite mixture models, generalized Dirichlet, high-dimensional data, hybrid stochastic expectation maximization algorithm (HSEM), image database summarization, image object recognition, image restoration, maximum likelihood (ML), SEM, Vistex.**

I. Introduction

HIGH-DIMENSIONAL data appears in many areas, such as pattern recognition, computer vision, and signal processing [2]. In problems involving the detection of real-world objects (e.g., faces, vehicles, buildings), for example, very high-dimensional feature vectors are necessary. In these conditions, the use of probabilistic approaches is always delicate. The complexity grows exponentially with the dimension, rapidly outstripping the computational and memory storage capabilities of computers. It is known that the support vector machine (SVM) [3] does not suffer from the high-dimensionality limitation and allows the use of the full dimensionality of the data. However, probabilistic models are still important for their theoretical and predictive properties. To avoid problems of dimensionality when using probabilistic models, the most common approaches involve the use of dimensionality reduction, which has been the subject of much research for the past few decades. Some good overviews of dimensionality reduction techniques are available in [4] and [5].

The approach presented in this paper is not aimed at dimensionality reduction. In fact, it is based on the statistical properties of the data through the use of generalized Dirichlet finite mixture models. The data is transformed in such a way that density estimation in the transformed space is simpler and more accurate. This transformation is possible thanks to the convenient mathematical properties of the generalized Dirichlet density. Finite mixtures of distributions have provided a mathematics based approach to the modeling of a wide variety of random phenomena [6]. In statistical pattern recognition, finite mixtures permit a formal approach to unsupervised learning. Indeed, finite mixtures can be viewed as a superimposition of a finite number of component densities and, thus, adequately model situations in which each data element is assumed to have been generated by one (unknown) component. The choice of the component model is very critical in mixture decomposition. The number of components required to model the mixture and the modeling capabilities are directly related to the component model used. For multivariate data, attention has focused on the use of multivariate normal components because of their isotropic nature, along with their capability to represent the distribution compactly by a mean vector and a covariance matrix. This choice offers some computational convenience, too. Indeed, normal components can be easily fitted by maximum likelihood (ML) via the expectation maximization (EM) algorithm proposed by Dempster *et al.* [7] and widely studied by McLachlan and Krishnan [8]. However, for many applications, the Gaussian distribution is not the best choice and will fail to discover *true* structure where the partitions are clearly non-Gaussian [9], [10]. In fact, due to its definition, the Gaussian cannot approximate asymmetric distributions

Manuscript received December 23, 2004; revised January 18, 2006. An earlier, shorter version of this work was presented at the 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), in part by Heritage Canada, and in part by Bell Canada through its Bell University Laboratories R&D program. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Joachim M. Buhmann.

N. Bouguila is with the Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montréal, QC H3G IT7 Canada (e-mail: bouguila@ciise.concordia.ca).

D. Ziou is with the Département d'Informatique, Sherbrooke University, Sherbrooke, QC J1K 2R1 Canada (e-mail: djemel.ziou@usherbrooke.ca).

Digital Object Identifier 10.1109/TIP.2006.877379

well. Another problem arises with high-dimensional data. One solution is the consideration of a diagonal covariance matrix, but in some applications we may be obliged to use a full covariance matrix. In previous works, we have shown that the Dirichlet [11] and multinomial Dirichlet [12] distributions can be good choices to overcome the disadvantages of the Gaussian in modeling continuous and discrete data, respectively. Despite its flexibility, the Dirichlet distribution has a very restrictive negative covariance structure, as we will show in the next section. In this paper, we present a generalization of the Dirichlet distribution which has a more general covariance structure than the Dirichlet distribution. This makes the generalized Dirichlet distribution more practical and useful in Bayesian learning scenarios in general and finite mixture modeling in particular.

The standard method used to fit finite mixture models is the well-known EM algorithm. The EM algorithm is used to locate a ML estimate of the mixture parameters [13]. However, the EM algorithm for finite mixture fitting has several drawbacks. It is sensitive to initialization because the likelihood function of a mixture model is generaly multimodal. Besides, the increase of the likelihood function at each step of the algorithm ensures its convergence to the maximum likelihood estimator in the case of unimodal likelihoods but implies a local convergence for multimodal likelihoods. Another problem concerns estimation in high-dimensional settings. Indeed, the number of parameters to estimate rises dramatically with the dimension, increasing the complexity of the model.

In this paper, we deal simultaneously with the above-mentioned problems. We propose a hybrid stochastic expectation maximization (HSEM) algorithm which is a generalization of the EM algorithm. The algorithm is called stochastic because it contains a step in which the data elements are randomly assigned to components in order to avoid convergence to a saddle point. The adjective "hybrid" is justified by the introduction of a Newton–Raphson step. Moreover, the HSEM algorithm allow high-dimensional probabilistic modeling and autonomously selects the number of components by the introduction of an agglomerative term.

The rest of the paper is organized as follows. In Section II, we present the generalized Dirichlet distribution in detail. Section III discusses the basic concepts of the EM algorithm, proposes the HSEM as a method to fit the generalized Dirichlet mixture and gives the complete estimation algorithm. Section IV is devoted to experimental results, and Section V ends the paper with some concluding remarks.

II. Generalized Dirichlet Distribution

If the random vector $\vec{X} = (X_1, \dots, X_d)$ follows a Dirichlet distribution the joint density function is given by [14], [15]

$$p(X_1, \dots, X_d) = \frac{\Gamma(|\vec{\alpha}|)}{\prod_{i=1}^{d+1} \Gamma(\alpha_i)} \prod_{i=1}^{d+1} X_i^{\alpha_i - 1} \tag{1}$$

where

$$\sum_{i=1}^{d} X_i < 1 \tag{2}$$

$$|\vec{X}| = \sum_{i=1}^{d} X_i, \qquad 0 < X_i < 1 \quad \forall i = 1, \dots, d \tag{3}$$

$$X_{d+1} = 1 - |\vec{X}| \tag{4}$$

$$|\vec{\alpha}| = \sum_{i=1}^{d+1} \alpha_i, \qquad \alpha_i > 0 \quad \forall i = 1, \dots, d+1. \tag{5}$$

This distribution is the multivariate extension of the 2-parameter Beta distribution, which is given by

$$p_{\text{Beta}}(X_1) = \frac{\Gamma(\alpha_1 + \alpha_2)}{\Gamma(\alpha_1)\Gamma(\alpha_2)} X_1^{\alpha_1 - 1}(1 - X_1)^{\alpha_2 - 1}. \tag{6}$$

The mean and variance of the Dirichlet distribution satisfy the following conditions:

$$E(X_i) = \frac{\alpha_i}{|\vec{\alpha}|} \tag{7}$$

$$\text{Var}(X_i) = \frac{\alpha_i(|\vec{\alpha}| - \alpha_i)}{|\vec{\alpha}|^2(|\vec{\alpha}| + 1)} \tag{8}$$

and the covariance between X_i and X_j is

$$\text{Cov}(X_i, X_j) = -\frac{\alpha_i \alpha_j}{|\vec{\alpha}|^2(|\vec{\alpha}| + 1)}. \tag{9}$$

Thus, any two random variables in $\vec{X}$ are negatively correlated. In some practical cases, two random variables may be positively correlated. Connor and Mosimann [16] have generalized the Dirichlet distribution as follows:

$$p(X_1, \dots, X_d) = \prod_{i=1}^{d} \frac{\Gamma(\alpha_i + \beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} X_i^{\alpha_i - 1} \left(1 - \sum_{j=1}^{i} X_j\right)^{\gamma_i} \tag{10}$$

for $\sum_{i=1}^{d} X_i < 1$ and $0 < X_i < 1$ for $i = 1, \dots, d$, where $\gamma_i = \beta_i - \alpha_{i+1} - \beta_{i+1}$ for $i = 1, \dots, d-1$ and $\gamma_d = \beta_d - 1$. Note that the generalized Dirichlet distribution is reduced to a Dirichlet distribution when $\beta_i = \alpha_{i+1} + \beta_{i+1}$. The mean and variance of the generalized Dirichlet distribution satisfy the following conditions:

$$E(X_i) = \frac{\alpha_i}{\alpha_i + \beta_i} \prod_{k=1}^{i-1} \frac{\beta_k + 1}{\alpha_k + \beta_k} \tag{11}$$

$$Var(X_i) = E(X_i) \left(\frac{\alpha_i + 1}{\alpha_i + \beta_i + 1} \prod_{k=1}^{i-1} \frac{\beta_k + 1}{\alpha_k + \beta_k} + 1 - E(X_i) \right) \tag{12}$$

and the covariance between X_i and X_j is

$$\mathrm{Cov}(X_i,X_j)=E(X_j)\left(\frac{\alpha_i}{\alpha_i+\beta_i+1}\prod_{k=1}^{i-1}\frac{\beta_k+1}{\alpha_k+\beta_k}+1-E(X_i)\right). \tag{13}$$

Numerous other properties of this distribution are given in [17].

III. Learning the Finite Generalized Dirichlet Mixture

A. The Finite Generalized Dirichlet Mixture and the EM Algorithm

A generalized Finite Dirichlet mixture with M components is defined as

$$p(\vec{X}|\Theta)=\sum_{j=1}^{M}p(\vec{X}|\vec{\Theta}_j)P(j) \tag{14}$$

where the $P(j)$ are the mixing probabilities and $p(\vec{X}|\vec{\Theta}_j)$ is the generalized Dirichlet distribution.

Each $\vec{\Theta}_j=(\alpha_{j1},\beta_{j1},\dots,\alpha_{jd},\beta_{jd})$ is the set of parameters defining the jth component, and Θ is the complete set of parameters, $\Theta=(\vec{\Theta}_1,\dots,\vec{\Theta}_M,P(1),\dots,P(M))$, needed to specify the mixture. Of course, being probabilities, the $P(j)$ must satisfy

$$0<P(j)\le 1,\quad j=1,\dots,M \tag{15}$$

$$\sum_{j=1}^{M}P(j)=1. \tag{16}$$

Given a set of N independent vectors $\mathcal{X}=(\vec{X}_1,\dots,\vec{X}_N)$, the log-likelihood corresponding to a M-component is

$$L(\Theta,\mathcal{X})=\log\prod_{i=1}^{N}p(\vec{X}_i|\Theta)=\sum_{i=1}^{N}\log\sum_{j=1}^{M}p(\vec{X}_i|\vec{\Theta}_j)P(j). \tag{17}$$

It is well known that the maximum likelihood (ML) estimate

$$\hat{\Theta}_{\mathrm{ML}}=\arg\max_{\Theta}\{L(\Theta,\mathcal{X})\} \tag{18}$$

which cannot be found analytically. The maximization defining the ML estimates is subject to the constraints in (15) and (16). The ML estimates of the mixture parameters can be obtained using EM and related techniques [8]. The EM algorithm [7] is a general approach to maximum likelihood in the presence of incomplete data. In EM, the "complete" data are considered to be $Y_i=\{\vec{X}_i,\vec{Z}_i\}$, where $\vec{Z}_i=(Z_{i1},\dots,Z_{iM})$, with

$$Z_{ij}=\begin{cases}1, & \text{if } \vec{X}_i \text{ belongs to class } j\\ 0, & \text{otherwise}\end{cases} \tag{19}$$

constituting the "missing" data. The relevant assumption is that the density of an observation $\vec{X}_i$, given $\vec{Z}_i$, is given by $\prod_{j=1}^{M}p(\vec{X}_i|j,\vec{\Theta}_j)^{Z_{ij}}$. The resulting *complete-data log-likelihood* is

$$L(\Theta,\mathcal{Z},\mathcal{X})=\sum_{i=1}^{N}\sum_{j=1}^{M}Z_{ij}\log(p(\vec{X}_i|\vec{\Theta}_j)P(j)). \tag{20}$$

The EM algorithm produces a sequence of estimates $\{\Theta^t,t=0,1,2\dots\}$ by applying two steps in alternation until some convergence criterion is satisfied.

1) **E-step:** Compute $\hat{Z}_{ij}$ given the parameter estimates from the initialization

$$\hat{Z}_{ij}=\frac{p(\vec{X}_i|\vec{\Theta}_j)P(j)}{\sum_{l=1}^{M}p(\vec{X}_i|\vec{\Theta}_l)P(l).}$$

2) **M-step:** Update the parameter estimates according to

$$\hat{\Theta}=\arg\max_{\Theta}L(\Theta,\mathcal{Z},\mathcal{X}).$$

The quantity $\hat{Z}_{ij}$ is the conditional expectation of Z_{ij} given the observation $\vec{X}_i$ and parameter vector Θ. The value Z^*_{ij} of $\hat{Z}_{ij}$ at a maximum of (20) is the conditional probability that observation i belongs to class j (the *a posteriori* probability); the classification of an observation X_i is taken to be $\{k/Z^*_{ik}=\max_j Z^*_{ij}\}$, which is the Bayes rule.

B. Proposed Approach: The Hybrid SEM Algorithm

The EM algorithm has been shown to monotonically increase the log-likelihood function [8]. However, it suffers from some drawbacks. Indeed, it is sensitive to initialization and converges slowly in the case of mixture models estimation. Besides, an important problem in mixture modeling is the selection of the number of components. We now address these issues. A number of so-called deterministic approaches based on criterion functions have been proposed such as the akaike information criterion (AIC) [18], the minimum description length (MDL) [19], and Schwartz's Bayesian inference criterion (BIC) [20]. These methods start by obtaining a set of candidate models by running EM with different numbers of components each time. After convergence, the optimal value is selected according to

$$\hat{M}=\arg\min\{C(\hat{\Theta}(M),M),M=M_{\min},\dots,M_{\max}\} \tag{21}$$

where $C(\hat{\Theta}(M),M)$ is some model selection criterion and $\hat{\Theta}(M)$ is an estimate of the mixture parameters assuming that there are M components. Usually, $C(\hat{\Theta}(M),M)=-\log p(\mathcal{X}|\hat{\Theta}(M))+f(M)$, where $f(M)$ is an increasing function penalizing higher values of M. Criterion functions of this type have been proposed under a Bayesian model selection framework. However, the majority of the criteria such as the MDL do not really take into account the prior information about the parameter vector Θ. In fact, the prior is

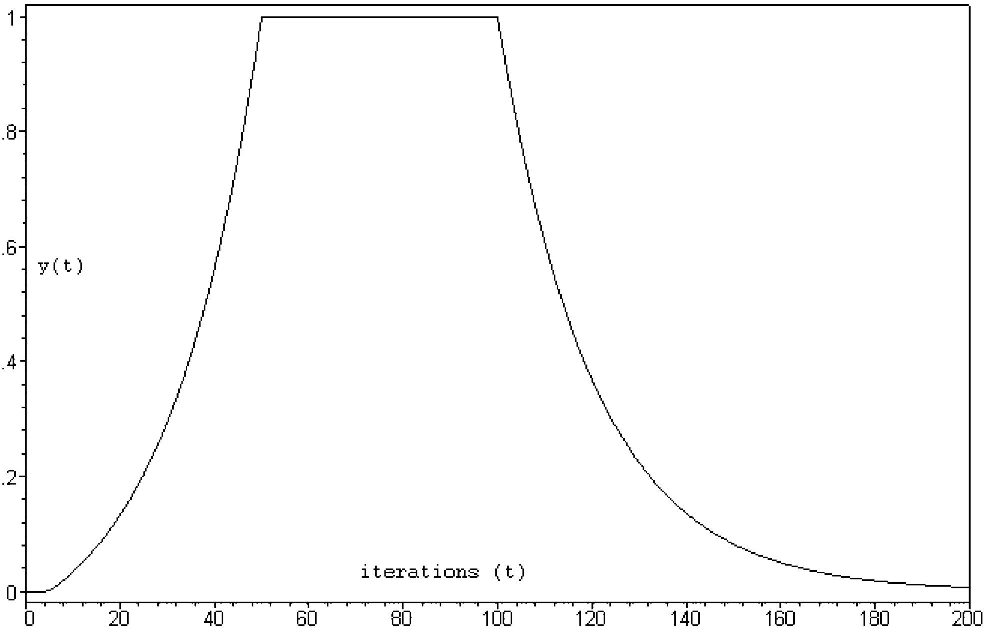


Fig. 1. Plot of the function $y(t)$ versus iterations t.

assumed to be flat. Besides, this particular method of using the criterion is time consuming, due to the need to run the algorithm several times. In our approach, we determine the appropriate number of components and the parameters of each component simultaneously by using an ad hoc agglomerative technique introduced by Frigui and Krishnapuram [21] and used in the case of Gaussian and Pearson mixtures [22]. The objective function, using the Lagrange multiplier Λ to incorporate the constraints in (15) and (16), is, thus

$$\Phi(\Theta,\mathcal{X}) = \sum_{i=1}^{N}\log\sum_{j=1}^{M} p(\vec{X}_i|\vec{\Theta}_j)P(j) + \Lambda\left(1-\sum_{j=1}^{M}P(j)\right)+\mu\sum_{j=1}^{M}P^2(j) \qquad (22)$$

which we maximize subject to (15) and (16). The first term in (22) is the log-likelihood function, and it assumes its global maximum value when each component represents only one of the feature vectors. The last term (the agglomerative term) reaches its maximum when all of the feature vectors are modeled by a single component, i.e., when $P(j_1) = 1$ for some j_1 and $P(j) = 0, \forall j, j \neq j_1$. The algorithm starts with an overspecified number of components in the mixture, and as it proceeds, components compete to model the data. The update for the mixture coefficients probabilities can be shown to be

$$P(j)^{(t)} = \frac{\sum_{i=1}^{N} \hat{Z}_{ij}^{(t-1)} + 2\mu p^2(j)^{(t-1)}}{N + 2\mu\sum_{j=1}^{M} p^2(j)^{(t-1)}}. \qquad (23)$$

The choice of μ is critical to the effective performance of the algorithm, since it specifies the tradeoff between the required likelihood of the data and the number of components to be found. If μ is too small, the second term will be neglected and the number of clusters will not be reduced. If μ is too large, the first term will be neglected and all points will be lumped into a single component. The value of μ should be chosen such that both terms are of the same order of magnitude [21]. Thus, we choose μ to be the ratio of the first term to the last term in (22) at each iteration t, i.e.,

$$\mu(t) = y(t)\frac{\sum_{i=1}^{N}\log\left(\sum_{j=1}^{M} p^{(t-1)}(\vec{X}_i|\vec{\Theta}_j)P^{(t-1)}(j)\right)}{\sum_{j=1}^{M} P^2(j)^{(t-1)}}. \qquad (24)$$

The function $y(t)$ is introduced in order to provide a good transition from the nonagglomerative scheme to the agglomerative one. The value of μ is, therefore, increased gradually and we choose the function $y(t)$ as follows [22]:

$$y(t) = \begin{cases} 0, & t \leq 5 \\ \frac{\exp(0.05t-0.25)-1}{\exp(2.25)-1}, & 5 < t \leq 50 \\ 1, & 50 < t \leq 100 \\ \exp(5.0 - .05t), & t > 100 \end{cases}. \qquad (25)$$

The plot of the profile of $y(t)$ versus iteration number is shown in Fig. 1.

The problem now is how to find the parameters Θ_j. In [23], we used the Fisher scoring method for the estimation of these parameters. This method involves the inverse of the $(2 \times d) \times (2 \times d)$ Fisher information matrix, which is not easy to compute especially for high-dimensional data. Here, we try to find an alternative method to overcome this drawback. As Scott and Thompson have observed, *the problem of density estimation in higher dimension involves first of all finding where the action is* [5]. We, therefore, begin by identifying the important classes by an efficient initialization algorithm (see Section III-C), and then use the convenient properties of the generalized Dirichlet distribution to refine the estimates. To estimate the parameters $\vec{\Theta}_j$, we will use an interesting property of the generalized Dirichlet distribution. If a vector $\vec{X}_i$ has a generalized Dirichlet distribution, then we can construct a vector $\vec{W}_i = (W_{i1}, \dots, W_{id})$ using the following geometric transformation $W_{ij} = T(X_{ij})$

$$T(X_{ij}) = \begin{cases} X_{ij}, & \text{if } j = 1 \\ \frac{X_{ij}}{(1-X_{i1}-\cdots-X_{ij-1})}, & \text{for } j = 2, 3, \dots, d \end{cases} \cdot \tag{26}$$

In this vector $\vec{W}_i$, each W_{il}, $l = 1, \dots, d$ has a Beta distribution with parameters α_{il} and β_{il} and the parameters $\{\alpha_{il}, \beta_{il}, l = 1, \dots, d\}$ define the generalized Dirichlet distribution which characterizes $\vec{X}_i$ [17]. The problem of estimating the parameters of a generalized Dirichlet mixture can, thus, be reduced to the estimation of the parameters of d Beta mixtures. To accomplish this, we must maximize this equation for every dimension

$$\Phi_W(\theta_l, \mathcal{W}) = \sum_{i=1}^{N} \log \left(\sum_{j=1}^{M} p_{\text{beta}}(W_{il}|\theta_{jl}) P(j) \right) \tag{27}$$

where $\mathcal{W} = (W_{1l}, \dots, W_{Nl})$, $0 < l < d$, $\theta_l = (\alpha_{1l}, \beta_{1l}, \dots, \alpha_{\text{ML}}, \beta_{\text{ML}})$, $\theta_{jl} = (\alpha_{jl}, \beta_{jl})$ and $P(j)$ are the mixing parameters established by (23). The maximization of (27) is equivalent to

$$\frac{\partial}{\partial \theta_{jl}} \Phi_W(\theta_l, \mathcal{W}) = 0 \quad \forall \quad 0 < l < d. \tag{28}$$

In order to estimate the θ_{jl} parameters we will use Fisher's scoring method [24]. The scoring method is based on the first, second and mixed derivatives of the function $\Phi_W(\theta_l, \mathcal{W})$. We will, therefore, compute these derivatives. Calculating the derivative of $\Phi_W(\theta_l, \mathcal{W})$ with respect to θ_{jl}, we obtain

$$\frac{\partial \Phi_W(\theta_l, \mathcal{W})}{\partial \theta_{jl}} = \sum_{i=1}^{N} p_{\text{beta}}(\theta_{jl}|W_{il}) \frac{\partial \log(p_{\text{beta}}(W_{il}|\theta_{jl}))}{\partial \theta_{jl}} \tag{29}$$

where $p_{\text{beta}}(\theta_{jl}|W_{il})$ is the posterior probability. Thus, we can deduce $\partial \Phi_W(\theta_l, \mathcal{W}) / \partial \alpha_{jl}$ and $\partial \Phi_W(\theta_l, \mathcal{W}) / \partial \beta_{jl}$. Since we require that the α_{jl} and β_{jl} be strictly positive, and we want the parameters upon which we will derive to be unconstrained, we reparametrize, setting $\alpha_{jl} = e^{\alpha'_{jl}}$ and $\beta_{jl} = e^{\beta'_{jl}}$, where α'_{jl} and β'_{jl} are unconstrained real numbers. Thus, the partial derivatives of $\Phi_W(\theta_l, \mathcal{W})$ with respect to α'_{jl} and β'_{jl} are as follows:

$$\begin{aligned} \frac{\partial \Phi_W(\theta_l, \mathcal{W})}{\partial \alpha'_{jl}} &= \frac{\partial \Phi_W(\theta_l, \mathcal{W})}{\partial \alpha_{jl}} \frac{\partial \alpha_{jl}}{\partial \alpha'_{jl}} \\ &= \alpha_{jl} \Bigg[(\Psi(\alpha_{jl} + \beta_{jl}) - \Psi(\alpha_{jl})) \sum_{i=1}^{N} p_{\text{beta}}(\theta_{jl}|W_{il}) \\ &\quad + \sum_{i=1}^{N} p_{\text{beta}}(\theta_{jl}|W_{il}) \log(W_{il}) \Bigg] \end{aligned} \tag{30}$$

$$\begin{aligned} \frac{\partial \Phi_W(\theta_l, \mathcal{W})}{\partial \beta'_{jl}} &= \frac{\partial \Phi_W(\theta_l, \mathcal{W})}{\partial \beta_{jl}} \frac{\partial \beta_{jl}}{\partial \beta'_{jl}} \\ &= \beta_{jl} \Bigg[(\Psi(\alpha_{jl} + \beta_{jl}) - \Psi(\beta_{jl})) \sum_{i=1}^{N} p_{\text{beta}}(\theta_{jl}|W_{il}) \\ &\quad + \sum_{i=1}^{N} p_{\text{beta}}(\theta_{jl}|W_{il}) (\log(1 - W_{il})) \Bigg]. \end{aligned} \tag{31}$$

By computing the second and mixed derivatives of $\Phi_W(\theta_l, \mathcal{W})$, we obtain

$$\begin{aligned} &\frac{\partial^2}{\partial {\alpha'_{jl}}^2} \Phi_W(\theta_l, \mathcal{W}) \\ &\quad = \frac{\partial}{\partial \alpha'_{jl}} \Phi_W(\theta_l, \mathcal{W}) \\ &\quad + \alpha_{jl}^2 (\Psi'(\alpha_{jl} + \beta_{jl}) - \Psi'(\alpha_{jl})) \sum_{i=1}^{N} p_{\text{beta}}(\theta_{jl}|W_{il}) \\ &\quad + \alpha_{jl} \Bigg[(\Psi(\alpha_{jl} + \beta_{jl}) - \Psi(\alpha_{jl})) \sum_{i=1}^{N} \frac{\partial p_{\text{beta}}(\theta_{jl}|W_{il})}{\partial \alpha'_{jl}} \\ &\quad + \sum_{i=1}^{N} \frac{\partial}{\partial \alpha'_{jl}} p_{\text{beta}}(\theta_{jl}|W_{il}) (\log(W_{il})) \Bigg] \end{aligned} \tag{32}$$

$$\begin{aligned} &\frac{\partial^2}{\partial \alpha'_{jl} \partial \beta'_{jl}} \Phi_W(\theta_l, \mathcal{W}) \\ &\quad = \alpha_{jl} \beta_{jl} \Psi'(\alpha_{jl} + \beta_{jl}) \sum_{i=1}^{N} p_{\text{beta}}(\theta_{jl}|W_{il}) \\ &\quad + \alpha_{jl} \Bigg[(\Psi(\alpha_{jl} + \beta_{jl}) - \Psi(\alpha_{jl})) \sum_{i=1}^{N} \frac{\partial p_{\text{beta}}(\theta_{jl}|W_{il})}{\partial \beta'_{jl}} \\ &\quad + \sum_{i=1}^{N} \frac{\partial p_{\text{beta}}(\theta_{jl}|W_{il})}{\partial \beta'_{jl}} (\log(W_{il})) \Bigg] \end{aligned} \tag{33}$$

where $\Psi'(.)$ is the trigamma function. The computation of the other mixed derivatives $(\partial^2 / \partial^2 \beta'_{jl}) \Phi_W(\theta_l, \mathcal{W})$ and $(\partial^2 / \partial \beta'_{jl} \partial \alpha'_{jl}) \Phi_W(\theta_l, \mathcal{W})$ is straightforward. Note that we need to compute the derivative of the a *posteriori* probability $p_{\text{beta}}(\theta_{jl}|W_{il})$ with respect to α'_{jl} and β'_{jl}

$$\begin{aligned} &\frac{\partial}{\partial \alpha'_{jl}} p_{\text{beta}}(\theta_{jl}|W_{il}) \\ &\quad = \alpha_{jl} \times p_{\text{beta}}(\theta_{jl}|W_{il}) (1 - p_{\text{beta}}(\theta_{jl}|W_{il})) \\ &\quad\quad \times (\Psi(\alpha_{jl} + \beta_{jl}) - \Psi(\alpha_{jl}) + \log(W_{il})) \end{aligned} \tag{34}$$

$$\begin{aligned} &\frac{\partial}{\partial \beta'_{jl}} p_{\text{beta}}(\theta_{jl}|W_{il}) \\ &\quad = \beta_{jl} \times p_{\text{beta}}(\theta_{jl}|W_{il}) (1 - p_{\text{beta}}(\theta_{jl}|W_{il})) \\ &\quad\quad \times (\Psi(\alpha_{jl} + \beta_{jl}) - \Psi(\beta_{jl}) + \log(1 - W_{il})). \end{aligned} \tag{35}$$

Given a set of initial estimates, Fisher's scoring method can now be used. The iterative scheme of the Fisher method is given by the following equation:

$$\begin{pmatrix} \hat{\alpha'}_{jl} \\ \hat{\beta'}_{jl} \end{pmatrix}^{(t)} = \begin{pmatrix} \hat{\alpha'}_{jl} \\ \hat{\beta'}_{jl} \end{pmatrix}^{(t-1)} + V^{(t-1)} \times \begin{pmatrix} \frac{\partial \Phi_W}{\partial \hat{\alpha'}_{jl}} \\ \frac{\partial \Phi_W}{\partial \hat{\beta'}_{jl}} \end{pmatrix}^{(t-1)} \quad (36)$$

where j is the class number: $1 \leq j \leq M$ and l is the current dimension: $1 \leq l \leq d$.

The matrix V is obtained as the inverse of the Fisher information matrix $\mathbf{I}$. The information matrix $\mathbf{I}$ is

$$\mathbf{I} = \begin{pmatrix} -E\left[\frac{\partial^2}{\partial^2 \alpha'_{jl}} \Phi_W(\theta_l, \mathcal{W})\right] & -E\left[\frac{\partial^2}{\partial \alpha'_{jl} \partial \beta'_{jl}} \Phi_W(\theta_l, \mathcal{W})\right] \\ -E\left[\frac{\partial^2}{\partial \beta'_{jl} \partial \alpha'_{jl}} \Phi_W(\theta_l, \mathcal{W})\right] & -E\left[\frac{\partial^2}{\partial {\beta'_{jl}}^2} \Phi_W(\theta_l, \mathcal{W})\right] \end{pmatrix}. \quad (37)$$

Given sufficiently accurate starting values, the sequence of iterates produced by the Fisher scoring method displays local quadratic convergence to a solution $\hat{\theta}_{jl}$. That is, given a norm $||.||$ on the parameter space, there is a constant h such that

$$||\theta_{jl}^{(t)} - \hat{\theta}_{jl}|| \leq h||\theta_{jl}^{(t-1)} - \hat{\theta}_{jl}||^2 \quad (38)$$

holds for $t = 1, 2, \ldots$ Quadratic convergence is very fast, which is regarded as the major strength of the Fisher scoring method, allowing it to overcome the slow convergence of the EM algorithm. This rapid convergence can be improved by introducing a stochastic step in the EM algorithm [25], [26]. In fact, the stochastic step prevents the sequence of estimates Θ^t from staying near an unstable stationary point of the likelihood function [26].

Algorithm

In order to make our algorithm less sensitive to local maxima, we have used some initialization schemes including the Fuzzy C-means and the method of moments (MM) [11]. Our initialization method can be resumed as follows.

INITIALIZATION Algorithm

1) INPUT: d-dimensional data $\vec{X}_i$, $i = 1, \ldots, N$ and the number of clusters M.
2) Apply the Fuzzy C-means to obtain the elements, covariance matrix and mean of each component.
3) Compute the $\vec{W}_i = (W_{i1}, \ldots, W_{id})$ form the $\vec{X}_i$. $W_{i1} = X_{i1}$ and $W_{il} = X_{il}/V_{il-1}$ for $l = 2, 3, \ldots, d$, where $V_{il} = 1 - X_{i1} - X_{i2} - \cdots - X_{il}$.
4) Apply MM for each component j and for each dimension l to obtain the vector of parameters $\vec{\theta}_{jl}$.
5) Assign the data to clusters, assuming that the current model is correct.
6) If the current model and the new model are sufficiently close to each other, terminate, else go to 4.

With this initialization method in hand, our Hybrid SEM (HSEM) algorithm for estimating of generalized Dirichlet mixture can be summarized as follows.

HSEM Algorithm

1) INPUT: d-dimensional data $\vec{X}_i, i = 1, \ldots, N$ and an over-specified number of clusters M.
2) Apply the INITIALIZATION Algorithm.
3) E-Step: Compute the a *posteriori* probabilities: $\hat{Z}_{ij} = p(\vec{X}_i|\vec{\Theta}_j)P(j)/\sum_{l=1}^{M} p(\vec{X}_i|\vec{\Theta}_l)P(l)$
4) S-Step: For each sample value $\vec{X}_i$ draw $\vec{Z}_i$ from the multinomial distribution of order one with M categories having probabilities specified by the $\hat{Z}_{ij}$.
5) M-Step:
   a) update the $\vec{\theta}_{jl}$ using **(36)**, $j = 1, \ldots, M.$ and $l = 1, \ldots, d.$;
   b) update the $P(j)$ using **(23)**, $j = 1, \ldots, M$.
6) If $P(j) < \epsilon$ discard component j, go to 3.
7) If the convergence test is passed, terminate, else go to 3.

Convergence tests could involve testing the stabilization of the $\vec{\Theta}_j$ or the value of the maximum likelihood function.

IV. Experimental Results

This section has two main goals: comparing the performance of the generalized Dirichlet mixture and the Gaussian mixture, and testing the algorithm in applications that involve high-dimensional data. For the estimation of the parameters of the Gaussian mixture, we have added a stochastic step to the well known EM algorithm. We have used the same agglomerative term presented in the previous section to estimate the number of clusters. In addition, we have considered a diagonal covariance matrix to avoid numerical problems.

A. Data Clustering

In the first experiment, our method was used to model the class-conditional densities in multidimensional data sets which differ in dimension, size and complexity. These data sets were obtained from the machine learning repository at the University of California, Irvine [27]. The data sets are Ruspini, Chevron, Diabetes, Iris, Breast Cancer data, Pima Indian Diabetes, Heart Disease, Bupa Liver Disorders, German Credit Card and Australian Credit Card data sets. These data sets will be used to compare the performance of the generalized Dirichlet mixture and the Gaussian mixture. The classification was performed using the Bayes rule ($\vec{X}_i$ is assigned to class j_1 if $Z^*_{ij_1} > Z^*_{ij}, \forall j \neq j_1$) after the class-conditional densities were estimated. The Ruspini data set contains two-dimensional data in four groups [see Fig. 2(a)]. Chevron is another two-dimensional data set. The data in Chevron arise from the processing of a series of images taken by a reconnaissance aircraft in which a large number of points are identified as representing possible mines, but many of these are in fact noise [see Fig. 2(b)]. Diabetes is a three-dimensional data set involving 145 observations used for diabetes diagnosis. The data set is composed of three clusters which are overlapping and are far from spherical in shape [see Fig. 2(c)]. Breast Cancer is a nine-dimensional data set which contains two classes and involves 683 samples. 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 [see Fig. 2(d)]. Using the generalized Dirichlet mixture for the Ruspini data, convergence was reached with an accuracy of 98.67%. We also plotted the results the presence of 4 classes can be

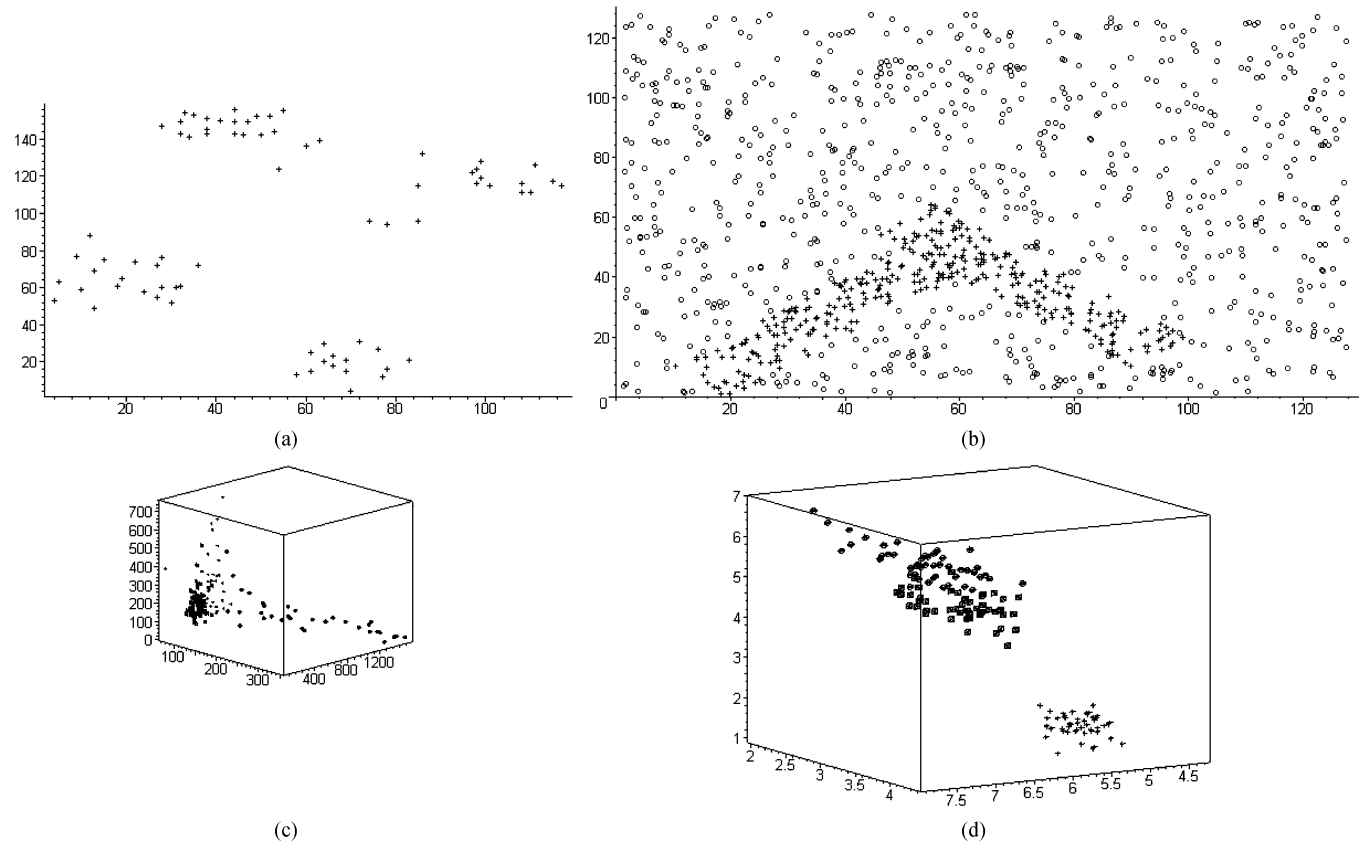


Fig. 2. Examples of the data sets used. (a) Ruspini data set. (b) Chevron data set. (c) Diabetes data set. (d) Iris data set when we take the three first dimensions.

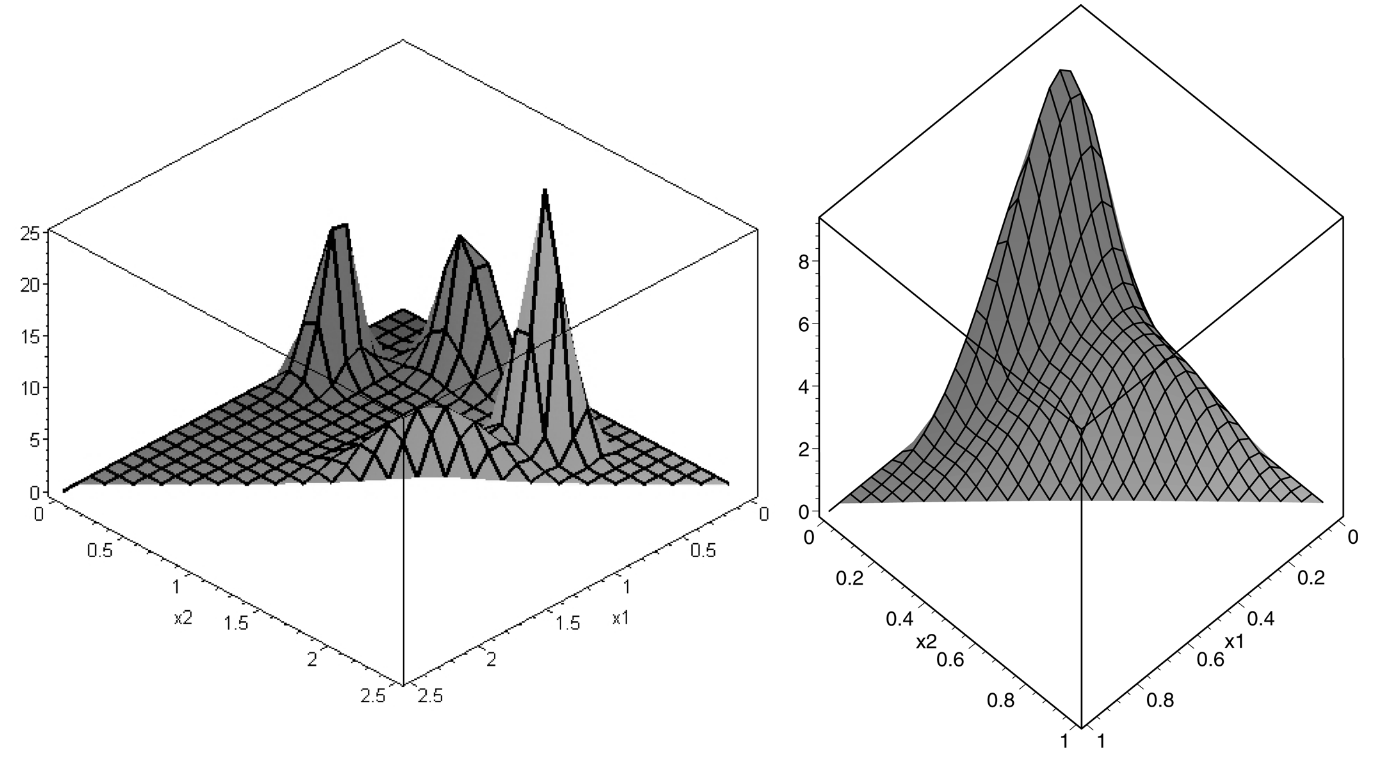
clearly observed [see Fig. 3(a)]. The HSEM algorithm also gave good results for the Chevron data set (an accuracy of 91.95%). In Fig. 3(b), two classes can be observed clearly. For the Diabetes data set, the accuracy was 94.22%. The accuracy was 98.66% for the Iris data set. In Table I, we present all the results and we compare them with the classification given by the Gaussian. From Table I, we can clearly see that the classification accuracies given by the HSEM algorithm are better.

B. Image Restoration

The second application concerns image restoration. The aim of the application is to demonstrate the modeling capabilities of the HSEM algorithm in image analysis. Suppose an image has been degraded and we wish to restore it. For this goal, we compare two approaches based on the HSEM algorithm and Wiener filtering. We assume that we have a large number of image pairs (original, degraded) for training. In our test, we consider images which are degraded by additive white noise, but the procedure can also be applied for other type of distortion which cannot be expressed simply in mathematical form. For this application, we use an approach proposed in [28]. First, vectors are formed for each original pixel in the training set (see Fig. 4), as shown in Fig. 5. Next, we apply our algorithm to these vectors. After the mixture's parameters are estimated, they can be used to restore a previously unseen degraded test image in the following way. For each pixel location in a given degraded image, let $X_1, \ldots, X_{d-1}$ denote the observed values of the neighborhood pixels, and let X_d be the original (unknown) pixel value. The value of X_d is then chosen according to this rule

$$X_d = \arg\max_{X_d} p((X_1, \ldots, X_d)|\Theta). \tag{39}$$

Fig. 6 shows the result of the HSEM-based restoration in an example where two 256×256 images [see Fig. 6(a) and (e)] are degraded by additive white Gaussian noise with a variance of 100 [see Fig. 6(b) and (f)]. The result of Wiener filtering is shown in Fig. 6(c) and (g). For the HSEM algorithm, a 3×3 neighborhood was used ($\mathrm{d} = 9 + 1 = 10$). We see clearly that the restored images Fig. 6(d) and (h) exhibit significant noise reduction and maintain good quality. In order to compare the two approaches (Wiener filtering and our algorithm), we used the SSIM (Structural SIMilarity) index, which is a method for measuring the similarity between two images, presented by Wang, Bovick and collaborators [29]. Note that the SSIM value decreases to zero when the images are less similar. We applied this method to compare the original images to the restored ones. The SSIM was 0.72 and 0.8 when we compared the original image in Fig. 6(a) with the images restored by Wiener filtering [see Fig. 6(c)] and the image restored using our method [see Fig. 6(d)], respectively. We also compared the results given by the two methods for the second images [see Fig. 6(e)]. The Wiener filtering and our method gave SSIM indices of 0.74 and 0.78, respectively.



(a) (b)

Fig. 3. (a) Representation of the Ruspini data set by the generalized Dirichlet mixture. (b) Representation of the Chevron data set by the generalized Dirichlet mixture.

TABLE I
RESULTS GIVEN BY THE HSEM AND THE GAUSSIAN FOR THE PATTERN RECOGNITION APPLICATION

Data set	Dimension	Samples	Classes	Accuracy	
				Generalized Dirichlet	Gaussian
Ruspini	2	75	4	98.67	97.34
chevron	2	350	2	91.95	88.72
Diabetes	3	145	3	94.22	91.88
Iris	4	150	3	98.66	97.33
Breast Cancer	9	683	2	98.97	97.65
Pima Indian Diabetes	8	768	2	94.80	74.01
Heart Disease	13	270	2	96.00	89.70
Bupa liver disorders	6	345	2	94.40	77.90
German Credit Card	24	750	2	87.33	70.01
Australian Credit Card	14	690	2	94.21	85.66

C. Handwritten Digit Recognition

We will start with the handwriting recognition, which has been the subject of intense interest in recent years [30]. Here, we are interested in the recognition of handwritten digits, which has many application scenarios such as automatic mail classification according to the zip code and signature date recognition [31]. Results were gathered for the UCI database obtained from the University of California, Irvine [27]. The UCI database contains 3823 training handwritten digit images and 1797 handwritten digit test images. The original images were processed using a program made available by the National Institute of Standards and Technology [32] to extract normalized bitmaps of handwritten digits. Each normalized bitmap includes a 32×32 matrix in which each element indicates one pixel with a value of white or black. Fig. 7 shows an example of the

Fig. 4. Examples from the training set used in the experiments.

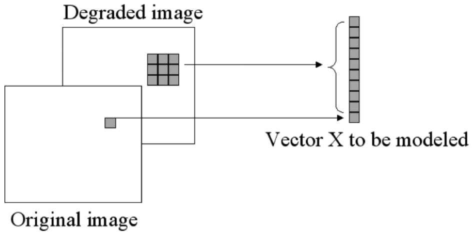


Fig. 5. Formation of data vector X from degraded and original pixels, for use in the training step.

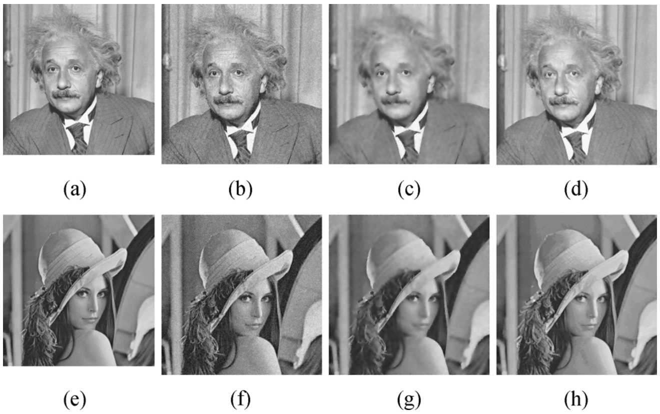


Fig. 6. Image restoration examples. (a), (e) Original images; (b), (f) degraded; (c), (g) Wiener filtered; (d), (h) restored using the generalized Dirichlet mixture.

normalized bitmaps. Each 32×32 bitmap is divided into overlapping blocks of 4×4 and the number of white pixels are counted in each block. This generates an 8×8 input matrix where each element is an integer in the range $0, \ldots, 16$. Interpreting each pixel in the 8×8 matrix as a feature results in a 64-dimensional feature vector. With the feature vectors in hand, each class k, $k = 0, \ldots, 9$ in the training set is represented by a mixture of generalized Dirichlet distributions $p(\vec{X}|\Theta_k)$ with parameter vector Θ_k. We call this the training step. The testing step consists of assigning each image in the testing set to one of the ten classes. For this purpose, we begin by representing each image by a 64-dimensional vector $\vec{X}$ as explained above. We then use the following rule to determine the class of the image

Fig. 7. Example of normalized bitmaps.

$$\vec{X} \longmapsto \arg\max_{k} p(\vec{X}|\Theta_k). \tag{40}$$

The training and the testing schemes are given in Fig. 8. The number of images misclassified was 14 (an error of 0.78%) as compared to 1.11% when we use a Gaussian mixture, 2% when we use the Euclidean distance, and 1.5% for the procedure reported in [33].

D. Texture Image Database Summarization for Efficient Retrieval

The fourth application concerns the summarization of texture image databases. Interactions between users and multimedia databases can involve queries like "Retrieve images that are similar to this image." A number of techniques have been developed to handle pictorial queries, e.g., QBIC [34], Blobworld [35], and Atlas [36]. Summarizing the database is very important because it simplifies the task of retrieval by restricting the search for similar images to a smaller domain of the database. Summarization is also very efficient for browsing. Knowing the categories of images in a given database allows the user to find the images he or she is looking for more quickly. Using mixture decomposition, we can find natural groupings of images and represent each group by the most representative image in the group. In other words, after appropriate features are extracted from the images, the feature space can be partitioned into regions that are relatively homogeneous with respect to the chosen set of features. By identifying the homogeneous regions in the feature space, the task of summarization is accomplished. For the experiment described in this paper, we used the *Vistex* color texture database obtained from the MIT Media Lab. In our experimental framework, each of the 512×512 images from the *Vistex* database was divided into 64×64 images. Since each 512×512 "mother image" contributes 64 images to our database, ideally all of the 64 images should be classified in the same class. In the experiment, six homogeneous texture groups, "Bark," "Fabric," "Food," "Metal," "Water," and "Sand" were used to create a new database. A database with 1920 images was obtained. Four images from each of the Bark, Fabric, and Metal texture groups were used to obtain 256 images for each of these categories, and six images from Water, Food, and Sand were used to obtain 384 images for these categories. Examples of images from each of the categories are shown in Fig. 9. In order to determine the vector of characteristics for each image,

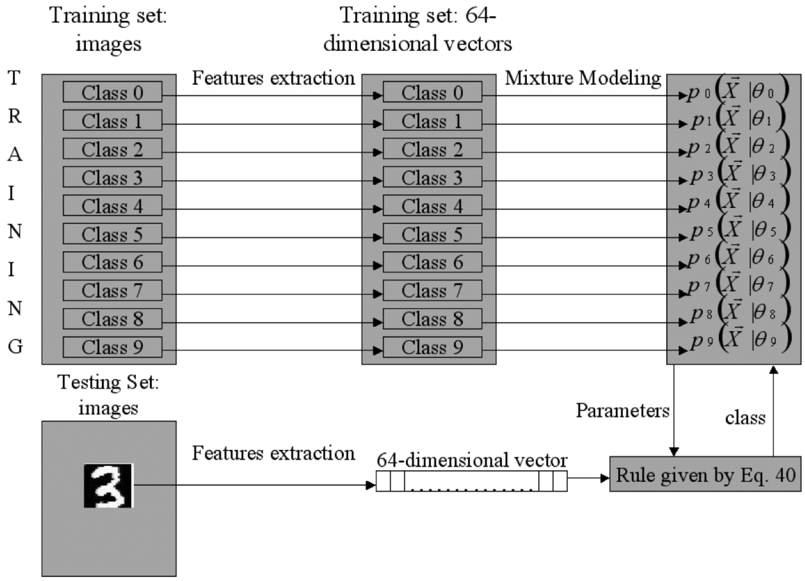


Fig. 8. Steps in handwritten digit recognition.

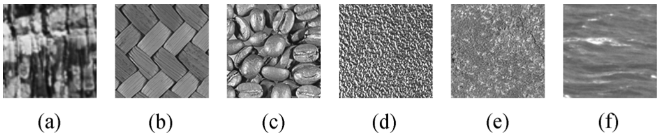


Fig. 9. Sample images from each group. (a) Bark, (b) Fabric, (c) Food, (d) Metal, (e) Sand, and (f) Water.

TABLE II
CONFUSION MATRIX FOR IMAGE CLASSIFICATION BY A GENERALIZED DIRICHLET MIXTURE

	Bark	Fabric	Food	Metal	Sand	Water
Bark	254	0	0	0	2	0
Fabric	0	251	5	0	0	0
Food	0	8	376	0	0	0
Metal	0	0	0	250	0	6
Sand	3	0	0	0	381	0
Water	3	0	0	6	2	373

TABLE III
CONFUSION MATRIX FOR IMAGE CLASSIFICATION BY A GAUSSIAN MIXTURE

	Bark	Fabric	Food	Metal	Sand	Water
Bark	240	0	0	3	8	5
Fabric	0	236	12	0	4	4
Food	0	12	365	4	0	3
Metal	0	2	2	242	4	6
Sand	8	2	0	0	370	4
Water	5	1	0	10	5	363

we have computed a set a features derived from the correlogram [37]. It has been noted that to obtain good results, many correlograms should be computed, each one considering a given neighborhood and direction. Some studies show that considering the following four neighborhoods is sufficient for co-ocurrence matrices, in the case of gray level images, to obtain good results in general: (1; 0), $(1;\pi/4)$, $(1;\pi/2)$, and $(1;3\pi/4)$ [38]. For each of these neighborhoods, we calculated the corresponding correlogram, then derived from it the following features which have been proposed for co-occurrence matrices: Mean, Variance, Energy, Correlation, Entropy, Contrast, Homogeneity, and Cluster Prominence [39]. Then, each image was characterized by a 36D vector. Applying our algorithm to the texture database, we found six categories. The classification was performed using the Bayesian decision rule after the class-conditional densities were estimated. The confusion matrix for the texture image classification application is given in Table II. 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 small: 35 in all, which represents an accuracy of 98.18%. Table III shows the confusion matrix for the Gaussian mixture (an accuracy of 94.59). After the database was summarized, we conducted another experiment designed to retrieve images similar to a query. First, we defined a measure to determine the closest component to the query vector. Next, another distance measure was used to determine the similarity between the query vector and the feature vectors in the closest component. The *a posteriori* probabilities were used to choose the component nearest to the query. After selecting the closest component, the 2-norm was applied to find the images most similar to the query. To measure the retrieval rates, each image was used as a query and the number of relevant images among those that were retrieved was noted. Precision and recall,

TABLE IV
RECALL AND PRECISION OBTAINED FOR THE TEXTURE DATABASE BY USING THE HSEM ALGORITHM

	No. of retrieved images				
Measure	16	48	64	80	96
Recall	0.24	0.74	0.93	0.95	0.98
Precsion	0.97	0.98	0.93	0.84	0.73

which are the measures most commonly used by the information retrieval community, were then computed using (41) and (42). These measures were then averaged over all the queries

$$\text{precision} = \frac{\text{number of relevant retrieved images}}{\text{total number of retrieved images}} \quad (41)$$

$$\text{recall} = \frac{\text{number of relevant retrieved images}}{\text{total number of relevant images}}. \quad (42)$$

As each 512 × 512 image from *Vistex* contributes 64 images to our database, given a query image, ideally all 64 images should be retrieved and are considered to be relevant. Table IV presents the retrieval rates obtained in terms of precision and recall. The results are shown when 16, 48, 64, 80, and 96 images were retrieved from the database in response to a query.

V. CONCLUSION

In this paper, we have focused on high-dimensional data clustering. The algorithm proposed is motivated by the great number of pattern recognition and image processing applications which involve such types of data. In contrast with other methods which use dimensionality reduction, our algorithm uses the full dimensionality of the data. In fact, it is based on the statistical properties of the data, through the use of generalized Dirichlet finite mixture models. The data is transformed in such a way that density estimation in the transformed space is simpler and more accurate. This transformation is possible thanks to the convenient mathematical proprieties of the generalized Dirichlet density. The generalized Dirichlet distribution has the advantage that by varying its parameters, it permits multiple modes and asymmetry and can, thus, approximate a wide variety of shapes. Besides, it has a more general covariance structure than the Dirichlet. We estimated the parameters of this mixture using the maximum likelihood and Fisher scoring methods, and by introducing a stochastic step. From the experimental results which involve data set clustering, image restoration, handwritten digit recognition and image database summarization for efficient retrieval, we can say that the generalized Dirichlet distribution offers strong modeling capabilities for both low and high-dimensional data. Future works can be devoted to a Bayesian approach to estimate the parameters of a generalized Dirichlet mixture and the introduction of Dirichlet Process priors with the Chinese restaurant process to estimate the number of components.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees and the associate editor for their helpful comments.

REFERENCES

[1] N. Bouguila and D. Ziou, "Dirichlet-based probability model applied to human skin detection," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Montreal, Canada, May 2004, pp. 521–524.

[2] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.

[3] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[4] S. J. Raudys and A. K. Jain, "Small sample size effects in statistical pattern recognition: Recommandations for practitioners," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 3, pp. 252–264, Mar. 1991.

[5] D. W. Scott and J. R. Thompson, "Probability density estimation in higher dimensions," *Comput. Sci. Statist.*, pp. 173–179, 1983.

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

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B*, vol. 39, pp. 1–38, 1977.

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

[9] A. E. Raftery and J. D. Banfield, "Model-based Gaussian and non-Gaussian clustering," *Biometrics*, vol. 49, pp. 803–821, 1993.

[10] N. Bouguila, D. Ziou, and J. Vaillancourt, "Novel mixtures based on the Dirichlet distribution: Application to data and image classification," in *Proc. Machine Learning and Data Mining in Pattern Recognition*, Leipzig, Germany, Jul. 2003, vol. 2734, LNCS, pp. 172–181.

[11] ——, "Unsupervised learning of a finite mixture model based on the Dirichlet distribution and its application," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1533–1543, Nov. 2004.

[12] N. Bouguila and D. Ziou, "Unsupervised learning of a finite discrete mixture model based on the multinomial Dirichlet distribution: Application to texture modeling," in *Proc. Pattern Recognition in Information Systems*, Porto, Portugal, Apr. 2004, pp. 118–127.

[13] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and EM algorithm," *SIAM Rev.*, vol. 26, no. 2, pp. 195–239, Apr. 1984.

[14] K. Samuel, K. W. Ng, and K. Fang, *Symmetric Multivariate and Related Distributions*. London, U.K.: Chapman & Hall, 1990.

[15] K. Samuel, N. Balakrishman, and J. Norman, *Continous Multivariate Distributions*. New York: Wiley, 2000, vol. 1.

[16] R. J. Connor and J. E. Mosimann, "Concepts of independence for proportions with a generalization of the Dirichlet distribution," *Amer. Statist. Assoc. J.*, vol. 64, pp. 194–206, 1969.

[17] T. Wong, "Generalized Dirichlet distribution in Bayesian analysis," *Appl. Math. Comput.*, vol. 97, pp. 165–181, 1998.

[18] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. AC-19, no. 6, pp. 716–723, Jun. 1974.

[19] J. Rissanen, "Modeling by shortest data description," *Biometrika*, vol. 14, pp. 465–471, 1978.

[20] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, 1978.

[21] H. Frigui and R. Krishnapuram, "A robust competitive clustering algorithm with applications in computer vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 450–465, May 1999.

[22] S. Medasani and R. Krishnapuram, "A comparison of Gaussian and Pearson mixture modeling for pattern recognition and computer vision applications," *Pattern Recognit. Lett.*, vol. 20, pp. 305–313, 1999.

[23] N. Bouguila and D. Ziou, "A powerful finite mixture model based on the generalized Dirichlet distribution: Unsupervised learning and applications," in *Proc. 17th Int. Conf. Pattern Recognition*, Cambridge, U.K., 2004, pp. 280–283.

[24] C. R. Rao, *Advanced Statistical Methods in Biomedical Research*. New York: Wiley, 1952.

[25] G. Celeux and J. Diebolt, "The SEM algorithm: A probabilistic teacher algorithm derived from the EM algorithm for the mixture problem," *Comput. Statist. Quart.*, vol. 2, no. 1, pp. 73–82, 1985.

[26] ——, "A stochastic approximation type EM algorithm for the mixture problem," *Stochast. Stochast. Rep.*, vol. 41, pp. 119–134, 1992.

[27] C. L. Blake and C. J. Merz, Repository of Machine Learning Databases Dept. Inf. Comput. Sci., Univ. California, Irvine, 1998 [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[28] K. Popat and R. W. Picard, "Cluster-based probability model and its application to image and texture processing," *IEEE Trans. Image Process.*, vol. 6, no. 2, pp. 268–284, Feb. 1997.

[29] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[30] R. Plamondon and S. N. Srihari, "On-line and off-line handwriting recognition: A comprehensive survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 63–84, Jan. 2000.

[31] D. Keysers, W. Macherey, H. Ney, and J. Dahmen, "Adaptation in statistical pattern recognition using tangent vectors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 269–274, Feb. 2004.

[32] M. Garris, J. Blue, G. Candela, P. Grother, S. Janet, and C. Wilson, "NIST Form-Based Handprint Recognition System," Tech. Rep., Nat. Inst. Standards Technol.

[33] H. Kim, D. Kim, and S. Y. Bang, "A numeral character recognition using the PCA mixture model," *Pattern Recognit. Lett.*, vol. 33, pp. 103–111, 2002.

[34] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. H. Glasman, D. Yanker, P. Faloutsos, and G. Taubin, "The QBIC project: Querying images by content using color, texture and shape," in *Proc. SPIE Conf. Storage and Retrieval for Images and Video Databases*, 1993, pp. 173–187.

[35] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik, "Blobworld: A system for region-based image indexing and retrieval," presented at the 3rd Int. Conf. Visual Information Systems 1999.

[36] M. L. Kherfi, D. Ziou, and A. Bernardi, "Combining positive and negative examples in relevance feedback for content-based image retrieval," *J. Vis. Commun. Image Represent.*, vol. 14, pp. 428–457, 2003.

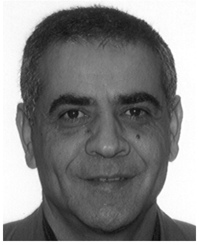
[37] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Juan, PR, 1997, p. 762.

[38] T. Randen and J. H. Husoy, "Filtering for texture classification: A comparative study," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 4, pp. 291–310, Apr. 1999.

[39] M. Unser, "Sum and difference histograms for texture classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 1, pp. 118–125, Jan. 1986.

Nizar Bouguila (M'06) received the Engineer degree from the University of Tunis in 2000 and the M.Sc. and Ph.D. degrees from Sherbrooke University, Sherbrooke, QC, Canada, in 2002 and 2006, respectively, all in computer science.

He is currently an Assistant Professor with the Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montréal, QC. His research interests include image processing, machine learning, 3-D graphics, computer vision, and pattern recognition.



Djemel Ziou received the B.Eng. degree in computer science from the University of Annaba, Algeria, in 1984 and the Ph.D. degree in computer science from the Institut National Polytechnique de Lorraine (INPL), Lorraine, France, in 1991.

From 1987 to 1993, he served as a Lecturer at several universities in France. During the same time period, he was a Researcher in the Centre de Recherche en Informatique de Nancy (CRIN), Nancy, France, and the Institut National de Recherche en Informatique et Automatique (INRIA), France. Presently, he is a Full Professor at the Department of Computer Science, Université de Sherbrooke, Sherbrooke, QC, Canada. 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.