Unsupervised Learning of Finite Mixture Models

Mario A.T. Figueiredo, *Senior Member*, *IEEE*, and Anil K. Jain, *Fellow*, *IEEE*

Abstract—This paper proposes an unsupervised algorithm for learning a finite mixture model from multivariate data. The adjective "unsupervised" is justified by two properties of the algorithm: 1) it is capable of selecting the number of components and 2) unlike the standard *expectation-maximization* (EM) algorithm, it does not require careful initialization. The proposed method also avoids another drawback of EM for mixture fitting: the possibility of convergence toward a singular estimate at the boundary of the parameter space. The novelty of our approach is that we do not use a model selection criterion to choose one among a set of preestimated candidate models; instead, we seamlessly integrate estimation and model selection in a single algorithm. Our technique can be applied to any type of parametric mixture model for which it is possible to write an EM algorithm; in this paper, we illustrate it with experiments involving Gaussian mixtures. These experiments testify for the good performance of our approach.

Index Terms—Finite mixtures, unsupervised learning, model selection, minimum message length criterion, Bayesian methods, expectation-maximization algorithm, clustering.

1 Introduction

Finite mixtures are a flexible and powerful probabilistic modeling tool for univariate and multivariate data. The usefulness of mixture models in any area which involves the statistical modeling of data (such as pattern recognition, computer vision, signal and image analysis, machine learning) is currently widely acknowledged.

In statistical pattern recognition, finite mixtures allow a formal (probabilistic model-based) approach to unsupervised learning (i.e., clustering) [28], [29], [35], [37], [57]. In fact, finite mixtures naturally model observations which are assumed to have been produced by one (randomly selected and unknown) of a set of alternative random sources. Inferring (the parameters of) these sources and identifying which source produced each observation leads to a clustering of the set of observations. With this model-based approach to clustering (as opposed to heuristic methods like k-means or hierarchical agglomerative methods [28]), issues like the selection of the number of clusters or the assessment of the validity of a given model can be addressed in a principled and formal way.

The usefulness of mixture models is not limited to unsupervised learning applications. Mixture models are able to represent arbitrarily complex probability density functions (pdf's). This fact makes them an excellent choice for representing complex class-conditional pdf's (i.e., likelihood functions) in (Bayesian) supervised learning scenarios [25], [26], [55], or priors for Bayesian parameter estimation [16]. Mixture models can also be used to perform feature selection [43].

- *M.A.T. Figueiredo is with the Institute of Telecommunications and the Department of Electrical and Computer Engineering, Instituto Superior Técnico, 1049-001 Lisboa, Portugal. E-mail: mtf@lx.it.pt.*
- *A.K. Jain is with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824. E-mail: jain@cse.msu.edu.*

Manuscript received 5 July 2000; revised 8 Feb. 2001; accepted 30 July 2001. Recommended for acceptance by W.T. Freeman.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 112382.

The standard method used to fit finite mixture models to observed data is the *expectation-maximization* (EM) algorithm [18], [36], [37], which converges to a *maximum likelihood* (ML) estimate of the mixture parameters. However, the EM algorithm for finite mixture fitting has several drawbacks: it is a local (greedy) method, thus sensitive to initialization because the likelihood function of a mixture model is not unimodal; for certain types of mixtures, it may converge to the boundary of the parameter space (where the likelihood is unbounded) leading to meaningless estimates.

An important issue in mixture modeling is the selection of the number of components. The usual trade off in model order selection problems arises: With too many components, the mixture may over-fit the data, while a mixture with too few components may not be flexible enough to approximate the true underlying model.

In this paper, we deal simultaneously with the above mentioned problems. We propose an inference criterion for mixture models and an algorithm to implement it which: 1) automatically selects the number of components, 2) is less sensitive to initialization than EM, and 3) avoids the boundary of the parameters space.

Although most of the literature on finite mixtures focuses on mixtures of Gaussian densities, many other types of probability density functions have also been considered. The approach proposed in this paper can be applied to any type of parametric mixture model for which it is possible to write an EM algorithm.

The rest of paper is organized as follows: In Section 2, we review finite mixture models and the EM algorithm; this is standard material and our purpose is to introduce the problem and define notation. In Section 3, we review previous work on the problem of learning mixtures with an unknown number of components and dealing with the drawbacks of the EM algorithm. In Section 4, we describe the proposed inference criterion, while the algorithm which implements it is presented in Section 5. Section 6 reports experimental results and Section 7 ends the paper by presenting some concluding remarks.

2 Learning Finite Mixture Models

2.1 Finite Mixture Models

Let $\mathbf{Y} = [Y_1, \ldots, Y_d]^T$ be a d-dimensional random variable, with $\mathbf{y} = [y_1, \ldots, y_d]^T$ representing one particular outcome of $\mathbf{Y}$. It is said that $\mathbf{Y}$ follows a k-component finite mixture distribution if its probability density function can be written as

$$p(\mathbf{y}|\boldsymbol{\theta}) = \sum_{m=1}^{k} \alpha_m \, p(\mathbf{y}|\boldsymbol{\theta}_m), \tag{1}$$

where $\alpha_1, \ldots, \alpha_k$ are the *mixing probabilities*, each $\boldsymbol{\theta}_m$ is the set of parameters defining the mth component, and $\boldsymbol{\theta} \equiv \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k, \alpha_1, \ldots, \alpha_k\}$ is the complete set of parameters needed to specify the mixture. Of course, being probabilities, the α_m must satisfy

$$\alpha_m \geq 0, \quad m = 1, \ldots, k, \quad \text{and} \quad \sum_{m=1}^{k} \alpha_m = 1. \tag{2}$$

In this paper, we assume that all the components have the same functional form (for example, they are all d-variate Gaussian), each one being thus fully characterized by the parameter vector $\boldsymbol{\theta}_m$. For detailed and comprehensive accounts on mixture models, see [35], [37], [57]; here, we simply review the fundamental ideas and define our notation.

Given a set of n independent and identically distributed samples $\mathcal{Y} = \{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n)}\}$, the log-likelihood corresponding to a k-component mixture is

$$\log p(\mathcal{Y}|\boldsymbol{\theta}) \; = \; \log \prod_{i=1}^{n} p(\mathbf{y}^{(i)}|\boldsymbol{\theta}) = \sum_{i=1}^{n} \log \sum_{m=1}^{k} \alpha_m p(\mathbf{y}^{(i)}|\boldsymbol{\theta}_m). \tag{3}$$

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

$$\widehat{\boldsymbol{\theta}}_{\text{ML}} = \arg\max_{\boldsymbol{\theta}} \{\log p(\mathcal{Y}|\boldsymbol{\theta})\}$$

cannot be found analytically. The same is true for the Bayesian *maximum* a posteriori (MAP) criterion,

$$\widehat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} \{\log p(\mathcal{Y}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})\},$$

given some prior $p(\theta)$ on the parameters. Of course, the maximizations defining the ML or MAP estimates are under the constraints in (2).

2.2 The EM Algorithm

The usual choice for obtaining ML or MAP estimates of the mixture parameters is the EM algorithm [18], [35], [36], [37]. EM is an iterative procedure which finds local maxima of $\log p(\mathcal{Y}|\boldsymbol{\theta})$ or $[\log p(\mathcal{Y}|\theta) + \log p(\theta)]$. For the case of Gaussian mixtures, the convergence behavior of EM is well studied [37], [63]. It was recently shown that EM belongs to a class of iterative methods called *proximal point algorithms* (PPA; for an introduction to PPA and a comprehensive set of references see [4], chapter 5) [13]. Seeing EM under this new light opens the door to several extensions and generalizations. An earlier related result, although without identifying EM as a PPA, appeared in [41].

The EM algorithm is based on the interpretation of $\mathcal{Y}$ as *incomplete* data. For finite mixtures, the *missing* part is a set of n labels $\mathcal{Z} = \{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(n)}\}$ associated with the n samples, indicating which component produced each sample. Each label is a binary vector $\mathbf{z}^{(i)} = [z_1^{(i)}, \ldots, z_k^{(i)}]$, where $z_m^{(i)} = 1$ and $z_p^{(i)} = 0$, for $p \neq m$, means that sample $\mathbf{y}^{(i)}$ was produced by the mth component. The complete log-likelihood (i.e., the one from which we could estimate θ if the *complete* data $\mathcal{X} = \{\mathcal{Y}, \mathcal{Z}\}$ was observed [36]) is

$$\log p(\mathcal{Y}, \mathcal{Z}|\theta) = \sum_{i=1}^{n} \sum_{m=1}^{k} z_m^{(i)} \log \Big[\alpha_m p(\mathbf{y}^{(i)}|\boldsymbol{\theta}_m)\Big]. \tag{4}$$

The EM algorithm produces a sequence of estimates $\{\widehat{\boldsymbol{\theta}}(t), \; t = 0, 1, 2, \ldots\}$ by alternatingly applying two steps (until some convergence criterion is met):

- **E-step**: Computes the conditional expectation of the complete log-likelihood, given $\mathcal{Y}$ and the current estimate $\widehat{\boldsymbol{\theta}}(t)$. Since $\log p(\mathcal{Y}, \mathcal{Z}|\boldsymbol{\theta})$ is linear with respect to the missing $\mathcal{Z}$, we simply have to compute the conditional expectation $\mathcal{W} \equiv E[\mathcal{Z}|\mathcal{Y}, \widehat{\boldsymbol{\theta}}(t)]$, and plug it into $\log p(\mathcal{Y}, \mathcal{Z}|\boldsymbol{\theta})$. The result is the so-called Q-function:

$$\begin{aligned} Q(\theta, \widehat{\boldsymbol{\theta}}(t)) &\equiv E\Big[\log p(\mathcal{Y}, \mathcal{Z}|\boldsymbol{\theta}) \;\Big|\; \mathcal{Y}, \widehat{\boldsymbol{\theta}}(t)\Big] \\ &= \log p(\mathcal{Y}, \mathcal{W}|\boldsymbol{\theta}). \end{aligned} \tag{5}$$

  Since the elements of $\mathcal{Z}$ are binary, their conditional expectations are given by

$$\begin{aligned} w_m^{(i)} &\equiv E\Big[z_m^{(i)}|\; \mathcal{Y}, \widehat{\boldsymbol{\theta}}(t)\Big] = \Pr\Big[z_m^{(i)} = 1|\; y^{(i)}, \widehat{\boldsymbol{\theta}}(t)\Big] \\ &= \frac{\widehat{\alpha}_m(t) \; p(\mathbf{y}^{(i)}|\widehat{\boldsymbol{\theta}}_m(t))}{\sum_{j=1}^{k} \widehat{\alpha}_j(t) \; p(\mathbf{y}^{(i)}|\widehat{\boldsymbol{\theta}}_j(t))}, \end{aligned} \tag{6}$$

  where the last equality is simply Bayes law (α_m is the a priori probability that $z_m^{(i)} = 1$, while $w_m^{(i)}$ is the a posteriori probability that $z_m^{(i)} = 1$, after observing $\mathbf{y}^{(i)}$).
- **M-step**: Updates the parameter estimates according to

$$\widehat{\boldsymbol{\theta}}(t+1) = \arg\max_{\boldsymbol{\theta}} \{Q(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}(t)) + \log p(\boldsymbol{\theta})\},$$

  in the case of MAP estimation, or

$$\widehat{\boldsymbol{\theta}}(t+1) = \arg\max_{\boldsymbol{\theta}} \; Q(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}(t)),$$

  for the ML criterion, in both cases, under the constraints in (2).

3 Previous Work

3.1 Estimating the Number of Components

Let us start by defining $\mathcal{M}_k$ as the class of all possible k-component mixtures built from a certain type of pdf's (e.g., all d-variate Gaussian mixtures with unconstrained covariance matrices). The ML criterion cannot be used to estimate k, the number of mixture components, because $\mathcal{M}_k \subseteq \mathcal{M}_{k+1}$, that is, these classes are nested. As an illustration, let $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k, \alpha_1, \ldots, \alpha_{k-1}, \alpha_k\}$, define a mixture in $\mathcal{M}_k$, and $\boldsymbol{\theta}' = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k, \boldsymbol{\theta}_{k+1}, \alpha_1, \ldots, \alpha_{k-1}, \alpha'_k, \alpha'_{k+1}\}$, define a

mixture in $\mathcal{M}_{k+1}$. If $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k$ and $\alpha_k = \alpha'_k + \alpha'_{k+1}$, then $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ represent the same probability density function. Consequently, the maximized likelihood $p(\mathcal{Y}|\widehat{\boldsymbol{\theta}}_{\mathrm{ML}})$ is a nondecreasing function of k, thus useless as a criterion to estimate the number of components.

Several model selection methods have been proposed to estimate the number of components of a mixture. The vast majority of these methods can be classified, from a computational point of view, into two classes: deterministic and stochastic.

3.1.1 Deterministic Methods

The methods in this class start by obtaining a set of candidate models (usually by EM) for a range of values of k (from $k_{\min}$ to $k_{\max}$) which is assumed to contain the true/optimal k. The number of components is then selected according to

$$\widehat{k} = \arg\min_k \left\{ \mathcal{C}\left(\widehat{\boldsymbol{\theta}}(k), k\right), k = k_{\min}, \ldots, k_{\max} \right\}, \tag{7}$$

where $\mathcal{C}\left(\widehat{\boldsymbol{\theta}}(k), k\right)$ is some model selection criterion, and $\widehat{\boldsymbol{\theta}}(k)$ is an estimate of the mixture parameters assuming that it has k components. Usually, these criteria have the form $\mathcal{C}\left(\widehat{\boldsymbol{\theta}}(k), k\right) = -\log p\left(\mathcal{Y}|\widehat{\boldsymbol{\theta}}(k)\right) + \mathcal{P}(k)$, where $\mathcal{P}(k)$ in an increasing function penalizing higher values of k. Examples of such criteria that have been used for mixtures include:

- Approximate Bayesian criteria, like the one in [50] (termed *Laplace-empirical criterion*, LEC, in [37]), and Schwarz's *Bayesian inference criterion* (BIC) [10], [17], [22], [53].
- Approaches based on information/coding theory concepts, such as Rissanen's *minimum description length* (MDL) [49], which formally coincides with BIC, the *minimum message length* (MML) criterion [42], [60], [61], Akaike's *information criterion*(AIC) [62], and the *informational complexity criterion* (ICOMP) [8].
- Methods based on the complete likelihood (4), which is also called *classification likelihood*), such as the *approximate weight of evidence* (AWE) [1], the *classification likelihood criterion* (CLC) [7], the *normalized entropy criterion* (NEC) [6], [12], and the *integrated classification likelihood* (ICL) criterion [5].

A more detailed review of these methods is found in [37] (chapter 6) which also includes a comparative study where ICL and LEC are found to outperform the other criteria.

3.1.2 Stochastic and Resampling Methods

Markov chain Monte Carlo (MCMC) methods can be used in two different ways for mixture inference: to implement model selection criteria (e.g., [2], [39], [51]); or, in fully Bayesian way, to sample from the full a posteriori distribution with k considered unknown [40], [45], [48]. Despite their formal appeal, we think that MCMC-based techniques are still far too computationally demanding to be useful in pattern recognition applications.

Resampling-based schemes [33] and cross-validation approaches [54] have also been used to estimate the number of mixture components. In terms of computational load, these methods are closer to stochastic techniques than to deterministic ones.

3.2 The Drawbacks of EM-Based Methods

Basically, all deterministic algorithms for fitting mixtures with unknown numbers of components use the EM-algorithm. Although some of these methods perform well, a major draw-back remains: a whole set of candidate models has to be obtained, and the following well-known problems associated with EM emerge.

3.2.1 The Initialization Issue

EM is highly dependent on initialization. Common (time-consuming) solutions include one (or a combination of several) of the following strategies: using multiple random starts and choosing the final estimate with the highest likelihood [25], [36], [37], [50], and initialization by clustering algorithms [25], [36], [37]. Recently, a modified EM algorithm using split and merge operations to escape from local maxima of the log-likelihood has been proposed [59].

Deterministic annealing (DA) has been used with success to avoid the initialization dependence of k-means type algorithms for hard-clustering [27], [38], [52]. The resulting algorithm is similar to EM for Gaussian mixtures under the constraint of covariance matrices of the form $T\mathbf{I}$, where T is called the *temperature* and $\mathbf{I}$ is the identity matrix. DA clustering algorithms begin at high temperature (corresponding to $w_m^{(i)} \simeq 1/k$, a high entropy, uninformative initialization); T is then lowered according to some *cooling schedule* until $T \simeq 0$. The heuristic behind DA is that forcing the entropy of the assignments to decrease slowly avoids premature (hard) decisions that may correspond to poor local minima. The constraint on the covariance matrix makes DA clustering unapplicable to mixture model fitting, when seen as a density estimation problem. It is also not clear how it could be applied to non-Gaussian mixtures. However, it turns out that it is possible to obtain deterministic annealing versions of EM for mixtures, without constraining the covariance matrices, by modifying the E-step [31], [58].

Recently, we have shown (see [20]) that the EM algorithm exhibits a self-annealing behavior [44], that is, it works like a DA algorithm without a prespecified cooling schedule. Basically, all that is necessary is a uninformative (high entropy) initialization of the type $w_m^{(i)} \simeq 1/k$ (called *random starting*, in [37]), and EM will automatically *anneal* without the need for externally imposing a cooling schedule. This fact explains the good performance of the *random starting* method, recently reported in [37].

3.2.2 The Boundary of the Parameter Space

EM may converge to the boundary of the parameter space. For example, when fitting a Gaussian mixture with unconstrained covariance matrices, one of the α_m's may approach zero and the corresponding covariance matrix may become arbitrarily close to singular. When the number of components assumed is larger than the optimal/true one, this tends to happen frequently, thus being a serious problem for methods that require mixture estimates for various values of k. This problem can be avoided through the use of soft constraints on the covariance matrices, as suggested in [31].

4 The Proposed Criterion

The well-known deterministic methods (see (7)) are model-class selection criteria: They select a model-class ($\mathcal{M}_k$) based on its "best" representative ($\widehat{\boldsymbol{\theta}}(k)$). However, in mixture models, the distinction between model-class selection and model estimation is unclear, e.g., a 3-component mixture in which one of the mixing probabilities is zero is undistinguishable from a 2-component mixture. These observations suggest a shift of approach: Let k be some arbitrary large value and infer the structure of the mixture by letting the estimates of some of the mixing probabilities be zero. This approach coincides with the MML philosophy [61], [60], which does not adopt the "model-class/model" hierarchy, but directly aims at finding the "best" overall model in the entire set of available models,

$$\bigcup_{k=k_{\min}}^{k_{\max}} \mathcal{M}_k,$$

rather than selecting one among a set of candidate models $\{\widehat{\boldsymbol{\theta}}(k),\ k = k_{\min}, \ldots, k = k_{\min}\}$. Previous uses of MML for mixtures do not strictly adhere to this perspective and end up using MML as a model-class selection criterion [42].

Rather than using EM to compute a set of candidate models (with the drawbacks mentioned above), we will be able to directly implement the MML criterion using a variant of EM. The proposed algorithm turns out to be much less initialization dependent than standard EM and automatically avoids the boundary of the parameter space.

4.1 The Minimum Message Length Criterion

The rationale behind minimum encoding length criteria (like MDL and MML) is: if you can build a short code for your data, that means that you have a good data generation model [49], [60], [61]. To formalize this idea, consider some data-set $\mathcal{Y}$, known to have been generated according to $p(\mathcal{Y}|\boldsymbol{\theta})$, which is to be encoded and transmitted. Following Shannon theory [15], the shortest code length (measured in bits, if base-2 logarithm is used, or in *nats*, if natural logarithm is adopted [15]) for $\mathcal{Y}$ is $\lceil -\log p(\mathcal{Y}|\boldsymbol{\theta}) \rceil$, where $\lceil a \rceil$ denotes "the smallest integer no less than a." Since even for moderately large data-sets $-\log p(\mathcal{Y}|\boldsymbol{\theta}) \gg 1$, the $\lceil \cdot \rceil$ operator is usually dropped. If $p(\mathcal{Y}|\boldsymbol{\theta})$ is fully known to both the transmitter and the receiver, they can both build the same code and communication can proceed. However, if $\boldsymbol{\theta}$ is a priori unknown, the transmitter has to start by estimating and transmitting $\boldsymbol{\theta}$. This leads to a two-part message, whose total length is given by

$$\text{Length}(\boldsymbol{\theta}, \mathcal{Y}) = \text{Length}(\boldsymbol{\theta}) + \text{Length}(\mathcal{Y}|\boldsymbol{\theta}). \tag{8}$$

All minimum encoding length criteria (like MDL and MML) state that the parameter estimate is the one minimizing $\text{Length}(\boldsymbol{\theta}, \mathcal{Y})$.

A key issue of this approach, which the several flavors of the minimum encoding length principle (e.g., MDL and MML) address differently, is that since $\boldsymbol{\theta}$ is a vector of real parameters, a finite code-length can only be obtained by quantizing $\boldsymbol{\theta}$ to finite precision. The central idea involves the following trade off. Let $\widetilde{\boldsymbol{\theta}}$ be a quantized version of $\boldsymbol{\theta}$. If a fine precision is used, $\text{Length}(\widetilde{\boldsymbol{\theta}})$ is large, but $\text{Length}(\mathcal{Y}|\widetilde{\boldsymbol{\theta}})$ can be made small because $\widetilde{\boldsymbol{\theta}}$ can come close to the optimal value. Conversely, with a coarse precision, $\text{Length}(\widetilde{\boldsymbol{\theta}})$ is small, but $\text{Length}(\mathcal{Y}|\widetilde{\boldsymbol{\theta}})$ can be very far from optimal. There are several ways to formalize and solve this trade off; see [32] for a comprehensive review and pointers to the literature.

The fact that the data itself may also be real-valued does not cause any difficulty; simply truncate $\mathcal{Y}$ to some arbitrary fine precision δ and replace the density $p(\mathcal{Y}|\boldsymbol{\theta})$ by the probability $p(\mathcal{Y}|\boldsymbol{\theta})\delta^d$ (d is the dimensionality of $\mathcal{Y}$). The resulting code-length is $-\log p(\mathcal{Y}|\boldsymbol{\theta}) - d\log\delta$, but $-d\log\delta$ is an irrelevant additive constant.

The particular form of the MML approach herein adopted is derived in Appendix A and leads to the following criterion (where the minimization with respect to $\boldsymbol{\theta}$ is to be understood as simultaneously in $\boldsymbol{\theta}$ and c, the dimension of $\boldsymbol{\theta}$):

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \Big\{ -\log p(\boldsymbol{\theta}) - \log p(\mathcal{Y}|\boldsymbol{\theta}) + \frac{1}{2}\log|\mathbf{I}(\boldsymbol{\theta})| + \frac{c}{2}\left(1 + \log\frac{1}{12}\right)\Big\}, \tag{9}$$

where[1] $\mathbf{I}(\boldsymbol{\theta}) \equiv -E[D^2_{\boldsymbol{\theta}} \log p(\mathcal{Y}|\boldsymbol{\theta})]$ is the (expected) Fisher information matrix, and $|\mathbf{I}(\boldsymbol{\theta})|$ denotes its determinant.

The MDL criterion (which formally, though not conceptually, coincides with BIC) can be obtained as an approximation to (9). Start by assuming a flat prior $p(\boldsymbol{\theta})$ and drop it. Then, since $\mathbf{I}(\boldsymbol{\theta}) = n\mathbf{I}^{(1)}(\boldsymbol{\theta})$ (where $\mathbf{I}^{(1)}(\boldsymbol{\theta})$ is the Fisher information corresponding to a single observation), $\log|\mathbf{I}(\boldsymbol{\theta})| = c\log n + \log|\mathbf{I}^{(1)}(\boldsymbol{\theta})|$. For large n, drop the order-1 terms $\log|\mathbf{I}^{(1)}(\boldsymbol{\theta})|$ and $\frac{c}{2}\left(1 + \log\frac{1}{12}\right)$. Finally, for a given c, take $-\log p(\mathcal{Y}|\boldsymbol{\theta}) \simeq -\log p(\mathcal{Y}|\widehat{\boldsymbol{\theta}}(c))$, where $\widehat{\boldsymbol{\theta}}(c)$ is the corresponding ML estimate. The result is the well-known MDL criterion,

$$\widehat{c}_{\text{MDL}} = \arg\min_{c}\Big\{-\log p(\mathcal{Y}|\widehat{\boldsymbol{\theta}}(c)) + \frac{c}{2}\log n\Big\}, \tag{10}$$

whose two-part code interpretation is clear: the data code-length is $-\log p(\mathcal{Y}|\widehat{\boldsymbol{\theta}}(c))$, while each of the c components of $\widehat{\boldsymbol{\theta}}(c)$ requires a code-length proportional to $(1/2)\log n$. Intuitively, this means that the encoding precision of the parameter estimates is made inversely proportional to the estimation error standard deviation, which, under regularity conditions, decreases with $\sqrt{n}$, leading to the $(1/2)\log n$ term [49].

4.2 The Proposed Criterion for Mixtures

For mixtures, $\mathbf{I}(\boldsymbol{\theta})$ cannot, in general, be obtained analytically [37], [42], [57]. To side-step this difficulty, we replace $\mathbf{I}(\boldsymbol{\theta})$ by the complete-data Fisher information matrix $\mathbf{I}_c(\boldsymbol{\theta}) \equiv -E[D^2_{\boldsymbol{\theta}} \log p(\mathcal{Y}, \mathcal{Z}|\boldsymbol{\theta})]$, which upper-bounds $\mathbf{I}(\boldsymbol{\theta})$ [57]. $\mathbf{I}_c(\boldsymbol{\theta})$ has block-diagonal structure

$$\mathbf{I}_c(\boldsymbol{\theta}) = n\ \text{block-diag}\Big\{\alpha_1\mathbf{I}^{(1)}(\boldsymbol{\theta}_1), \ldots, \alpha_k I^{(1)}(\boldsymbol{\theta}_k), \mathbf{M}\Big\},$$

where $\mathbf{I}^{(1)}(\boldsymbol{\theta}_m)$ is the Fisher matrix for a single observation known to have been produced by the mth component, and $\mathbf{M}$ is the Fisher matrix of a multinomial distribution (recall

1. Here, $D^2_{\boldsymbol{\theta}}$ denotes the matrix of second derivatives, or Hessian.

that $|\mathbf{M}| = (\alpha_1\alpha_2\cdots\alpha_k)^{-1}$) [57]. The approximation of $\mathbf{I}(\boldsymbol{\theta})$ by $\mathbf{I}_c(\boldsymbol{\theta})$ becomes exact in the limit of nonoverlapping components.

We adopt a prior expressing lack of knowledge about the mixture parameters. Naturally, we model the parameters of different components as a priori independent and also independent from the mixing probabilities, i.e.,

$$p(\boldsymbol{\theta}) = p(\alpha_1,\ldots,\alpha_k)\prod_{m=1}^{k} p(\boldsymbol{\theta}_m).$$

For each factor $p(\boldsymbol{\theta}_m)$ and $p(\alpha_1,\ldots,\alpha_k)$, we adopt the standard noninformative Jeffreys' prior (see, for example, [3])

$$p(\boldsymbol{\theta}_m) \propto \sqrt{|\mathbf{I}^{(1)}(\boldsymbol{\theta}_m)|} \tag{11}$$

$$p(\alpha_1,\ldots,\alpha_k) \propto \sqrt{|\mathbf{M}|} \;=\; (\alpha_1\alpha_2\cdots\alpha_k)^{-1/2} \tag{12}$$

for $0 \leq \alpha_1,\alpha_2,\ldots,\alpha_k \leq 1$ and $\alpha_1+\alpha_2+\cdots+\alpha_k = 1$. With these choices and noticing that for a k-component mixture, $c = Nk+k$, where N is the number of parameters specifying each component, i.e., the dimensionality of $\boldsymbol{\theta}_m$, (9) becomes

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta},\mathcal{Y}), \tag{13}$$

with

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta},\mathcal{Y}) = &\frac{N}{2}\sum_{m=1}^{k}\log\Big(\frac{n\,\alpha_m}{12}\Big) + \frac{k}{2}\,\log\frac{n}{12} \\ &+ \frac{k(N+1)}{2} - \log p(\mathcal{Y}|\boldsymbol{\theta}).\end{aligned} \tag{14}$$

Apart from the order-1 term $\frac{k(N+1)}{2}(1-\log 12)$, this criterion has the following intuitively appealing interpretation in the spirit of the standard two-part code formulation of MDL and MML: 1) As usual, $-\log p(\mathcal{Y}|\boldsymbol{\theta})$ is the code-length of the data. 2) The expected number of data points generated by the mth component of the mixture is $n\alpha_m$; this can be seen as an effective sample size from which $\boldsymbol{\theta}_m$ is estimated; thus, the "optimal" (in the MDL sense) code length for each $\boldsymbol{\theta}_m$ is $(N/2)\log(n\alpha_m)$. 3) The α_ms are estimated from all the n observations, giving rise to the $(k/2)\log(n)$ term.

The objective function in (14) does not make sense if we allow any of the α_ms to be zero (it becomes $-\infty$). However, this difficulty is removed by invoking its code-length interpretation: To specify the mixture model, we only need to code the parameters of those components whose probability is nonzero. Letting k_{nz} denote the number of non-zero-probability components, we have

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta},\mathcal{Y}) = &\frac{N}{2}\sum_{m:\,\alpha_m>0}\log\Big(\frac{n\,\alpha_m}{12}\Big) + \frac{k_{nz}}{2}\,\log\frac{n}{12} \\ &+ \frac{k_{nz}(N+1)}{2} - \log p(\mathcal{Y}|\boldsymbol{\theta}).\end{aligned} \tag{15}$$

An additional term is needed to encode k_{nz}, but its code length is constant (specifically, $\log(k)$, since $k_{nz} \in \{1,2,\ldots,k\}$), thus it is irrelevant. This is the final cost function, whose minimization with respect to $\boldsymbol{\theta}$ will constitute our mixture estimate.

5 Algorithm

5.1 Minimization of the Cost Function via EM

From a Bayesian point of view, (15) is equivalent, for fixed k_{nz}, to an a posteriori density resulting from the adoption of Dirichlet-type prior for the α_m's,

$$p(\alpha_1,\ldots,\alpha_k) \propto \exp\left\{-\frac{N}{2}\sum_{m=1}^{k}\log\alpha_m\right\}, \tag{16}$$

(with negative parameters, thus improper [3]), and a flat prior leading to ML estimates for the $\boldsymbol{\theta}_m$s. Since Dirichlet priors are conjugate to multinomial likelihoods [3], the EM algorithm to minimize the cost function in (15), with k_{nz} fixed, has the following M-step (recall the constraints in (2)):

$$\widehat{\alpha}_m(t+1) = \frac{\max\left\{0,\left(\sum_{i=1}^{n} w_m^{(i)}\right) - \frac{N}{2}\right\}}{\sum_{j=1}^{k}\max\left\{0,\left(\sum_{i=1}^{n} w_j^{(i)}\right) - \frac{N}{2}\right\}}, \quad \text{for } m = 1,2,\ldots,k, \tag{17}$$

$$\widehat{\boldsymbol{\theta}}_m(t+1) = \arg\max_{\boldsymbol{\theta}_m} Q(\boldsymbol{\theta},\widehat{\boldsymbol{\theta}}(t)), \quad \text{for } m : \widehat{\alpha}_m(t+1) > 0, \tag{18}$$

where the $w_m^{(i)}$ are given by the E-step equation in (6). The $\boldsymbol{\theta}_m$s corresponding to components for which $\widehat{\alpha}_m(t+1) = 0$ become irrelevant; notice in (3) that any component for which $\alpha_m = 0$ does not contribute to the log-likelihood.

We stress that, for $N > 1$, this Dirichlet "prior" with negative exponents $-N/2$ is not the original Jeffreys prior on the mixing probabilities ((12), which is itself a Dirichlet prior but with exponent -1/2). Although Dirichlet priors with negative exponents (e.g., Jeffreys' prior) have been adopted in several contexts, to our knowledge Dirichlet (improper) "priors" with exponents less than -1 have not been used before.

We now highlight some aspects of this algorithm and its relationship with other work.

5.1.1 Component Annihilation

An important feature of the **M-step** defined by (17) is that it performs component annihilation, thus being an explicit rule for moving from the current value of k_{nz} to a smaller one. Notice that this prevents the algorithm from approaching the boundary of the parameter space: When one of the components becomes "too weak," meaning that it is not supported by the data, it is simply annihilated. One of the drawbacks of standard EM for mixtures is thus avoided.

5.1.2 Robustness Regarding Initialization

By starting with $k_{nz} = k$, where k is much larger than the true/optimal number of mixture components, this algorithm is robust with respect to initialization. Local maxima of the likelihood arise when there are too many components in one region of the space, and too few in another (see, e.g., [59]) because EM is unable to move components across low-likelihood regions. By starting with "too many" components all over the space, this problem is avoided, and all

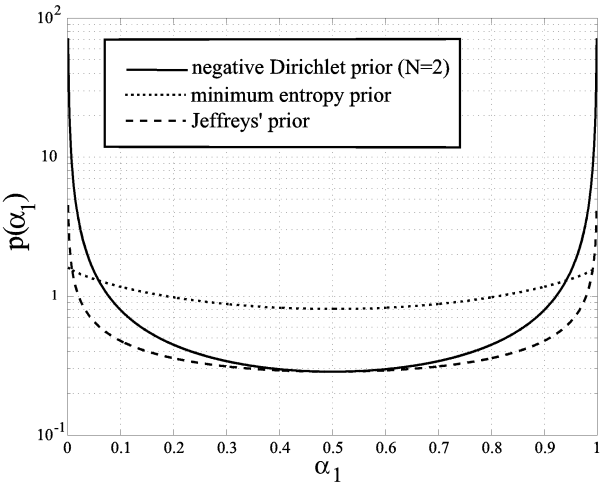

Fig. 1. Plot of the "negative Dirichlet prior" in (16) (solid line), for $k = 2$ and $N = 2$, thus $\alpha_2 = 1 - \alpha_1$ and $p(\alpha_1) \propto (\alpha_1(1-\alpha_1))^{-1}$); observe how this prior encourages configurations where α_1 equals either zero or one. For comparison, we also plot the corresponding Jeffreys prior (as given by (12), dashed line) and the minimum entropy prior $p(\alpha_1) = \alpha_1^{\alpha_1}(1-\alpha_1)^{(1-\alpha_1)}$ [9], [64] (dotted line). Notice how these other priors also favor estimates where α_1 equals either zero or one, though not as strongly.

that has to be done is to remove the unnecessary ones. We have previously exploited this idea in [21].

Another type of local minimum from which standard EM may not escape corresponds to situations where two (or more) components have similar parameters, thus sharing (approximately) the same data points. The Dirichlet-type prior with negative exponents in (16) makes these situations unstable and promotes the competition among these components; as a result, one of them will eventually "win" and the others will be annihilated. The unstable nature of this negative-parameter Dirichlet-type prior is clear in the plot shown in Fig. 1, where $N = 2$, $k = 2$, thus $\alpha_2 = 1 - \alpha_1$, and $p(\alpha_1) \propto (\alpha_1(1-\alpha_1))^{-1}$.

5.1.3 Relation with Supervised Learning

It may seem that the component annihilation behavior of the new M-step in (17) is too strong. However, although we have no proof of optimality, there are some interesting connections in the particular case of Gaussian mixtures. Consider a d-dimensional Gaussian mixture, with arbitrary covariance matrices; then $N = d + d(d+1)/2$, meaning that the minimum value of $\sum_i w_m^{(i)}$ needed to support component m grows quadratically with dimension d (notice that $\sum_i w_m^{(i)}$ can be seen as an equivalent number of points assigned to the mth component). This is in accordance with known results on the relation between sample size, dimensionality, and error probability in supervised classification [46], [47]; namely, in learning quadratic discriminants, the training sample size needed to guarantee a given error probability grows (approximately) quadratically with the dimensionality of the feature space. This connection still holds in the case of a Gaussian mixture with components sharing a common covariance matrix; in this case, $N = d$, in agreement with the fact that for linear discriminants, the sample size needed to guarantee a given error probability grows linearly with the dimensionality [46], [47]. This connection with supervised classification is not surprising due to the use of the complete-data Fisher information.

5.1.4 Relation with Minimum-Entropy Priors

It can be shown that the log of the "prior" in (16) verifies

$$-\frac{N}{2}\sum_{m=1}^{k}\log\alpha_k \propto -\frac{N}{2}\,\mathcal{D}_{\mathrm{KL}}[\{1/k\} \parallel \{\alpha_m\}],$$

where $\mathcal{D}_{\mathrm{KL}}[\{1/k\} \parallel \{\alpha_m\}]$ is the Kullback-Leibler divergence between a uniform distribution (all probabilities equal to $1/k$) and the one specified by the α_ms. This shows that our criterion penalizes uniform distributions, thus being related to minimum entropy priors $p(\boldsymbol{\theta}) \propto \exp\{-\beta H(\boldsymbol{\theta})\}$, where $H(\boldsymbol{\theta})$ is the entropy of the likelihood $p(\mathbf{y}|\boldsymbol{\theta})$ [9], [64].

In [9], a minimum entropy prior (with $\beta = 1$) is used to learn the structure of probabilistic models, such as the number of components of a mixture. The method proposed in [9], like ours, starts with a large number of components, some of which are then annihilated under the influence of the minimum-entropy prior. However, as shown in [9], the M-step resulting from the minimum entropy prior on the mixing probabilities does not have a closed-form solution, and does not explicitly annihilate components; annihilation requires an additional test. In Fig. 1, our prior is compared with a minimum entropy prior (for $k = 2$ and $N = 2$) showing that it favors component annihilation more strongly. On a more fundamental level, the approach in [9] raises the following questions: 1) Why should $\beta = 1$ be chosen, and what is its influence on the results? 2) For mixture models, why consider only the entropy of the multinomial distribution defined by the mixing probabilities, which is but a lower bound on the full entropy of the mixture?

5.2 The Component-Wise EM Algorithm

Direct use of EM with the **M-step** in (17) and (18) has a failure-mode: if k is too large, it can happen that no component has enough initial support ($\sum_{i=1}^{n} w_m^{(i)} < N/2$, for $m = 1, 2, \ldots, k$) and all $\widehat{\alpha}_m$s will be undetermined. We avoid this problem by using the recent *component-wise EM for mixtures* (CEM^2) algorithm [11]. Basically, rather than simultaneously updating all the α_ms and $\boldsymbol{\theta}_m$s, CEM^2 updates them sequentially: update α_1 and $\boldsymbol{\theta}_1$, recompute $\mathcal{W}$, update α_2 and $\boldsymbol{\theta}_2$, recompute $\mathcal{W}$, and so on. For our purposes, the key feature of CEM^2 is the following: If one component dies ($\widehat{\alpha}_m(t+1) = 0$), immediate redistribution of its probability mass to the other components increases their chance of survival. This allows initialization with an arbitrarily large k without any problems.

Convergence of CEM^2 was shown in [11] with the help of its *proximal point algorithm* interpretation. The order of updating does not affect the theoretical monotonicity properties of CEM^2, although it may affect the final result since the objective function has multiple local minima. Although better results may in principle be achieved with adaptive schedules, for simplicity, we adopt a simple cyclic updating procedure, as in [11].

Finally, notice that although it may seem that CEM^2 is computationally much heavier than standard EM, due to the multiple E-steps to recompute $\mathcal{W}$, that is not so. Suppose

Inputs: $k_{\min}$, $k_{\max}$, ϵ, initial parameters $\widehat{\boldsymbol{\theta}}(0) = \{\widehat{\boldsymbol{\theta}}_1, ..., \widehat{\boldsymbol{\theta}}_{k_{\max}}, \widehat{\alpha}_1, ..., \widehat{\alpha}_{k_{\max}}\}$
Output: Mixture model in $\widehat{\boldsymbol{\theta}}_{\text{best}}$

$t \leftarrow 0, \quad k_{nz} \leftarrow k_{\max}, \quad \mathcal{L}_{\min} \leftarrow +\infty$
$u_m^{(i)} \leftarrow p(\mathbf{y}^{(i)}|\widehat{\boldsymbol{\theta}}_m)$, for $m = 1, ..., k_{\max}$, and $i = 1, ..., n$
while $k_{nz} \geq k_{\min}$ **do**
  repeat
    $t \leftarrow t + 1$
    for $m = 1$ to $k_{\max}$ **do**
      $w_m^{(i)} \leftarrow \widehat{\alpha}_m \, u_m^{(i)} \left(\sum_{j=1}^{k_{\max}} \widehat{\alpha}_j \, u_j^{(i)} \right)^{-1}$, for $i = 1, ..., n$
      $\widehat{\alpha}_m \leftarrow \max\left\{0, \left(\sum_{i=1}^{n} w_m^{(i)}\right) - \frac{N}{2}\right\} \left(\sum_{j=1}^{k} \max\left\{0, \left(\sum_{i=1}^{n} w_j^{(i)}\right) - \frac{N}{2}\right\}\right)^{-1}$
      $\{\widehat{\alpha}_1, ..., \widehat{\alpha}_{k_{\max}}\} \leftarrow \{\widehat{\alpha}_1, ..., \widehat{\alpha}_{k_{\max}}\} \left(\sum_{m=1}^{k_{\max}} \widehat{\alpha}_m \right)^{-1}$
      if $\widehat{\alpha}_m > 0$ **then**
        $\widehat{\boldsymbol{\theta}}_m \leftarrow \arg\max_{\boldsymbol{\theta}_m} \log p\left(\mathcal{Y}, \mathcal{W} | \, \boldsymbol{\theta}\right)$
        $u_m^{(i)} \leftarrow p(\mathbf{y}^{(i)}|\widehat{\boldsymbol{\theta}}_m)$, for $i = 1, ..., n$
      else
        $k_{nz} \leftarrow k_{nz} - 1$
      end if
    end for
    $\widehat{\boldsymbol{\theta}}(t) \leftarrow \{\widehat{\boldsymbol{\theta}}_1, ..., \widehat{\boldsymbol{\theta}}_{k_{\max}}, \widehat{\alpha}_1, ..., \widehat{\alpha}_{k_{\max}}\}$,
    $\mathcal{L}[\widehat{\boldsymbol{\theta}}(t), \mathcal{Y}] \leftarrow \frac{N}{2} \sum_{m:\, \widehat{\alpha}_m > 0} \log \frac{n \, \widehat{\alpha}_m}{12} + \frac{k_{nz}}{2} \log \frac{n}{12} + \frac{k_{nz} N + k_{nz}}{2} - \sum_{i=1}^{n} \log \sum_{m=1}^{k} \widehat{\alpha}_m u_m^{(i)}$
  until $\mathcal{L}[\widehat{\boldsymbol{\theta}}(t-1), \mathcal{Y}] - \mathcal{L}[\widehat{\boldsymbol{\theta}}(t), \mathcal{Y}] < \epsilon \left| \mathcal{L}[\widehat{\boldsymbol{\theta}}(t-1), \mathcal{Y}] \right|$
  if $\mathcal{L}[\widehat{\boldsymbol{\theta}}(t), \mathcal{Y}] \leq \mathcal{L}_{\min}$ **then**
    $\mathcal{L}_{\min} \leftarrow \mathcal{L}[\widehat{\boldsymbol{\theta}}(t), \mathcal{Y}]$
    $\widehat{\boldsymbol{\theta}}_{\text{best}} \leftarrow \widehat{\boldsymbol{\theta}}(t)$
  end if
  $m^* \leftarrow \arg\min_m \{\widehat{\alpha}_m > 0\}, \quad \widehat{\alpha}_{m^*} \leftarrow 0, \quad k_{nz} \leftarrow k_{nz} - 1$
end while

Fig. 2. The complete algorithm ($\epsilon = 10^{-5}$ in all the examples presented ahead).

that $\boldsymbol{\theta}_m$ and α_m were just updated; of course, via normalization, some other α_js, for $j \neq m$, also change. Updating all the $w_j^{(i)}$ variables only requires full computation of (6) for $j = m$. For $j \neq m$, the terms $p(\mathbf{y}^{(i)}|\boldsymbol{\theta}_j)$, which involve most of the computational effort of the E-step, remain unchanged and only have to be computed once per sweep, like in standard EM. In conclusion, CEM^2 is only slightly computationally heavier than EM.

5.3 The Complete Algorithm

After convergence of CEM^2, i.e., when the relative decrease in $\mathcal{L}(\widehat{\boldsymbol{\theta}}(t), \mathcal{Y})$ falls below a threshold ϵ (e.g., $\epsilon = 10^{-5}$), there is no guarantee that we have found a minimum of $\mathcal{L}(\theta, \mathcal{Y})$. In fact, the component annihilation in (17) does not take into account the additional decrease in $\mathcal{L}(\boldsymbol{\theta}, \mathcal{Y})$ caused by the decrease in k_{nz}. Consequently, we must check if smaller values of $\mathcal{L}(\boldsymbol{\theta}, \mathcal{Y})$ are achieved by setting to zero components that were not annihilated by (17). To this end, we simply annihilate the least probable component (with smallest $\widehat{\alpha}_m$) and rerun CEM^2 until convergence. This procedure is repeated until $k_{nz} = 1$. In the end, we choose the estimate that led to the minimum value of $\mathcal{L}(\boldsymbol{\theta}, \mathcal{Y})$. Of course, if we know that the number of components is no less than some $k_{\min} > 1$, we stop when $k_{nz} = k_{\min}$. Fig. 2 contains a detailed pseudocode description of the algorithm.

6 Experiments

Although our algorithm can be used for any type of mixture model, our experiments focus only on Gaussian mixtures, which are by far the most commonly used. In d-variate Gaussian mixture models with arbitrary covariance matrices, we have

$$p(\mathbf{y}|\boldsymbol{\theta}_m) = \frac{(2\pi)^{-\frac{d}{2}}}{\sqrt{|\mathbf{C}_m|}} \exp\left\{ -\frac{1}{2} (\mathbf{y} - \boldsymbol{\mu}_m)^T \mathbf{C}_m^{-1} (y - \boldsymbol{\mu}_m) \right\} \equiv \mathcal{N}(\boldsymbol{\mu}_m, \mathbf{C}_m).$$

Thus, $\boldsymbol{\theta}_m = (\boldsymbol{\mu}_m, \mathbf{C}_m)$, $N = d + d(d+1)/2$, and the M-step (18) is

$$\widehat{\boldsymbol{\mu}}_m(t+1) = \left(\sum_{i=1}^{n} w_m^{(i)} \right)^{-1} \sum_{i=1}^{n} \mathbf{y}^{(i)} \, w_m^{(i)} \tag{19}$$

$$\widehat{\mathbf{C}}_m(t+1) = \left(\sum_{i=1}^{n} w_m^{(i)} \right)^{-1} \sum_{i=1}^{n} (\mathbf{y}^{(i)} - \widehat{\boldsymbol{\mu}}_m(t+1)) (\mathbf{y}^{(i)} - \widehat{\boldsymbol{\mu}}_m(t+1))^T \, w_m^{(i)}. \tag{20}$$

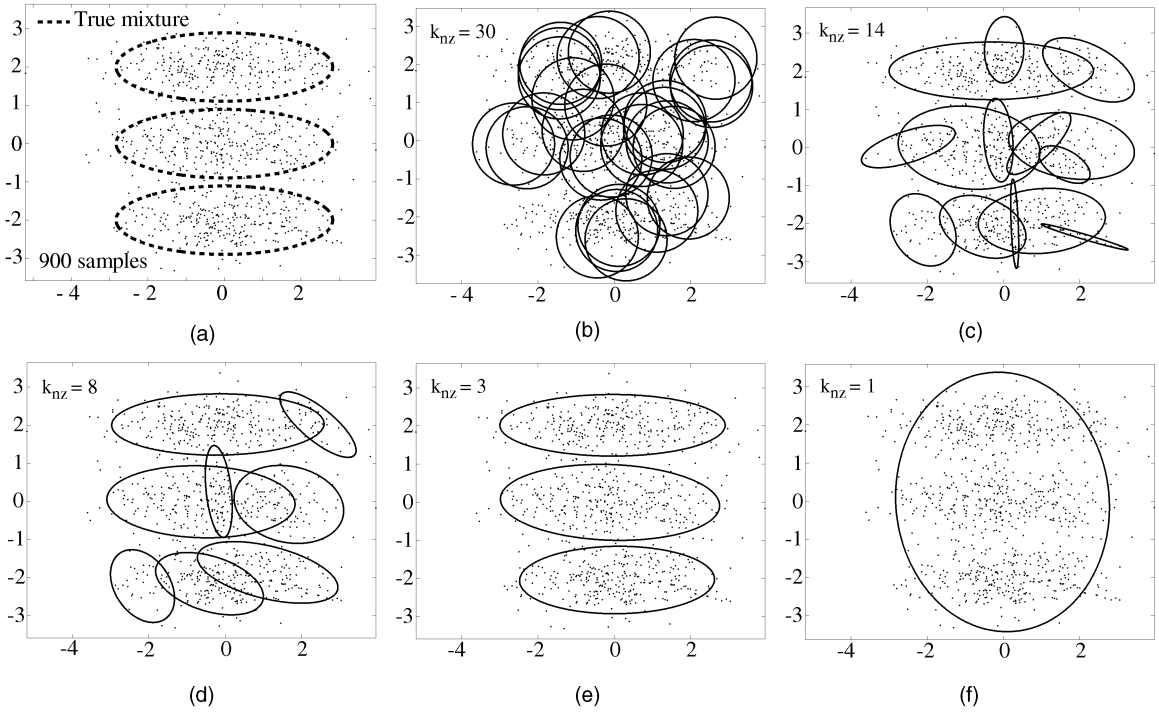

Fig. 3. Fitting a Gaussian mixture: (a) the dashed ellipses show the true mixture; (b) initialization with $k_{nz} = 30$; (c), (d), (e), and (f) four estimates (for $k_{nz} = 14, 8, 3,$ and 1). Our algorithm selects a mixture with $k_{nz} = 3$. The solid ellipses are level-curves of each component estimate; only those components for which $\widehat{\alpha} \neq 0$ are shown.

6.1 Initialization for Gaussian Mixtures

We initialize the $k_{\max}$ mean vectors to randomly chosen data points. The initial covariances are made proportional to the identity matrix, $\widehat{\mathbf{C}}(t=0) = \sigma^2\mathbf{I}$, where σ^2 is a fraction (e.g., $1/5$ or $1/10$) of the mean of the variances along each dimension of the data,

$$\sigma^2 = \frac{1}{10\,d}\ \mathrm{trace}\left(\frac{1}{n}\sum_{i=1}^{n}(\mathbf{y}^{(i)} - \mathbf{m})(\mathbf{y}^{(i)} - \mathbf{m})^T\right)$$

($\mathbf{m} \equiv \frac{1}{n}\sum_{i=1}^{n}\mathbf{y}^{(i)}$ is the global data mean). Other initialization strategies are possible; in fact, we have verified experimentally that any method that spreads a large enough number of components throughout the data space leads to good results. To quantify what "large enough" means, notice that the condition for success of this initialization is that all components of the true mixture are represented; that is, for each component, there is at least one data point such that one of the initial means is on that data point. Assume that the sample size n is large enough so that the proportion of points that were generated by each component is very close to their mixing probabilities $\{\alpha_1, \ldots, \alpha_k\}$. Let $\alpha_{\min} = \min\{\alpha_1, \ldots, \alpha_k\}$ be the probability of the least probable component, i.e., the one which will more probably be left out of the initialization. The probability that this component is unrepresented in the initialization is (approximately, for large n) given by $(1 - \alpha_{\min})^k$. Then, if a probability of successful initialization of at least $1 - \varepsilon$ is desired, it is necessary to have

$$k > \frac{\log \varepsilon}{\log(1 - \alpha_{\min})}.$$

For example, for $\varepsilon = 0.05$ and $\alpha_{\min} = 0.1$, we obtain $k > 28$.

6.2 First Examples

In the first example, we use 900 samples from a 3-component bivariate mixture from [58]:

$$\alpha_1 = \alpha_2 = \alpha_3 = 1/3$$

mean vectors at $[0, -2]^T$, $[0,\ 0]^T$, $[0,\ 2]^T$, and equal covariance matrices $\mathrm{diag}\{2,\ 0.2\}$. With $k_{\max} = 30$, Fig. 3 shows an initialization, two intermediate configurations, and the (successful) final estimate with $k_{nz} = 3$. The plot of $\mathcal{L}(\widehat{\boldsymbol{\theta}}(t), \mathbf{y})$ shown in Fig. 4a reveals that mixtures with $k_{nz} = 2$ and 1 have higher values of $\mathcal{L}(\widehat{\boldsymbol{\theta}}(t), \mathbf{y})$ and, consequently, they were discarded. We repeated this experiment 100 times with 100 percent success. In conclusion, for this mixture, our method successfully solves the initialization issue, like the DA version of EM proposed in [58], without using any cooling schedule. More importantly, our method automatically selects the number of components.

In the second example, we consider a situation where the mixture components overlap. In fact, two of the four components share a common mean, but have different covariance matrices. The parameters are:

$$\alpha_1 = \alpha_2 = \alpha_3 = 0.3,$$

and

$$\alpha_4 = 0.1; \boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = [-4,\ -4]^T, \boldsymbol{\mu}_3 = [2,\ 2]^T,$$

and $\boldsymbol{\mu}_4 = [-1,\ -6]^T$; and

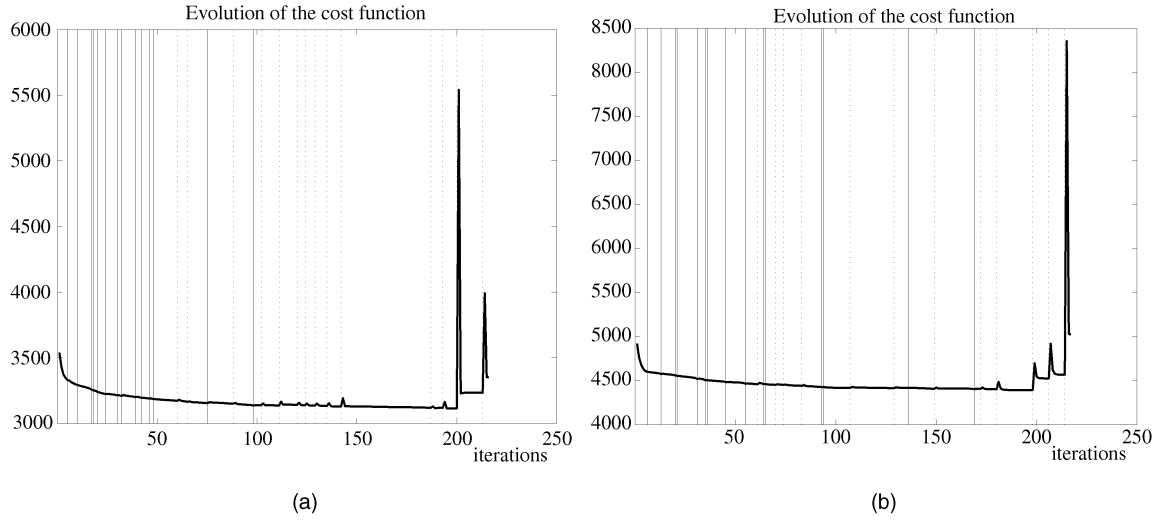

Fig. 4. Evolution of the cost function $\mathcal{L}(\widehat{\boldsymbol{\theta}}(t), \mathbf{y})$ for (a) the example in Fig. 3, and (b) the example of Fig. 5. The vertical solid lines signal the annihilation of one (or more) components inside the CEM^2 algorithm; the vertical dotted lines indicate the least probable component being forced to zero after the convergence of CEM^2.

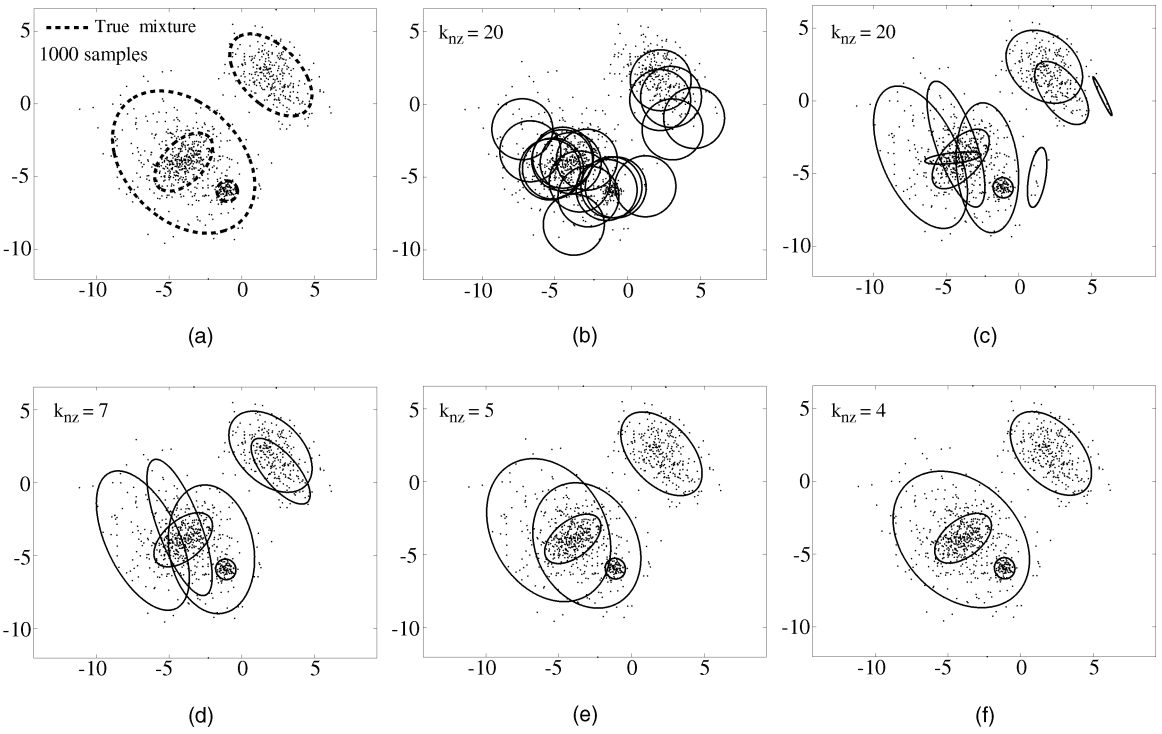

Fig. 5. Fitting a Gaussian mixture with overlapping components: (a) the dashed ellipses show the true mixture; (b) initialization with $k_{nz} = 20$; (c), (d), and (e) three immediate estimates (for $k_{nz} = 10, 7$, and 5); (f) the final estimate (with $k_{nz} = 4$).

$$\mathbf{C}_1 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \quad \mathbf{C}_2 = \begin{bmatrix} 6 & -2 \\ -2 & 6 \end{bmatrix}$$
$$\mathbf{C}_3 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \quad \mathbf{C}_4 = \begin{bmatrix} 0.125 & 0 \\ 0 & 0.125 \end{bmatrix}.$$

Fig. 5a shows the true mixture and 1,000 samples of it, while Figs. 5b, 5c, 5d, 5e, and 5f show the evolution of the algorithm. The cost function is plotted in Fig. 4b.

Next, we consider the well-known Iris data set (150 four-dimensional points from three classes, 50 per class), to which we fit a Gaussian mixture with arbitrary covariance matrices. Using $k_{\max} = 20$, the algorithm was run 100 times (to study the robustness with respect to the random initialization), and every time it correctly identified the three classes, using 30 to 50 iterations. The data and the estimated components (from one of the 100 runs) are shown in Fig. 6, projected on the two principal components of the data.

Finally, we consider two univariate data sets that were studied in [48] using MCMC: the *enzyme* data set ($n = 245$) and the *acidity* data-set ($n = 155$). Figs. 7 and 8 show histograms of these data sets together with the mixture densities obtained by our algorithm. Interestingly, the selected numbers of mixture components ($k_{nz} = 3$ and

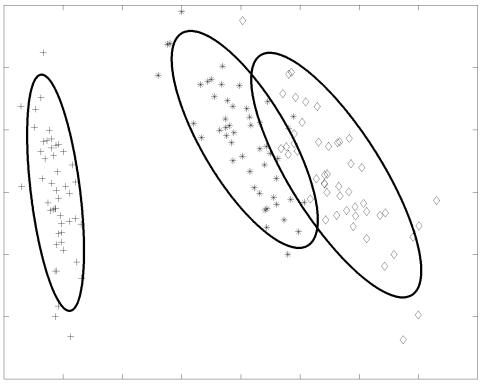

Fig. 6. Iris data (projected on the two principal components) and the estimated Gaussian mixture. The three classes are plotted with different symbols ("+," "*," and "◇"); of course, the mixture was obtained without knowledge of the class of each data point.

$k_{nz} = 4$, for the acidity and enzyme data sets, respectively) coincide with the location of the maxima of the a posteriori marginal probabilities $p(k|\mathcal{Y})$ obtained via MCMC [48].

6.3 Comparison with EM-Based Methods

To compare our algorithm with EM-based methods referred in Section 3.1.1, we chose MDL/BIC, as the most commonly used model selection criterion, and LEC and ICL which are reported in [37] and [50] as outperforming all other criteria.

As described in Section 3.1.1, mixture estimates are obtained by EM for a range of values of k, from $k_{\min}$ to $k_{\max}$; the "best" one is then selected according to (7), where $\mathbf{C}\left(\widehat{\boldsymbol{\theta}}(k), k\right)$ is either the MDL/BIC, LEC, or ICL criterion. The MDL/BIC criterion is given by (10), with $c = Nk + k$. For details about LEC, see [37] and [50]. ICL is described in [5] and [37]. To deal with the initialization issue of EM, a standard approach was used: start from 10 different random initial conditions and pick the solution with the highest likelihood. When one of the covariances approaches singularity (as measured by its condition number), we abort that run of EM and start a new one. The convergence criterion is the same which is used in our algorithm, also with $\epsilon = 10^{-5}$.

Like our method, MDL/BIC and LEC were 100 percent successful in identifying the three components in 100 simulations from the mixture of Fig. 3, and the four components in 100 simulations of the mixture of Fig. 5. The ICL criterion was 100 percent successful for the mixture of Fig. 3, but with the mixture of Fig. 5 it failed completely, choosing two components 82 times, and three components 18 times. The poor performance of ICL when there are overlapping components, also reported in [5], is due to the fact that it is tailored to clustering applications. In these tests, the EM-based algorithms used $k_{\min} = 1$ and $k_{\max} = 5$; obtaining the five mixture candidates requires a total of around 1,200 to 1,400 EM iterations, versus the 200 to 250 iterations typically required by our method (for a much larger range, $k_{\min} = 1$ to $k_{\max} = 30$) for these two mixtures.

With the Iris data set, MDL/BIC and LEC were again able to find the three components (confirming the results in [50]), while ICL fails, always selecting two components.

For the univariate data sets, MDL/BIC, LEC, and ICL, all chose mixtures with two components. According to the marginal a posteriori probabilities $p(k|\mathcal{Y})$ obtained in [48] via MCMC, this number has little support from the data: respectively, $p(2|\mathcal{Y}) = 0.082$ and $p(2|\mathcal{Y}) = 0.024$, for the acidity and enzyme data sets. As mentioned above, the numbers of components selected by our algorithm have the highest values: respectively, $p(3|\mathcal{Y}) = 0.244$ and $p(4|\mathcal{Y}) = 0.317$, for the acidity and enzyme data (see [48]).

To study how the several methods perform when the degree of component overlap varies, we used a bivariate Gaussian mixture with two equiprobable components ($\boldsymbol{\mu}_1 = [0, 0]^T$, $\boldsymbol{\mu}_2 = [\delta, 0]^T$, and $\mathbf{C}_1 = \mathbf{C}_2 = \mathbf{I}$). Fig. 9a shows the percentage (over 50 simulations, with $n = 800$) of correct selections achieved by each method, assuming free covariance matrices. Notice that, for $\delta < 2$, the mixture density is not even bimodal [57]. ICL only performs acceptably for $\delta > 3$, so we left it out of this plot. When not correct, all the methods selected just one component, never more than 2. These results reveal an excellent performance of our method and of the LEC criterion, with our algorithm requiring roughly $5 \sim 7$ times fewer iterations. Fig. 9b reports a similar study, now for 10-dimensional data ($d = 10$, thus each component has $N = 65$ parameters), with $\boldsymbol{\mu}_1 = [0, \ldots, 0]^T$, $\boldsymbol{\mu}_2 = [\delta, \ldots, \delta]^T$, and $\mathbf{C}_1 = \mathbf{C}_2 = \mathbf{I}$. In this case, the distance between the two components is $\delta\sqrt{10}$. The results for the LEC criterion (not shown) are now disastrous; it always chooses $k_{\max}$ components (regardless of how large $k_{\max}$ is).

The results in Fig. 10 are from a similar test, now under the constraint that the covariance matrix of each component is diagonal. Notice the marked performance improvement for the 10-dimensional data (compare the numbers in the

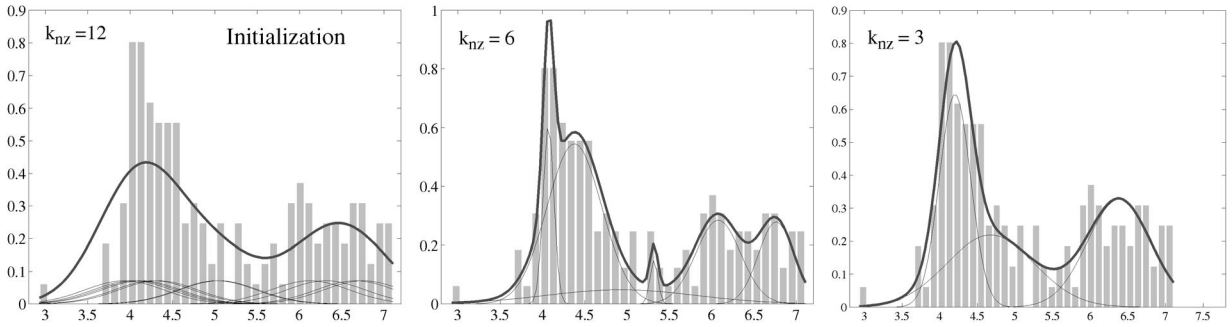

Fig. 7. Fitting a Gaussian mixture to the acidity data set. Our algorithm selects $k_{nz} = 3$.

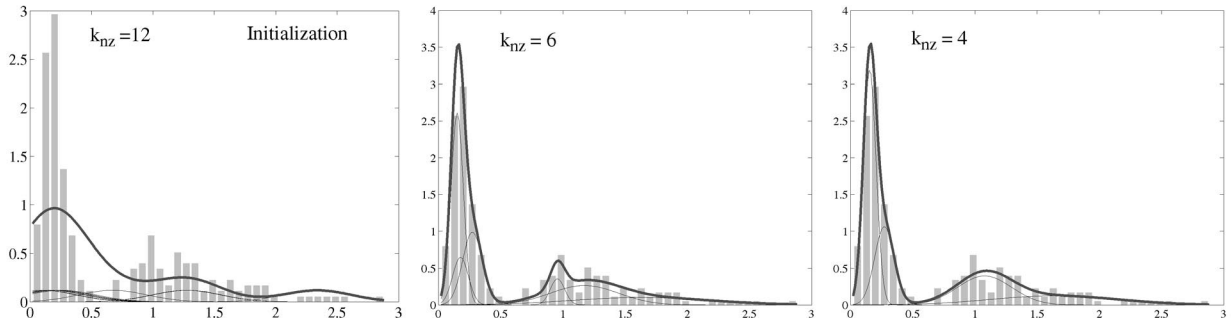

Fig. 8. Fitting a Gaussian mixture to the enzyme data set. Our algorithm selects $k_{nz} = 4$.

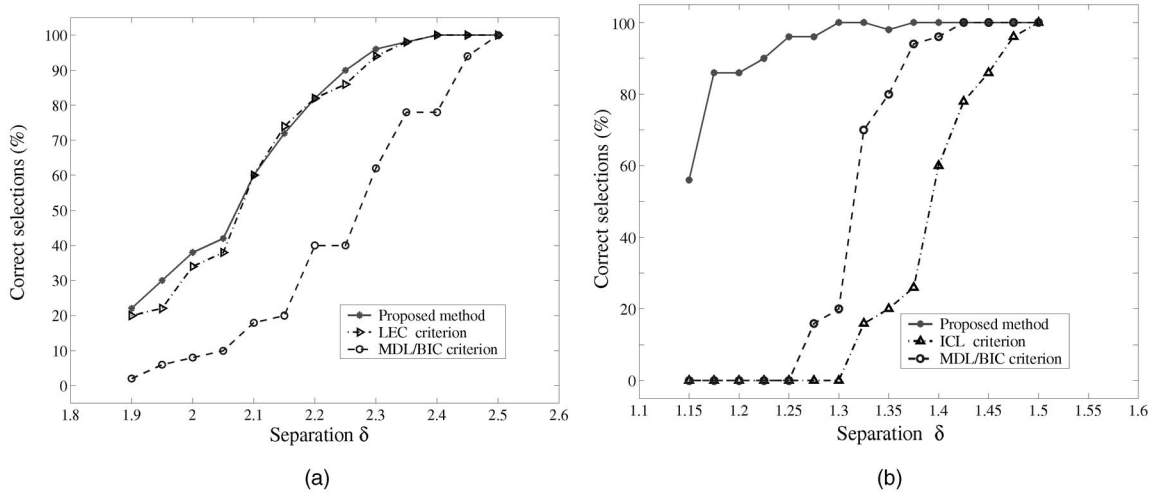

Fig. 9. Percentage of success (over 50 simulations) of various methods as a function of separtation between components; (a) for a bivariate two-component Gaussian mixture (distance equals δ), (b) for a 10-dimensional two components Gaussian mixture (distance equals $\delta\sqrt{10}$).

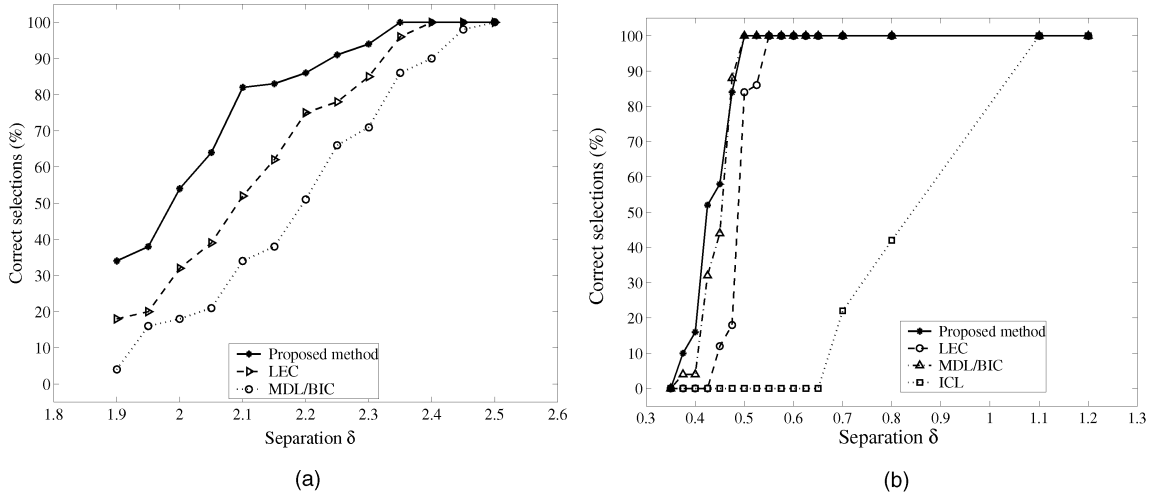

Fig. 10. Percentage of success versus separation between components under the constraint of diagonal covariances. (a) Bivarate data (distance equals δ). (b) Ten-dimensional data (distance equals $\delta\sqrt{10}$.

horizontal axis of Figs. 9b and 10b). In any case, our method always outperforms the others.

6.4 Mixtures as Class-Conditional Densities

Using Gaussian mixtures to model class-conditional densities has been suggested several times, e.g., [25], [26], [34], [55]. In our first example (following [25]), there are three (equiprobable) classes in $I\!R^{21}$. Each observation is $\mathbf{y}^{(i)} = [y_1^{(i)}, y_2^{(i)}, \ldots, y_{21}^{(i)}]^T$, with

$$
\begin{aligned}
y_j^{(i)} &= u^{(i)} h_{1,\,j} + (1 - u^{(i)}) h_{2,\,j} + n_j^{(i)}, \quad \text{Class 1,} \\
y_j^{(i)} &= u^{(i)} h_{1,\,j} + (1 - u^{(i)}) h_{3,\,j} + n_j^{(i)}, \quad \text{Class 2,} \\
y_j^{(i)} &= u^{(i)} h_{2,\,j} + (1 - u^{(i)}) h_{3,\,j} + n_j^{(i)}, \quad \text{Class 3,}
\end{aligned}
$$

TABLE 1
Average Test Error Rates (Over 50 Simulations) for the Texture Classification Example

Method	mean error rate	error rate standard deviation
Mixture-based, our method	0.0074	0.0020
Mixture-based, EM and MDL/BIC	0.0075	0.0019
Linear discriminant	0.0185	0.0024
Quadratic discriminant	0.0155	0.0027

where the $u^{(i)}$ are i.i.d. uniform in $(0,1)$, the $n_j^{(i)}$ are i.i.d. zero-mean unit-variance Gaussian samples, and h_1, h_2, and h_3 define three points in $\mathbb{R}^{21}$ given by

$$h_{1,j} = \max(0, 6 - |j - 11|), \quad j = 1, 2, \ldots, 21$$
$$h_{2,j} = h_{1,j-4}, \quad j = 1, 2, \ldots, 21$$
$$h_{3,j} = h_{1,j+4}, \quad j = 1, 2, \ldots, 21.$$

Each class is roughly one side of a noisy triangle in $\mathbb{R}^{21}$, whose vertices are $[h_{1,1}, \ldots, h_{1,21}]^T$, $[h_{2,1}, \ldots, h_{2,21}]^T$, and $[h_{3,1}, \ldots, h_{3,21}]^T$.

As in [25], the class-conditional mixtures are fitted to sets of 100 samples per class; the resulting MAP classifier is tested on an independent set of 500 samples. In [25], the class-conditionals are fitted to 3-component mixtures with a common covariance, using EM with 10 random starts; the resulting classifier outperforms linear discriminants, quadratic discriminants, *classification and regression trees* (CART), and other classifiers. The mean error rate (over 10 simulations) reported in [25] is 0.169. Using a common covariance matrix for each class (like in [25]), we found a mean error rate of 0.162, using our method to estimate the class-conditional mixtures. Notice that, while exhibiting a slightly better error rate, our method does not require multiple random starts and it adaptively selects the number of components (in this problem, usually three, but also two and four several times).

To further exemplify the use of mixtures as class-conditional densities, we considered a real texture classification problem. From a collage of four Brodatz textures, we obtained 4,000 ($\sim 1,000$ per class) randomly located 19-dimensional Gabor filter features (see [30]). Using the proposed algorithm, we then fitted mixtures (with free covariance matrices) to 800 samples from each class, leaving the remaining 200 samples per class to serve as test data. In Table 1, we report the results (over 50 simulations with random train/test data partitions) in terms of error rate, comparing the mixture-based methods with linear discriminants and quadratic discriminants. Our method achieves a similar performance as the EM-based method using the MDL/BIC criterion, at a fraction (~ 0.1) of the computational cost. The ICL and LEC criteria yielded very bad results in this problem. Finally, Fig. 11 shows the best 2D projection (obtained using discriminant analysis [19]) of 800 points from each class, together with the projections of the mixtures that were fitted to each class-conditional density and of the corresponding decision regions.

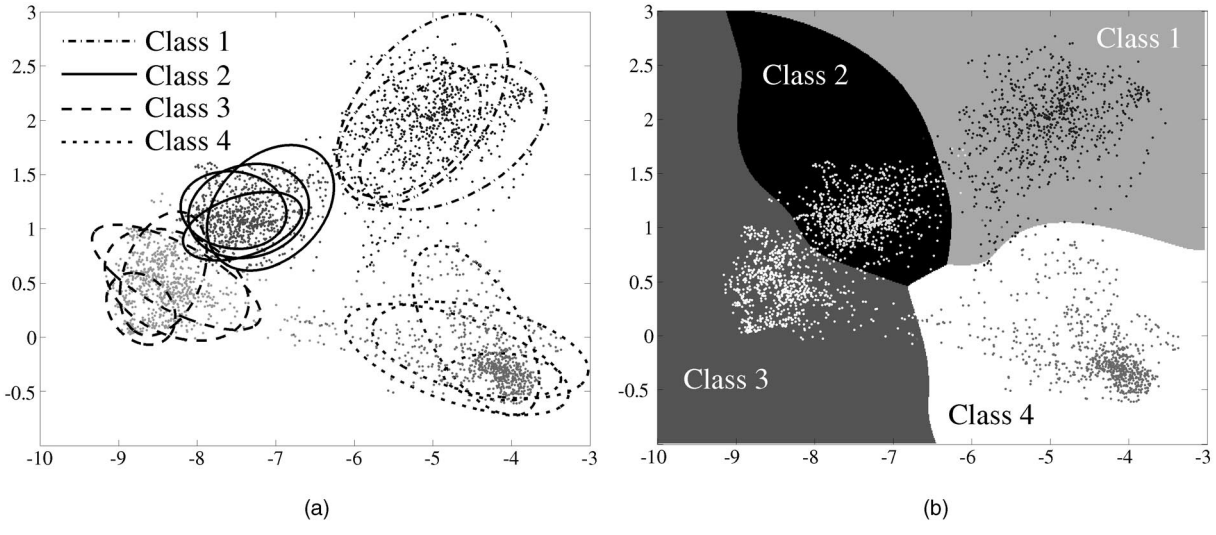

Fig. 11. Best 2D projection of the texture data (using discriminant analysis) together with (a) the projections of the mixtures that were fitted to each class-conditional density and (b) the projections of the corresponding decision regions.

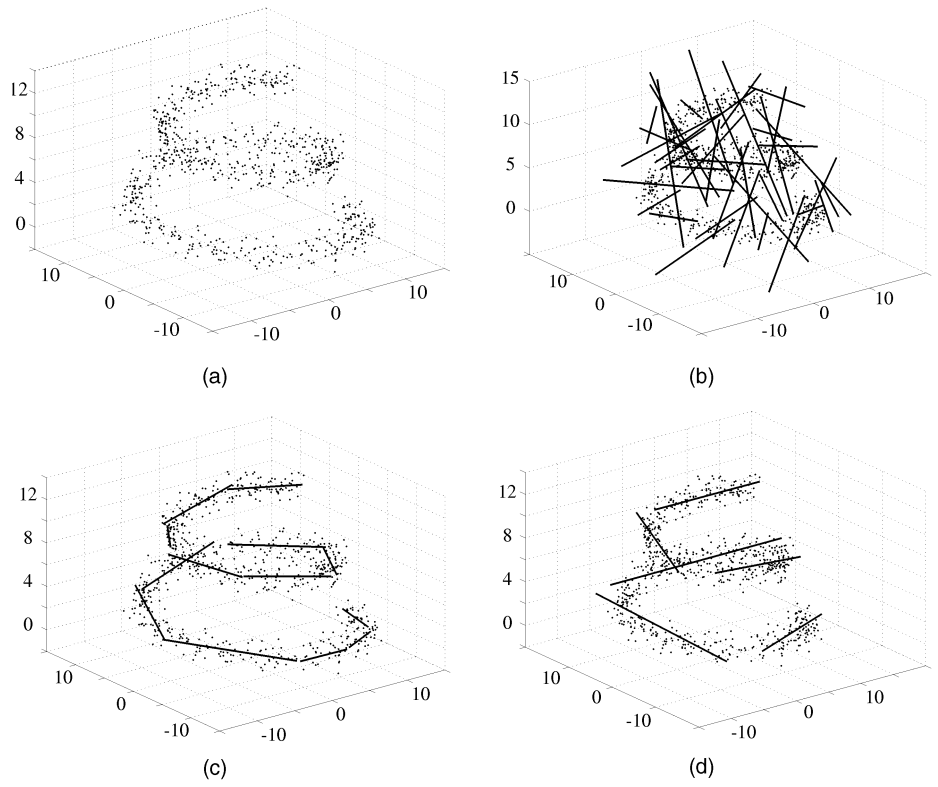

Fig. 12. Fitting a mixture of factor analyzers to the noisy shrinking spiral: (a) data, (b) initial configuration at $k_{nz} = k = 40$, (c) estimate at $k_{nz} = 13$ (the selected one), and (d) at $k_{nz} = 6$. The line segment ends are $(\boldsymbol{\mu}_m - 2\boldsymbol{\Lambda}_m, \boldsymbol{\mu}_m + 2\boldsymbol{\Lambda}_m)$.

6.5 Mixtures of Factor Analyzers

Mixtures of factor analyzers (MFA), proposed in [24], are basically mixtures of Gaussians with a reduced parameterization of the covariance matrices:

$$p(\mathbf{y}^{(i)}|\boldsymbol{\theta}) = \sum_{m=1}^{k} \alpha_m \mathcal{N}(\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m \boldsymbol{\Lambda}_m^T + \boldsymbol{\Psi}_m).$$

The data produced by component m is modeled as $\mathbf{y} = \Lambda_m \mathbf{v} + \mathbf{n}$, where $\mathbf{v}$ is $\mathcal{N}(0, \mathbf{I})$, $\mathbf{n}$ is $\mathcal{N}(0, \boldsymbol{\Psi}_m)$, and $\boldsymbol{\Psi}_m$ is a diagonal matrix. Since $\mathbf{v}$ may be of lower dimension than $\mathbf{y}$, MFA are able to perform local dimensionality reduction. MFA are closely related to the mixtures of probabilistic principal component analyzers proposed in [56]. An EM algorithm for MFA was derived in [24]. The split and merge EM algorithm [59] was also applied to MFA, successfully overcoming most of the initialization sensitivity of EM. A recently proposed variational Bayesian approach estimates the number of components and also the dimensionality of each component [23].

We tested the algorithm proposed here on the noisy shrinking spiral data. As described in [59], the goal is to extract a piece-wise linear approximation to a one-dimensional non-linear manifold from three-dimensional data. In this case, $\boldsymbol{\theta}_m \equiv (\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m, \boldsymbol{\Psi}_m)$, where each $\boldsymbol{\Lambda}_m$ is 3×1, thus $N = 9$ (three components of each $\boldsymbol{\mu}_m$, plus three components of each $\boldsymbol{\Lambda}_m$, plus the three diagonal elements of each $\boldsymbol{\Psi}_m$). The data is generated according to

$$\begin{bmatrix} x_1^i \\ x_2^i \\ x_3^i \end{bmatrix} = \begin{bmatrix} (13 - 0.5t_i)\cos t_i \\ (0.5t_i - 13)\sin t_i, \\ t_i \end{bmatrix} + \begin{bmatrix} n_1^i \\ n_2^i \\ n_3^i \end{bmatrix}, \quad i = 1, 2, \ldots 900,$$

the t_i are uniformly distributed in $[0, 4\pi]$, and n_1^i, n_2^i, and n_3^i are i.i.d. $\mathcal{N}(0, 1)$.

Fig. 12 shows the data set, the initial mixture with $k_{nz} = k = 40$, and estimates at $k_{nz} = 13$ (the selected one) and $k_{nz} = k_{\min} = 6$. We repeated the test 30 times with different data sets; the algorithm selected $k_{nz} = 13$, 28 times, and $k_{nz} = 12$ and $k_{nz} = 11$, once each, never getting trapped in poor local minima. With $k_{\min} = 6$, the number of iterations is typically between 300 and 400, similar to the algorithm in [59]. Notice however, that the algorithm in [59] does not select the number of components and would have to be run several times to select an optimal number of components.

6.6 Failure Conditions

The most difficult mixtures for our algorithm are those with components of very different weights. It may happen that a component that is already well adjusted to a subset of data produced by a low weight component gets prematurely forced to zero, instead of an unnecessary heavier component almost completely overlapping another one. For instance, in the example of Fig. 5, if $\alpha_4 < 0.05$, the algorithm sometimes drops the smaller component. A possible solution to this problem consists in using different criteria to reduce the mixture, e.g., the weighted Kullback-Leibler criterion that we have used in [21].

7 Discussion and Outlook

In this paper, we have proposed a method for learning finite mixtures from data which is able to select the number of components in an unsupervised way. The proposed algorithm also avoids several drawbacks of the standard EM algorithm: sensitivity to initialization and possible convergence to the boundary of the parameter space. The method is based on a MML-like criterion which is directly implemented by a modified EM algorithm. The novelty in our approach is that we do not use MML as a model selection criterion to choose one among a set of candidate models; instead, we seamlessly integrate estimation and model selection in a single algorithm. Experimental results showed the good performance of the approach in learning mixtures of Gaussians and mixtures of factor analyzers.

An important issue in mixture fitting is the detection of outliers (i.e., observations that are not well modeled by any mixture component). Outliers can be handled by an extra component (uniform or Gaussian of very high variance) whose role is to "absorb" these anomalous observations [37]. We are currently investigating how this idea can be incorporated in the technique proposed in this paper.

Acknowledgments

The authors acknowledge Dr. Naonori Ueda for providing the noisy shrinking spiral code. This research was partially supported by the US Department of the Navy, Office of Naval Research (ONR), and by the Portuguese Foundation for Science and Technology (FCT), under project POSI/33143/SRI/2000.

Appendix

Derivation of the MML Criterion

This appendix presents the derivation of the MML criterion (9) which plays a central role in this paper. We closely follow [32], and start with the scalar parameter case. Consider a prior $p(\theta)$ and the likelihood function $p(\mathcal{Y}|\theta)$. Let $\tilde{\theta}$ be a finite precision version of θ, with quantization step Δ. The complete description length is

$$\text{Length}(\tilde{\theta}, \mathcal{Y}) \simeq \underbrace{-\log(\Delta p(\tilde{\theta}))}_{\text{Length}(\tilde{\theta})} \underbrace{-\log p(\mathcal{Y}|\tilde{\theta})}_{\text{Length}(\mathcal{Y}|\tilde{\theta})}, \tag{21}$$

since, a priori, $P(\theta \in [\tilde{\theta} - \Delta/2,\ \tilde{\theta} + \Delta/2]) \simeq \Delta p(\tilde{\theta})$. Assuming the prior is smooth enough, we write $\Delta p(\tilde{\theta}) \simeq \Delta p(\theta)$. To obtain $-\log p(\mathcal{Y}|\tilde{\theta})$, a second order Taylor expansion is used:

$$\begin{aligned} -\log p(\mathcal{Y}|\tilde{\theta}) \simeq & -\log p(\mathcal{Y}|\theta) - (\theta - \tilde{\theta}) \frac{\partial \log p(\mathcal{Y}|\theta)}{\partial \theta} \\ & - \frac{1}{2}\ (\theta - \tilde{\theta})^2\ \frac{\partial^2 \log p(\mathcal{Y}|\theta)}{\partial \theta^2}. \end{aligned} \tag{22}$$

This can now be used to obtain the expected value of the description length:

$$\begin{aligned} E[\text{Length}(\tilde{\theta}, \mathcal{Y})] \simeq & -\log \Delta - \log p(\theta) - \log p(\mathcal{Y}|\theta) \\ & + \frac{\Delta^2}{24}\ \mathcal{I}(\mathcal{Y}, \theta), \end{aligned} \tag{23}$$

where $\mathcal{I}(\mathcal{Y}, \theta) \equiv -\partial^2 \log p(\mathcal{Y}|\theta)/\partial \theta^2$ is the (observed) Fisher information, and where the two following facts were invoked: $E[\theta - \tilde{\theta}] = 0$ and $E[(\theta - \tilde{\theta})^2] = \Delta^2/12$, both well-known consequences of assuming that the quantization error is uniformly distributed in $[-\Delta/2,\ \Delta/2]$. To find the optimal Δ, the corresponding derivative is set to zero leading to $\Delta_{\text{opt}} = \sqrt{12/\mathcal{I}(\mathcal{Y}, \theta)}$. Inserting Δ_{opt} into (23), and approximating the observed Fisher information by the expected Fisher information $I(\theta) \equiv -E[\partial^2 \log p(\mathcal{Y}|\theta)/\partial \theta^2]$, we finally have the MML criterion for a single parameter (compare with (9)):

$$\begin{aligned} \widehat{\theta} = \arg\min_{\theta} \Big\{ & -\log p(\theta) - \log p(\mathcal{Y}|\theta) \\ & + \frac{1}{2} \log I(\theta) + \frac{1}{2}\ \left(1 + \log \frac{1}{12}\right) \Big\}. \end{aligned} \tag{24}$$

The derivation for the vector parameter case is somewhat more complicated, and we omit most of the details (see [32]). The main difference arises in the definition of the quantization regions, which are no longer simply intervals as in the scalar case. Wallace and Freeman have proposed using optimal quantization lattices. In one dimension, these are simply intervals, in two dimensions, the optimal regions are hexagonal, while in three dimensions they are truncated octahedrons [14]. With Δ denoting the volume of the quantization region (the length of the quantization interval in the scalar case), the mean squared quantization error is now given by

$$E[\| \boldsymbol{\theta} - \tilde{\boldsymbol{\theta}} \|^2] = c\ \kappa_c\ \Delta^{2/c}, \tag{25}$$

where c is the dimension of $\boldsymbol{\theta}$, and κ_c is the so-called *optimal quantizing lattice constant* for $\mathbb{R}^c$ [14]. Due to this difference, the MML criterion for a c-dimensional $\boldsymbol{\theta}$ becomes

$$\begin{aligned} \widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \Big\{ & -\log p(\boldsymbol{\theta}) - \log p(\mathcal{Y}|\boldsymbol{\theta}) \\ & + \frac{1}{2} \log |\mathbf{I}(\boldsymbol{\theta})| + \frac{c}{2}\ (1 + \log \kappa_c) \Big\}. \end{aligned} \tag{26}$$

Notice that, since $\kappa_1 = 1/12 \simeq 0.083(3)$, for $c = 1$ we recover (24). As c grows, κ_c approaches an asymptotic value, $\kappa_c \to (2\pi e)^{-1} \simeq 0.05855$ [14]. Since κ_c does not vary much, we approximate κ_c by $1/12$ (which corresponds to hypercubic quantization regions) [32], finally obtaining (9).

References

[1] J. Banfield and A. Raftery, "Model-Based Gaussian and Non-Gaussian Clustering," *Biometrics*, vol. 49, pp. 803-821, 1993.
[2] H. Bensmail, G. Celeux, A. Raftery, and C. Robert, "Inference in Model-Based Cluster Analysis," *Statistics and Computing*, vol. 7, pp. 1-10, 1997.
[3] J. Bernardo and A. Smith, *Bayesian Theory*. Chichester, UK: J. Wiley & Sons, 1994.
[4] D. Bertsekas, *Nonlinear Programming*. Belmont, Mass.: Athena Scientific, 1999.

[5] C. Biernacki, G. Celeux, and G. Govaert, "Assessing a Mixture Model for Clustering with the Integrated Classification Likelihood," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 7, pp. 719-725, July 2000.

[6] C. Biernacki, G. Celeux, and G. Govaert, "An Improvement of the NEC Criterion for Assessing the Number of Clusters in a Mixture Model," *Pattern Recognition Letters,* vol. 20, pp. 267-272, 1999.

[7] C. Biernacki and G. Govaert, "Using the Classification Likelihood to Choose the Number of Clusters," *Computing Science and Statistics,* vol. 29, pp. 451-457, 1997.

[8] H. Bozdogan, "Choosing the Number of Component Clusters in the Mixture Model Using a New Informational Complexity Criterion of the Inverse-Fisher Information Matrix," *Information and Classification,* O. Opitz, B. Lausen, and R. Klar, eds., pp. 40-54, Springer Verlag, 1993.

[9] M. Brand, "Structure Learning in Conditional Probability Models Via Entropic Prior and Parameter Extinction," *Neural Computation,* vol. 11, pp. 1155-1182, 1999.

[10] J. Campbell, C. Fraley, F. Murtagh, and A. Raftery, "Linear Flaw Detection in Woven Textiles Using Model-Based Clustering," *Pattern Recognition Letters,* vol. 18, pp. 1539-1548, 1997.

[11] G. Celeux, S. Chrétien, F. Forbes, and A. Mkhadri, "A Component-Wise EM Algorithm for Mixtures," Technical Report 3746, INRIA Rhône-Alpes, France, 1999. Available at http://www.inria.fr/RRRT/RR-3746.html.

[12] G. Celeux and G. Soromenho, "An Entropy Criterion for Assessing the Number of Clusters in a Mixture Model," *Classification J.,* vol. 13, pp. 195-212, 1996.

[13] S. Chrétien and A. Hero III, "Kullback Proximal Algorithms for Maximum Likelihood Estimation," *IEEE Trans. Information Theory,* vol. 46, pp. 1800-1810, 2000.

[14] J. Conway and N. Sloane, *Sphere Packings, Lattices, and Groups.* New York: Springer Verlag, 1993.

[15] T. Cover and J. Thomas, *Elements of Information Theory.* New York: John Wiley & Sons, 1991.

[16] S. Dalal and W. Hall, "Approximating Priors by Mixtures of Natural Conjugate Priors," *J. Royal Statistical Soc. (B),* vol. 45, 1983.

[17] A. Dasgupta and A. Raftery, "Detecting Features in Spatial Point Patterns with Clutter Via Model-Based Clustering," *J. Am. Statistical Assoc.,* vol. 93, pp. 294-302, 1998.

[18] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood Estimation from Incomplete Data Via the EM Algorithm," *J. Royal Statistical Soc. B,* vol. 39, pp. 1-38, 1977.

[19] R. Duda and P. Hart, *Pattern Classification and Scene Analysis.* New York: John Wiley & Sons, 1973.

[20] M. Figueiredo and A.K. Jain, "Unsupervised Selection and Estimation of Finite Mixture Models," *Proc. Int'l Conf. Pattern Recognition—ICPR-2000,* pp. 87-90, 2000.

[21] M. Figueiredo, J. Leitão, and A.K. Jain, "On Fitting Mixture Models," *Energy Minimization Methods in Computer Vision and Pattern Recognition,* E. Hancock and M. Pellilo, eds., pp. 54-69, Springer Verlag, 1999.

[22] C. Fraley and A. Raftery, "How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis," Technical Report 329, Dept. Statistics, Univ. Washington, Seattle, WA, 1998.

[23] Z. Ghahramani and M. Beal, "Variational Inference for Bayesian Mixtures of Factor Analyzers," *Advances in Neural Information Processing Systems 12,* S. Solla, T. Leen, and K.-R. Müller, eds., pp. 449-455, MIT Press, 2000.

[24] Z. Ghahramani and G. Hinton, "The EM Algorithm for Mixtures of Factor Analyzers," Technical Report CRG-TR-96-1, Univ. of Toronto, Canada, 1997.

[25] T. Hastie and R. Tibshirani, "Discriminant Analysis by Gaussian Mixtures," *J. Royal Statistical Soc. (B),* vol. 58, pp. 155-176, 1996.

[26] G. Hinton, P. Dayan, and M. Revow, "Modeling the Manifolds of Images of Handwritten Digits," *IEEE Trans. Neural Networks,* vol. 8, pp. 65-74, 1997.

[27] T. Hofmann and J. Buhmann, "Pairwise Data Clustering by Deterministic Annealing," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 1, pp. 1-14, Jan. 1997.

[28] A.K. Jain and R. Dubes, *Algorithms for Clustering Data.* Englewood Cliffs, N.J.: Prentice Hall, 1988.

[29] A.K. Jain, R. Duin, and J. Mao, "Statistical Pattern Recognition: A Review," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 1, pp. 4-38, Jan. 2000.

[30] A.K. Jain and F. Farrokhnia, "Unsupervised Texture Segmentation Using Gabor Filters," *Pattern Recognition,* vol. 24, pp. 1167-1186, 1991.

[31] M. Kloppenburg and P. Tavan, "Deterministic Annealing for Density Estimation by Multivariate Normal Mixtures," *Physical Rev. E,* vol. 55, pp. R2089-R2092, 1997.

[32] A. Lanterman, "Schwarz, Wallace, and Rissanen: Intertwining Themes in Theories of Model Order Estimation," *Int'l Statistical Rev.,* vol. 69, pp. 185-212, Aug. 2001.

[33] G. McLachlan, "On Bootstrapping the Likelihood Ratio Test Statistic for the Number of Components in a Normal Mixture," *J. Royal Statistical Soc. Series (C),* vol. 36, pp. 318-324, 1987.

[34] G. McLachlan, *Discriminant Analyis and Statistical Pattern Recognition.* New York: John Wiley & Sons, 1992.

[35] G. McLachlan and K. Basford, *Mixture Models: Inference and Application to Clustering.* New York: Marcel Dekker, 1988.

[36] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions.* New York: John Wiley & Sons, 1997.

[37] G. McLachlan and D. Peel, *Finite Mixture Models.* New York: John Wiley & Sons, 2000.

[38] P. Meinicke and H. Ritter, "Resolution-Based Complexity Control for Gaussian Mixture Models," *Neural Computation,* vol. 13, no. 2, pp. 453-475, 2001.

[39] K. Mengersen and C. Robert, "Testing for Mixtures: A Bayesian Entropic Approach," *Proc. Fifth Valencia Int'l Meeting Bayesian Statistsics 5,* J. Bernardo, J. Berger, A. Dawid, and F. Smith, eds., pp. 255-276, 1996.

[40] R. Neal, "Bayesian Mixture Modeling," *Proc. 11th Int'l Workshop Maximum Entropy and Bayesian Methods of Statistical Analysis,* pp. 197-211, 1992.

[41] R. Neal and G. Hinton, "A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants," *Learning in Graphical Models,* M.I. Jordan, ed., pp. 355-368, Kluwer Academic Publishers, 1998.

[42] J. Oliver, R. Baxter, and C. Wallace, "Unsupervised Learning Using MML," *Proc. 13th Int'l Conf. Machine Learning,* pp. 364-372, 1996.

[43] P. Pudil, J. Novovicova, and J. Kittler, "Feature Selection Based on the Approximation of Class Densities by Finite Mixtures of the Special Type," *Pattern Recognition,* vol. 28, no. 9, pp. 1389-1398, 1995.

[44] A. Rangarajan, "Self Annealing: Unifying Deterministic Annealing and Relaxation Labeling," *Energy Minimization Methods in Computer Vision and Pattern Recognition,* M. Pellilo and E. Hancock, eds., pp. 229-244, Springer Verlag, 1997.

[45] C. Rasmussen, "The Infinite Gaussian Mixture Model," *Advances in Neural Information Processing Systems 12,* S. Solla, T. Leen, and K.-R. Müller, eds., pp. 554-560, MIT Press, 2000.

[46] S. Raudys and A.K. Jain, "Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, pp. 252-264, 1991.

[47] S. Raudys and V. Pikelis, "On Dimensionality, Sample Size, Classification Error, and Complexity of Classification Algorithms in Pattern Recognition," *IEEE Trans. Pattern Analysis and Machice Intelligence,* vol. 2, pp. 243-252, 1980.

[48] S. Richardson and P. Green, "On Bayesian Analysis of Mixtures with Unknown Number of Components," *J. Royal Statistical Soc. B,* vol. 59, pp. 731-792, 1997.

[49] J. Rissanen, *Stochastic Complexity in Stastistical Inquiry.* Singapore: World Scientific, 1989.

[50] S. Roberts, D. Husmeier, I. Rezek, and W. Penny, "Bayesian Approaches to Gaussian Mixture Modelling," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 11, pp. 1133-1142, Nov. 1998.

[51] K. Roeder and L. Wasserman, "Practical Bayesian Density Estimation Using Mixtures of Normals," *J. Am. Statistical Assoc.,* vol. 92, pp. 894-902, 1997.

[52] K. Rose, "Deterministic Annealing for Clustering, Compression, Classification, Regression, and Related Optimization Problems," *Proc. IEEE,* vol. 86, pp. 2210-2239, 1998.

[53] G. Schwarz, "Estimating the Dimension of a Model, *Annals of Statistics,* vol. 6, pp. 461-464, 1978.

[54] P. Smyth, "Model Selection for Probabilistic Clustering Using Cross-Validated Likelihood," *Statistics and Computing,* vol. 10, no. 1, pp. 63-72, 2000.

[55] R. Streit and T. Luginbuhl, "Maximum Likelihood Training of Probabilistic Neural Networks," *IEEE Trans. Neural Networks,* vol. 5, no. 5, pp. 764-783, 1994.
[56] M. Tipping and C. Bishop, "Mixtures of Probabilistic Principal Component Analyzers," *Neural Computation,* vol. 11, no. 2, pp. 443-482, 1999.
[57] D. Titterington, A. Smith, and U. Makov, *Statistical Analysis of Finite Mixture Distributions.* Chichester, U.K.: John Wiley & Sons, 1985.
[58] N. Ueda and R. Nakano, "Deterministic Annealing EM Algorithm," *Neural Networks,* vol. 11, pp. 271-282, 1998.
[59] N. Ueda, R. Nakano, Z. Gharhamani, and G. Hinton, "SMEM Algorithm for Mixture Models," *Neural Computation,* vol. 12, pp. 2109-2128, 2000.
[60] C. Wallace and D. Dowe, "Minimum Message Length and Kolmogorov Complexity," *The Computer J.,* vol. 42, no. 4, pp. 270-283, 1999.
[61] C. Wallace and P. Freeman, "Estimation and Inference Via Compact Coding," *J. Royal Statistical Soc. (B),* vol. 49, no. 3, pp. 241-252, 1987.
[62] M. Whindham and A. Cutler, "Information Ratios for Validating Mixture Analysis," *J. Am. Statistcal Assoc.,* vol. 87, pp. 1188-1192, 1992.
[63] L. Xu and M. Jordan, "On Convergence Properties of the EM Algorithm for Gaussian Mixtures," *Neural Computation,* vol. 8, pp. 129-151, 1996.
[64] A. Zellner, "Maximal Data Information Prior Distributions," *New Developments in the Applications of Bayesian Methods,* A. Aykac and C. Brumat, eds., pp. 211-232, Amsterdam: North Holland, 1977.

Mário A.T. Figueiredo (S '87-M '95-SM '2000) received the EE, MSc, and PhD degrees in electrical and computer engineering, all from Instituto Superior Técnico (I.S.T., the Engineering School of the Technical University of Lisbon), Lisbon, Portugal, in 1985, 1990, and 1994, respectively. Since 1994, he has been an assistant professor with the Department of Electrical and Computer Engineering, I.S.T. He is also with the Communication Theory and Pattern Recognition Group, Institute of Telecommunications, Lisbon. In 1998, he held a visiting position with the Department of Computer Science and Engineering of the Michigan State University, East Lansing. His scientific interests include image processing and analysis, computer vision, statistical pattern recognition, statistical learning, and information theory. Dr. Figueiredo received the Portuguese IBM Scientific Prize in 1995. He is on the editorial board of the journal *Pattern Recognition Letters* and he is cochair of the 2001 International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition. He is a senior member of the IEEE.

Anil K. Jain (S '70-M '72-SM '86-F '91) is a university distinguished professor with the Department of Computer Science and Engineering, Michigan State University, East Lansing. He served as the department chair from 1995 to 1999. His research interests include statistical pattern recognition, Markov random fields, texture analysis, neural networks, document image analysis, fingerprint matching, and 3D object recognition. He is coauthor of *Algorithm for Clustering Data* (Englewood Cliffs, NJ: Prentice-Hall, 1988), editor of the book *Real-Time Object Measurement and Classification* (Berlin, Germany: Springer Verlag, 1988), and coeditor of the books *Analysis and Interpretation of Range Images* (Berlin, Germany: Springer Verlag, 1989), *Markov Random Fields* (New York: Academic Press, 1992), *Artificial Neural Networks and Pattern Recognition* (Amsterdam, The Netherlands: Elsevier, 1993), *3D Object Recognition* (Amsterdam, The Netherlands: Elsevier, 1993), and *BIOMETRICS: Personal Identification in Networked Society* (Boston, Mass.: Kluwer Academic Publishers, 1999). Dr. Jain received the Best Paper Awards in 1987 and 1991 and certificates of outstanding contributions in 1976, 1979, 1992, and 1997, from the Pattern Recognition Society. He also received the 1996 *IEEE Transactions on Neural Networks* outstanding paper award. He was the editor-in-chief of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1990-1994). He is a fellow of the IEEE and of the International Association of Pattern Recognition. He received a Fullbright Research Award in 1998 and a Guggenheim Fellowship in 2001.