On Fitting Mixture Models

Mário A. T. Figueiredo[1], José M. N. Leitão[1], and Anil K. Jain[2]

[1] Instituto de Telecomunicações, and
Departamento de Engenharia Electrotécnica e de Computadores.
Instituto Superior Técnico, 1049-001 Lisboa, PORTUGAL
E-mail: mtf@lx.it.pt and jleitao@red.lx.it.pt

[2] Department of Computer Science and Engineering
Michigan State University, East Lansing, MI 48824, U.S.A.
E-mail: jain@cse.msu.edu

Abstract. Consider the problem of fitting a finite Gaussian mixture, with an unknown number of components, to observed data. This paper proposes a new minimum description length (MDL) type criterion, termed MMDL (for *mixture* MDL), to select the number of components of the model. MMDL is based on the identification of an "equivalent sample size", for each component, which does not coincide with the full sample size. We also introduce an algorithm based on the standard *expectation-maximization* (EM) approach together with a new agglomerative step, called *agglomerative* EM (AEM). The experiments here reported have shown that MMDL outperforms existing criteria of comparable computational cost. The good behavior of AEM, namely its good robustness with respect to initialization, is also illustrated experimentally.

1 Introduction

Finite mixtures are a flexible and powerful probabilistic modeling tool. In statistical pattern recognition, mixtures allow a formal (model-based) approach to (unsupervised) clustering [7]; in fact, mixtures adequately describe situations where each observation is modeled as having been produced by one of a set of alternative mechanisms [31]. However, strict adherence to this interpretation is not required. Mixtures can simply be seen as models able to represent arbitrarily complex probability density functions (pdf's); this makes them an ideal tool for representing complex class-conditional pdf's in supervised learning scenarios (see[22] and references therein).

This paper is devoted to the problem of fitting Gaussian mixtures with unknown number of components to multivariate observations. The two fundamental issues to be dealt with are: **(a)** how to estimate the number of components, for which several techniques (reviewed below) have been proposed; and **(b)** how to estimate the parameters defining the mixture model. For this second question,

Published in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, E. Hancock and M. Pellilo (Eds.), pp. 54-69, Springer-Verlag, 1999.

the standard answer is the *expectation-maximization* (EM) algorithm, but several authors have also advocated the (much more computationally demanding) *Markov chain Monte-Carlo* (MCMC) method.

We propose a new criterion to estimate the number of components which is shown experimentally to outperform existing methods of comparable computational cost. Our criterion is a modified version of the *minimum description length* (MDL) principle, based on what can be called the "equivalent sample size". We also introduce an (EM-based) algorithm aimed at mitigating the initialization dependence that makes EM difficult to use in practice. From a clustering perspective, our algorithm can be seen as an agglomerative hierarchical-type scheme, thus we term it *agglomerative* EM (AEM): we start with a large number of components (clusters) and evolve towards a small number of components. From a density estimation perspective, our algorithm has a multi-scale flavor: we go from a fine-scale representation with a large number of components, thus potentially irregular, to a smoother/coarser one with fewer components.

We review relevant previous work on mixture model fitting in Section 2, which also serves to introduce notation and the EM algorithm. Section 3 presents the MMDL criterion, while Section 4 is devoted to AEM. Section 5 reports experimental results, and Section 6 presents our conclusions.

2 Fitting Mixture Models

2.1 Introduction

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

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

where k is the number of components, each $f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta}_m)$ is called a component density function, and the α_m ($\sum_{m=1}^{k} \alpha_m = 1$) are the mixing probabilities. Assuming that all the components have the same functional form (*e.g.*, they are all d-variate Gaussian), each one is fully characterized by the parameter vector $\boldsymbol{\theta}_i$. Let $\boldsymbol{\Theta}_{(k)} = \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_k, \alpha_1, ..., \alpha_{k-1}\}$ be the parameter set defining a given mixture (notice that $\alpha_k = 1 - \sum_{m=1}^{k-1} \alpha_m$), and $\mathcal{M}_{(k)}$ be the space of all possible k-component mixtures built from a certain class of pdf's. This paper focuses on mixtures of Gaussian components, denoted as $f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta}_m) = \mathcal{N}(\mathbf{y}|\boldsymbol{\mu}_m, \mathbf{C}_m)$, where $\boldsymbol{\theta}_m = (\boldsymbol{\mu}_m, \mathbf{C}_m)$, if arbitrary covariance $\mathbf{C}_m$ and mean $\boldsymbol{\mu}_m$ are assumed; if a common covariance $\mathbf{C}$ is adopted, we simply write $\boldsymbol{\theta}_m = \boldsymbol{\mu}_m$.

The maximum likelihood (ML) estimate of $\boldsymbol{\Theta}_{(k)}$, based on a set of n independent observations $\mathbf{y}_{\text{obs}} = \{\mathbf{y}^{(1)}, ..., \mathbf{y}^{(n)}\}$, is

$$\widehat{\boldsymbol{\Theta}}_{(k)} = \arg\max_{\boldsymbol{\Theta}_{(k)}} L\left(\boldsymbol{\Theta}_{(k)}, \mathbf{y}_{\text{obs}}\right), \tag{2}$$

where $L\left(\boldsymbol{\Theta}_{(k)}, \mathbf{y}_{\text{obs}}\right)$ is the log-likelihood function

$$L\left(\boldsymbol{\Theta}_{(k)}, \mathbf{y}_{\text{obs}}\right) = \log \prod_{i=1}^{n} f_{\mathbf{Y}}(\mathbf{y}^{(i)}|\boldsymbol{\Theta}_{(k)}) = \sum_{i=1}^{n} \log \sum_{m=1}^{k} \alpha_m f_{\mathbf{Y}}(\mathbf{y}^{(i)}|\boldsymbol{\theta}_m). \quad (3)$$

In general, Eq. (2) has no closed-form solution but it can be approached quite easily via the *expectation-maximization* (EM) algorithm [16], [31].

2.2 The EM Algorithm for Gaussian Mixtures

Behind the EM algorithm is the interpretation of the set of observations $\mathbf{y}_{\text{obs}} = \{\mathbf{y}^{(1)}, ..., \mathbf{y}^{(n)}\}$ as incomplete data, with the missing information being a corresponding set of labels $\mathbf{z}_{\text{miss}} = \{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(n)}\}$ [16], [31]. Each of these labels has the form $\mathbf{z}^{(i)} = [z_1^{(i)}, ..., z_k^{(i)}]^T$, with $z_m^{(i)} = 1$ and $z_p^{(i)} = 0$, for $p \neq m$, if and only if $\mathbf{y}^{(i)}$ was produced by the m-th component of the mixture. This complete data setup agrees with the interpretation of a mixture density as a model of a two-step data generation process: first, randomly choose one of the k available "data generators" with probabilities $\{\alpha_1, ..., \alpha_k\}$; then, produce a sample from the chosen "generator".

The loglikelihood function based on the complete data $\{\mathbf{y}_{\text{obs}}, \mathbf{z}_{\text{miss}}\}$, denoted $L_c(\boldsymbol{\Theta}_{(k)}, \mathbf{y}_{\text{obs}}, \mathbf{z}_{\text{miss}})$, is easily found to be (for details see [31])

$$L_c\left(\boldsymbol{\Theta}_{(k)}, \mathbf{y}_{\text{obs}}, \mathbf{z}_{\text{miss}}\right) = \sum_{j=1}^{n} \sum_{m=1}^{k} z_m^{(j)} \log \left[\alpha_m f_{\mathbf{Y}}(\mathbf{y}^{(j)}|\boldsymbol{\theta}_m)\right]. \quad (4)$$

In its general form, the EM algorithm proceeds by successively applying two steps to produce a sequence of parameter estimates $\{\widehat{\boldsymbol{\Theta}}_{(k)}^{(1)}, \widehat{\boldsymbol{\Theta}}_{(k)}^{(2)}, ..., \widehat{\boldsymbol{\Theta}}_{(k)}^{(t)}, ...\}$:

E-step: Compute the expected value of the complete loglikelihood, conditioned on the observed data and on the current parameter estimate $\widehat{\boldsymbol{\Theta}}_{(k)}^{(t)}$,

$$Q\left(\boldsymbol{\Theta}_{(k)}, \widehat{\boldsymbol{\Theta}}_{(k)}^{(t)}\right) = \int L_c\left(\boldsymbol{\Theta}_{(k)}, \mathbf{y}_{\text{obs}}, \mathbf{z}_{\text{miss}}\right) f_{\mathbf{Z}_{\text{miss}}}(\mathbf{z}_{\text{miss}}|\widehat{\boldsymbol{\Theta}}_{(k)}^{(t)}, \mathbf{y}_{\text{obs}})\, d\mathbf{z}_{\text{miss}}.$$

M-step: Update the parameter estimates according to

$$\widehat{\boldsymbol{\Theta}}_{(k)}^{(t+1)} = \arg\max_{\boldsymbol{\Theta}_{(k)}} Q\left(\boldsymbol{\Theta}_{(k)}, \widehat{\boldsymbol{\Theta}}_{(k)}^{(t)}\right). \quad (5)$$

Under mild conditions [16], EM converges to a (local) maximum of $L\left(\boldsymbol{\Theta}_{(k)}, \mathbf{y}_{\text{obs}}\right)$.

The key to the efficient implementation of this algorithm is the choice of an observed/missing data structure, *i.e.*, the function $L_c\left(\boldsymbol{\Theta}_{(k)}, \mathbf{y}_{\text{obs}}, \mathbf{z}_{\text{miss}}\right)$, such that the E and M steps have simple closed-form expressions. This is the case in

Eq. (4), which is linear in the missing variables, thus reducing the E-step to the computation of the conditional expectation of the $z_m^{(i)}$ variables [16], [31],

$$w_m^{(i,t)} \equiv E\left[z_m^{(i)} | \widehat{\boldsymbol{\Theta}}_{(k)}^{(t)}, \mathbf{y}_{\text{obs}}\right] = \frac{\widehat{\alpha}_m^{(t)} f_{\mathbf{Y}}(\mathbf{y}^{(i)} | \widehat{\boldsymbol{\theta}}_m^{(t)})}{\sum_{j=1}^{k} \widehat{\alpha}_j^{(t)} f_{\mathbf{Y}}(\mathbf{y}^{(i)} | \widehat{\boldsymbol{\theta}}_j^{(t)})}. \tag{6}$$

The M-step also has a simple closed form solution (recall that $\boldsymbol{\theta}_m = \{\boldsymbol{\mu}_m, \mathbf{C}_m\}$):

$$\widehat{\alpha}_m^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} w_m^{(i,t)} \tag{7}$$

$$\widehat{\boldsymbol{\mu}}_m^{(t+1)} = \left(\sum_{i=1}^{n} w_m^{(i,t)}\right)^{-1} \sum_{i=1}^{n} \mathbf{y}^{(i)}\, w_m^{(i,t)} \tag{8}$$

$$\widehat{\mathbf{C}}_m^{(t+1)} = \left(\sum_{i=1}^{n} w_m^{(i,t)}\right)^{-1} \sum_{i=1}^{n} \left(\mathbf{y}^{(i)} - \widehat{\boldsymbol{\mu}}_m^{(t+1)}\right)\left(\mathbf{y}^{(i)} - \widehat{\boldsymbol{\mu}}_m^{(t+1)}\right)^T w_m^{(i,t)}. \tag{9}$$

The main difficulties in using EM for mixture model fitting are: its critical dependence on initialization; the possibility of convergence to a point on the boundary of the parameter space with unbounded likelihood (*i.e.*, one of the α_m parameters approaching zero with the corresponding covariance becoming arbitrarily close to singular).

2.3 Estimating the Number of Components

It is well known that the ML criterion can not be used to estimate the number of mixture components because $\mathcal{M}_{(k)} \subseteq \mathcal{M}_{(k+1)}$; for example, $\boldsymbol{\Theta}_{(k)} = \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_k, \alpha_1, ..., \alpha_{k-1}\}$ and $\boldsymbol{\Theta}'_{(k+1)} = \{\boldsymbol{\theta}'_1, ..., \boldsymbol{\theta}'_k, \boldsymbol{\theta}'_{k+1}, \alpha'_1, ..., \alpha'_{k-1}, \alpha'_k\}$, such that $\boldsymbol{\theta}_k = \boldsymbol{\theta}'_k = \boldsymbol{\theta}'_{k+1}$ and $\alpha_k = \alpha'_{k+1} + \alpha'_k$ (where, of course, $\alpha_k = 1 - \sum_{j=1}^{k-1} \alpha_j$, and $\alpha'_{k+1} = 1 - \sum_{j=1}^{k} \alpha'_j$) represent intrinsically indistinguishable mixture densities. Consequently, the maximized likelihood $L(\widehat{\boldsymbol{\Theta}}_{(k)}, \mathbf{y}_{\text{obs}})$ is a non-decreasing function of k, thus useless as a model selection criterion. This is a particular instance of the *identifiability* problem (see, *e.g.*, [31]). As also pointed out in [31], classical (χ^2 based) hypothesis testing is not useful here because the necessary regularity conditions are not met.

Several approaches are available to estimate the number of components of a mixture; from an algorithmic standpoint, they can be divided into two main classes: EM-based techniques and stochastic techniques.

EM-based approaches use the (fixed k) EM algorithm to obtain a sequence of parameter estimates for a range of values of k, $\{\widehat{\boldsymbol{\Theta}}_{(k)},\ k = k_{\min}, ..., k_{\max}\}$, with the estimate of k being defined as the minimizer of some cost function,

$$\widehat{k} = \arg\min_k \left\{\mathcal{C}\left(\widehat{\boldsymbol{\Theta}}_{(k)}, k\right),\ k = k_{\min}, ..., k_{\max}\right\}. \tag{10}$$

Most often, this cost function includes the maximized log-likelihood function plus an additional term whose role is to penalize large values of k.

Under this general formulation, we find the MDL criterion [23] in which the cost function is

$$\mathcal{C}_{\text{MDL}}\left(\widehat{\boldsymbol{\Theta}}_{(k)}, k\right) = -L\left(\widehat{\boldsymbol{\Theta}}_{(k)}, \mathbf{y}_{\text{obs}}\right) + \frac{N(k)}{2}\log n, \tag{11}$$

where $N(k)$ is the number of parameters needed to specify a k-component mixture. For arbitrary means and covariances, $N(k) = (k-1) + k(d + d(d+1)/2)$ (recall that d is the dimension of $\mathbf{Y}$); if a common covariance is assumed, then $N(k) = (k-1) + kd + d(d+1)/2$.

Several EM-based approaches also use approximate versions of the *Bayes factor* (the correct Bayesian model selection criterion [9]), such as the evidence-based Bayesian (EBB) criterion [25], the *approximate weight of evidence* (AWE) [1], and Schwarz's *Bayesian inference criterion* (BIC) [5]. Although derived in a different framework, BIC formally coincides with MDL and is also given by Eq. (11). The *minimum message length* (MML) criterion [20], Akaike's *information criterion* (AIC) [35], and Bezdek's *partition coefficient* (PC) [3] are other approaches in this class. As pointed out in [25], EBB, MDL/BIC, and MML perform comparably and outperform all other methods against which they were tested. Concerning AWE, it is argued in [5] that MDL/BIC provides a better approximation to the true Bayes factor. The AIC and PC criteria were shown in [20] (based on tests on 20 different mixtures) to be outperformed by MML and MDL/BIC. Accordingly, any new method in this class need only be compared against EBB, MDL/BIC, or MML. Finally, drawbacks of MML and EBB are: MML can not be used for certain values of d (for example $d = 9$ and $d > 24$) [25]; both EBB and MML depend on arbitrarily chosen parameters which can critically influence its results.

Resampling-based schemes [14] (which have also been used in a clustering framework [8]) and cross-validation approaches [30] are (computationally) much closer to stochastic techniques (see below) than to the methods in the previous paragraph and will not be further considered here.

Stochastic approaches involve Markov chain Monte Carlo (MCMC) sampling and are far more computationally intensive than EM. MCMC is used in two different ways: to implement model selection criteria to actually estimate k (*e.g.*, [2], [18], [26]); and, in a more "fully Bayesian" way, to sample from the full *a posteriori* distribution with k considered unknown [19], [21]. Despite their formal appeal, we think that MCMC-based techniques are still far too computationally demanding to be useful in pattern recognition applications. For example, tests reported in [21], using small samples ($n = 245, 155, 82$) of univariate data, require 100000 MCMC sweeps following a so-called *burn-in* period of another 100000 sweeps; this is a huge amount of computation for such small problems.

2.4 Initialization of EM

The EM algorithm requires an initial parameter setting $\widehat{\boldsymbol{\Theta}}_{(k)}^{(1)}$ or an initial association of each observation to one of the components (*i.e.*, an initial setting of $w_m^{(i,1)}$) [16], [31]. This is a critical issue because EM converges to a local maximum of the likelihood function: the final estimate depends on the initialization. There are several different approaches to deal with this difficulty. Running EM several times, from random initializations, and then choosing the final estimate that leads to the highest local maximum of the likelihood is a commonly used technique (*e.g.*, [17] and [25]). Another common procedure is to use some clustering method to provide an initial partition of the data [17]. Finally, we mention the deterministic annealing (DA) EM algorithm (DAEM); DA is a fast surrogate of the (stochastic) simulated annealing approach to global optimization, which has been successfully applied in several problems [27]. In particular, for mixture estimation, DAEM avoids some of the initialization dependence of EM [10], [32], [36]. All these choices pay a high price in terms of computational efficiency.

3 The MMDL Criterion

It was shown in [25] that MDL/BIC (although simpler) performs comparably with EBB and MML, although it sometimes slightly underestimates the true k. A similar conclusion can be obtained from the many (20) tests described in [20]. It was also reported in [11] and [29] that MDL/BIC tends to slightly underestimate the true order. In order to overcome this problem, let us look again at the MDL criterion in Eq. (11). The meaning of the MDL cost function is the total code length of a two-part code for the observed data $\mathbf{y}_{\text{obs}}$ and the parameter estimate $\widehat{\boldsymbol{\Theta}}_{(k)}$ (see [23], for details and motivation): first encode the data, given $\widehat{\boldsymbol{\Theta}}_{(k)}$; then, encode $\widehat{\boldsymbol{\Theta}}_{(k)}$. Formally, Eq. (11) is of the form

$$\begin{aligned} \mathcal{C}_{\text{MDL}}\left(\mathbf{y}_{\text{obs}}, \widehat{\boldsymbol{\Theta}}_{(k)}\right) &= \mathcal{L}\left(\mathbf{y}_{\text{obs}}, \widehat{\boldsymbol{\Theta}}_{(k)}\right) \\ &= \mathcal{L}\left(\mathbf{y}_{\text{obs}} | \widehat{\boldsymbol{\Theta}}_{(k)}\right) + \mathcal{L}\left(\widehat{\boldsymbol{\Theta}}_{(k)}\right), \end{aligned} \tag{12}$$

where $\mathcal{L}(\mathbf{y}_{\text{obs}}|\widehat{\boldsymbol{\Theta}}_{(k)}) = -L(\widehat{\boldsymbol{\Theta}}_{(k)}, \mathbf{y}_{\text{obs}})$ is the well-known Shannon's optimal code length[1]. The second code-length, $\mathcal{L}(\widehat{\boldsymbol{\Theta}}_{(k)})$, results from the following reasoning. To obtain finite-length codewords for $\widehat{\boldsymbol{\Theta}}_{(k)}$, its (real-valued) elements are truncated to some finite precision. With a coarse precision, $\mathcal{L}(\widehat{\boldsymbol{\Theta}}_{(k)})$ is small but the encoded parameters may be far from the optimal ones and so the first part of the code may become longer. With a finer resolution, the encoded parameters will be close to the optimal ones, but longer codewords are required. As shown

[1] As is usually done, we are ignoring the integer constraint on code-lengths and disregarding that we are dealing with densities, not probability masses. Discretization would lead to probability masses and a common (thus irrelevant) additional code length term [23].

in [23], the optimal code-length for each real parameter, asymptotically for large n, is $(1/2)\log n$; this leads to Eq. (11).

In most problems where the MDL/BIC criterion is used, all data points have equal importance in estimating each component of the parameter vector. This is not the case in mixtures, where each data point has its own weight in estimating different parameters, as is clear from Eqs. (8) and (9). This fact is revealed if we compute the Fisher information of a parameter of the m-th mode of the mixture (denoted generically as θ_m) which leads to (see [31])

$$I(\theta_m) = n\,\alpha_m\,I_1(\theta_m), \tag{13}$$

where $I_1(\theta_m)$ denotes the Fisher information associated with a single observation known to have been produced by the m-th component density, *i.e.*,

$$I_1(\theta_m) = -E\left[\frac{\partial^2}{\partial\theta_m^2}\log f_Y\left(y|\theta_m\right)\right].$$

What Eq. (13) shows is that a parameter θ_m "sees" an *equivalent sample size* equal to $n\alpha_m$, rather than n. This is intuitively acceptable because θ_m will basically be estimated from the data that "was generated" by the m-th component of the mixture; the expected amount of this data is precisely $n\,\alpha_m$. Applying this fact, while keeping the classical MDL code-length for the mixing probabilities (because these are estimated from all the data), we finally obtain the MMDL cost function

$$\begin{aligned}\mathcal{C}_{\text{MMDL}}\left(\widehat{\boldsymbol{\Theta}}_{(k)},k\right) &= -L\left(\widehat{\boldsymbol{\Theta}}_{(k)},\mathbf{y}_{\text{obs}}\right)+\frac{k-1}{2}\log n+\frac{N(1)}{2}\sum_{m=1}^{k}\log\left(n\alpha_m\right)\\ &= -L\left(\widehat{\boldsymbol{\Theta}}_{(k)},\mathbf{y}_{\text{obs}}\right)+\frac{N(k)}{2}\log n+\underbrace{\frac{N(1)}{2}\sum_{m=1}^{k}\log\alpha_m}_{<\,0}\end{aligned} \tag{14}$$

where $N(1)$ is the number of real parameters defining each component (see the paragraph after Eq. (11)). The MMDL cost function can also be interpreted from a BIC-type perspective as the inclusion of some of the $o(1)$ terms that are dropped to obtain the classical form.

In summary, the MMDL criterion introduces a lower penalty than MDL/BIC; notice that the new term that appears in Eq. (14) when compared with Eq. (11) is necessarily negative. This is a result of the identification of the amount of data which is effectively used in estimating the parameters of each component of the mixture.

4 The Algorithm

To implement the MMDL criterion we propose a new (EM-based) algorithm. Let $k_{\max}$ be some number known to be considerably larger than the true/optimal k

(say, k_{true}) and $k_{\min}$ be another number such that, for sure, $k_{\min} < k_{\text{true}}$. The basic structure of the algorithm is as follows:

Initialization:

Set $k \leftarrow k_{\max}$.

Let $\widehat{\boldsymbol{\Theta}}_{(k)}^{(1)}$ be some initial k-component mixture estimate.

Main Loop:

While $k \geq k_{\min}$, repeat:

- Run EM, using $\widehat{\boldsymbol{\Theta}}_{(k)}^{(1)}$ as initialization, until a stopping condition is met. Store the resulting mixture parameter estimate $\widehat{\boldsymbol{\Theta}}_{(k)}$.
- Compute and store $\mathcal{C}_{\text{MMDL}}\left(\widehat{\boldsymbol{\Theta}}_{(k)}, k\right)$.
- Obtain a $(k-1)$-component mixture, "close" (in a sense to be specified below) to the k-component one specified by $\widehat{\boldsymbol{\Theta}}_{(k)}$.
  Let $\widehat{\boldsymbol{\Theta}}_{(k-1)}^{(1)}$ represent this $(k-1)$-component mixture.
- Set $k \leftarrow k-1$

Choosing the optimal k:

Find the minimum of the stored MMDL cost function values:

$$\widehat{k}_{\text{MMDL}} = \arg\min_{k} \left\{ \mathcal{C}_{\text{MMDL}}\left(\widehat{\boldsymbol{\Theta}}_{(k)}, k\right),\ k = k_{\min}, k_{\min}+1, ..., k_{\max} \right\}.$$

The final mixture parameter estimate is $\widehat{\boldsymbol{\Theta}}_{(\widehat{k}_{\text{MMDL}})}$.

The crucial aspect of the algorithm is the use of a $(k-1)$-component mixture, "close" to the current k-component one, to initialize the next run of EM. This is done by looking for the pair of components that are closer to each other and less probable and merge them into a single new component (see details below). For this reason, our algorithm shares some of the spirit of agglomerative hierarchical clustering schemes [7], thus we call it *agglomerative* EM (AEM). The first run of EM, due to the excessive number of components, is somewhat insensitive to initialization. Of course we are not claiming that AEM is guaranteed to find the globally optimal mixture estimate; it is known that even MCMC may have difficulties escaping from local maxima of the likelihood function [24].

AEM can be used with any criterion other than MMDL, or even when k_{true} is known: in this case, simply set $k_{\min} = k_{\text{true}}$ and skip the phase where the optimal k is chosen. Naturally, AEM can also be based on modified versions of EM [16]. Finally, observe that the computational requirements of AEM are the minimum possible for any EM-based method doing unknown order mixture fitting. EM only has to be applied once for each value of k, instead of the common approach of using a set of random initializations for each k.

4.1 Initialization.

For low dimensions ($d = 1, 2$), the initial mixture is composed of $k_{\max}$ components uniformly spread over the region occupied by the observed data (defined

by the minimum and maximum observed values of each coordinate). For higher dimensions, a better initialization is obtained by clustering the data into $k_{\max}$ groups using successive binary splitting and K-means optimization at each stage [7]. As long as $k_{\max}$ is large enough, AEM is quite insensitive to initialization.

4.2 Stopping Conditions for EM.

Each run of EM is stopped if at least one of the following two conditions is true:

$$\text{Condition 1:} \begin{cases} \max\left\{ \dfrac{\| \widehat{\boldsymbol{\mu}}_m^{(t)} - \widehat{\boldsymbol{\mu}}_m^{(t-1)} \|}{\| \widehat{\boldsymbol{\mu}}_m^{(t)} \|},\ m = 1, 2, ..., k \right\} < \delta_\mu \\ \text{and} \\ \max\left\{ \dfrac{\| \widehat{\mathbf{C}}_m^{(t)} - \widehat{\mathbf{C}}_m^{(t-1)} \|}{\| \widehat{\mathbf{C}}_m^{(t)} \|},\ m = 1, 2, ..., k \right\} < \delta_C \end{cases} \tag{15}$$

$$\text{Condition 2: } \min\left\{ \widehat{\alpha}_m^{(t)},\ m = 1, 2, ..., k \right\} < \alpha_{\min}. \tag{16}$$

Condition 1 checks if consecutive parameter estimates do not differ significantly; in all the examples below, we set $\delta_\mu = \delta_C = 0.001$ and use infinity norms $\| \cdot \|_\infty$. Condition 2 looks for a component whose probability is becoming too small; we typically use $\alpha_{\min} = 5\, d/n$. Condition 2 avoids one of the known problems of EM mentioned earlier (convergence to the boundary of the parameter space).

4.3 Obtaining the $(k-1)$-Component Mixture.

The $(k-1)$-component mixture is obtained by merging two components of the k-component one. We start by locating the pair of mixture components, say m_1 and m_2, that are closer to each other and, simultaneously, less probable. Specifically, we choose m_1 and m_2 as

$$(m_1, m_2) = \arg\min_{(i,j)} \left\{ (\widehat{\alpha}_i + \widehat{\alpha}_j)\, \mathcal{D}_s \left[f_{\mathbf{Y}}(\mathbf{y}|\widehat{\boldsymbol{\theta}}_i), f_{\mathbf{Y}}(\mathbf{y}|\widehat{\boldsymbol{\theta}}_j) \right],\ i \neq j \right\}, \tag{17}$$

where $\mathcal{D}_s[f_{\mathbf{Y}}(\mathbf{y}|\widehat{\boldsymbol{\theta}}_i), f_{\mathbf{Y}}(\mathbf{y}|\widehat{\boldsymbol{\theta}}_j)]$ is the *symmetric Kullback-Leibler* (KL) *divergence* [12], the standard dissimilarity measure between probability densities [12]. The Jensen-Shannon divergence (see [13]) would be a natural candidate, because it allows weighting differently the two probability functions being compared; however, it does not have a closed form expression for Gaussian densities and so we settled for the KL divergence. In the Gaussian case, the symmetric KL divergence is [12]:

$$\begin{aligned} \mathcal{D}_s \left[\mathcal{N}(\mathbf{y}|\boldsymbol{\mu}_i, \mathbf{C}_i), \mathcal{N}(\mathbf{y}|\boldsymbol{\mu}_j, \mathbf{C}_j) \right] = & \frac{1}{2} \operatorname{tr}\left[(\mathbf{C}_i - \mathbf{C}_j)\left(\mathbf{C}_j^{-1} - \mathbf{C}_i^{-1} \right) \right] \\ & + \frac{1}{2} \left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j \right)^T \left[\mathbf{C}_i^{-1} + \mathbf{C}_j^{-1} \right]^{-1} \left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j \right). \end{aligned}$$

If EM was stopped by Condition 2 (Eq. (16)), we force m_1 to be the component responsible for making it true. We then choose m_2 by Eq. (17), fixing $i = m_1$.

Consider now the sub-mixture $\alpha'_{m_1} f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta}_{m_1}) + \alpha'_{m_2} f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta}_{m_2})$, where $\alpha'_{m_1} = \alpha_{m_1}/(\alpha_{m_1} + \alpha_{m_2})$ and $\alpha'_{m_2} = 1 - \alpha'_{m_1}$. Merging the two components of this submixture is equivalent to finding the parameters $\boldsymbol{\mu}^*$ and $\mathbf{C}^*$ of the "closest" Gaussian density. If "closeness" is taken in the KL sense, then

$$(\boldsymbol{\mu}^*, \mathbf{C}^*) = \arg\min_{\boldsymbol{\mu},\mathbf{C}} \mathcal{D}\left[\alpha'_{m_1}\mathcal{N}(\mathbf{y}|\boldsymbol{\mu}_{m_1}, \mathbf{C}_{m_1}) + \alpha'_{m_2}\mathcal{N}(\mathbf{y}|\boldsymbol{\mu}_{m_1}, \mathbf{C}_{m_1}), \mathcal{N}(\mathbf{y}|\boldsymbol{\mu}, \mathbf{C})\right],$$

which has a simple solution (see [34], Chapp. 12): $\boldsymbol{\mu}^*$ and $\mathbf{C}^*$ are the global mean and covariance of the given two-component mixture, *i.e.*,

$$\boldsymbol{\mu}^* = \alpha'_{m_1}\boldsymbol{\mu}_{m_1} + \alpha'_{m_2}\boldsymbol{\mu}_{m_2} \tag{18}$$

$$\mathbf{C}^* = \alpha'_{m_1}(\mathbf{C}_{m_1} + \boldsymbol{\mu}_{m_1}\boldsymbol{\mu}_{m_1}^T) + \alpha'_{m_2}(\mathbf{C}_{m_2} + \boldsymbol{\mu}_{m_2}\boldsymbol{\mu}_{m_2}^T) - \boldsymbol{\mu}^*\boldsymbol{\mu}^{*T}. \tag{19}$$

This means that when merging components m_1 and m_2 of the mixture, the resulting component must retain the combined probability, mean, and covariance. Assume, without loss of generality, that $m_2 = k$, which can always be achieved by resorting the components. Merging component $m_1 < k$ and $m_2 = k$ of the k-component mixture given by $\{\alpha_m, \boldsymbol{\mu}_m, \mathbf{C}_m,\ m = 1, ..., k\}$ then yields a $(k-1)$-component mixture defined by $\{\alpha'_m, \boldsymbol{\mu}'_m, \mathbf{C}'_m,\ m = 1, ..., k-1\}$, where

$$\alpha'_m = \begin{cases} \alpha_m, & m \neq m_1 \\ \alpha_{m_1} + \alpha_{m_2}, & m = m_1, \end{cases}$$

$$\boldsymbol{\mu}'_m = \begin{cases} \boldsymbol{\mu}_m, & m \neq m_1 \\ \dfrac{\alpha_{m_1}\boldsymbol{\mu}_{m_1} + \alpha_{m_2}\boldsymbol{\mu}_{m_2}}{\alpha_{m_1} + \alpha_{m_2}}, & m = m_1, \end{cases}$$

$$\mathbf{C}'_m = \begin{cases} \mathbf{C}_m, & m \neq m_1 \\ \dfrac{\alpha_{m_1}(\mathbf{C}_{m_1} + \boldsymbol{\mu}_{m_1}\boldsymbol{\mu}_{m_1}^T) + \alpha_{m_2}(\mathbf{C}_{m_2} + \boldsymbol{\mu}_{m_2}\boldsymbol{\mu}_{m_2}^T)}{\alpha_{m_1} + \alpha_{m_2}} - \boldsymbol{\mu}'_{m_1}\boldsymbol{\mu}'^T_{m_1}, & m = m_1. \end{cases}$$

5 Experimental Results

This section is divided into two parts: the first one basically illustrates the working of the AEM algorithm showing how it evolves from a redundant mixture to successively lower order ones, and how this avoids the need for careful initialization. The second part focuses on MMDL by presenting examples (with synthetic and real data) where it overcomes the under-fitting tendency of MDL/BIC.

5.1 The AEM Algorithm

The first example uses 1000 samples from a mixture of 3 univariate Gaussians with means $\mu_1 = \mu_2 = 0$, and $\mu_3 = 6$, and standard deviations $\sigma_1 = 1$, $\sigma_2 = \sqrt{6}$, and $\sigma_3 = 1$; mixing probabilities are $\alpha_1 = 0.3$, $\alpha_2 = 0.4$, and $\alpha_3 = 0.3$. Fig. 1 shows AEM evolving from an 8-component mixture (after starting at $k_{\max} = 12$) to just two components. Observe the above mentioned multi-scale flavor of the method in the evolution from more erratic density estimates to smoother ones.

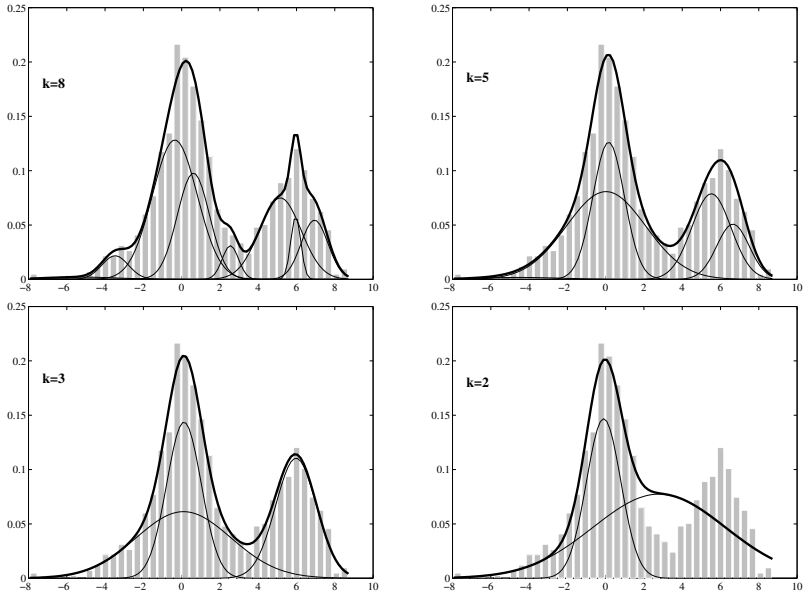

Fig. 1. Mixture estimates for $k = 8, 5, 3$ (the true value), and 2, obtained by AEM. Thin lines show the component densities multiplied by the corresponding probabilities, while the thick line plots the resulting mixture. The gray bars represent a (normalized) histogram of the observations.

The MMDL estimates are $\widehat{k} = 3$, $\widehat{\mu}_1 = 0.09$, $\widehat{\mu}_2 = 0.11$, $\widehat{\mu}_3 = 5.97$, $\widehat{\sigma}_1 = \sqrt{0.87}$, $\widehat{\sigma}_2 = \sqrt{6.12}$, $\widehat{\sigma}_3 = \sqrt{1.11}$, $\widehat{\alpha}_1 = 0.32$, $\widehat{\alpha}_2 = 0.38$, and $\widehat{\alpha}_3 = 0.30$.

For the next example, 1500 samples were drawn from a mixture of 3 bivariate Gaussians with $\alpha_1 = \alpha_3 = 0.3$, $\alpha_2 = 0.4$, $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = [-4,\ -4]^T$, $\boldsymbol{\mu}_3 = [3,\ 3]^T$,

$$\mathbf{C}_1 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}, \quad \mathbf{C}_2 = \begin{bmatrix} 6 & -2 \\ -2 & 6 \end{bmatrix}, \text{ and } \mathbf{C}_3 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}.$$

Fig. 2 shows the algorithm evolving from its initialization (a set of $k_{\max} = 9$ similar and uniformly spread Gaussians) to the correct 3-component mixture. Notice how different the initial mixture is from the true one and how AEM was able to overcome this poor initialization. The final parameter estimates are $\widehat{\boldsymbol{\mu}}_1 = [-4.03,\ -4.12]^T$, $\widehat{\boldsymbol{\mu}}_2 = [-4.01,\ -3.90]^T$, $\widehat{\boldsymbol{\mu}}_3 = [3.08,\ 2.91]^T$,

$$\widehat{\mathbf{C}}_1 = \begin{bmatrix} 1.07 & 0.56 \\ 0.56 & 0.88 \end{bmatrix}, \quad \widehat{\mathbf{C}}_2 = \begin{bmatrix} 5.4 & -1.89 \\ -1.89 & 6.12 \end{bmatrix}, \text{ and } \widehat{\mathbf{C}}_3 = \begin{bmatrix} 2.10 & -1.14 \\ -1.14 & 2.17 \end{bmatrix}.$$

Finally, we study the well known IRIS data set[2] that consists of 50 (4-dimensional) samples of each of the three classes present: *Versicolor*, *Virginica*, and *Setosa*. Starting with $k_{\max} = 8$, both MMDL and MDL/BIC correctly selected $\widehat{k} = 3$. Using the corresponding parameter estimates to build a *maximum*

[2] Available, *e.g.*, at `http://www.ics.uci.edu/pub/machine-learning-databases/`

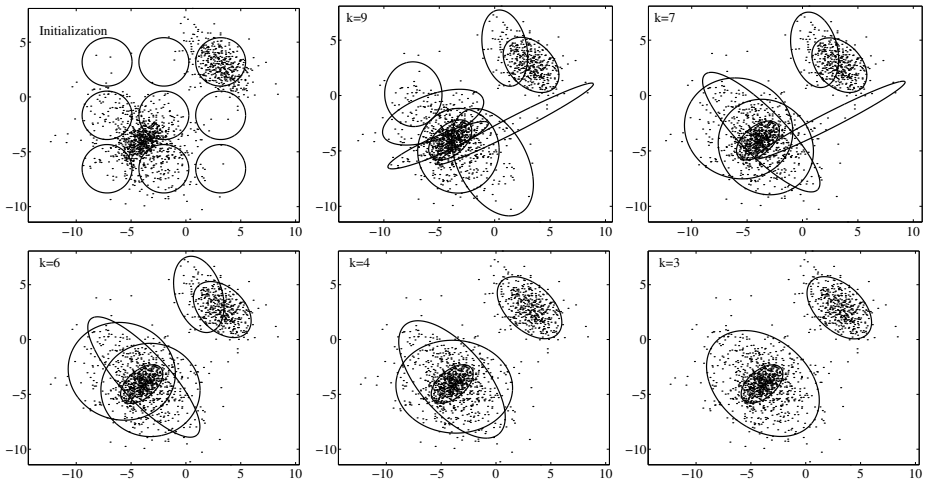

Fig. 2. Initialization and sequence of mixture estimates for $k = 9, 7, 6, 4$, and 3 (the ellipses are isodensity curves of each component).

a posteriori classifier according to

$$\widehat{m}(\mathbf{y}^{(i)}) = \arg\max_{m} \left\{ \widehat{\alpha}_m \; f_{\mathbf{Y}}(\mathbf{y}^{(i)} | \widehat{\boldsymbol{\theta}}_m) \right\},$$

we find that only two samples get misclassified (one *Versicolor* is classified as *Virginica* and one *Virginica* as *Versicolor*). This is even a little better than the three errors reported in [25]; more importantly, it is obtained without multiple random starts of EM.

5.2 Comparing MMDL versus MDL/BIC

Univariate Data. We start by considering two real univariate data sets for which MMDL and MDL/BIC yield different estimates of the number of Gaussian components: the Old Faithful geyser eruption durations (well known in the density estimation literature [28]), and the enzyme activity data from [21]. Table 1 reports the values of $\mathcal{C}_{\mathrm{MMDL}}(\cdot)$ and $\mathcal{C}_{\mathrm{MDL/BIC}}(\cdot)$ for several values of k for these two data sets. Fig. 3 shows the resulting mixture density estimates. For the Old Faithful data, MMDL allows an extra component ($\widehat{k}_{\mathrm{MMDL}} = 4$) with which the resulting mixture adjusts better to the skewness of the right portion of the histogram. For the enzyme data, the additional component in the mixture selected by MMDL yields a clearly better fit to the observed histogram. Of course, in this real data cases, there is no underlying true mixture, and so there is no way to tell what is the correct number of components and we must rely on visual evaluation. An alternative would be to perform a (leave-one-out type) cross validation study comparing the MDL/BIC and MMDL criteria.

	k	1	2	3	4	5
Old Faithful	MMDL	429.8	288.8	283.6	**282.2**	286.7
	MDL/BIC	429.8	293.2	**289.4**	291.1	287.8
	k	1	2	3	4	5
Enzyme	MMDL	236.3	66.9	**65.8**	67.4	73.9
	MDL/BIC	236.3	**71.2**	72.5	77.4	87.5

Table 1. MMDL and MDL/BIC cost function values for several values of k for the Old Faithful and enzyme data sets.

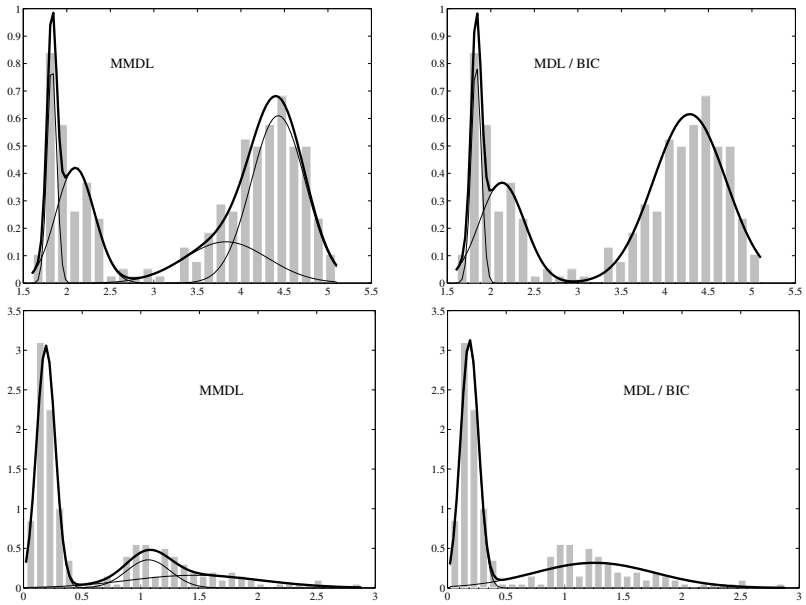

Fig. 3. Mixture estimates produced by the MMDL and MDL/BIC criteria for the Old Faithful (top row) and the enzyme data (bottom row).

Multivariate Data. To test the MMDL criterion on multivariate data, we have considered a mixture with 8 components on a 3D sample space. The component means are located at the vertices of a cube of side Δ,

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{\mu}_2 = \begin{bmatrix} \Delta \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{\mu}_3 = \begin{bmatrix} 0 \\ \Delta \\ 0 \end{bmatrix}, \ldots, \boldsymbol{\mu}_7 = \begin{bmatrix} 0 \\ \Delta \\ \Delta \end{bmatrix}, \quad \boldsymbol{\mu}_8 = \begin{bmatrix} \Delta \\ \Delta \\ \Delta \end{bmatrix}$$

and all have unit covariance matrix $\mathbf{C}_i = \text{diag}\{1,1,1\}$, for $i = 1,2,...,8$. We obtained 50 sets of 1200 samples each, for three different separations among the mixture components: $\Delta = 3$, 3.5, and 4. Figure 4 shows, for these three values of Δ, the number of times that each value of k was chosen by MMDL and MDL/BIC. Notice how the performance of MDL/BIC degrades faster than that of MMDL. For this test, since the goal is to study the behavior of the MMDL

and MDL/BIC criteria, not of the AEM algorithm, we have used $k_{\max} = 8$ and the true parameters as initialization.

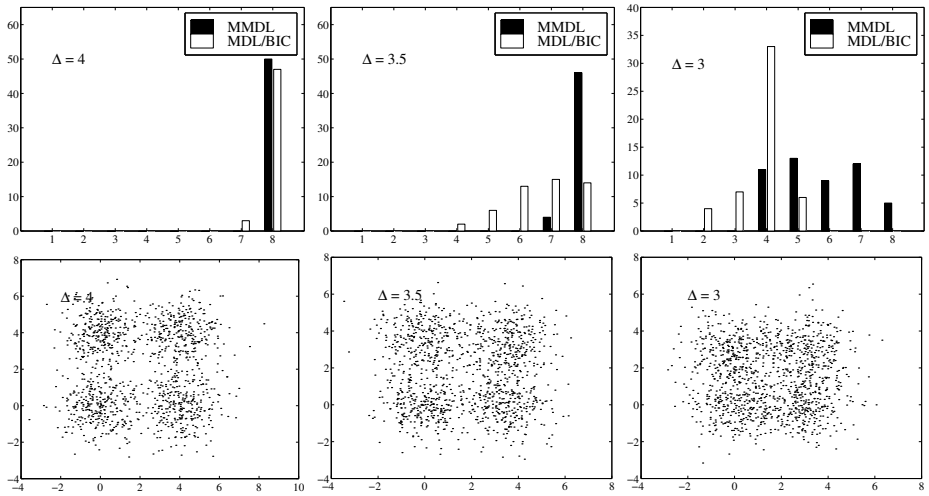

Fig. 4. Top row: histograms of the estimates of k (true value is 8) for $\Delta = 4$, 3.5, 3. Bottom row: examples of the first two components of the sample sets.

The MMDL criterion was also used in [33], for a Bayesian image classification problem. The class-conditional densities are represented by Gaussian mixtures, learned via a vector-quantization (VQ) approach, with the MMDL criterion controlling the size of each VQ. Given the very high dimensionality of the feature space (> 100), $N(1)$ is very high and MDL/BIC always yielded uselessly small estimates of k. With the estimates provided by MMDL, the resulting Bayesian classifier exhibited very good performance.

6 Conclusions and Further Work

We have proposed a new criterion to select the number of components in Gaussian mixtures and a new algorithm specially suited for mixture model estimation with an unknown number of components. The new criterion, called *mixture MDL* (MMDL), is a simple modification of the standard MDL/BIC, resulting from the identification of what can be called the *equivalent sample size* for each component. The proposed algorithm is based on EM together with an agglomerative step, thus it is called *agglomerative* EM (AEM). We have presented examples illustrating the behavior of AEM and its robustness with respect to initialization (although a more complete set of tests is still required). To compare MMDL versus MDL/BIC, we have performed experiments on real and synthetic data. All the experiments confirm that MMDL allows a better fit to the observed data.

Finally, we mention the parameterization of the covariance matrices (based on eigen-decomposition), introduced in [1] (see also [4]). That parameterization allows taking selected characteristics of the components to be common (for example, same shape, arbitrary orientation). MMDL can also be used to perform model selection among the options provided by that approach. The goal is to simultaneously choose the number of components and decide which characteristics (if any) should be assumed common.

References

1. J. Banfield and A. Raftery. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49:803–821, 1993.
2. H. Bensmail, G. Celeux, A. Raftery, and C. Robert. Inference in model-based cluster analysis. *Statistics and Computing*, 7:1–10, 1997.
3. J. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum Press, New York, 1981.
4. G. Celeux and G. Govaert. Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781–793, 1995.
5. C. Fraley and A. Raftery. How many clusters? Which clustering method? Answers via model-based cluster analysis. Technical Report 329, Department of Statistics, University of Washington, Seattle, WA, 1998.
6. T. Hastie and R. Tibshirani. Discriminant analysis by Gaussian mixtures. *Journal of the Royal Statistical Society (B)*, 58:155–176, 1996.
7. A. Jain and R. Dubes. *Algorithms for Clustering Data.* Prentice Hall, Englewood Cliffs, N. J., 1988.
8. A. Jain and J. Moreau. Bootstrap techniques in cluster analysis. *Pattern Recognition*, 20(5):547–568, 1987.
9. R. Kass and A. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90:733–795, 1995.
10. M. Kloppenburg and P. Tavan. Deterministic annealing for density estimation by multivariate normal mixtures. *Physical Review E*, 55:R2089–R2092, 1997.
11. P. Kontkanen, P. Myllymäki, and H. Tirri. Comparing bayesian model class selection criteria in discrete finite mixtures. In *Proceedings of Information, Statistics, and Induction in Science – ISIS'96*, pp. 364-374, Singapore, 1996. World Scientific.
12. S. Kullback. *Information Theory and Statistics.* J. Wiley & Sons, N. York, 1959.
13. J. Lin. Divergence measures based on the Shannon entropy. *IEEE Trans. Information Theory*, 37:145–151, 1991.
14. G. McLachlan. On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture. *Jour. Roy. Stat. Soc. (C)*, 36:318–324, 1987.
15. G. McLachlan and K. Basford. *Mixture Models: Inference and Application to Clustering.* Marcel Dekker, New York, 1988.
16. G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions.* John Wiley & Sons, New York, 1997.
17. G. McLachlan and D. Peel. MIXFIT: an algorithm for the automatic fitting and testing of normal mixture models. In *Proceedings of the 14th IAPR International Conference on Pattern Recognition*, volume II, pages 553–557, 1998.
18. K. Mengersen and C. Robert. Testing for mixtures: a Bayesian entropic approach. In J. Bernardo, J. Berger, A Dawid, and F. Smith, editors, *Bayesian Statistsics 5: Proceedings of the Fifth Valencia International Meeting*, pages 255–276. Oxford University Press, 1996.

19. R. Neal. Bayesian mixture modeling. In *Proceedings of the 11th International Workshop on Maximum Entropy and Bayesian Methods of Statistical Analysis*, pages 197–211. Kluwer, Dordrecht, The Netherlands, 1992.
20. J. Oliver, R. Baxter, and C. Wallace. Unsupervised learning using MML. In *Proceedings of the Thirtheenth International Conference on Machine Learning*, pages 364–372. Morgan Kaufmann, San Francisco, CA, 1996.
21. S. Richardson and P. Green. On Bayesian analysis of mixtures with unknown number of components. *Jour. of the Royal Statist. Soc. B*, 59:731–792, 1997.
22. B. Ripley. *Pattern Recognition and Neural Networks.* Cambridge University Press, Cambridge, U.K., 1996.
23. J. Rissanen. *Stochastic Complexity in Stastistical Inquiry.* World Scientific, 1989.
24. C. Robert. Mixtures of distributions: Inference and estimation. In W. Gilks, S. Richardson, and D. Spiegelhalter, editors, *Markov Chain Monte Carlo in Practice*, London, 1996. Chapman & Hall.
25. S. Roberts, D. Husmeier, I. Rezek, and W. Penny. Bayesian approaches to gaussian mixture modelling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11), November 1998.
26. K. Roeder and L. Wasserman. Practical Bayesian density estimation using mixtures of normals. *Journal of the American Statistical Association*, 92:894–902, 1997.
27. K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proc. of IEEE*, 86:2210–2239, 1998.
28. B. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman & Hall, London, 1986.
29. P. Smyth. Clustering using Monte-Carlo cross-validation. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 126–133. AAAI Press, Menlo Park, CA, 1996.
30. P. Smyth. Model selection for probabilistic clustering using cross-validated likelihood. Technical Report UCI-ICS 98-09, Information and Computer Science, University of California, Irvine, CA, 1998.
31. D. Titterington, A. Smith, and U. Makov. *Statistical Analysis of Finite Mixture Distributions.* John Wiley & Sons, Chichester (U.K.), 1985.
32. N. Ueda and R. Nakano. Deterministic annealing EM algorithm. *Neural Networks*, 11:271–282, 1998.
33. A. Vailaya, M. Figueiredo, A. K. Jain, and H. Jiang Zhang. A bayesian framework for semantic classification of outdoor vacation images. In *Proceedings of the 1999 SPIE Conference on Storage and Retrieval for Image and Video Databases VII*, pages 415–426. San Jose, CA, 1999.
34. M. West and J Harrison. *Bayesian Forecasting and Dynamic Models.* Springer-Verlag, New York, 1989.
35. M. Whindham and A. Cutler. Information ratios for validating mixture analysis. *Journal of the American Satistical Association*, 87:1188–1192, 1992.
36. A. Yuille, P. Stolorz, and J. Utans. Statistical physics, mixtures of distributions, and the EM algorithm. *Neural Computation*, 6:332–338, 1994.