On choosing a mixture model for clustering

Joseph Ngatchou-Wandji, Jan Bulla

On choosing a mixture model for clustering

Joseph Ngatchou-Wandji
Département BIOSTAT, EHESP
35043 Rennes
and
Université Henri Poincaré
54505 Vandoeuvre-lès-Nancy, France
E-Mail: Joseph.Ngatchou-Wandji@iecn.u-nancy.fr
Phone: +33 (0) 383684564

Jan Bulla
LMNO
Université de Caen
CNRS UMR 6139
14032 Caen Cedex, France
E-Mail: bulla@math.unicaen.fr
Phone: +33 (0) 231 567450

Abstract

Two methods for clustering data and choosing a mixture model are proposed. First, we derive a new classification algorithm based on the classification likelihood. Then, the likelihood conditional on these clusters is written as the product of likelihoods of each cluster, and AIC- respectively BIC-type approximations are applied. The resulting criteria turn out to be the sum of the AIC or BIC relative to each cluster plus an entropy term. The performances of our methods are evaluated by Monte-Carlo methods and on a real data set, showing in particular that the iterative estimation algorithm converges quickly in general, and thus the computational load is rather low.

Key Words: Mixtures models, clustering, AIC, BIC, ICL.

Acknowledgments: We would like to thank the associate editor and two anonymous referees whose comments led to improve the paper.

1 Introduction

Because of their ability to represent relationships in data, finite mixture models are commonly used for summarizing distributions. In cluster analysis, they can provide a framework for assessing the partitions of the data, and for choosing the number of clusters. A finite mixture model is characterized by its form denoted by m, and the number of components K, which can be interpreted as the number of species in the population from which the data has been collected. For optimizing a mixture, one often uses a scoring function on which the comparison between the competing models with different values of K is carried out. Such scoring functions are, for example, penalized likelihoods computing the likelihood on a single training set and provide a penalty for model complexity. The AIC (Akaike 1973, 1974) and the BIC (Schwarz 1978) criteria are based on such likelihoods, as well as the algorithm provided by Figueiredo et al. (1993) for estimating a mixture model.

For assessing the number of clusters arising from a Gaussian mixture model, Biernacki & Govaert (1997, 1999) used a penalized completed likelihood (CL). However, the associated criterion tends to overestimate the correct number of clusters when there is no restriction on the mixing proportions. The reason for this shortcoming is that the CL does not penalize the number of parameters in a mixture model. A penalization is provided in a Bayesian framework by Biernacki et al. (2000), who proposed a criterion based on the integrated completed likelihood (ICL). Their method consists in approximating the integrated completed likelihood by the BIC. This approximation, however, suffers from a lack of a theoretical justification, although their numerical simulations show satisfactory performance. Other procedures for determining the clusterings of data and a mixture model can be found, for instance, in Kazakos (1977), Engelman & Hartigan (1969), Bozdogan (1992), Medvedovic et al. (2001), Fraley & Raftery (1998), or McCullagh & Yang (2008).

In this paper, we propose two alternative approaches, based on AIC and BIC criteria applied to the classification likelihood. In a certain sense, these are close to Fraley & Raftery (1998), whose method is rather based on the BIC criterion applied to the mixture likelihood. Concretely, we first construct a new classification algorithm allowing to estimate the clusters of the data. On the basis of this classification, we define two new criteria based on AIC- and BIC-like approximations, which turn out to be the sum of the AIC or BIC approximations relative to each cluster plus an entropy term. On the one hand, this method avoids a number of technical difficulties encountered by ICL. On the other hand, the iterative estimation algorithm converges quickly in general, and thus the computational load is rather low.

This paper is organized as follows. In Section 2, we summarize clustering methods. Subsequently, Section 3 recalls a number of existing methods for choosing a mixture, and we describe our new approaches. Finally, Section 4 contains numerical examples to evaluate the performance of our methods.

2 Model-based clustering

A d-variate finite mixture model assumes that the data $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^{dn}$ are a sample from a probability distribution with density of the form

$$f(\mathbf{u}|m, K, \theta) = \sum_{k=1}^{K} p_k \phi_k(\mathbf{u}|\mathbf{a}_k), \quad \mathbf{u} \in \mathbb{R}^d, \tag{1}$$

where K is the number of components of the mixture, the p_k's represent the mixing proportions and the components $\phi_k(\cdot|\mathbf{a}_k)$'s are density functions, possibly of different nature[1], each with a known form and depending on the parameter vector $\mathbf{a}_k$. The notation m stands for the joint components, which characterize the nature of the mixture. Finally, $\theta := (\theta_1, \theta_2) := ((p_1, \ldots, p_K), (\mathbf{a}_1, \ldots, \mathbf{a}_K))$ represents the full parameter vector of the mixture (m, K) at hand. The most popular mixture is the Gaussian mixture model, where $\phi_k(\cdot|\cdot)$ are Gaussian densities with mean μ_k and covariance matrix Σ_k. More precisely, $\phi_k(\cdot|\mathbf{a}_k) = \phi(\cdot|\mathbf{a}_k)$ is a d-variate Gaussian density with $\mathbf{a}_k = (\mu_k, \Sigma_k)$ for $k = 1, \ldots, K$.

It is well known that the mixture model can be seen as an incomplete data structure model, where the complete data is given by

$$\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n) = ((\mathbf{x}_1, \mathbf{z}_1), \ldots, (\mathbf{x}_n, \mathbf{z}_n)),$$

with $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)$ standing for the missing data. The reader can report to Titterington et al. (1985) for more details. Note that $\mathbf{z}_i = (\mathbf{z}_{i1}, \ldots, \mathbf{z}_{iK})$ is a K-dimensional vector such that $\mathbf{z}_{ik}$ takes the value 1 if $\mathbf{x}_i$ arises from the component k, and takes the value 0 if not for $i = 1, \ldots, n$. Obviously, the vector $\mathbf{z}$ defines a partition $C = \{C_1, \ldots, C_K\}$ of the data $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, with $C_k = \{\mathbf{x}_i | \mathbf{z}_{ik} = 1, i = 1, \ldots, n\}$. If $\mathbf{z}$ was observed, the clusters would be known and the data in each class C_k could be assumed to be drawn from a distribution with density $\phi_k(\cdot; \mathbf{a}_k)$. Therefore, the likelihood conditional on $\mathbf{z}$ would have a form allowing for easy inference. Unfortunately, $\mathbf{z}$ is in general

[1]For example, a Gaussian and a Student distribution are of different natures, while two Gaussian distributions are of the same nature.

not observed and has to be estimated.

There are many ways for estimating $\mathbf{z}$. For instance, Rayment (1972), Kazakos (1977), Scott & Symons (1971), Symons (1981) treat the vector $\mathbf{z}$ as a parameter, which is estimated jointly with K and θ by maximizing the likelihood function

$$f(\mathbf{x}, \mathbf{z}|m, K, \theta) = \prod_{i=1}^{n} f(\mathbf{x}_i, \mathbf{z}_i|m, K, \theta), \tag{2}$$

where we recall that K is the number of components of the mixture, m stands for the joint components and

$$f(\mathbf{x}_i, \mathbf{z}_i|m, K, \theta) = \prod_{k=1}^{K} p_k^{\mathbf{z}_{ik}} [\phi_k(\mathbf{x}_i|\mathbf{a}_k)]^{\mathbf{z}_{ik}}, \ i = 1, \ldots, n. \tag{3}$$

The drawback of this method is that all possible clusterings of the data in K groups are considered, which may be computationally costly. Additionally, Marriott (1975) points out an inconsistency of the parameter estimates, and, $\mathbf{z}$ is formally treated as a parameter rather than a vector of missing observations. A Bayesian estimator of $\mathbf{z}$ is also defined in Symons (1981). Another, more popular method is the so-called MAP (maximum a posteriori) method, described as follows. For $i = 1, \ldots, n$ and $k = 1, \ldots, K$, let $t_{ik}(\theta)$ denote the conditional probability that $\mathbf{x}_i$ arises from the k^{th} mixture component. Then, one can easily show that

$$t_{ik}(\theta) = \frac{p_k \phi_k(\mathbf{x}_i|\mathbf{a}_k)}{\sum_{\ell=1}^{K} p_\ell \phi_\ell(\mathbf{x}_i|\mathbf{a}_\ell)}. \tag{4}$$

Let $\hat{\theta}$ be the maximum likelihood estimate of θ. Under some regularity conditions the so-called EM algorithm (Titterington et al. 1985) allows the computation of this estimator, by means of which, $\mathbf{z}_{ik}$ can be derived by

$$\hat{\mathbf{z}}_{ik} = \begin{cases} 1 & \text{if } \arg\max_{\ell \in \{1,\ldots,K\}} t_{i\ell}(\hat{\theta}) = k \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

for $i = 1, \ldots, n$ and $k = 1, \ldots, K$. For more approaches for estimating $\mathbf{z}$, see, e.g., McLachlan (1992), Fraley & Raftery (1998), McLachlan & Peel (2000), Medvedovic et al. (2001), Fraley & Raftery (2006) or McCullagh & Yang (2008). The estimates $\hat{\mathbf{z}}$ provided by either of these methods serve to determine the clusters of the data. Based on these clusters, it is possible to express likelihood for further inference. In the following section, we propose a new clustering algorithm based on the so-called classification likelihood.

3 Choosing a mixture model

3.1 Existing methods

Several methods exist for choosing a mixture model among a given number of models. One of these, consisting in maximizing (2), has already been recalled and commented in the previous section (see Symons 1981, for details). However, the most popular approaches are based on the AIC and BIC criteria as well as their extensions, or other criteria such as that presented by Figueiredo et al. (1993). In a Bayesian framework, one selects the model having the largest posterior probability. This is tantamount to choosing the model with the largest integrated completed likelihood (ICL), provided that all the models have equal prior probabilities (Biernacki et al. 2000). This corresponds to the model $(\hat{m}, \hat{K})$ such that

$$(\hat{m}, \hat{K}) = \arg\max_{m,K} f(\mathbf{x}, \mathbf{z}|m, K),$$

where

$$f(\mathbf{x}, \mathbf{z}|m, K) = \int_{\Theta_{m,K}} f(\mathbf{x}, \mathbf{z}|m, K, \theta)\pi(\theta|m, K)d\theta, \tag{6}$$

with $\Theta_{m,K}$ the parameter space, $\pi(\theta|m, K)$ a non-informative or weakly informative prior distribution on $\theta \in \Theta_{m,K}$ for the same model, and $f(\mathbf{x}, \mathbf{z}|m, K, \theta)$ the likelihood function defined by (2). From a BIC-like approximation of the right-hand side of (6), Biernacki et al. (2000) propose to select the model

which maximizes

$$\log f(\mathbf{x}, \mathbf{z}|m, K, \hat{\theta}^*) - \frac{d_{m,K}}{2} \log(n), \tag{7}$$

where $d_{m,K}$ stands for the dimension of the space $\Theta_{m,K}$, and $\hat{\theta}^* = \arg\max_\theta f(\mathbf{x}, \mathbf{z}|m, K, \theta)$. Since $\mathbf{z}$ is not observed, it is substituted by $\hat{\mathbf{z}}$ given by (5), and $\hat{\theta}$ is utilized instead of $\hat{\theta}^*$ in the above formula. Thus, their ICL criterion selects the $(\hat{m}, \hat{K})$ maximizing

$$\mathrm{ICL}(m, K) = \log f(\mathbf{x}, \hat{\mathbf{z}}|m, K, \hat{\theta}) - \frac{d_{m,K}}{2} \log(n). \tag{8}$$

It is important to note that the approximation (7) is not valid in general for mixture models. Moreover, even if this approximation was valid, the accuracy of (8) obtained by substituting $\mathbf{z}$ for $\hat{\mathbf{z}}$ and $\hat{\theta}$ for $\hat{\theta}^*$ may be hard to quantify.

3.2 Some new approaches

In the following, we adopt different techniques for finding the mixture model leading to the greatest evidence of clustering given data $\mathbf{x}$. Our approaches consist in first estimating the clusters, and secondly applying AIC-/BIC-like criteria to the likelihood derived from these clusters. More precisely, we consider the likelihood defined by equation (2), given that the vector $\mathbf{z}$ and thus $\theta_1 = (p_1, \ldots, p_K)$ are assumed to be known. Indeed, with this assumption it is easy to derive that the resulting conditional likelihood can be expressed as a product of the likelihoods of each component of the mixture model to which AIC or BIC approximations can be applied.

Assuming $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)$ given, the data are partitioned into K classes $C_1, C_2, \ldots, C_K$. Moreover, let $n_k = \sum_{i=1}^n z_{ik} = |C_k|$ for all $k = 1, \ldots, K$, where z_{ik} is the k^{th} component of $\mathbf{z}_i$, $i = 1, \ldots, n$. Then, the p_k can be consistently estimated by the natural estimators $\hat{p}_k = n_k/n$, which are also asymptotically normal. Thus, a consistent and asymptotically normal estimator of θ_1 is given by $\hat{\theta}_1 = (\hat{p}_1, \ldots, \hat{p}_K)$. Then, the likelihood and log-

likelihood functions of θ_2 given $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ can be approximated by

$$\ell(m, K, \theta_2|\hat{\theta}_1, \mathbf{z}) = \prod_{k=1}^{K} \prod_{\mathbf{x}_j \in C_k} \hat{p}_k \phi_k(\mathbf{x}_j|\mathbf{a}_k), \tag{9}$$

$$L(m, K, \theta_2|\hat{\theta}_1, \mathbf{z}) = \sum_{k=1}^{K} \left(\sum_{\mathbf{x}_j \in C_k} \log[\phi_k(\mathbf{x}_j|\mathbf{a}_k)] + n_k \log \hat{p}_k \right). \tag{10}$$

What remains is the estimation of $\theta_2 = (\mathbf{a}_1, \ldots, \mathbf{a}_K)$. This can be done by maximizing either (9) or (10). Note that the estimator of $\mathbf{a}_k$ depends only on the n_k observations within the k^{th} group C_k for $k = 1, \ldots, K$. Henceforth, we denote $\ell(m, K, \theta_2|\theta_1, \mathbf{z})$ by $\ell(m, K, \theta_2)$ and $L(m, K, \theta_2|\theta_1, \mathbf{z})$ by $L(m, K, \theta_2)$.

Let $d_{\mathbf{a}_k}$ denote the length of the vector $\mathbf{a}_k$, and $\Theta_{m,K}^{(k)} \subset \mathbb{R}^{d_{a_k}}$ for all $k = 1, \ldots, K$. In what follows, we suppose that $\theta_2 \in \Theta_{m,K}^{*} = \Theta_{m,K}^{(1)} \times \ldots \times \Theta_{m,K}^{(K)}$, and that the $\phi_k(\cdot|\mathbf{a}_k)$ are identifiable and differentiable up to order 2. Then, the integrated likelihood is defined by

$$\begin{aligned} \ell(m, K) &= \int_{\Theta_{m,K}^{*}} \ell(m, K, \theta_2) \pi(\theta_2|m, K) d\theta_2 \\ &= \prod_{k=1}^{K} \hat{p}_k \int_{\Theta_{m,K}^{(k)}} \prod_{\mathbf{x}_j \in C_k} \phi_k(\mathbf{x}_j|\mathbf{a}_k) \pi_k(\mathbf{a}_k|m, K) d\mathbf{a}_k, \end{aligned} \tag{11}$$

which follows from the likelihood function (9).

Theorem 1 *Assume that* $\mathbf{z}$ *is known, and that the* n_k*'s are large enough for* $k = 1, 2, \ldots, K$*. Then, the following approximation for the log-likelihood function holds:*

$$L(m, K, \theta_2) \approx \sum_{k=1}^{K} \left(\sum_{\mathbf{x}_j \in C_k} \log \phi_k(\mathbf{x}_j|\hat{\mathbf{a}}_k) - d_{\mathbf{a}_k} \right) + \sum_{k=1}^{K} n_k \log \hat{p}_k. \tag{12}$$

Proof. Given $\mathbf{z}$, the deviance of the model can be approximated by

$$\begin{aligned} & L(m, K, \theta_2) - \sum_{k=1}^{K} \left(\sum_{\mathbf{x}_j \in C_k} \log \phi_k(\mathbf{x}_j | \hat{\mathbf{a}}_k) + n_k \log \hat{p}_k \right) \\ = & \sum_{k=1}^{K} \sum_{\mathbf{x}_j \in C_k} \Big[\log \phi_k(\mathbf{x}_j | \mathbf{a}_k) - \log \phi_k(\mathbf{x}_j | \hat{\mathbf{a}}_k) \Big], \end{aligned}$$

which is the sum of the deviances relative to the components of the mixture. As n_k is large, it follows

$$\sum_{\mathbf{x}_j \in C_k} \Big[\log \phi_k(\mathbf{x}_j | \mathbf{a}_k) - \log \phi_k(\mathbf{x}_j | \hat{\mathbf{a}}_k) \Big] \approx -d_{\mathbf{a}_k},$$

for each $k = 1, \ldots, K$ (see, e.g., Akaike 1974). □

Theorem 2 *Assume that* $\mathbf{z}$ *is known, that the* n_k *are large enough for* $k = 1, 2, \ldots, K$*, and that the prior on* θ_2 *has the form*

$$\pi(\theta_2 | m, K) = \pi_1(\mathbf{a}_1 | m, K) \times \ldots \times \pi_K(\mathbf{a}_K | m, K). \tag{13}$$

Then, the logarithm of the integrated likelihood can be approximated by

$$\log \ell(m, K) \approx \sum_{k=1}^{K} \left(\sum_{\mathbf{x}_j \in C_k} \log \phi_k(\mathbf{x}_j | \hat{\mathbf{a}}_k) - \frac{1}{2} d_{\mathbf{a}_k} \log(n_k) \right) + \sum_{k=1}^{K} n_k \log \hat{p}_k. \tag{14}$$

Proof. See the appendix.

The first term on the right-hand sides of (12) and (14), respectively, looks like sums of AIC and BIC respectively, and depend on $\mathbf{z}$ and θ_1. Therefore, we denote these quantities by $\mathrm{SAIC}(m, K | \theta_1, \mathbf{z})$ and $\mathrm{SBIC}(m, K | \theta_1, \mathbf{z})$, which

stands for "Sum of AIC/BIC". They can be written as

$$\mathrm{SAIC}(m, K|\hat{\theta}_1, \mathbf{z}) = \sum_{k=1}^{K} \left(\sum_{\mathbf{x}_j \in C_k} \log \phi_k(\mathbf{x}_j|\hat{\mathbf{a}}_k) + n_k \log \hat{p}_k - d_{\mathbf{a}_k} \right) \tag{15}$$

$$\mathrm{SBIC}(m, K|\hat{\theta}_1, \mathbf{z}) = \sum_{k=1}^{K} \left[\sum_{\mathbf{x}_j \in C_k} \log \phi_k(\mathbf{x}_j|\hat{\mathbf{a}}_k) + n_k \log \hat{p}_k - \frac{d_{\mathbf{a}_k}}{2} \log(n_k) \right]. \tag{16}$$

We would like to remark that some penalties related to those in SAIC and SBIC can be found for instance in Pauler (1998) and Raftery & Krivitsky (2007).

Before describing a technique for model selection based on equation (15) respectively (16), we first provide an algorithm for parameter estimation given the number of clusters, denoted by K. Let $\mathbf{z}_K$ denote the corresponding missing data, and θ_{1K} the corresponding parameter vector θ_1. Given the mixture components ϕ_k, $k = 1, \ldots, K$, the algorithm is described as follows:

- Initialize $\mathbf{z}$ (for example, by the k-means algorithm)
- **Repeat**
  - for $k = 1, \ldots, K$, compute $n_k = \sum_{i=1}^{n} z_{ik}$ and $p_k = \frac{n_k}{n}$, thus $\theta_{1K} = (p_1, \ldots, p_K)$
  - maximize the log-likelihood given in (10) with respect to $\theta_2 = (\mathbf{a}_1, \ldots, \mathbf{a}_K)$ and denote by θ_{2K} the vector for which the likelihood reaches the maximum.
  - for $i = 1, \ldots, n$ and $k = 1, \ldots, K$, compute

$$\mathbf{z}_{ik} = \begin{cases} 1 & \text{if } \arg\max_{\ell \in \{1,\ldots,K\}} t_{i\ell}(\theta_K) = k \\ 0 & \text{otherwise} \end{cases},$$

    where $\theta_K = (\theta_{1K}, \theta_{2K})$ and t_{ik} is defined by (4)

Until the log-likelihood remains constant

- Return $\mathbf{z}_K$ and θ_{1K}.

For choosing the relevant model, and thus determining its form, its parameters, and the number of clusters, we propose to proceed as follows.

- Set the maximum number of components $K_{\max}$
- For $K = 2, \ldots, K_{\max}$
  - Compute θ_{1K} and $\mathbf{z}_K$ (with the above algorithm)
  - Compute $\text{SAIC}(m, K|\theta_{1K}, \mathbf{z}_K)$ or $\text{SBIC}(m, K|\theta_{1K}, \mathbf{z}_K)$
- Select $(\hat{m}, \hat{K})$ and $\mathbf{z}_{\hat{K}}$ by

$$(\hat{m}, \hat{K}) = \arg\max_{m,K} \text{SAIC}(m, K|\theta_{1K}, \mathbf{z}_K).$$

  or

$$(\hat{m}, \hat{K}) = \arg\max_{m,K} \text{SBIC}(m, K|\theta_{1K}, \mathbf{z}_K).$$

4 Numerical examples

In this section, we mainly study the performance of the SAIC and SBIC criteria in comparison to the BIC resulting from a second model-based clustering algorithm by Fraley & Raftery (2006). Four different settings are considered: an application to data from the Old Faithful Geyser (Yellowstone National Park, USA) and three Monte Carlo experiments. Moreover, we present brief results on the robustness of our algorithm towards its initialization, the speed of convergence, and the classification performance. The software utilized is `R` 2.10.1 (R Development Core Team 2010), the package `Mclust` has version number 3.3.1. All code is available from the authors upon request.

Table 1: Model selection by SAIC/SBIC

This table displays log-likelihood, SAIC, and SBIC of the estimated models with 2, 3, and 4 components, initialized by k-means or random paths.

no. comp.	2	3	4
logL	-1131	-1125	-1120
SAIC	-1141	-1140	-1140
SBIC	-1155	-1157	-1158

4.1 Old Faithful Geyser

The data analyzed are waiting times between eruptions and the durations of the eruption for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA. This data set with 272 observations is included in the `datasets` package of `R`. In order to initialize our clustering algorithm, termed mb1 in the following, we follow two approaches. On the one hand, we utilize the k-means algorithm (function `kmeans` in `R`) to estimate an initial trajectory of $\mathbf{z}$, where the k-means itself is started by 100 different random sets, and estimate models with two, three, and four components. On the other hand, we generate 1000 random paths for $\mathbf{z}$ (identical sampling probability for each component). The initialization by random paths requires higher computational effort, however, also attains higher likelihoods. Therefore, this method is preferred for this example with relatively small sample size, and we do not further comment results from the k-means initialization. Fitting the 2-component model, the algorithm estimates clusters containing less than 5% of the sample for only 5% of the initial paths. However, this figure rises to $\sim$30% for the models with three/four components. These models have been removed, as they do not really entail three respectively four components. Table 1 presents the results, showing an almost constant SAIC. Thus, according to this criterion, the parsimonious 2-component model should be selected. The SBIC also attains the highest value for two components, therefore the same model is chosen.

For comparison with a standard algorithm for model-based clustering, we also fitted models using the `R` package `mclust` (Fraley & Raftery 2006). This

Figure 1: Clustering of Old Faithful Geyser data
The figure shows bivariate data from the Old Faithful Geyser, clustered by mb1. The preferred model has two components, the centers of which are marked by filled circles. Contours result from the two estimated Gaussian densities.

algorithm, termed mb2 in the following, is initialized by hierarchical clustering and selects an appropriate model by the BIC. The result of the mb2 is a model with 3 components, which might be attributed to "model deviation to normality in the two obvious groups rather than a relevant additional group" (Biernacki et al. 2000). Note that the number of components preferred by the SAIC/SBIC corresponds to that of the ICL criterion of the before mentioned authors. Figure 1 displays the data, the estimated densities of the two components and the mapping of the observations to the components.

The estimated parameters are

$$\mu_1 = \begin{pmatrix} 2.04 \\ 54.5 \end{pmatrix}, \mu_2 = \begin{pmatrix} 4.29 \\ 80.0 \end{pmatrix},$$

$$\Sigma_1 = \begin{pmatrix} 0.0712 & 0.452 \\ 0.452 & 34.1 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 0.169 & 0.918 \\ 0.918 & 35.9 \end{pmatrix}.$$

The estimated values of $\mathbf{z}$ indicate that 35.7% and 64.3% of the observations belong to the respective components.[2]

Finally, the speed of convergence of the algorithm and its stability towards the initialization is of interest. The number of iterations required by the algorithm is rather manageable in the majority of cases. Considering the

[2]For the model with three components, the estimated means and covariances are $\mu_1 = \begin{pmatrix} 1.86 \\ 53.2 \end{pmatrix}$, $\mu_2 = \begin{pmatrix} 2.31 \\ 56.6 \end{pmatrix}$, $\mu_3 = \begin{pmatrix} 4.29 \\ 80.0 \end{pmatrix}$, $\Sigma_1 = \begin{pmatrix} 0.00971 & -0.00502 \\ -0.00502 & 25.6 \end{pmatrix}$, $\Sigma_2 = \begin{pmatrix} 0.0464 & 0.235 \\ 0.235 & 41.5 \end{pmatrix}$, $\Sigma_3 = \begin{pmatrix} 0.169 & 0.918 \\ 0.918 & 35.9 \end{pmatrix}$. For four components, the respective estimates equal $\mu_1 = \begin{pmatrix} 2.09 \\ 61.3 \end{pmatrix}$, $\mu_2 = \begin{pmatrix} 1.87 \\ 54.4 \end{pmatrix}$, $\mu_3 = \begin{pmatrix} 4.29 \\ 80.0 \end{pmatrix}$, $\mu_4 = \begin{pmatrix} 2.07 \\ 49.7 \end{pmatrix}$, $\Sigma_1 = \begin{pmatrix} 0.0851 & 0.414 \\ 0.414 & 9.74 \end{pmatrix}$, $\Sigma_2 = \begin{pmatrix} 0.0150 & 0.157 \\ 0.157 & 2.15 \end{pmatrix}$, $\Sigma_3 = \begin{pmatrix} 0.169 & 0.918 \\ 0.918 & 35.9 \end{pmatrix}$. $\Sigma_4 = \begin{pmatrix} 0.0733 & 0.505 \\ 0.505 & 10.2 \end{pmatrix}$.

random initializations, the third quartile of the number of iterations lies at 14, 16, and 15 for models with 2, 3, and 4 components, respectively. The corresponding figures for the k-means initialization are 3, 9, and 13, confirming low computational load. Concerning the stability of the algorithm towards initialization, it should be noted that mb1 failed to converge in 12% of the cases in the 2-component case. This may be attributed to very poor initialization of the components. Moreover, the algorithm converged to the maximum likelihood of -1131 in 69.6% of the cases, which corresponds to the maximum attained by the k-means initialization. For 3 respectively 4 components, the results are less satisfactory: First, almost all estimated models are (slightly) different to each other. Moreover, in 48%/76% of the samples the algorithm does not converge properly, determines components with very few observation (< 10), or estimates two or more components with (almost) identical parameters. This behaviour may, however, be viewed keeping in mind "Garbage in, garbage out", as the initialization paths are purely random and may also underline the preference for the model with two components. Summarizing, random path initialization does not seem to provide better results than the k-means initialization, but rather entails convergence problems. Therefore, if not stated differently, our algorithm is always initialized by the k-means in the following.

4.2 Monte Carlo experiment 0

In order to examine the performance of SAIC/SBIC in situations with smaller sample size and overlapping clusters, we carry out Monte Carlo experiments in the style of Situation 1 and 2 described by (Biernacki et al. 2000, Section 4.1.1). More precisely, in each case we simulate 500 samples from a Gaussian mixture with three components having equal volume and shape, but different orientation. The common parameters of both situations are

$$\mu_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \mu_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \mu_3 = \begin{pmatrix} 8 \\ 0 \end{pmatrix},$$

Figure 2: Examples for Situation 1 and 2
The figure shows examples for Situation 1 (left panel) and 2 (right panel) for the Monte Carlo experiment inspired by Biernacki et al. (2000). The contours result from the two underlying true Gaussian densities.

$$\Sigma_1 = \Sigma_3 = \begin{pmatrix} 0.11 & 0 \\ 0 & 9 \end{pmatrix}.$$

The two situations differ in Σ_2. In Situation 1, the covariance matrix equals

$$\Sigma_2 = \begin{pmatrix} 2.33 & 3.85 \\ 3.85 & 6.78 \end{pmatrix},$$

resulting in an angle of 30° between the first eigenvectors of Σ_1 and Σ_2. In Situation 2 the respective angle equals 18° leading to

$$\Sigma_2 = \begin{pmatrix} 0.96 & 2.61 \\ 2.61 & 8.15 \end{pmatrix}.$$

The number of observations per cluster are $n_1 = n_2 = 100$ and $n_3 = 200$, Figure 2 shows two of the simulated data sets.

For each simulated sample, we executed our algorithm mb1. In order to initialize mb1, we utilize the k-means algorithm as stated in the previous section. Additionally, the algorithm mb2 by (Fraley & Raftery 2006) is fitted as presented above, selecting the preferred model by the BIC. In order to estimate similar models by the two algorithms, mb2 is constrained to treat the ellipsoidal and unconstrained case (argument `modelNames = "VVV"`).
For the Situation 1, BIC and SBIC/SAIC exhibit a different behavior. While the BIC prefers a model with two components in 52.0% of all cases (followed by four and three components with 39.1% and 8.83%, respectively), SAIC and SBIC show a strong preference for the two components. The SAIC selects $K = 2$ components in 92.4% ($K = 3$: 5.98%, $K = 4$: 1.65%), and the SBIC attains even 95.3% for $K = 2$ ($K = 3$: 4.12%, $K = 4$: 0.619%). In Situation 2, similar tendencies occur. The preferred model of the BIC

resulting from mb2 again has two components (52.4% of all cases), followed by $K = 4$ and $K = 3$ (32.1 % and 15.5%, respectively). However, SAIC and SBIC prefer $K = 2$ in 94.4% respectively 95.8% of all simulation runs, almost ignoring models with three and four components. These results indicate that both mb1 and mb2 tend to select select $K = 2$ instead of the correct model with $K = 3$. However, the SAIC/SBIC exhibits a much stronger tendency towards parsimonious 2-component-models than it is the case for the BIC. In particular, the results differ from those of the ICL reported by Biernacki et al. (2000), selecting $K = 3$ in the majority of cases for Situation 1, but mostly $K = 2$ for Situation 2.
To further investigate this effect, we repeated the previous experiment with reduced sample size, more precisely $n_1 = n_2 = 50$ and $n_3 = 100$ and similar tendencies occur. The BIC mainly prefers $K = 2$ components in both situations (52% and 77.4% for Situation 1 and 2, respectively). However, SAIC and SBIC show preference for the same number of components in 88.8% respectively 90.6% of all cases for Situation 1, and 94.6% respectively 96.4% in Situation 2.

4.3 Monte Carlo experiment 1

For the first three settings of this experiment we simulate in each case 500 samples from a two-component Gaussian mixture with the following common parameters:

$$\mu_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \mu_2 = \begin{pmatrix} 3 \\ 0 \end{pmatrix},$$

$$\Sigma_1 = \begin{pmatrix} 0.5 & 0 \\ 0 & 1 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 3 & -0.8 \\ -0.8 & 1 \end{pmatrix}$$

The three settings differ with respect to the number of observations per component. Setting 1a is subject to equal proportions with $n_1 = n_2 = 250$. The other two settings are subject to unequal proportions: Setting 1b deals with a bigger sample, i.e. $n_1 = 750$, $n_2 = 250$, and Setting 1c with a smaller sample, i.e. $n_1 = 100$, $n_2 = 200$. Figure 3 displays examples for the two

Figure 3: Examples for Settings 1a and 1b
The figure shows examples for the first Monte Carlo experiment. The left panel shows Setting 1b, the right panel Setting 1c, both subject to unequal proportions. The contours result from the two underlying true Gaussian densities.

settings.

For each simulated sample, we executed our algorithm mb1 for models with 2 and 3 components initialized by the k-means algorithm. In Setting 1a and 1c, the performance of SAIC and SBIC is equivalent, both select the model with two components in 99.6% of all cases. The results in Setting 1b show a similar tendency, SAIC and SBIC select two components for 95.8% respectively 95.6% of the samples. These results indicate that both SAIC and SBIC exhibit a strong tendency to select the correct model and do not differ much from the BIC resulting from mb2, which selects $K = 3$ components in 100% of all cases in all three settings. Besides, further analysis of those cases in which the model with three components is chosen by SAIC or SBIC reveals that the third (erroneous) component always contains only a small number of mostly outlying observations. On average, this number equals 21.3 in Setting 1a and less than 10 in Setting 1b, which means that we do not observe a proper third component. It may be subject to further investigation whether this results from the fact that the algorithm has been initialized by only one path $\mathbf{z}$, and different initializations may improve the results. However, for the sake of readability we do not follow this path here as the selection shows a clear preference of the correct model.
Last, we address the number of iterations required for the algorithm to converge. In Setting 1a, the average number of iterations equals 7.72 (s.d. 2.68) and 13.5 (6.72) for two and three components, respectively. The corresponding figures equal 5.63 (1.69) respectively 27.8 (15.9) in Setting 1b , and 8.27 (2.79) respectively 9.49 (5.84) in Setting 1c. Note that the number of iterations is rather low, in particular when considering the slow convergence in the neighbourhood of a maximum of the commonly used EM-algorithm (Dempster et al. 1977, Redner & Walker 1984).

Figure 4: Examples for Settings 2a and 2b
The figure shows examples for the second Monte Carlo experiment. The left panel (Setting 2a) displays the case $n = 1500$, and in the right panel (Setting 2b) n equals 3000. The contours result from the three underlying true Gaussian densities.

4.4 Monte Carlo experiment 2

The second experiment treats two settings with three Gaussian components. As for the first experiment, we simulate two times 500 samples having the common parameters:

$$\mu_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \mu_2 = \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \mu_3 = \begin{pmatrix} 3 \\ -2 \end{pmatrix},$$

$$\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 2 & -1.5 \\ -1.5 & 2 \end{pmatrix}, \Sigma_3 = \begin{pmatrix} 2 & 1.9 \\ 1.9 & 2 \end{pmatrix}$$

The two settings differ with respect to the number of observations per component: in Setting 2a, the numbers of observations per component equal $n_1 = 250$, $n_2 = 500$, and $n_3 = 1000$, whereas in Setting 2b the sample sizes are doubled, i.e., $n_1 = 500$, $n_2 = 1000$, and $n_3 = 2000$. Figure 4 displays an example for each of the settings.

The estimation procedure is carried out just as in the previous experiment. The number of iterations required by the algorithm to converge is still low, being equal to 14.8 (7.53) respectively 14.6 (7.14) iterations for Setting 2a and 2b in the 3-component case. For two and four components, the figure roughly halves and doubles, respectively. However, in some cases the algorithm fails to converge (2a: 1.0% / 3.8%, 2b: 2.2% / 5.5% for three/four components), and we exclude these cases from all further analysis to avoid any bias.

In Setting 2a, the performance of SAIC and SBIC is almost identical, selecting the 3-component-model in 78.6% and 78.1%, respectively. The results in Setting 2b show a similar tendency, SAIC and SBIC select three components

for 79.1% respectively 78.5% of the samples. The performance of mb2 is not much different: In Setting 2a, the BIC correctly selects $K = 3$ components in 89.2%, and the corresponding figure for Setting 2b equals 92% (in the remaining cases, $K = 4$ is chosen). As before, further analysis of those cases in which four components are selected reveals that the number of observations in the fourth component is small. In 75% of these cases, the number of observations is less than 26 and 15 in Setting 2a and 2b, respectively, and the same comments as in the previous section w.r.t. the initialization of the algorithm apply.

4.5 Classification performance

The aim of this section is to present the classification performance of our algorithm and a comparison to the previously introduced mb2 by Fraley & Raftery (2006). The two main questions addressed are: Firstly, is the classification performance of mb1 satisfactory? Secondly, how is the classification performance of mb1 compared to mb2?

As the previous Monte Carlo experiments revealed, both algorithms tend to select the correct model in the context of large samples. Therefore, we consider the settings described in the previous Section 4.3 and 4.4, each comprising 500 simulated samples. Our algorithm is initialized a) by the k-means and b) by the path estimated by mb2, whereas mb2 utilizes the default setting, a hierarchical clustering. The first three columns of Table 2 report the average classification errors of mb1 - initialized by kmeans and the mb2-path - and mb2. All figures are rather small and, at first glance, the classification error of mb1 initialized by k-means seems to be a little higher than that of mb2. However, as the first entries of the last columns indicate, the difference is significant at 5%-level only in Settings 1b/1c. As to the classification error of mb2 and mb1 initialized by the mb2-path, the second entry of the last columns shows no significant difference in any setting. Thus, the classification error of mb1 may be considered satisfactory, and not necessarily inferior or superior to mb2.

Table 2: Classification error

This table displays the classification error by Monte-Carlo simulation scenario. The columns display, from left to right: Average classification error of mb1 (initialized by the k-means and mb2), average classification error of mb2, p-values of Wilcoxon's signed rank test.

scenario	mb1 (kmeans)	mb1 (mb2)	mb2	p-values
1a	2.75	2.62	2.63	0.117/0.972
1b	2.34	2.29	2.25	0.0194/0.342
1c	2.84	2.51	2.52	0.0455/0.741
2a	1.94	1.93	1.91	0.305/0.425
2b	1.88	1.88	1.87	0.576/0.758

Summarizing, for the classification performance of mb1 initialized by the k-means seems satisfactory given the examples treated in this section. In particular, cases where the separation of the mixing distributions is not too obvious, i.e. small sample sizes and/or close centers may be subject to further studies.

5 Discussion and concluding remarks

In this section, we discuss the similitudes and differences between our procedure and existing relevant ones, and we conclude our work.

1. Our classification procedure is a CEM algorithm (see Celeux & Govaert 1992) based on the *classification* likelihood, which can be initialized with an arbitrary $\mathbf{z}$. For its derivation, we have used the idea of Fraley & Raftery (1998) who propose a procedure based on the *mixture* likelihood. One of the main advantages of such algorithms is that good estimators of $\mathbf{z}$, such as those given by hierarchical classification or k-means methods, are available, and can therefore be used as starting point. The classical CEM algorithm is initialized with an arbitrary value of the parameters whose *preliminary* estimator, which could be used as the starting point, is generally not easy to obtain.

2. At each step of our CEM algorithm, the parameters of the k^{th} component of the mixture are estimated based on the observations from the k^{th} class, while the mixing proportions are estimated empirically. These features are quite different to those of the classical MAP method used in Biernacki et al. (2000) and the CEM algorithm described in Fraley & Raftery (1998).

3. The ICL procedure uses the maximum likelihood (ML) estimator of the parameters from the incomplete likelihood instead of the ML estimator from the complete likelihood without any theoretical justification. Such problems are not encountered with the SBIC respectively SAIC, in which $\mathbf{z}$ is estimated iteratively from our CEM-like algorithm.

4. The SBIC procedure is constructed under the assumption that the prior distribution of the parameter vector is the product of the priors of each of its components. This gives rise to a different penalization terms than those of BIC and ICL.

5. With respect to the procedure for selecting the number of components, our method has some similarities with that of Fraley & Raftery (1998). However, their approach utilizes the mixture likelihood and ML estimator, whereas we use the classification likelihood and empirical estimators of the mixing proportions combined/and with ML estimators of the parameters of the mixture components.

6. The numerical examples show that SAIC and SBIC show a satisfactory selection performance. In particular, in the context of small samples and overlapping components, parsimonious models are selected. Additionally, an appealing property of the mb1 algorithm is the low number of iterations required to converge.

6 Appendix: Proof of Theorem 2

Let $k \in \{1, \ldots, K\}$, and denote

$$\Lambda^{(k)}(m, K) = \int_{\Theta^{(k)}_{m,K}} \prod_{\mathbf{x}_j \in C_k} \phi_k(\mathbf{x}_j|\mathbf{a}_k)\pi_k(\mathbf{a}_k|m, K)d\mathbf{a}_k$$

and

$$g(\mathbf{a}_k) = \sum_{\mathbf{x}_j \in C_k} \log \phi_k(\mathbf{x}_j|\mathbf{a}_k) + \log \pi_k(\mathbf{a}_k|m, K).$$

Moreover, define the vector $\mathbf{a}^*_k$ and the matrix $A_{\mathbf{a}^*_k}$ as follows:

$$\mathbf{a}^*_k = \arg \max_{\mathbf{a}_k \in \Theta^{(k)}_{m,K}} \left(\frac{1}{n_k} g(\mathbf{a}_k) \right)$$

and

$$A_{\mathbf{a}^*_k} = -\frac{1}{n_k} \left(\frac{\partial^2 g(\mathbf{a}^*_k)}{\partial \mathbf{a}^{(i)}_k \partial \mathbf{a}^{(j)}_k} : 1 \leq i, j \leq d_{\mathbf{a}_k} \right).$$

Then we obtain

$$\begin{aligned} \Lambda^{(k)}(m, K) &= \int_{\Theta^{(k)}_{m,K}} \exp[g(\mathbf{a}_k)]d\mathbf{a}_k \\ &= \exp[g(\mathbf{a}^*_k)] \left(\frac{2\pi}{n_k} \right)^{d_{\mathbf{a}_k}/2} |A_{\mathbf{a}^*_k}|^{-1/2} + O(n_k^{-1/2}). \end{aligned}$$

by the Laplace transformation (see, e.g., Kass & Raftery 1995). Since $\log g(\mathbf{a}_k)$ behaves like the product $\prod_{\mathbf{x}_j \in C_k} \phi_k(\mathbf{x}_j|\mathbf{a}_k)$ for all $k = 1, \ldots, K$, which increases whilst $\pi_k(\mathbf{a}_k|m, K)$ is constant, one can substitute the vector $\mathbf{a}^*_k$ by $\hat{\mathbf{a}}_k = \arg\max\{(1/n_k) \prod_{\mathbf{x}_j \in C_k} \phi_k(\mathbf{x}_j|\mathbf{a}_k)\}$ and the matrix $A_{\mathbf{a}^*_k}$ by the Fisher information matrix $I_{\hat{\mathbf{a}}_k}$ defined by

$$\begin{aligned} I_{\hat{\mathbf{a}}_k} &= -\left(\sum_{\mathbf{x}_j \in C_k} E\left[\frac{\partial^2 \log \phi_k(\mathbf{x}_j|\hat{\mathbf{a}}_k)}{\partial \mathbf{a}^{(i)}_k \partial \mathbf{a}^{(j)}_k} \right] : 1 \leq i, j \leq d_{\mathbf{a}_k} \right) \\ &= -\left(n_k E\left[\frac{\partial^2 \log \phi_k(\mathbf{x}_j|\hat{\mathbf{a}}_k)}{\partial \mathbf{a}^{(i)}_k \partial \mathbf{a}^{(j)}_k} \right] : 1 \leq i, j \leq d_{\mathbf{a}_k} \right). \end{aligned}$$

Then follows:

$$\begin{aligned}\log \Lambda^{(k)}(m,K) &= \sum_{\mathbf{x}_j \in C_k} \log \phi_k(\mathbf{x}_j|\hat{\mathbf{a}}_k) + \log \pi_k(\hat{\mathbf{a}}_k|m,K) - \frac{d_{\mathbf{a}_k}}{2}\log(n_k) \\ &\quad + \frac{d_{\mathbf{a}_k}}{2}\log(2\pi) - \frac{1}{2}\log(|I_{\hat{\mathbf{a}}_k}|) + O(n_k^{-1/2}).\end{aligned}$$

Neglecting the $O(n_k^{-1/2})$ and $O(1)$ terms, one obtains the approximation

$$\log \Lambda^{(k)}(m,K) \approx \sum_{\mathbf{x}_j \in C_k} \log \phi_k(\mathbf{x}_j|\hat{\mathbf{a}}_k) - \frac{d_{\mathbf{a}_k}}{2}\log(n_k).$$

Thus, from this approximation and (11) follows

$$\log \ell(m,K) \approx \sum_{k=1}^{K} \left(\sum_{\mathbf{x}_j \in C_k} \log \phi_k(\mathbf{x}_j|\hat{\mathbf{a}}_k) + n_k \log \hat{p}_k - \frac{d_{\mathbf{a}_k}}{2}\log(n_k) \right).$$

□

References

Akaike, H. (1973), *in* 'Proceedings of the Second International Symposium on Information Theory', Budapest: Akademiai Kiado 1973, pp. 267–281.

Akaike, H. (1974), 'A new look at the statistical model identification', *IEEE T. Automat. Contr.* **19**(6), 716–723.

Biernacki, C., Celeux, G. & Govaert, G. (2000), 'Assessing a mixture model for clustering with the integrated completed likelihood', *IEEE T. Pattern Anal.* **22**(7), 719–725.

Biernacki, C. & Govaert, G. (1997), 'Using the classification likelihood to choose the number of clusters', *Comp. Sci. Stat.* **29**(2), 451–457.

Biernacki, C. & Govaert, G. (1999), 'Choosing models in model-based clustering and discriminant analysis', *J. Stat. Comput. Sim.* **64**(1), 49–71.

Bozdogan, H. (1992), *Choosing the number of components clusters in the mixture model using a new informational complexity criterion of the inverse Fisher information matrix. In O. Opitz, B. Lausen, and R. Klar, editors, Information and classification*, Springer-Verlag.

Celeux, G. & Govaert, G. (1992), 'A classification em algorithm for clustering and two stochastic versions', *Comput. Statist. Data Anal.* **14**(3), 315–332.

Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977), 'Maximum likelihood from incomplete data via the EM algorithm', *J. Roy. Statist. Soc. Ser. B* **39**(1), 1–38. With discussion.

Engelman, L. & Hartigan, J. (1969), 'Percentage points for a test for clusters', *J. Amer. Statist. Assoc.* **64**, 1647.

Figueiredo, M., Leitão, J. & Jain, A. (1993), *On fitting mixture models*, Lecture Notes in Computer Science, Springer Berlin / Heidelberg.

Fraley, C. & Raftery, A. E. (1998), 'How many clusters? which clustering method? answer via model-based cluster analysis', *Comput. J.* **41**, 578–588.

Fraley, C. & Raftery, A. E. (2006), MCLUST version 3 for R: Normal mixture modeling and model-based clustering, Technical report, Department of Statistics, University of Washington. Technical Report no. 504.

Kass, R. & Raftery, A. E. (1995), 'Bayes factors', *J. A. Stat. Assoc.* **90**(430), 773–795.

Kazakos, D. (1977), 'Recursive estimation of prior probabilities using a mixture', *IEEE T. Inform. Theory* **23**(2), 203–211.

Marriott, F. (1975), 'Separating mixtures of normal distributions', *Biometrics* **31**(3), 767–769.

McCullagh, P. & Yang, J. (2008), 'How many clusters?', *Bayesian Analysis* **3**(1), 101–120.

McLachlan, G. (1992), *Discriminant Analysis and Statistical Pattern recognition*, Wiley Series in Probability and Statistics, John Wiley & Son.

McLachlan, G. & Peel, D. (2000), *Finite Mixture Models*, Wiley Series in Probability and Statistics, John Wiley & Son.

Medvedovic, M., Succop, P., Shukla, R. & Dixon, K. (2001), 'Clustering mutational spectra via classification likelihood and Markov chain Monte Carlo algorithms', *J. Agricultural, Biological, Environmental Statist.* **6**(1), 19–37.

Pauler, D. K. (1998), 'The schwartz criterion and related methods for normal linear models', *Biometrika* **85**(1), 13–27.

R Development Core Team (2010), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *http://www.r-project.org*

Raftery, Adrian E., N. M. N. S. J. M. & Krivitsky, P. N. (2007), 'Estimating the integrated likelihood via posterior simulation using the harmonic mean identity', *Bayesian Statistics* **8**, 1–45.

Rayment, P. (1972), 'The identification problem for a mixture of observations from two normal populations', *Technometrics* **14**(4), 911–918.

Redner, R. A. & Walker, H. F. (1984), 'Mixture densities, maximum likelihood and the EM algorithm', *SIAM Rev.* **26**(2), 195–239.

Schwarz, G. (1978), 'Estimating the dimension of a model', *Ann. Stat.* **6**(2), 461–464.

Scott, A. & Symons, M. (1971), 'Clustering methods based on likelihood ratio criteria', *Biometrics* **27**(2), 387–397.

Symons, M. (1981), 'Clustering criteria and multivariate normal mixtures', *Biometrics* **37**(1), 35–43.

Titterington, D., Smith, A. & Makov, U. (1985), *Statistical analysis of finite mixture distributions*, Wiley series in probability and mathematical statistics, John Wiley & Son.