Journal of Data Science **11**(2013), 157-179

On Choosing a Mixture Model for Clustering

Joseph Ngatchou-Wandji[1]* and Jan Bulla[2]
[1] *Université de Lorraine and EHESP de Rennes and* [2] *Université de Caen*

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 performance of our methods is 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: AIC, BIC, clustering, ICL, mixtures models.

1. Introduction

Because of their ability to represent relationships in data, finite mixture models are commonly used for summarizing distributions. In the field of 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 comprising a penalty for model complexity. The AIC [1, 2] and the BIC [26] criteria are based on such likelihoods, as well as the algorithm provided by [11] for estimating a mixture model.

For assessing the number of clusters arising from a Gaussian mixture model, [5, 6] used a penalized completed likelihood (CL). However, the associated criterion tends to overestimate the correct number of clusters when there is no

*Corresponding author.

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 [4], 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 methods for determining the clusterings of data and a mixture model can be found, for instance, in [15], [10], [7], [20], [12], or [17].

In this paper, we propose two alternative approaches, based on the AIC and BIC criteria applied to the classification likelihood. In a certain sense, these are close to [12], 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. 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 $\boldsymbol{x} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n) \in \mathbb{R}^{dn}$ are a sample from a probability distribution with density of the form

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

where K is the number of components of the mixture, the p_k's represent the mixing proportions and the components $\phi_k(\cdot|\boldsymbol{a}_k)$'s are density functions, possibly of different nature[1], each with a known form and depending on the parameter vector $\boldsymbol{a}_k$. The notation m stands for the joint components, which characterize the nature of the mixture. Finally, $\theta := (\theta_1, \theta_2) := ((p_1, \cdots, p_K), (\boldsymbol{a}_1, \cdots, \boldsymbol{a}_K))$ represents the full parameter vector of the mixture (m, K) at hand. The most

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

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, $\phi_k(\cdot|\boldsymbol{a}_k) = \phi(\cdot|\boldsymbol{a}_k)$ is a d-variate Gaussian density with $\boldsymbol{a}_k = (\mu_k, \Sigma_k)$ for $k = 1, \cdots, K$.

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

$$\boldsymbol{y} = (\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n) = ((\boldsymbol{x}_1, \boldsymbol{z}_1), \cdots, (\boldsymbol{x}_n, \boldsymbol{z}_n)),$$

with $\boldsymbol{z} = (\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n)$ representing the missing data. For more details, we refer the reader to [29]. Note that $\boldsymbol{z}_i = (\boldsymbol{z}_{i1}, \cdots, \boldsymbol{z}_{iK})$ is a K-dimensional vector such that $\boldsymbol{z}_{ik}$ takes the value 1 if $\boldsymbol{x}_i$ arises from the component k, and takes the value 0 if not for $i = 1, \cdots, n$. Obviously, the vector $\boldsymbol{z}$ defines a partition $C = \{C_1, \cdots, C_K\}$ of the data $\boldsymbol{x} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$, with $C_k = \{\boldsymbol{x}_i | \boldsymbol{z}_{ik} = 1, i = 1, \cdots, n\}$. If $\boldsymbol{z}$ 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 $\phi_k(\cdot; \boldsymbol{a}_k)$. Therefore, the likelihood conditional on $\boldsymbol{z}$ would have a form allowing for easy inference. Unfortunately, $\boldsymbol{z}$ is in general not observed and has to be estimated.

There are many ways for estimating $\boldsymbol{z}$. For instance, [24, 15, 27, 28] treat the vector $\boldsymbol{z}$ as a parameter, which is estimated jointly with K and θ by maximizing the likelihood function

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

where

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

The drawback of this method is that all possible clusters of the data in K groups have to be considered, which may be computationally costly. Additionally, [16] points out an inconsistency of the parameter estimates, and, $\boldsymbol{z}$ is formally treated as a parameter rather than a vector of missing observations. A Bayesian estimator of $\boldsymbol{z}$ is also defined in [28]. Another, more popular method, is the so-called MAP (maximum a posteriori) method, described as follows. For $i = 1, \cdots, n$ and $k = 1, \cdots, K$, let $t_{ik}(\theta)$ denote the conditional probability that $\boldsymbol{x}_i$ arises from the k^{th} mixture component. Then, one can easily show that

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

Let $\hat{\theta}$ be the maximum likelihood estimate of θ. Under some regularity conditions, the so-called EM algorithm [29] allows the computation of this estimator, by means of which, $\boldsymbol{z}_{ik}$ can be derived by

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

for $i = 1, \cdots, n$ and $k = 1, \cdots, K$. Additionally, more approaches for estimating $\boldsymbol{z}$ exist, see, e.g., [18], [12], [19], [20], [13] or [17]. The estimates $\hat{\boldsymbol{z}}$ 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 the likelihood function (2), has already been recalled and commented in the previous section (see [28] 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 [11]. 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 [4]. This corresponds to the model $(\hat{m}, \hat{K})$ such that

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

where

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

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 $f(\boldsymbol{x}, \boldsymbol{z}|m, K, \theta)$ the likelihood function (2). For a BIC-like approximation of the right-hand side of (6), [4] proposed to select the model which maximizes

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

where $d_{m,K}$ stands for the dimension of the space $\Theta_{m,K}$, and $\hat{\theta}^* = \arg\max_\theta f(\boldsymbol{x}, \boldsymbol{z}|m, K, \theta)$. Since $\boldsymbol{z}$ is not observed, it is substituted by $\hat{\boldsymbol{z}}$ given in (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

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

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 $\boldsymbol{z}$ for $\hat{\boldsymbol{z}}$ 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 the clustering of given data $\boldsymbol{x}$. 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 (2), given that the vector $\boldsymbol{z}$ and thus $\theta_1 = (p_1, \cdots, p_K)$ 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 $\boldsymbol{z} = (\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n)$ given, the data are partitioned into K classes $C_1, C_2, \cdots, C_K$. Moreover, let $n_k = \sum_{i=1}^n z_{ik} = |C_k|$ for all $k = 1, \cdots, K$, where z_{ik} is the k^{th} component of $\boldsymbol{z}_i$, $i = 1, \cdots, n$. 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 $\hat{\theta}_1 = (\hat{p}_1, \cdots, \hat{p}_K)$. Then, the likelihood and log-likelihood functions of θ_2 given $(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$ can be approximated by

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

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

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

Let $d_{\boldsymbol{a}_k}$ denote the length of the vector $\boldsymbol{a}_k$, and $\Theta_{m,K}^{(k)} \subset \mathbb{R}^{d_{a_k}}$ for all $k =$

$1, \cdots, K$. In what follows, we suppose that $\theta_2 \in \Theta^*_{m,K} = \Theta^{(1)}_{m,K} \times \cdots \times \Theta^{(K)}_{m,K}$ and

$$\pi(\theta_2|m,K) = \pi_1(\boldsymbol{a}_1|m,K) \times \cdots \times \pi_K(\boldsymbol{a}_K|m,K). \tag{11}$$

We also suppose that the $\phi_k(\cdot|\boldsymbol{a}_k)$ are identifiable and differentiable up to order 2. Then, the integrated likelihood is defined by

$$\begin{aligned}\ell(m,K) &= \int_{\Theta^*_{m,K}} \ell(m,K,\theta_2)\pi(\theta_2|m,K)d\theta_2 \\ &= \prod_{k=1}^{K} \hat{p}_k \int_{\Theta^{(k)}_{m,K}} \prod_{\boldsymbol{x}_j \in C_k} \phi_k(\boldsymbol{x}_j|\boldsymbol{a}_k)\pi_k(\boldsymbol{a}_k|m,K)d\boldsymbol{a}_k, \end{aligned} \tag{12}$$

which follows from the likelihood function (9).

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

$$L(m,K,\theta_2) \approx \sum_{k=1}^{K} \left(\sum_{\boldsymbol{x}_j \in C_k} \log \phi_k(\boldsymbol{x}_j|\hat{\boldsymbol{a}}_k) - d_{\boldsymbol{a}_k} \right) + \sum_{k=1}^{K} n_k \log \hat{p}_k. \tag{13}$$

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

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

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

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

follows for each $k = 1, \cdots, K$ (see, e.g., [2]). □

Theorem 2. Assume that $\boldsymbol{z}$ is known, that the n_k are large enough for $k = 1, 2, \cdots, K$, and that the prior on θ_2 has the form (11) with noninformative $\pi_k(\boldsymbol{a}_k|m,K)$'s. Then, the logarithm of the integrated likelihood can be approximated by

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

Proof. See the Appendix.

The first term on the right-hand sides of (13) and (14) resemble sums of AIC and BIC, respectively, and depend on $\boldsymbol{z}$ and θ_1. Therefore, we denote these quantities by SAIC$(m, K|\theta_1, \boldsymbol{z})$ and SBIC$(m, K|\theta_1, \boldsymbol{z})$, which stands for "Sum of AIC/BIC". They can be represented as

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

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

We would like to remark that penalty terms related to those utilized for SAIC and SBIC can be found, for instance, in [21] and [23].

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

```
• Initialize $\boldsymbol{z}$ (for example, by the $k$-means algorithm)
• Repeat
    – for $k = 1, \cdots, K$, compute $n_k = \sum_{i=1}^{n} z_{ik}$ and $p_k = n_k/n$, thus $\theta_{1K} = (p_1, \cdots, p_K)$
    – maximize the log-likelihood given in (10) with respect to $\theta_2 = (\boldsymbol{a}_1, \cdots, \boldsymbol{a}_K)$ and denote by $\theta_{2K}$ the vector for which the likelihood reaches the maximum.
    – for $i = 1, \cdots, n$ and $k = 1, \cdots, K$, compute
          $\boldsymbol{z}_{ik} = \begin{cases} 1, & \text{if} \arg\max_{\ell \in \{1, \cdots, K\}} t_{i\ell}(\theta_K) = k, \\ 0, & \text{otherwise,} \end{cases}$
      where $\theta_K = (\theta_{1K}, \theta_{2K})$ and $t_{ik}$ is defined by (4)
  Until the log-likelihood remains constant
• Return $\boldsymbol{z}_K$ and $\theta_{1K}$.
```

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 $K = 2, \cdots, K_{\max}$
    - Compute θ_{1K} and $\boldsymbol{z}_K$ (with the above algorithm)
    - Compute $\mathrm{SAIC}(m, K|\theta_{1K}, \boldsymbol{z}_K)$ or $\mathrm{SBIC}(m, K|\theta_{1K}, \boldsymbol{z}_K)$
- Select $(\hat{m}, \hat{K})$ and $\boldsymbol{z}_{\hat{K}}$ by

$$(\hat{m}, \hat{K}) = \arg\max_{m,K} \mathrm{SAIC}(m, K|\theta_{1K}, \boldsymbol{z}_K),$$

or

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

The first step of the procedure above consists in defining a value for $K_{\max}$. However, for practical purposes another possibility might be to start with a "small" $K_{\max}$ and monitor the evolution of the SAIC/SBIC values. If SAIC/SBIC attains its highest value for $K_{\max}$, the user may step-wise increase this quantity, as the previous estimation results remain unaffected.

4. Numerical Examples

In this section, we primarily study the performance of the SAIC and SBIC criteria in comparison to the BIC resulting from a second model-based clustering algorithm by [13]. 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 analysis was carried out with R 2.10.1 [22], using version 3.3.1 of package Mclust. All code is available from the authors upon request.

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, called mb1 in the following, we follow two approaches. On the one hand, we use the k-means algorithm (function kmeans in R) to estimate an initial trajectory of $\boldsymbol{z}$, 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 $\boldsymbol{z}$ (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 ∼30% for the models with three/four components. These models have been removed, as they do not really utilize three respectively four components. Table 1 presents the results, showing an almost constant SAIC. Thus, according to this criterion, the parsimonious 3-component model should be selected. The SBIC attains the highest value for two components, therefore the model with two components is chosen. Here, we set $K_{\max} = 4$ because both SBIC and SAIC do not increase anymore when increasing the number of states from three to four.

Table 1: Model selection by SAIC/SBIC

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

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.

For comparison with a standard algorithm for model-based clustering, we also fitted models using the R package mclust [13]. This algorithm, called mb2 in the following, is initialized by hierarchical clustering and selects an appropriate model using 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” [4]. 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

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

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

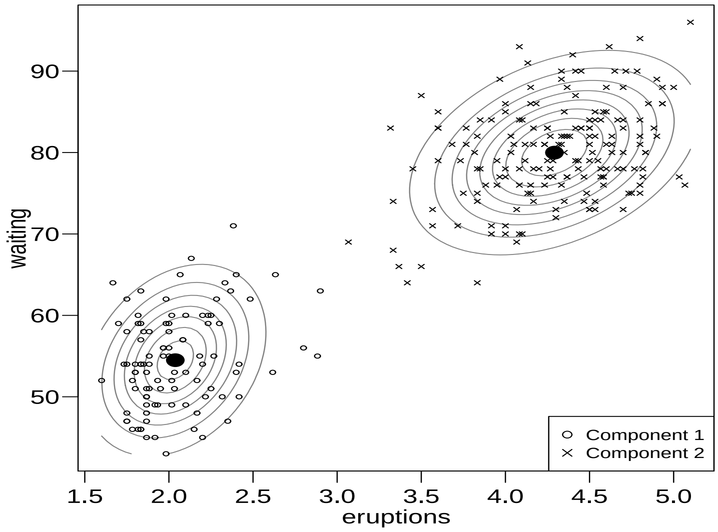

Figure 1: Clustering of Old Faithful Geyser data

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.

The estimated values of $\boldsymbol{z}$ 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 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 a 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 a very poor initialization of the components. Convergence problems mainly occur because less than three observations belong to one of the components, such that the variance-covariance matrix cannot be estimated

[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 $\mu_1 = \begin{pmatrix} 2.09 \\ 61.3 \end{pmatrix}$, $\mu_2 = \begin{pmatrix} 1.87 \\ 54.4 \end{pmatrix}$, $\mu_3 = \begin{pmatrix} 4.29 \\ 80.0 \end{pmatrix}$, $\mu_4 = \begin{pmatrix} 2.07 \\ 49.7 \end{pmatrix}$, $\Sigma_1 = \begin{pmatrix} 0.0851 & 0.414 \\ 0.414 & 9.74 \end{pmatrix}$, $\Sigma_2 = \begin{pmatrix} 0.0150 & 0.157 \\ 0.157 & 2.15 \end{pmatrix}$, $\Sigma_3 = \begin{pmatrix} 0.169 & 0.918 \\ 0.918 & 35.9 \end{pmatrix}$. $\Sigma_4 = \begin{pmatrix} 0.0733 & 0.505 \\ 0.505 & 10.2 \end{pmatrix}$.

anymore. This phenomenon is mostly present at the initialization stage, but also happens rarely during the iteration steps. 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. Keeping in mind "Garbage in, garbage out", this behaviour may however be viewed 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 [3, 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

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

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

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.

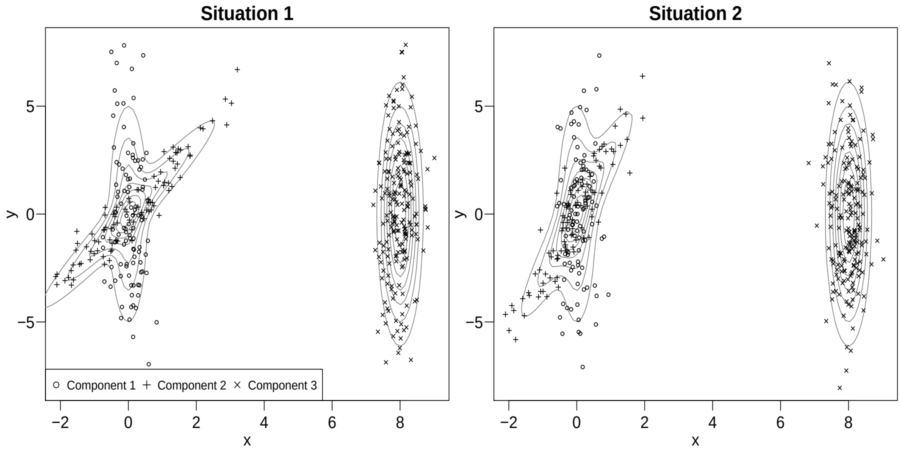

Figure 2: Examples for Situation 1 and 2

The figure shows examples for Situation 1 (left panel) and 2 (right panel) for the Monte Carlo experiment inspired by [4]. The contours result from the two underlying true Gaussian densities.

For each simulated sample, we executed our algorithm mb1. In order to initialize mb1, we use the k-means algorithm as stated in the previous section. Additionally, the algorithm mb2 by [13] 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").

Table 2 shows the results for the three criteria and the two algorithms. If an algorithm did not converge for either 2, 3, or 4 components, this sample was excluded from the analysis. The second and sixth column, respectively, of the table contain the number of excluded samples. The remaining columns present the proportions with which 2, 3, and 4 components were selected by BIC, SAIC, and SBIC in the two situations. These values result from evaluating only those samples without convergence difficulties.

Table 2: Number of components selected by BIC/SAIC/SBIC

	Situation 1				Situation 2			
	excl.	2	3	4	excl.	2	3	4
BIC	2	52.0%	8.8%	39.2%	2	52.4%	15.5%	32.1%
SAIC	1	92.6%	5.8%	1.6%	1	94.2%	5.4%	0.4%
SBIC	1	92.6%	5.8%	1.6%	1	94.2%	5.4%	0.4%

This table displays the proportion with which 2, 3, and 4 components were selected by BIC, SAIC, and SBIC in Situation 1 (left) and 2 (right). Here, $n_1 = n_2 = 100$ and $n_3 = 200$. Additionally, it shows the number of samples excluded due to non-convergence.

In Situation 1, both SAIC and SBIC show a strong preference for a model with two components. This model is also the preferred model by the BIC, however, it selects the models with more components more often than SAIC/SBIC do. In Situation 2, similar tendencies occur. The preferred model by all criteria has two components, but SAIC/SBIC exhibits a much stronger tendency towards parsimonious 2-component-models than BIC. Furthermore, BIC selects the model $K = 4$ less often than in the previous situation. In view of the results of [3], one may confirm that "BIC selects the number of mixture components needed to provide a good approximation to the density, rather than the number of clusters as such". On the contrary, SAIC/SBIC seem to be more suitable to identify the number of clusters in the two situations. Finally, it may be noted that the results differ slightly from those of the BIC reported by [4]. These authors demonstrate the selection of $K = 3$ in the majority of cases for Situation 1 (92% for BIC and 88% for ICL), whereas in Situation 2 BIC mostly choses $K = 3$ (92%) and ICL mainly prefers $K = 3$ (88%). Still, both their estimation algorithms and samples are different to ours.

To further investigate this setting, we repeated the previous experiment with reduced sample size. More precisely, $n_1 = n_2 = 50$ and $n_3 = 100$, and tendencies similar to those in the previous situation occur. SAIC/SBIC prefer the more parsimonious model with $K = 2$ more often than BIC. However, BIC selects $K = 2$ more often in Situation 2 than in Situation 1.

Table 3: Number of components selected by BIC/SAIC/SBIC

	Situation 1				Situation 2			
	excl.	2	3	4	excl.	2	3	4
BIC	12	52.1%	15.8%	32.2%	17	77.4%	12.8%	9.7%
SAIC	8	89.0%	8.1%	2.9%	0	94.8%	4.6%	0.6%
SBIC	8	89.0%	8.1%	2.9%	0	96.4%	3.0%	0.6%

This table displays the proportion with which 2, 3, and 4 components were selected by BIC, SAIC, and SBIC in Situation 1 (left) and 2 (right). Here, $n_1 = n_2 = 50$ and $n_3 = 100$. Additionally, it shows the number of samples excluded due to non-convergence.

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 $n_1 = n_2 = 250$. In the other two settings the proportions differ: 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 settings.

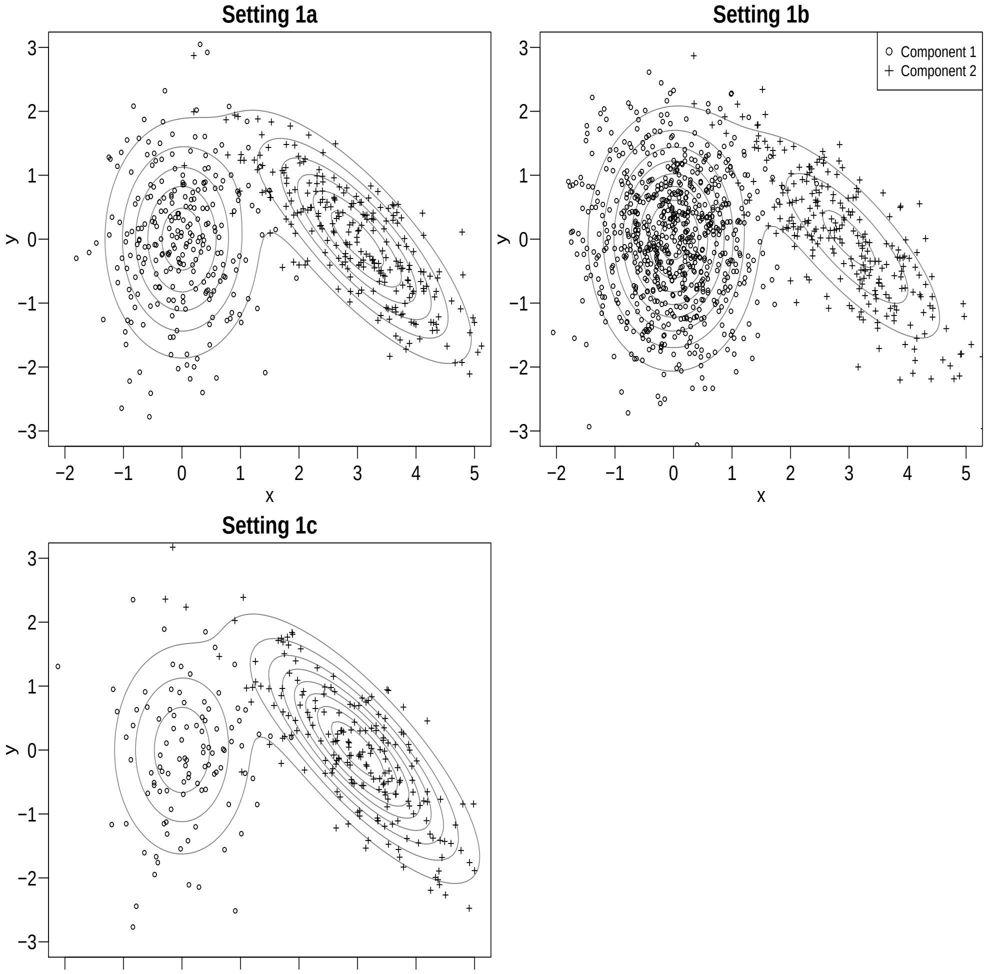

Figure 3: Examples for Settings 1a and 1b

The figure shows examples for the first Monte Carlo experiment. The upper left panel shows Setting 1a, the upper right panel Setting 1b, and the lower panel Setting 1c. The contours result from the two underlying true Gaussian densities.

For each simulated sample, we executed the algorithms mb1 and mb2 for models with 2 and 3 components, where mb1 was initialized by the k-means algorithm. Then, we calculated the frequencies for selecting each model, which Table 4 summarizes. None of the algorithms failed to converge for any sample. The 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 $K = 2$ components in 100% of the cases in all three settings. It may be noted that the performance of SAIC and SBIC are equivalent in Setting 1a and 1c, selecting the model with two components in 99.4% and 99.6%, respectively, of all cases. 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, 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 $\boldsymbol{z}$, 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 for the correct model.

Table 4: Number of components selected by BIC/SAIC/SBIC

	Setting 1a		Setting 1b		Setting 1c	
	2	3	2	3	2	3
BIC	100.0%	0.0%	100.0%	0.0%	100.0%	0.0%
SAIC	99.4%	0.6%	95.8%	4.2%	99.6%	0.4%
SBIC	99.4%	0.6%	95.6%	4.4%	99.6%	0.4%

This table displays the proportions for selecting 2 and 3 components by BIC, SAIC, and SBIC in Setting 1a (left), 1b (middle) and 1c (right).

Last, we address the number of iterations required for the algorithm mb1 to converge. Table 5 displays the average number of iterations for all settings and number of components. 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 [9, 25].

Table 5: Number of iterations required by mb1

	Setting 1a		Setting 1b		Setting 1c	
	2	3	2	3	2	3
number iter.	7.7	13.5	5.6	27.8	8.3	9.5
s.d.	2.7	6.7	1.7	15.9	2.8	5.8

This table displays number of iterations required by the algorithm mb1 to converge in Setting 1a (left), 1b (middle) and 1c (right), respectively.

4.4 Monte Carlo Experiment 2

This 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 are $n_1 = 250$, $n_2 = 500$, 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.

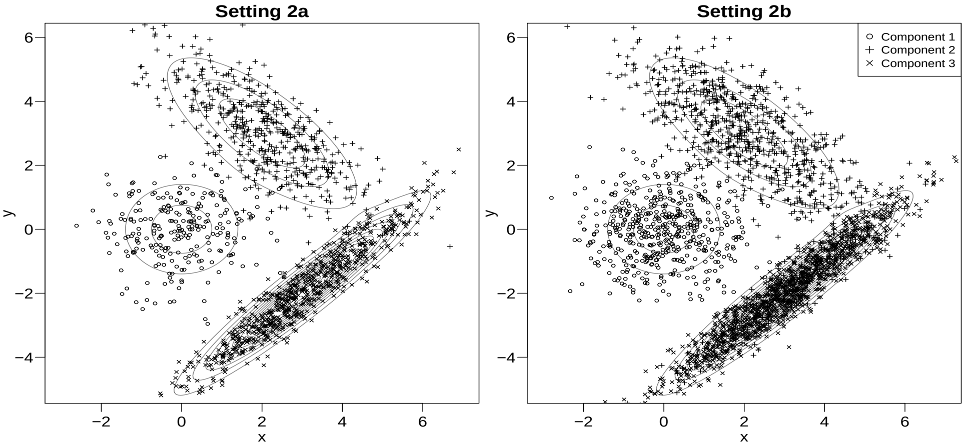

Figure 4: Examples for Settings 2a and 2b

The figure shows examples for the second Monte Carlo experiment. The left panel (Setting 2a) displays the case $n = 1750$, and in the right panel (Setting 2b) $n = 3500$. The contours result from the three underlying true Gaussian densities.

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, with 14.8 (7.53) and 14.6 (7.14) iterations for Setting 2a and 2b, respectively, in the 3-component case. For two components, the corresponding figure roughly halves, and for four components it doubles. Moreover, the algorithm sometimes fails to converge in these settings (see Table 6). It may be noted that the large majority of convergence problems can be attributed to the (wrong) models with $K = 4$. We excluded these cases from further analysis to avoid any bias.

Table 6: Number of components selected by BIC/SAIC/SBIC

	Setting 2a				Setting 2b			
	excl.	2	3	4	excl.	2	3	4
BIC	0	0.0%	91.0%	9.0%	0	0.0%	90.4%	9.6%
SAIC	39	0.0%	79.2%	20.8%	8	0.0%	72.4%	27.6%
SBIC	39	0.0%	78.5%	21.5%	8	0.2%	71.5%	28.3%

This table displays the proportions with which 2, 3, and 4 components were selected by BIC, SAIC, and SBIC in Setting 2a (left) and 2b (right). Here, $n_1 = n_2 = 100$ and $n_3 = 200$. Additionally, it shows the number of samples excluded due to non-convergence.

Table 6 also summarizes the model selection results. The performance of SAIC and SBIC is almost identical, selecting the 3-component-model in approximately 79% and 72% of all samples in Setting 2a and 2b, respectively. The performance of mb2 is slightly better, selects $K = 3$ in approximately 91% of all cases in both settings. As before, further analysis of those cases in which mb1 selects four components reveals that the number of observations in the fourth component is rather small. In 75% of these cases, the number of observations is less than 26 and 15 in Setting 2a and 2b, respectively. Thus, 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 [13]. In what follows, we define the classification performance as the fraction of correctly classified observations, which naturally only makes sense in a sample where the memberships to a component are known. 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?

In this section, we consider the settings described in the previous Sections 4.3 and 4.4, each comprising 500 simulated samples. For each setting, we analyzed those fitted models with the correct number of components, that is $K = 2$ and $K = 3$ for the Settings 1a/1b/1c and 2a/2b, respectively. The algorithm mb1 is initialized in two different ways: a) by the k-means and b) by the path estimated by mb2. Then again, mb2 uses the default setting, i.e., a hierarchical clustering. The first three columns of Table 7 report the average classification errors of mb1 - initialized by k-means 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 the 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 7: Classification error

scenario	mb1 (k-means)	mb1 (mb2)	mb2	p-values
1a	2.75	2.62	2.63	0.117/0.972
1b	2.34	2.29	2.25	0.019/0.342
1c	2.84	2.51	2.52	0.046/0.741
2a	1.94	1.93	1.91	0.305/0.425
2b	1.88	1.88	1.87	0.576/0.758

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.

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 [8]) based on the classification likelihood, which can be initialized with an arbitrary $\boldsymbol{z}$. For its derivation, we have used the idea of [12] who propose a procedure based on the mixture likelihood. One of the main advantages of such algorithms is that good estimators of $\boldsymbol{z}$, 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 parameter. A preliminary estimator of this value, which could be used as the starting point, is not easy to obtain in general.

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 [4] and the CEM algorithm described in [12].

3. The ICL procedure uses the maximum likelihood (ML) estimator of the pa-

rameters 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 and SAIC, respectively, because $\boldsymbol{z}$ 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 penalization term that is different to 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 [12]. However, their approach utilizes the mixture likelihood and ML estimator, whereas we use the classification likelihood and empirical estimators of the mixing proportions combined 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.

Appendix: Proof of Theorem 2

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

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

and

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

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

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

and

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

Then we obtain

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

by the Laplace transformation [see, e.g., 14]. Since $\exp\{g(\boldsymbol{a}_k)\}$, the likelihood of the class C_k behaves like the product $\prod_{\boldsymbol{x}_j \in C_k} \phi_k(\boldsymbol{x}_j|\boldsymbol{a}_k)$ for all $k = 1, \cdots, K$, which increases with n_k whilst $\pi_k(\boldsymbol{a}_k|m,K)$ is constant, one can substitute the vector $\boldsymbol{a}^*_k$ by $\hat{\boldsymbol{a}}_k = \arg\max\{(1/n_k)\prod_{\boldsymbol{x}_j \in C_k} \phi_k(\boldsymbol{x}_j|\boldsymbol{a}_k)\}$ and the matrix $A_{\boldsymbol{a}^*_k}$ by the Fisher information matrix $I_{\hat{\boldsymbol{a}}_k}$ defined by

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

Then follows:

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

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

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

follows. Thus, from this approximation and (12) follows

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

□

Acknowledgements

We would like to thank the associate editor and two anonymous referees whose comments led to improve the paper. Moreover, thanks to I. Bulla for his attentive proofreading and corrections.

References

[1] Akaike, H. (1973). Information theory as an extension of the maximum likelihood principle. In *Proceedings of the Second International Symposium on Information Theory* (Edited by B. N. Petrov and F. Csaki), 267-281. Akademial Kiado, Budapest.

[2] Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control* **19**, 716-723.

[3] Baudry, J. P., Raftery, A. E., Celeux, G., Lo, K. and Gottardo, R. (2010). Combining mixture components for clustering. *Journal of Computational and Graphical Statistics* **19**, 332-353.

[4] Biernacki, C., Celeux, G. and Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, 719-725.

[5] Biernacki, C. and Govaert, G. (1997). Using the classification likelihood to choose the number of clusters. *Computing Science and Statistics* **29**, 451-457.

[6] Biernacki, C. and Govaert, G. (1999). Choosing models in model-based clustering and discriminant analysis. *Journal of Statistical Computation and Simulation* **64**, 49-71.

[7] 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 *Information and Classification* (Edited by O. Opitz, B. Lausen and R. Klar), 40-54. Springer-Verlag, Heidelberg.

[8] Celeux, G. and Govaert, G. (1992). A classification em algorithm for clustering and two stochastic versions. *Computational Statistics & Data Analysis* **14**, 315-332.

[9] Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B* **39**, 1-38.

[10] Englemann, L. and Hartigan, J. A. (1969). Percentage points for a test for clusters. *Journal of the American Statistical Association* **64**, 1647-1648.

[11] Figueiredo, M. A. T., Leitão, J. M. N. and Jain, A. K. (1993). On fitting mixture models. In *Lecture Notes in Computer Science* (Edited by E. R. Hancock and M. Pelillo), 54-69. Springer-Verlag, Heidelberg.

[12] Fraley, C. and Raftery, A. E. (1998). How many clusters? which clustering method? answer via model-based cluster analysis. *Computer Journal* **41**, 578-588.

[13] Fraley, C. and 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.

[14] Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association* **90**, 773-795.

[15] Kazakos, D. (1977). Recursive estimation of prior probabilities using a mixture. *IEEE Transactions on Information Theory* **23**, 203-211.

[16] Marriott, F. H. C. (1975). Separating mixtures of normal distributions. *Biometrics* **31**, 767-769.

[17] McCullagh, P. and Yang, J. (2008). How many clusters? *Bayesian Analysis* **3**, 101-120.

[18] McLachlan, G. J. (1992). *Discriminant Analysis and Statistical Pattern Recognition.* Wiley Series in Probability and Statistics. John Wiley & Sons, New York.

[19] McLachlan, G. J. and Peel, D. (2000). *Finite Mixture Models.* Wiley Series in Probability and Statistics. John Wiley & Sons, New York.

[20] Medvedovic, M., Succop, P., Shukla, R. and Dixon, K. (2000). Clustering mutational spectra via classification likelihood and Markov chain Monte Carlo algorithms. *Journal of Agricultural, Biological and Environmental Statistics* **6**, 19-37.

[21] Pauler, D. K. (1998). The Schwartz criterion and related methods for normal linear models. *Biometrika* **85**, 13-27.

[22] R Development Core Team. (2010). *R: A Language and Environment for Statis- tical Computing.* R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org.

[23] Raftery, A. E., Newton, M. A., Satagopan, J. M. and Krivitsky, P. N. (2007). Estimating the integrated likelihood via posterior simulation using the harmonic mean identity. *Bayesian Statistics* **8**, 1-45.

[24] Rayment, P. R. (1972). The identification problem for a mixture of observations from two normal populations. *Technometrics* **14**, 911-918.

[25] Redner, R. A. and Walker, H. F. (1984). Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review* **26**, 195-239.

[26] Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics* **6**, 461-464.

[27] Scott, A. J. and Symons, M. J. (1971). Clustering methods based on likelihood ratio criteria. *Biometrics* **27**, 387-397.

[28] Symons, M. J. (1981). Clustering criteria and multivariate normal mixtures. *Biometrics* **37**, 35-43.

[29] Titterington, D. M., Smith, A. F. M. and Makov, U. E. (1985). *Statistical Analysis of Finite Mixture Distributions.* Wiley Series in Probability and Mathematical Statistics. John Wiley & Son, New York.

Received July 19, 2012; accepted September 18, 2012.

Joseph Ngatchou-Wandji
Institut Élie Cartan de Nancy
Université de Lorraine and EHESP de Rennes
B.P. 70239 54506 Vandoeuvre-lès-Nancy Cedex, France
joseph.ngatchou-wandji@univ-lorraine.fr

Jan Bulla
LMNO
Université de Caen
CNRS UMR 6139 14032 Caen Cedex, France
bulla@math.unicaen.fr