Available online at www.sciencedirect.com

ScienceDirect

Computational Statistics & Data Analysis 51 (2006) 587 – 600

COMPUTATIONAL STATISTICS & DATA ANALYSIS

www.elsevier.com/locate/csda

Model-based cluster and discriminant analysis with the MIXMOD software

Christophe Biernacki[a,*], Gilles Celeux[b], Gérard Govaert[c], Florent Langrognet[d]

[a] *UMR 8524, CNRS & Université de Lille 1, 59655 Villeneuve d'Ascq, France*
[b] *INRIA Futurs, 91405 Orsay, France*
[c] *UMR 6599, CNRS & Université de Technologie de Compiègne, 60205 Compiègne, France*
[d] *UMR 6623, CNRS & Université de Franche-Comté, 25030 Besançon, France*

Received 8 January 2005; received in revised form 19 December 2005; accepted 19 December 2005
Available online 10 January 2006

Abstract

The Mixture Modeling (MIXMOD) program fits mixture models to a given data set for the purposes of density estimation, clustering or discriminant analysis. A large variety of algorithms to estimate the mixture parameters are proposed (EM, Classification EM, Stochastic EM), and it is possible to combine these to yield different strategies for obtaining a sensible maximum for the likelihood (or complete-data likelihood) function. MIXMOD is currently intended to be used for multivariate Gaussian mixtures, and fourteen different Gaussian models can be distinguished according to different assumptions regarding the component variance matrix eigenvalue decomposition. Moreover, different information criteria for choosing a parsimonious model (the number of mixture components, for instance) are included, their suitability depending on the particular perspective (cluster analysis or discriminant analysis). Written in C++, MIXMOD is interfaced with SCILAB and MATLAB. The program, the statistical documentation and the user guide are available on the internet at the following address: `http://www-math.univ-fcomte.fr/mixmod/index.php`.

Keywords: Gaussian models; EM-like algorithms; Model selection

1. Introduction

Because of their flexibility, finite mixture distributions have become a very popular approach when modeling a wide variety of random phenomena. In particular, finite mixture models provide powerful tools for density estimation, clustering and discriminant analysis. Mixture models are increasingly being used in a variety of disciplines including astronomy, biology, genetics, economics, engineering and marketing, and consequently computer programs for statistical modeling with finite mixture distributions are increasingly sought after. MIXMOD is one such program, designed principally for model-based cluster analysis and supervised classification. This article sets out to give a general presentation of the statistical features of this mixture program.

MIXMOD is publicly available under the GPL license and is distributed for different platforms (Linux, Unix, Windows). It is an object-oriented package built around the C++ language. It is interfaced with widely-used mathematical software

[*] Corresponding author. Tel.: +33 3204 36876; fax: +33 3204 34302.
E-mail addresses: christophe.biernacki@math.univ-lille1.fr (C. Biernacki), gilles.celeux@inria.fr (G. Celeux), gerard.govaert@utc.fr (G. Govaert), florent.langrognet@math.univ-fcomte.fr (F. Langrognet).

0167-9473/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2005.12.015

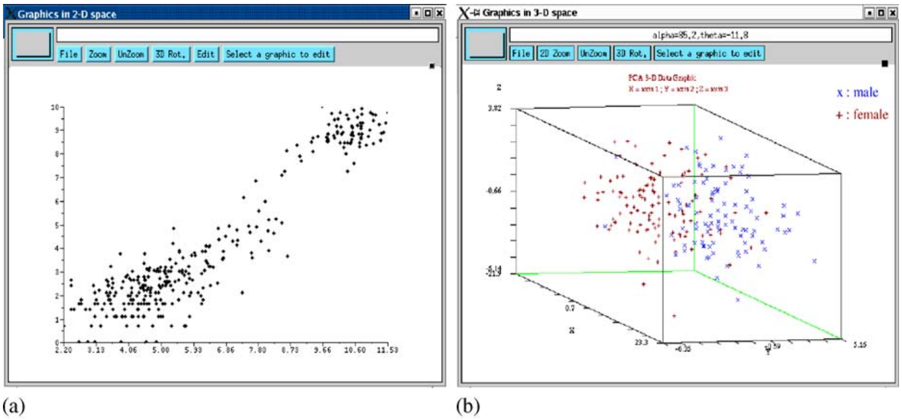

Fig. 1. Selected examples: (a) the French *départements* data set for clustering, and (b) the *borealis* data set for discriminant analysis.

MATLAB and SCILAB. It was developed jointly by INRIA, the Besançon math laboratory and the Heudiasyc laboratory at Compiègne.

In its current version MIXMOD includes only multivariate Gaussian mixture models, but a generalization to other types of mixture distributions, including the latent class model for the statistical analysis of discrete data, is planned for future versions. The main features of the current version are the following:

- Three levels of use from beginner to expert.
- Fourteen geometrically meaningful Gaussian mixture models derived from different variance matrix parameterizations.
- Estimation of mixture parameters with various EM and EM-like algorithms, provided with different initialization strategies.
- Availability of numerous criteria to select a reliable model depending on the particular density estimation, cluster or discriminant analysis perspective.
- Numerous graphical displays (in 1D, 2D and 3D) including densities, isodensities, classifier or cluster descriptions, etc. in canonical or principal component analysis (PCA) space.

This article is not intended to replace either the user guide or the statistical documentation that the reader can find on the web. It aims to provide a synthetic overview of MIXMOD's features by combining a short presentation of its statistical characteristics with some selected examples.

The first data set is intended to illustrate MIXMOD's features in a clustering context. Fig. 1(a) displays log-population versus log-density (in inhabitants/km^2) for 312 towns in three French *départements* (Biernacki et al., 2000), namely *Seine-Saint-Denis* and *Hauts de Seine*, which form part of the densely-populated Paris conurbation, along with the rural *département* of *Haute-Corse*.

The second data set is intended to illustrate MIXMOD's features in a discriminant analysis context. This data set concerns 204 seabirds belonging to the *borealis* subspecies of the Procellaridae (petrel) family, for which five morphological variable measurements were obtained (Biernacki et al., 2002): culmen (bill length), tarsus, wing and tail lengths, and culmen depth. Fig. 1(b) displays males (55%) and females (45%) in the first PCA 3D space.

2. Some technical features of MIXMOD

The development of the program began in 2001, and the latest release of MIXMOD (MIXMOD 1.6) is composed of 40 C++ classes and 20000 lines of code, and is interfaced with SCILAB and MATLAB.

The (`http://www-math.univ-fcomte.fr/mixmod/index.php`) website has recently been improved and includes the following sections: Download, Documentation, FAQ, Bugs, News, ...

2.1. MIXMOD operating modes

The MIXMOD program can be used in three different ways.

- MIXMOD as a GUI: the `mixmodGraph` function, available in SCILAB and MATLAB, brings up the MIXMOD Graphical User Interface. This function is the easiest way to get to know MIXMOD, but several MIXMOD features are not available in this mode.
- MIXMOD as a SCILAB or MATLAB function: the `mixmod` function can be called like any standard function in both the SCILAB and the MATLAB environments. It includes a number of optional inputs, and allows certain parameters to be specified more precisely than the `mixmodGraph` function allows. Moreover, graphical displays can be obtained with the `mixmodView` function.
- MIXMOD as a command line: this method of running MIXMOD, using input and output files, is not available in SCILAB and MATLAB. It is intended for users who are familiar with a shell environment (Linux, Unix, or Windows).

In this document examples are presented using the `mixmod` function in a SCILAB environment (the second method in the list above).

2.2. Data representation in MIXMOD

MIXMOD may handle up to three complementary data structures, depending on the available data:

- Individuals: each individual is represented by a row and each variable by a column.
- Partition: each row is the indicator vector of the different class memberships for an individual. Its *j* coordinate is `1` if the individual belongs to class *j*, otherwise `0`. A row of `0`s indicates an individual with an unknown class assignment.
- Weights: each row gives the weight of an individual.

2.3. Performance of successive versions of MIXMOD

Reducing CPU time remains a major objective in the design and implementation of MIXMOD. MIXMOD 1.6 is approximately 3 times faster than MIXMOD 1.1, and this trend is set to continue in future releases and versions.

3. Fourteen Gaussian mixture models

3.1. Eigenvalue parameterization of variance matrices

In MIXMOD the observations $\boldsymbol{x} \in \mathbb{R}^d$ are assumed to arise from a mixture distribution with density

$$f(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{k=1}^{K} p_k \varphi\left(\boldsymbol{x};\boldsymbol{\mu}_k,\Sigma_k\right), \tag{1}$$

where $p_k \geqslant 0$ for $k=1,\ldots,K$ and $\sum_{k=1}^{K} p_k = 1$ are the mixing proportions, $\varphi(\boldsymbol{x};\boldsymbol{\mu},\Sigma)$ is the density of a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and variance matrix Σ, and $\boldsymbol{\theta} = \left(p_1,\ldots,p_K,\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_K,\Sigma_1,\ldots,\Sigma_K\right)$ denotes the vector parameter to be estimated.

In this model, the density of the *k*th mixture component is the Gaussian density $\varphi\left(\boldsymbol{x};\boldsymbol{\mu}_k,\Sigma_k\right)$

$$\varphi\left(\boldsymbol{x};\boldsymbol{\mu}_k,\Sigma_k\right) = (2\pi)^{-d/2}|\Sigma_k|^{-1/2}\exp\left\{-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)'\Sigma_k^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}_k\right)\right\}. \tag{2}$$

This Gaussian density model leads to an ellipsoidal class with center $\boldsymbol{\mu}_k$ and whose geometric characteristics can be deduced from the eigenvalue decomposition of the variance matrix Σ_k.

Table 1
Characteristics and identifiers of the 28 Gaussian mixture models available in MIXMOD

Model	Family	Prop.	Volume	Shape	Orient.
$\left[p\lambda DAD'\right]$	General	Equal	Equal	Equal	Equal
$\left[p\lambda_k DAD'\right]$			Free	Equal	Equal
$\left[p\lambda DA_k D'\right]$			Equal	Free	Equal
$\left[p\lambda_k DA_k D'\right]$			Free	Free	Equal
$\left[p\lambda D_k AD'_k\right]$			Equal	Equal	Free
$\left[p\lambda_k D_k AD'_k\right]$			Free	Equal	Free
$\left[p\lambda D_k A_k D'_k\right]$			Equal	Free	Free
$\left[p\lambda_k D_k A_k D'_k\right]$			Free	Free	Free
$[p\lambda B]$	Diagonal	Equal	Equal	Equal	Axes
$\left[p\lambda_k B\right]$			Free	Equal	Axes
$\left[p\lambda B_k\right]$			Equal	Free	Axes
$\left[p\lambda_k B_k\right]$			Free	Free	Axes
$[p\lambda I]$	Spherical	Equal	Equal	Equal	NA
$\left[p\lambda_k I\right]$			Free	Equal	NA
$\left[p_k\lambda DAD'\right]$	General	Free	Equal	Equal	Equal
$\left[p_k\lambda_k DAD'\right]$			Free	Equal	Equal
$\left[p_k\lambda DA_k D'\right]$			Equal	Free	Equal
$\left[p_k\lambda_k DA_k D'\right]$			Free	Free	Equal
$\left[p_k\lambda D_k AD'_k\right]$			Equal	Equal	Free
$\left[p_k\lambda_k D_k AD'_k\right]$			Free	Equal	Free
$\left[p_k\lambda D_k A_k D'_k\right]$			Equal	Free	Free
$\left[p_k\lambda_k D_k A_k D'_k\right]$			Free	Free	Free
$\left[p_k\lambda B\right]$	Diagonal	Free	Equal	Equal	Axes
$\left[p_k\lambda_k B\right]$			Free	Equal	Axes
$\left[p_k\lambda B_k\right]$			Equal	Free	Axes
$\left[p_k\lambda_k B_k\right]$			Free	Free	Axes
$\left[p_k\lambda I\right]$	Spherical	Free	Equal	Equal	NA
$\left[p_k\lambda_k I\right]$			Free	Equal	NA

Following Banfield and Raftery (1993) and Celeux and Govaert (1995), each mixture component variance matrix can be written

$$\Sigma_k = \lambda_k D_k A_k D'_k, \tag{3}$$

where $\lambda_k = |\Sigma_k|^{1/d}$, D_k is the matrix of eigenvectors of Σ_k and A_k is a diagonal matrix, such that $|A_k| = 1$, with the normalized eigenvalues of Σ_k on the diagonal in a decreasing order. The parameter λ_k determines the *volume* of the kth cluster, D_k its *orientation* and A_k its *shape*. By allowing some of these quantities to vary between clusters, parsimonious and easily interpreted models useful in describing various clustering or classification situations can be obtained. Varying the assumptions concerning the parameters λ_k, D_k and A_k leads to eight general models of interest. For instance, different volumes and equal shapes and orientations are assumed by requiring that $A_k = A$ (A unknown) and $D_k = D$ (D unknown) for each mixture component. This model is denoted $\left[\lambda_k DAD'\right]$. With this convention, $\left[\lambda D_k AD'_k\right]$ indicates a model whose components have equal volumes and shapes and different orientations. Another family of interest uses the assumption that the variance matrices Σ_k are diagonal. For the parameterization (3), this means that the orientation matrices D_k are permutation matrices. In this paper these diagonal variance matrices are conventionally denoted $\Sigma_k = \lambda_k B_k$, where B_k is a diagonal matrix with $|B_k| = 1$. This particular parameterization gives rise to four models: $[\lambda B]$,$\left[\lambda_k B\right]$, $\left[\lambda B_k\right]$ and $\left[\lambda_k B_k\right]$. The final family of models assumes spherical shapes, namely $A_k = I$, I denoting the identity matrix. Here, two parsimonious models can be considered: $[\lambda I]$ and $\left[\lambda_k I\right]$. A total of 14 Gaussian models are obtained in this way.

Note that, in the following, models $[\lambda DAD']$ and $\left[\lambda D_k A_k D_k'\right]$ may also be written in the more compact forms $[\lambda C]$ and $\left[\lambda C_k\right]$, respectively. Similarly, models $\left[\lambda_k C\right]$ and $\left[\lambda_k C_k\right]$ are equivalent to models $\left[\lambda_k DAD'\right]$ and $\left[\lambda_k D_k A_k D_k'\right]$, respectively.

3.2. Constraints on proportions

Apart from these geometrical features, another important parameter of the kth mixture component is its mixing weight or proportion p_k. Two typical assumptions are considered with regard to the proportions: we assume either equal or free proportions over the mixture components. Combining these alternative proportion assumptions with the 14 previous models leads to 28 different models denoted $[p\lambda I]$, $\left[p_k\lambda I\right]$, $\left[p\lambda_k DAD'\right]$, etc., using the convention previously defined. All those models, summarized in Table 1, are available in MIXMOD in both the unsupervised and supervised contexts.

3.3. Links with some standard criteria

These different mixture models do not only have a simple geometric interpretation. They also reveal in a new light some standard clustering criteria that have been proposed without any reference to a statistical model. For instance, the K-means criterion of Ward (1963) can easily be derived from the simplest $[p\lambda I]$ model. The $[p\lambda DAD']$ model corresponds to the criterion suggested by Friedman and Rubin (1967), and the models $\left[p\lambda_k DAD'\right]$, $\left[p\lambda_k DA_k D'\right]$ and $\left[p\lambda_k D_k A_k D_k'\right]$ correspond to other documented clustering criteria (see for instance Scott and Symons, 1971; Diday and Govaert, 1974; Maronna and Jacovkis, 1974; Schroeder, 1976). In discriminant analysis, models $[p\lambda C]$ and $\left[p\lambda_k C_k\right]$ lead, respectively, to the standard linear and quadratic classifiers (see for instance McLachlan, 1992).

4. Model-based clustering

4.1. The clustering problem

Data considered in MIXMOD for clustering are n vectors $\mathbf{x}=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\}$ in $\mathbb{R}^d$. The aim is to estimate an unknown partition $\mathbf{z}$ of $\mathbf{x}$ into K clusters, $\mathbf{z}=\{\mathbf{z}_1,\ldots,\mathbf{z}_n\}$ denoting n indicator vectors or labels $\mathbf{z}_i=(z_{i1},\ldots,z_{iK})$, $i=1,\ldots,n$ with $z_{ik}=1$ if $\boldsymbol{x}_i$ belongs to the kth cluster and 0 otherwise. The underlying idea of model-based clustering is to link each cluster to each of the mixture components. Usually all the labels $\mathbf{z}_i$ are unknown. Nevertheless, partial labeling of data is possible, and MIXMOD is able to handle situations where the data set $\mathbf{x}$ is divided into two subsets $\mathbf{x}=\left\{\mathbf{x}^\ell,\mathbf{x}^u\right\}$ where $\mathbf{x}^\ell=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m\}$ $(1\leqslant m\leqslant n)$ are data with known labels $\mathbf{z}^\ell=\{\mathbf{z}_1,\ldots,\mathbf{z}_m\}$, whereas $\mathbf{x}^u=\{\boldsymbol{x}_{m+1},\ldots,\boldsymbol{x}_n\}$ are data with unknown labels $\mathbf{z}^u=\{\mathbf{z}_{m+1},\ldots,\mathbf{z}_n\}$. Moreover, MIXMOD allows a weight to be specified for each statistical unit. This option is useful, for instance, for handling grouped or frequency data.

In the Gaussian model-based clustering framework handled by MIXMOD, complete data $(\boldsymbol{x}_i,\mathbf{z}_i)$ $(i=1,\ldots,n)$ are assumed to arise from the joint probability distribution $\prod_{k=1}^K\left(p_k\varphi\left(\boldsymbol{x}_i;\boldsymbol{\mu}_k,\Sigma_k\right)\right)^{z_{ik}}$. In this statistical context, MIXMOD includes two commonly used maximum likelihood (m.l.) approaches: first the mixture approach, which involves maximizing over $\boldsymbol{\theta}$ the density of the observed data set, and secondly the classification approach which involves maximizing over $\boldsymbol{\theta}$ and $\mathbf{z}^u$ the density of the complete data set.

4.2. Estimation by the mixture approach

The mixture approach means maximizing over $\boldsymbol{\theta}=\left(p_1,\ldots,p_K,\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_K,\Sigma_1,\ldots,\Sigma_K\right)$ the observed loglikelihood

$$L\left(\boldsymbol{\theta};\mathbf{x},\mathbf{z}^\ell\right)=\sum_{i=1}^m\sum_{k=1}^K z_{ik}\ln\left(p_k\varphi\left(\boldsymbol{x}_i;\boldsymbol{\mu}_k,\Sigma_k\right)\right)+\sum_{i=m+1}^n\ln\left(\sum_{k=1}^K p_k\varphi\left(\boldsymbol{x}_i;\boldsymbol{\mu}_k,\Sigma_k\right)\right). \tag{4}$$

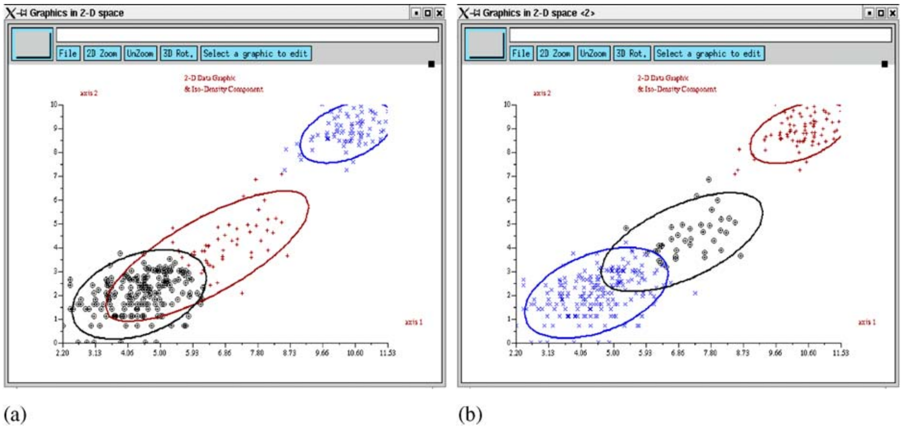

Fig. 2. Estimated partition and isodensity components for the French *départements* data set: (a) with the EM procedure and (b) with the CEM procedure.

A partition $\hat{\mathbf{z}}^u$ is derived from the m.l. estimator $\hat{\boldsymbol{\theta}}$ using a *Maximum A Posteriori* (MAP) procedure which consists of assigning each $\boldsymbol{x}_i$ in $\mathbf{x}^u$ to the component k providing the largest conditional probability

$$t_k\left(\boldsymbol{x}_i;\hat{\boldsymbol{\theta}}\right)=\frac{\hat{p}_k\varphi\left(\boldsymbol{x}_i;\hat{\boldsymbol{\mu}}_k,\hat{\Sigma}_k\right)}{\sum_{k'=1}^{K}\hat{p}_{k'}\varphi\left(\mathbf{x}_i;\hat{\boldsymbol{\mu}}_{k'},\hat{\Sigma}_{k'}\right)} \tag{5}$$

that $\boldsymbol{x}_i$ arises from it. Maximizing $L\left(\boldsymbol{\theta};\mathbf{x},\mathbf{z}^\ell\right)$ can be performed in MIXMOD via the EM algorithm of Dempster et al. (1977) or by a stochastic version of EM called SEM (see for instance Celeux and Diebolt, 1985; McLachlan and Krishnan, 1997). Three different ways of combining these algorithms are described in Section 7. Obviously, the estimator $\hat{\boldsymbol{\theta}}$, and consequently $\hat{\mathbf{z}}^u$, depend on both the chosen Gaussian mixture model and the number of clusters in question.

Example 1 (*French départements*). Fig. 2(a) displays the partition and isodensity component estimated with the EM algorithm for the Gaussian mixture model $\left[p_k\lambda_k DA_k D'\right]$ with three components.

4.3. Estimation using the classification approach

The second approach available in MIXMOD is the classification approach where the indicator vectors $\mathbf{z}^u$, identifying the mixture component origin, are treated as unknown parameters. This approach aims to maximize the complete loglikelihood

$$CL\left(\boldsymbol{\theta},\mathbf{z}^u;\mathbf{x},\mathbf{z}^\ell\right)=\sum_{i=1}^{n}\sum_{k=1}^{K}z_{ik}\ln\left(p_k\varphi\left(\boldsymbol{x}_i;\boldsymbol{\mu}_k,\Sigma_k\right)\right) \tag{6}$$

over both the parameter $\boldsymbol{\theta}$ and the labels $\mathbf{z}^u$. The CL criterion can be maximized by making use of a classification version of the EM algorithm, the so-called CEM algorithm (Celeux and Govaert, 1992) which includes a classification step (C-step) between the E and M steps. Section 7 looks at different strategies to derive the m.l. estimate of $\boldsymbol{\theta}$ which make use of this algorithm.

Example 2 (*French départements*). Fig. 2(b) displays the partition and isodensity component estimated with the CEM algorithm for the Gaussian mixture model $\left[p_k\lambda_k DA_k D'\right]$ with three components. The result should be compared with the solution obtained with the EM algorithm shown in Fig. 2(a).

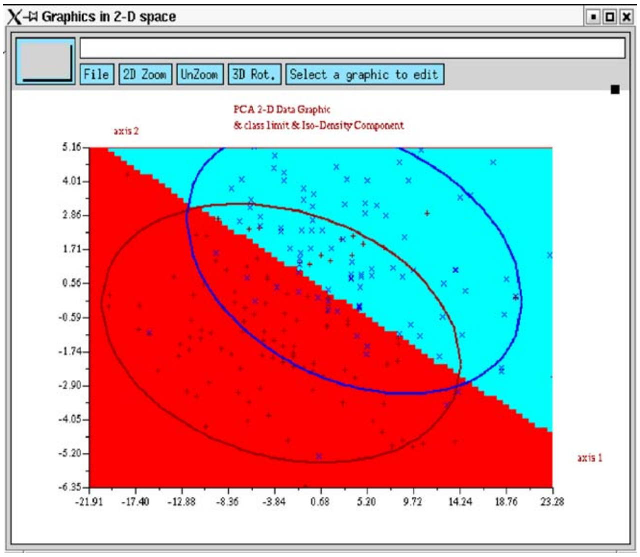

Fig. 3. Class limit, isodensity component and individuals for seabirds with model $\left[p_k \lambda_k D_k A_k D_k'\right]$ in the first PCA 2D space.

5. Model-based discriminant analysis

Data processed by MIXMOD for discriminant analysis consists of a training data set of n vectors $(\mathbf{x}, \mathbf{z}) = \{(\boldsymbol{x}_1, z_1), \ldots, (\boldsymbol{x}_n, z_n)\}$, where $\boldsymbol{x}_i$ belongs to $\mathbb{R}^d$, and z_i is the indicator vector of the class containing the statistical unit i. The aim is to design from this training set a classifier to estimate the class $\mathbf{z}_{n+1}$ of any new observation with vector $\boldsymbol{x}_{n+1}$ in $\mathbb{R}^d$ and an unknown label. It should be noted that weighting the data is also available in the discriminant analysis context.

The statistical assumptions are those used in the clustering situation, and the mixture parameter $\boldsymbol{\theta}$ is estimated by maximizing the complete loglikelihood (6). Since $\mathbf{z}$ is completely known, the m.l. estimate $\hat{\boldsymbol{\theta}}$ of the model parameter $\boldsymbol{\theta}$ reduces to a single maximization step. Any new point $\mathbf{x}_{n+1}$ can be assigned to one of the K classes using the MAP procedure with $\hat{\boldsymbol{\theta}}$.

In summary, discriminant analysis is performed in MIXMOD by the two following steps:

- **M-step:** Computation of the m.l. estimate $\hat{\theta}$ of $\boldsymbol{\theta}$ by maximizing the complete loglikelihood (6).
- **MAP-step:** Assignment of a new point $\mathbf{x}$ to one of the K classes by the following rule:

$$k(\mathbf{x}) = \underset{k}{\operatorname{argmax}}\, t_k\left(\mathbf{x}; \hat{\boldsymbol{\theta}}\right).$$

Example 3 (*seabirds*). Fig. 3 displays the classifier boundary, isodensity component and individuals in the first PCA 2D space for the most general Gaussian mixture model $\left[p_k \lambda_k D_k A_k D_k'\right]$.

6. An overview of MIXMOD algorithms

6.1. EM algorithm

The EM algorithm aims to maximize the mixture likelihood in an unsupervised context. Starting from an initial arbitrary parameter $\boldsymbol{\theta}^0$, the qth iteration of the EM algorithm consists of repeating the following E and M steps.

- **E-step:** Compute the conditional probabilities $t_{ik}^q = t_k\left(\boldsymbol{x}_i; \boldsymbol{\theta}^{q-1}\right)$ that $\boldsymbol{x}_i$ belongs to the kth cluster $(i = m+1, \ldots, n)$ by using the current value $\boldsymbol{\theta}^{q-1}$ of the parameter.

- **M-step:** The m.l. estimate $\boldsymbol{\theta}^q$ of $\boldsymbol{\theta}$ is updated using the conditional probabilities t_{ik}^q as conditional mixing weights. This step is highly dependent on the Gaussian model used. Detailed formulae for the 14 Gaussian mixture models available in MIXMOD are given in Celeux and Govaert (1995).

6.2. SEM algorithm

In the stochastic EM (SEM) algorithm an S-step is incorporated between the E- and the M-steps of EM. This is a restoration step for the unknown labels which are simulated according to their current conditional distribution. In the M-step the estimate of parameter $\boldsymbol{\theta}$ is updated by maximizing the completed loglikelihood corresponding to the restored labels.

SEM does not converge pointwise. It generates a Markov chain whose stationary distribution is more or less concentrated around the m.l. parameter estimator. A natural parameter estimate from a SEM sequence $\left(\boldsymbol{\theta}^q\right)_{q=1,\dots,Q}$ is the mean $\sum_{q=r+1}^{Q}\boldsymbol{\theta}^q/(Q-r)$ of the iteration values (the first r burn-in iterates are discarded in the calculation of this mean). An alternative estimate uses the parameter value leading to the highest likelihood in an SEM sequence.

6.3. CEM algorithm

The classification EM (CEM) algorithm incorporates a classification step between the E- and the M-steps of EM. This classification step involves assigning each point to one of the K components from a MAP procedure for the current parameter value. Unlike the stochastic step in SEM, this classification step is deterministic, since unknown labels are restored with the MAP procedure. As in SEM, the M-step consists of updating the parameter estimate $\boldsymbol{\theta}$ by maximizing the completed loglikelihood corresponding to the restored labels.

CEM is a *K-means*-like algorithm and, unlike EM, it converges in a finite number of iterations. CEM does not maximize the observed loglikelihood L (4), but maximizes in $\boldsymbol{\theta}$ and $\mathbf{z}^u$ the complete loglikelihood CL (6). As a consequence, CEM is not meant to converge to the m.l. estimate of $\boldsymbol{\theta}$, and yields inconsistent estimates of the parameters especially when the mixture components are overlapping or are in disparate proportions (McLachlan and Peel, 2000, Section 2.21).

6.4. M-step and MAP functions

These two functions are useful mainly in discriminant analysis. The M-step is devoted to the m.l. estimation of the mixture parameter $\boldsymbol{\theta}$ when the labels $\mathbf{z}$ are known. This maximization step is simply the M-step used in the SEM and the CEM algorithms. The MAP procedure has already been described in Section 4.2.

7. Strategies for using EM and related algorithms

7.1. Initialization strategies

There are five different ways to start an algorithm in MIXMOD. Other than for the first of these, which is deterministic, it is recommended that the set {starting strategy/running algorithm} be repeated several times in order to select the solution providing the best value of the criterion to be maximized. The criterion is the observed loglikelihood when the running algorithm is EM or SEM, and the completed loglikelihood when the running algorithm is CEM.

- An algorithm can be started from user specifications as a particular partition $\mathbf{z}^{u0}$ or a particular mixture parameter value $\boldsymbol{\theta}^0$. This possibility is available for EM, SEM and CEM.
- An algorithm can be started from a random mixture parameter value $\boldsymbol{\theta}^0$. In MIXMOD this random initial position is obtained by drawing component means from the data set at random, fixing proportions to equality and choosing a diagonal common variance matrix where the diagonal is equal to the empirical variance of each variable. Since this is probably the most frequently employed way of initiating EM, CEM or SEM, it can be regarded as a reference strategy.

- The EM algorithm can be started from the position providing the highest completed likelihood after many runs of CEM started with random positions and stopped with stability of the *CL* criterion (6). The number of restarts of CEM is a priori unknown and depends on the assignment of iterations chosen by the user (see Biernacki et al., 2003).
- The EM algorithm can be started from the position providing the highest likelihood after many short runs of EM started with random positions. By a *short run* of EM we mean that the algorithm is stopped as soon as $\left(L^q - L^{q-1}\right)/\left(L^q - L^0\right) \leqslant 10^{-2}$, L^q denoting the observed loglikelihood at the qth iteration. Here 10^{-2} represents a default threshold value which can be chosen on pragmatic grounds. The number of restarts of short runs of EM is a priori unknown and depends on the assignment of iterations chosen by the user (see Biernacki et al., 2003).
- The EM algorithm can be started from the position providing the highest likelihood in a sequence of SEM started with random positions and with an assignment of iterations chosen by the user (see Biernacki et al., 2003).

7.2. *Stopping rules*

In MIXMOD, there are three ways to stop an algorithm.

- The EM, SEM and CEM algorithms can be stopped after a predefined number of iterations.
- An algorithm can be stopped using a threshold for the relative change of the criterion in question (the loglikelihood L (4) or the completed loglikelihood CL (6)). When using EM this possibility is not recommended, since EM can encounter slow convergence situations. It is recommended that CEM, which converges in a finite number of iterations, be stopped at stationarity.
- An algorithm can be stopped as soon as one of the two previous criteria is satisfied.

7.3. *Chained algorithms*

In MIXMOD it is easy to combine the EM, SEM and CEM algorithms at will. This possibility can yield original and efficient initialization strategies, as presented in Biernacki et al. (2003).

8. Model selection

It is of obvious interest to be able to select automatically a Gaussian mixture model M and the number K of mixture components. However, choosing a sensible mixture model will depend very much on the particular modeling aim. We therefore make a distinction between the density estimation, the cluster and the discriminant analysis perspectives.

8.1. *Density estimation and cluster analysis perspective*

In MIXMOD three criteria are available in an unsupervised setting: BIC, ICL and NEC. If no information on K is available, it is recommended to vary it between $K = 1$ and the smallest integer larger than $n^{0.3}$ (see Bozdogan, 1993).

When estimating density Bayesian information criterion (BIC) must be preferred. Denoting by $v_{M,K}$ the number of free parameters in the Gaussian mixture model M with K clusters, BIC is expressed by the following penalization of the maximum loglikelihood $L_{M,K}$:

$$\mathrm{BIC}_{M,K} = -2L_{M,K} + v_{M,K}\ln(n). \tag{7}$$

The couple (M, K) yielding the lowest value for BIC is chosen. Although standard sufficient regularity conditions for deriving BIC (Schwarz, 1978) are not fulfilled for mixtures, it has been proved, for a large family of mixtures, that the BIC criterion is consistent (Kéribin, 2000), and BIC has been shown to be efficient on practical grounds (see for instance Fraley and Raftery, 1998).

In the context of cluster analysis ICL and NEC can provide more parsimonious and robust answers. To take into account the ability of the mixture model to reveal a clustering structure in the data, as an alternative to the BIC criterion

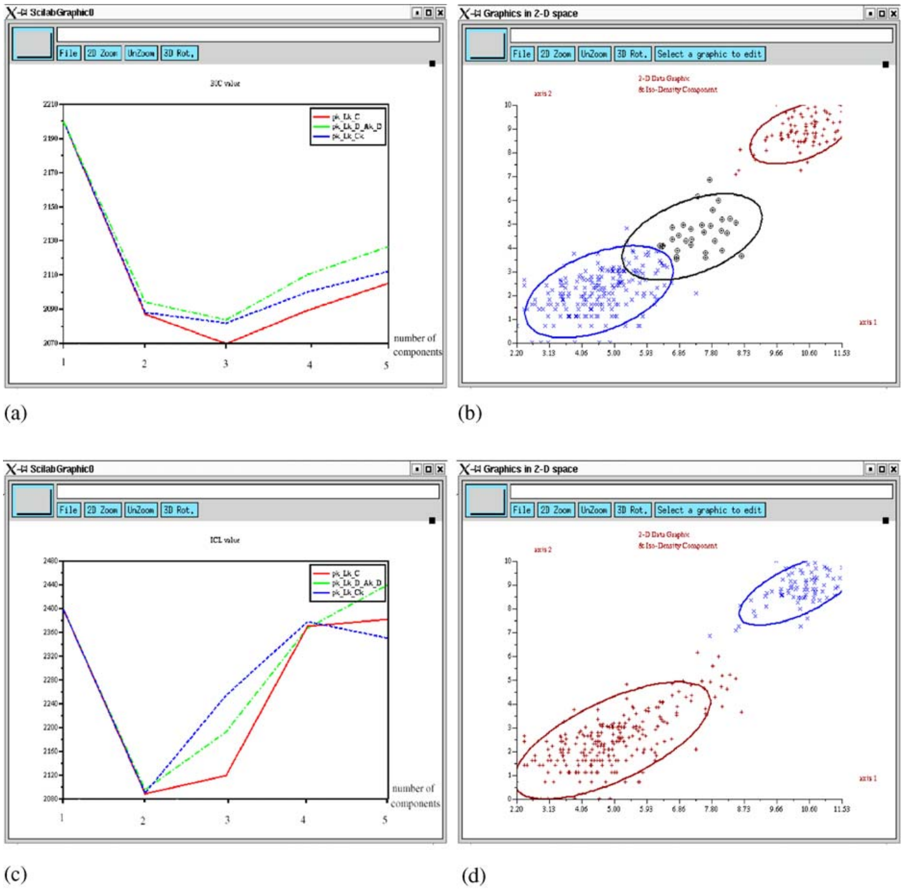

Fig. 4. Selection of a combination model-number of components for the French *départements* data set: (a) BIC values; (b) the associated optimal partition; (c) ICL values; (d) the associated optimal partition.

one may use the integrated complete-data likelihood (ICL) criterion (see Biernacki et al., 2000) expressed by

$$\mathrm{ICL}_{M,K} = \mathrm{BIC}_{M,K} - 2 \sum_{i=m+1}^{n} \sum_{k=1}^{K} \hat{z}_{ik} \ln\left(t_{ik}\right), \tag{8}$$

where $t_{ik} = t_k\left(\boldsymbol{x}_i; \hat{\boldsymbol{\theta}}_{M,K}\right)$ (with $\hat{\boldsymbol{\theta}}_{M,K}$ the m.l. parameter estimate for model M and number of components K) and where $\hat{\mathbf{z}} = \mathrm{MAP}\left(\hat{\boldsymbol{\theta}}_{M,K}\right)$. This criterion, to be minimized, is simply the BIC criterion penalized by an entropy term which measures the overlap of the clusters. The normalized entropy criterion (NEC) criterion proposed by Celeux and Soromenho (1996) uses a similar entropy term $E_K = -\sum_{i=m+1}^{n} \sum_{k=1}^{K} t_{ik} \ln\left(t_{ik}\right)$, but this criterion is intended to be used principally in determining the number of mixture components K, rather than the model parameterization M (Biernacki and Govaert, 1999). The criterion, to be minimized, is expressed by

$$\mathrm{NEC}_K = \frac{E_K}{L_K - L_1}. \tag{9}$$

Note that NEC_1 is not defined. Biernacki et al. (1999) proposed the following efficient rule for dealing with this problem: Let $K^\star$ be the value minimizing NEC_K $\left(2 \leqslant K \leqslant K_{\mathrm{sup}}\right)$, K_{sup} being an upper bound for the number of mixture components. $K^\star$ clusters are chosen if $\mathrm{NEC}_{K^\star} \leqslant 1$, otherwise no clustering structures in the data are declared.

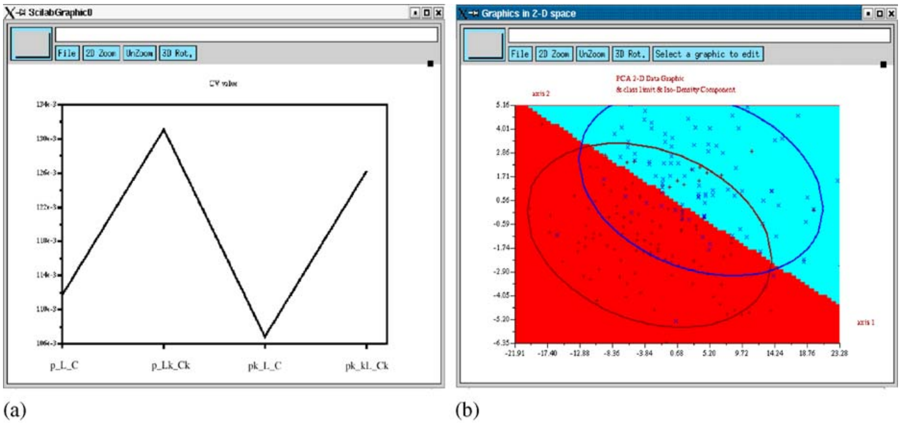

Fig. 5. Selection of a Gaussian mixture model for seabirds: (a) CV values and (b) associated optimal discriminant rule.

Example 4 (*French départements*). Five numbers of components ($K = 1 - 5$) and three Gaussian mixture models $\left[p_k\lambda_k DAD'\right]$, $\left[p_k\lambda_k DA_kD'\right]$ and $\left[p_k\lambda_k D_kA_kD_k\right]$ are considered. The EM algorithm is run for each combination model-K. Figs. 4 (a) and (b), respectively, display the BIC values for each of these combinations and the partition corresponding to the best combination selected by BIC. Figs. 4(c) and (d) give analogous displays for the ICL criterion.

8.2. Discriminant analysis perspective

In this situation the model M has to be selected but the number of mixture components is fixed. In MIXMOD two criteria are proposed in a supervised setting: BIC and the cross-validated error rate (CV). The CV criterion is specific to the supervised classification setting. It is defined by

$$\mathrm{CV}_M = \frac{1}{m}\sum_{i=1}^{m}\delta\left(\hat{\mathbf{z}}_i^{(i)}, \mathbf{z}_i\right), \tag{10}$$

where δ denotes the 0–1 cost and $\hat{\mathbf{z}}_i^{(i)}$ the group to which $\boldsymbol{x}_i$ is assigned when designing the classifier from the entire data set $(\mathbf{x}, \mathbf{z})$ without $(\boldsymbol{x}_i, \mathbf{z}_i)$. Fast estimation of the n discriminant rules is implemented in the Gaussian situation when $m = n$, i.e. when all labels are known (Biernacki and Govaert, 1999).

In MIXMOD, following an approach described in Bensmail and Celeux (1996), it is possible to select one of the 14 Gaussian mixture models by minimization of the cross-validated error rate. It should, however, be stressed that this cross-validated error rate is an optimistic estimate of the actual error rate. This is a situation where the method includes the selection of one model among several, and the actual error rate should therefore be assessed from an independent sample. Roughly speaking, three samples are needed: a *training* sample to estimate the parameters of the 14 models, a *validation* sample to choose one of the 14 models and a *test* sample to assess the actual error rate of the whole method. It means that when using cross validation to assess the error rate it is necessary to perform a *double* cross validation to get an unbiased estimate. In practice this kind of cross validation is painfully slow, and it is not currently implemented in MIXMOD. To assess a classifier involving the choice of a model in MIXMOD, it is necessary to discard at random a test sample from the whole data set. This test sample will be used to assess the actual error rate of the whole procedure.

Example 5 (*seabirds*). Four Gaussian mixture models $[p\lambda DAD']$, $\left[p\lambda_k D_kA_kD_k'\right]$, $\left[p_k\lambda DAD'\right]$ and $\left[p_k\lambda_k D_kA_kD_k'\right]$ are considered. Figs. 5 (a) and (b), respectively, display the CV values for each model and the classifier corresponding to the best model selected by CV.

9. Companion functions

The MATLAB and SCILAB environments provide high-level functions, typically generating graphical displays.

9.1. Graphical displays of criterion values

One of the optional outputs of the `mixmod` function is a 4D array providing values for all the requested criteria for all requested strategies, all requested numbers of mixture components and all requested Gaussian mixture models. From this array, simple criteria variations can be displayed in MIXMOD. Illustrations of this feature can be seen in Figs. 4(a), (c) and 5(a).

9.2. The MIXMODVIEW function for graphics

MIXMOD provides the `mixmodView` function for visualizing outputs. This function enables graphics generated from `mixmod` function outputs (density, isodensity, etc.) to be displayed in 1D, 2D and 3D space. The following graphics are available:

- Isodensity component and density mixture in the first PCA space.
- Class limit, isodensity component and individuals in the first PCA 2D space.
- Mixture density in the first PCA 2D space.
- Individuals and labels in the first PCA 3D space.

Many of these features have already been illustrated in previous examples. The following example shows the density display.

Example 6 (*French départements*). Figs. 6(a) and (b), respectively, display mixture density in the first PCA space and in the initial 2D space.

9.3. The PRINTMIXMOD function for summaries

The `printMixmod` function can be used to summarize `mixmod` function outputs. It displays a readable summary of output (input conditions, criterion value, loglikelihood, completed loglikelihood, parameter estimates, etc.).

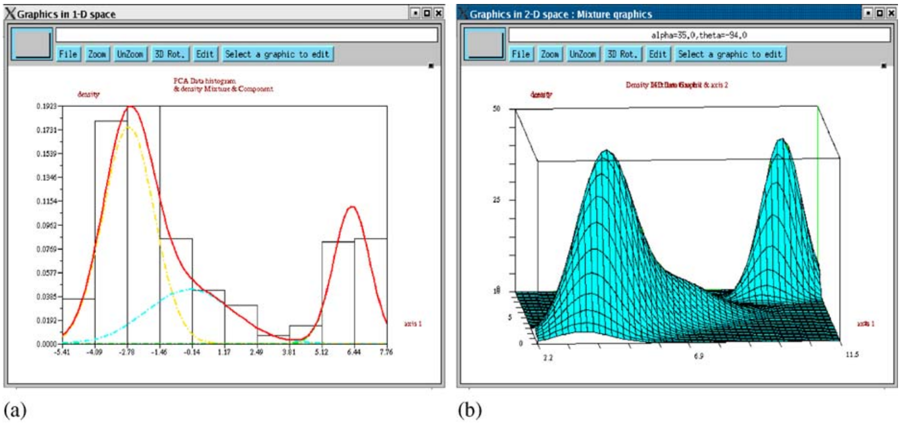

Fig. 6. Density mixture: (a) first PCA space and (b) initial 2D space.

9.4. The INPUTMIXMOD function for input facilities

The `inputMixmod` yields SCILAB or MATLAB structures which can be used by the `mixmod` function. It enables the criterion, Gaussian mixture models and strategy (initialization, algorithm, stopping rule) to be specified easily.

10. Further developments of MIXMOD

MIXMOD has become a relatively reliable and fast program for handling Gaussian mixtures. Users' remarks posted on the website have helped bugs to be identified and corrected, and the efficiency of the code has been improved with successive versions. Currently the emphasis is on reducing significantly the CPU time required by MIXMOD. All remarks and suggestions by users are appreciated, not only regarding the `mixmod` function, but also regarding secondary features such as `mixmodView`. The website is the ideal vehicle for collecting and exchanging this kind of information.

In the coming months version 2.0 of MIXMOD will become available. Version 2.0 adds clustering and discriminant analysis for multivariate binary or qualitative data, given that the use of such data is common in such important fields as ecology, psychology, text mining, and image analysis. In this context Bernoulli or multinomial distribution mixtures are employed, and some original parsimonious models are proposed. Looking further ahead, future versions will include a means of handling mixed data with both continuous and discrete variables in the same analysis.

Acknowledgments

The authors are grateful to Grégory Noulin and Yann Vernaz for their contribution in relation to the MIXMOD coding. They are also indebted to their respective institutions for their continued support. In particular, they are pleased to acknowledge the two-year ODL grant made by INRIA. They thank also David Lewis for the English language improvement and the reviewers for the suggestions that have helped to improve the presentation of this article.

References

Banfield, J.D., Raftery, A.E., 1993. Model-based Gaussian and non-Gaussian clustering. Biometrics 49, 803–821.

Bensmail, H., Celeux, G., 1996. Regularized Gaussian discriminant analysis through eigenvalue decomposition. J. Amer. Statist. Assoc. 91 (2), 1743–17448.

Biernacki, C., Govaert, G., 1999. Choosing models in model-based clustering and discriminant analysis. J. Statist. Comput. Simulation 64, 49–71.

Biernacki, C., Celeux, G., Govaert, G., 1999. An improvement of the NEC criterion for assessing the number of clusters in a mixture model. Pattern Recognition Lett. 20, 267–272.

Biernacki, C., Celeux, G., Govaert, G., 2000. Assessing a mixture model for clustering with the integrated completed likelihood. IEEE Trans. Pattern Analysis and Machine Intelligence 22 (7), 719–725.

Biernacki, C., Beninel, F., Bretagnolle, V., 2002. A generalized discriminant rule when training population and test population differ on their descriptive parameters. Biometrics 58 (2), 387–397.

Biernacki, C., Celeux, G., Govaert, G., 2003. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. Comput. Statist. Data Anal. 41, 561–575.

Bozdogan, H., 1993. Choosing the number of component clusters in the mixture-model using a new informational complexity criterion of the inverse-fisher information matrix. In: Opitz, O., Lauritzen, B., Klar, R. (Eds.), Information and Classification. Springer, Heidelberg, pp. 40–54.

Celeux, G., Diebolt, J., 1985. The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. Comput. Statist. Quart. 2, 73–82.

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

Celeux, G., Govaert, G., 1995. Gaussian parsimonious clustering models. Pattern Recognition 28 (5), 781–793.

Celeux, G., Soromenho, G., 1996. An entropy criterion for assessing the number of clusters in a mixture model. J. Classification 13, 195–212.

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

Diday, E., Govaert, G., 1974. Classification avec distance adaptative. C. R. Acad. Sci. Paris, Sér. A 278, 993–995.

Fraley, C., Raftery, A.E., 1998. How many clusters? Which clustering method? Answers via model-based cluster analysis. Comput. J. 41, 578–588.

Friedman, H.P., Rubin, J., 1967. On some invariant criteria for grouping data. J. Amer. Statist. Assoc. 62, 1159–1178.

Kéribin, C., 2000. Consistent estimation of the order of mixture models. Sankhyã Ser. A 1, 49–66.

Maronna, R., Jacovkis, P., 1974. Multivariate clustering procedure with variable metrics. Biometrics 30, 499–505.

McLachlan, G.J., 1992. Discriminant Analysis and Statistical Pattern Recognition. Wiley, New York.

McLachlan, G.J., Krishnan, K., 1997. The EM Algorithm. Wiley, New York.

McLachlan, G.J., Peel, D., 2000. Finite Mixture Models. Wiley, New York.

Schroeder, A., 1976. Analyse d'un mélange de distributions de probabilité de même type. Rev. Statist. Appl. 24 (1), 39–62.

Schwarz, G., 1978. Estimating the number of components in a finite mixture model. Ann. Statist. 6, 461–464.

Scott, A.J., Symons, M.J., 1971. Clustering methods based on likelihood ratio criteria. Biometrics 27, 387–397.

Ward, J., 1963. Hierarchical grouping to optimize an objective function. J. Amer. Statist. Assoc. 58, 236–244.