Supervised and unsupervised classification using mixture models

Stéphane Girard, Jerome Saracco

SUPERVISED AND UNSUPERVISED CLASSIFICATION USING MIXTURE MODELS

STÉPHANE GIRARD AND JÉRÔME SARACCO

ABSTRACT. This chapter is dedicated to model-based supervised and unsupervised classification. Probability distributions are defined over possible labels as well as over the observations given the labels. To this end, the basic tools are the mixture models. This methodology yields a posterior distribution over the labels given the observations which allows to quantify the uncertainty of the classification. The role of Gaussian mixture models is emphasized leading to Linear Discriminant Analysis and Quadratic Discriminant Analysis methods. Some links with Fisher Discriminant Analysis and logistic regression are also established. The Expectation-Maximization algorithm is introduced and compared to the K-means clustering method. The methods are illustrated both on simulated datasets as well as on real datasets using the **R** software.

1. INTRODUCTION

Classification is an important field of statistical learning which is usually divided into two main tasks. First, in the supervised learning approach, the goal is to estimate a function from inputs x to outputs y given a training set $\{(x_i, y_i), i = 1, \ldots, n\}$ where n is the number of examples. In most cases, $x \in \mathbb{R}^p$ and x is referred to as the features or the covariates. If the response variable is real-valued, then the estimation of the link function is a regression problem, see Fraix-Burnet and Valls-Gabaud (2014) for application to astrophysics. If the response variable is categorical, it is rather denoted by z, and it takes its value in a finite set: $z \in \{1, \ldots, K\}$. The estimation problem is called supervised classification, or discriminant analysis. Second, in the unsupervised learning approach, one only has $\{x_i, i = 1, \ldots, n\}$. In this case, the goal is not so well defined, it can be summarized as finding interesting patterns in the data. Unsupervised dimension reduction or unsupervised classification (also called clustering) can enter in this framework. We refer to Bishop (2006) and Hastie *et al.*, (2001) for accounts on statistical learning.

This chapter is dedicated to model-based supervised and unsupervised classification. We shall define a (prior) probability distribution $p(z)$ over possible labels z as well as over the observations x given the labels z, denoted by $p(x|z)$. To this end, the basic tools are the mixture models. This will allow us to build a posterior distribution $p(z|x)$ over the labels given the observations and thus to quantify the uncertainty of the classification. Supervised classification is addressed in Section 2. The role of Gaussian mixture models is emphasized leading to Linear Discriminant Analysis and Quadratic Discriminant Analysis methods. Some links with Fisher Discriminant Analysis and logistic regression are also established. Section 3 is dedicated to unsupervised classification. The Expectation-Maximization algorithm is introduced and compared to the K-means clustering method. Finally, some extensions are presented in Section 4.

2. Supervised classification

Recall that the dataset is denoted by $\{(x_1, z_1), \ldots, (x_n, z_n)\}$ where $x_i \in \mathbb{R}^p$ and $z_i \in \{1, ..., K\}$ for all $i = 1, \ldots, n$. It is assumed that this dataset is already split into K groups $\mathcal{C}_1, \ldots, \mathcal{C}_K$. The labels are thus supposed to be known and $z_i = k$ means $x_i \in \mathcal{C}_k$ for all $i = 1, \ldots, n$ and $k = 1, \ldots, K$. The goal of supervised classification is to affect a new data point $x \in \mathbb{R}^p$ to one of the K groups in an optimal way. Some further notations are introduced in the next subsubsection.

2.1. **Notations.** For all $k = 1, \ldots, K$, let us consider:

- the number of observations in class $\mathcal{C}_k$: $n_k = \sharp\{i = 1, \ldots, n \text{ s.t. } z_i = k\}$,
- the mean $\bar{x}_k \in \mathbb{R}^p$ of class $\mathcal{C}_k$ defined as

$$\bar{x}_k = \frac{1}{n_k} \sum_{z_i = k} x_i,$$

- the sample mean $\bar{x} \in \mathbb{R}^p$ which can be calculated using two equivalent formulas:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i = \frac{1}{n} \sum_{k=1}^{K} n_k \bar{x}_k,$$

- the sample $p \times p$ covariance matrix associated with class $\mathcal{C}_k$ defined as:

$$V_k = \frac{1}{n_k} \sum_{z_i = k} (x_i - \bar{x}_k)(x_i - \bar{x}_k)^t,$$

- the within-class $p \times p$ covariance matrix which is defined as the mean of the previous covariance matrices:

$$W = \frac{1}{n} \sum_{k=1}^{K} n_k V_k,$$

- the between-class $p \times p$ covariance matrix which is defined as the covariance matrix of the class means:

$$B = \frac{1}{n} \sum_{k=1}^{K} n_k (\bar{x}_k - \bar{x})(\bar{x}_k - \bar{x})^t,$$

- the sample $p \times p$ covariance matrix which can be calculated using two equivalent formulas:

$$V = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^t = B + W.$$

2.2. **Fisher Discriminant Analysis.** The goal of Fisher Discriminant Analysis (FDA) is to project the data in a subspace of dimension $d \ll p$ in order to visualize the classes structure. Let us denote by $a_1, \ldots, a_d$ the d axes in $\mathbb{R}^p$ spanning the Fisher discriminant subspace.

2.2.1. *Idea of the method in the case $d = 1$.* In this simplified framework, the Fisher discriminant subspace is one dimensional, and is spanned by only one axis a_1. The idea is find $a_1 \in \mathbb{R}^p$ such that the projected observations $a_1^t x_i$ belonging to different classes are well-separated. In other words, the goal is to maximize the ratio between the between projected variance and the sample projected variance. Mathematically, the sample projected variance on an axis $a \in \mathbb{R}^p$ is obtained as

$$\begin{aligned}
\text{var}(a^t x_1, \ldots, a^t x_n) &= \frac{1}{n}\sum_{i=1}^{n}[(a^t x_i) - (a^t \bar{x})]^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}[a^t(x_i - \bar{x})]^2 \\
&= \frac{1}{n}\sum_{i=1}^{n} a^t(x_i - \bar{x})(x_i - \bar{x})^t a \\
&= a^t\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^t\right)a \\
&= a^t V a.
\end{aligned}$$

A similar formula holds for the between projected variance and therefore, the axis a_1 is obtained by solving the following optimization problem:

$$a_1 = \underset{a \in \mathbb{R}^p}{\text{argmax}} \frac{a^t B a}{a^t V a}.$$

It is easily shown that the solution is closed-form: a_1 is the eigenvector of the matrix $V^{-1}B$ associated with the largest eigenvalue.

2.2.2. *Back to the general case $d \geq 1$.* As a straightforward extension of the case $d = 1$, one can show that Fisher discriminant subspace is spanned by $a_1, \ldots, a_d$ obtained as the d eigenvectors of $V^{-1}B$ associated with the d largest eigenvalues.

Remark. The dimension of the optimal projection subspace is at most $K - 1$.

This property is a consequence of $\text{Rank}(B) \leq K - 1$ (since B is the covariance matrix of K observations) which implies $\text{Rank}(V^{-1}B) \leq K - 1$.

Remark. Both matrices $V^{-1}B$ and $W^{-1}B$ have the same (ordered) eigenvectors.

This fact can be proved as follows. Let a be an eigenvector of $V^{-1}B$ associated with the eigenvalue λ. Thus $V^{-1}Ba = \lambda a$ which is equivalent to $Ba = \lambda V a$. Recalling that $V = W + B$, we have $Ba = \lambda(B + W)a$ or equivalently $W^{-1}Ba = (1 - \lambda)^{-1}a$.

As a consequence of the two above remarks, when there are $K = 2$ classes, Fisher discriminant subspace is one dimensional and is spanned by

$$a_1 = W^{-1}(\bar{x}_1 - \bar{x}_2),$$

since, in such a case, the eigenvector associated to the unique non-zero eigenvalue is closed form.

2.2.3. *Fisher Discriminant Analysis versus Principal Component Analysis.* Principal Component Analysis (PCA) is a dimension reduction technique, see for instance Jolliffe (2002). It aims at finding the linear subspace maximizing the sample projected variance. Similarly, it can be shown that it amounts to performing the eigendecomposition of V. PCA is usually an efficient way to reduce the dimension (van der Maaten *et al.*, 2009) but it may be sub-optimal in the supervised classification framework since it does not take into account the label information. As a comparison, FDA can be considered as a supervised dimension reduction method whereas PCA is unsupervised. This difference is illustrated on the USPS digit database, a standard dataset for handwritten digit recognition. It can be downloaded at `http://statweb.stanford.edu/~tibs/ElemStatLearn/data.html`. Each of the $n = 7291$ observations is a 16×16 grey-level image obtained by scanning a handwritten digit. The images are encoded as vectors of dimension $p = 16 \times 16 = 256$. The original dataset is made of $K = 10$ classes corresponding to the digits $0, 1, \ldots, 9$. Here, we focus on the three classes associated with the digits 3, 5 and 8 (see the top panel of Figure 1). The two-dimensional projections of the dataset obtained by PCA and FDA are compared on the bottom panel. The class structure is much clearer on the FDA projection than on the PCA one.

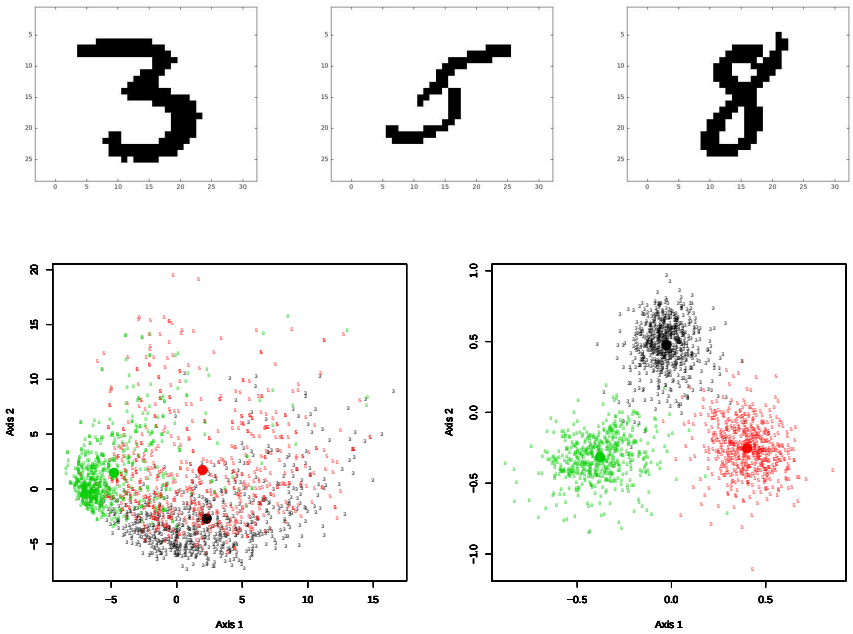


FIGURE 1. Top: A sample from the USPS dataset. Bottom: Two-dimensional projections (left: PCA axes, right: FDA axes).

2.3. **Mixture model.**

2.3.1. *Definition.* The observations $x_1, \ldots, x_n$ are supposed to be independent realizations from a random vector $X \in \mathbb{R}^p$ while the labels $z_1, \ldots, z_n$ are assumed to

be drawn from a discrete random variable $Z \in \{1, \ldots, K\}$. The distribution of the random pair (X, Z) is built assuming that:

- Z follows a multinomial distribution with parameters $\pi_1, \ldots, \pi_K$ called mixing proportions, *i.e.* $\mathbb{P}(Z = k) = \pi_k$, $k = 1, \ldots, K$. The mixing proportions should verify the constraints $\pi_k \in [0, 1]$ for all $k = 1, \ldots, K$ and $\sum_{k=1}^{K} \pi_k = 1$.
- Given $Z = k$, X has a density denoted by $f(\cdot, \theta_k)$, where θ_k represents the parameter (possibly multidimensional) of f. For the sake of simplicity, it is assumed here that the parametric form of f does not depend on the mixture component k. This assumption could be nevertheless easily be dropped.

From the law of total probability, one can show that the density of X is a linear combination of the densities associated with the K mixture components:

$$f(\cdot) = \sum_{k=1}^{K} \pi_k f(\cdot, \theta_k). \tag{1}$$

An illustration in dimension $p = 1$ is provided in Figure 2 in case of Gaussian densities. We refer to McLachlan and Peel (2000) for a general account on mixture models.

2.3.2. *Sampling from a mixture.* Sampling from a mixture model is an easy task which involves two steps:

- Pick a component k with probability π_k,
- Draw a sample from the density $f(\cdot, \theta_k)$.

An illustration is given in Figure 3, where the previous algorithm is implemented in **R** code on the same mixture model as in Figure 2.

2.3.3. *A priori vs a posteriori probabilities.* For all $k = 1, \ldots, K$, the mixing proportion $\pi_k = \mathbb{P}(Z = k)$ can be interpreted as the *a priori* probability that an observation x belongs to the class $\mathcal{C}_k$. Here, *a priori* means without using the observation x of the random variable X. At the opposite, the conditional probability $\mathbb{P}(Z = k|X = x)$ is called the *a posteriori* probability that an observation x belongs to $\mathcal{C}_k$. These two probabilities are linked through Bayes rule:

$$\mathbb{P}(Z = k|X = x) = \mathbb{P}(Z = k) f(x|Z = k)/f(x) = \pi_k f(x, \theta_k)/f(x). \tag{2}$$

2.3.4. *Bayes decision rule.* The aim of supervised classification is to build a decision rule *i.e.* a function $\delta : \mathbb{R}^p \to \{1, \ldots, K\}$ which affects a label $k \in \{1, \ldots, K\}$ to each observation $x \in \mathbb{R}^p$. It is possible to show that the rule which minimizes the probability of miss-classification is Bayes decision rule given by

$$\delta^\star(x) = \underset{k=1\ldots,K}{\operatorname{argmax}}\, \mathbb{P}(Z = k|X = x) = \underset{k=1,\ldots,K}{\operatorname{argmax}}\, \pi_k f(x, \theta_k)/f(x),$$

from (2). Remarking that the denominator does not depend on k, Bayes decision rule can be simplified as

$$\delta^\star(x) = \underset{k=1,\ldots,K}{\operatorname{argmax}}\, \pi_k f(x, \theta_k). \tag{3}$$

In practice, the parameters π_k and θ_k have to be estimated: $\hat{\pi}_k = n_k/n$ and $\hat{\theta}_k$ is the maximum likelihood estimator computed on the class $\mathcal{C}_k$ given by

$$\hat{\theta}_k = \underset{\theta}{\operatorname{argmax}} \prod_{z_i=k} f(x_i,\theta) = \underset{\theta}{\operatorname{argmax}} \sum_{z_i=k} \log f(x_i,\theta) =: \underset{\theta}{\operatorname{argmax}}\, \mathcal{L}(\theta, x_1, \ldots, x_n).$$

In the previous equation, $\mathcal{L}(\theta, x_1, \ldots, x_n)$ denotes the log-likelihood which is usually more convenient to optimize than the likelihood. The estimated Bayes decision rule is then obtained by plugging the estimated parameters in (3):

$$\hat{\delta}^\star(x) = \underset{k=1,\ldots,K}{\operatorname{argmax}} n_k f(x,\hat{\theta}_k). \tag{4}$$

2.3.5. *Binary classification.* In the particular case of $K = 2$ classes, the posterior probabilities $\mathbb{P}(Z = 1|X = x)$ and $\mathbb{P}(Z = 2|X = x)$ can be simplified. Since they sum to one, let us focus on the first one:

$$\begin{aligned} \mathbb{P}(Z=1|X=x) &= \frac{\pi_1 f(x,\theta_1)}{\pi_1 f(x,\theta_1) + \pi_2 f(x,\theta_2)} \\ &= \frac{1}{1+\frac{\pi_2 f(x,\theta_2)}{\pi_1 f(x,\theta_1)}} \\ &= \Psi(S(x)), \end{aligned} \tag{5}$$

where Ψ is called the logistic function defined by $\Psi(t) = 1/(1+\exp(-t))$ for all $t \in \mathbb{R}$ and where S is the so-called score defined for $x \in \mathbb{R}^p$ by

$$S(x) = \log\left(\frac{\pi_1 f(x,\theta_1)}{\pi_2 f(x,\theta_2)}\right). \tag{6}$$

The previous result is usually stated as "the posterior probability is a logistic function of the score". A straightforward consequence is that Bayes decision rule can be rewritten as $\delta^\star(x) = 1$ if and only if $S(x) > 0$.

2.4. **Gaussian Mixture Model.**

2.4.1. *Quadratic Discriminant Analysis.* In the Gaussian Mixture Model (GMM), it is further assumed that, for all $k = 1, \ldots, K$, the density $f(\cdot,\theta_k)$ is a Gaussian density with mean μ_k and covariance matrix Σ_k. We thus have $\theta_k = (\mu_k, \Sigma_k)$. Letting $L_k(\cdots) = -2\log(\pi_k f(\cdot,\theta_k))$, Bayes decision rule (3) can be rewritten as

$$\delta^*(x) = \underset{k=1,\ldots,K}{\operatorname{argmin}} L_k(x)$$

where

$$L_k(x) = (x-\mu_k)^t \Sigma_k^{-1}(x-\mu_k) + \log(\det \Sigma_k) - 2\log(\pi_k) + C.$$

Here, and in the sequel, C is a constant which does not depend on k. This classification method is referred to as Quadratic Discriminant Analysis (QDA). Note that $(x-\mu_k)^t\Sigma_k^{-1}(x-\mu_k)$ can be interpreted as the Mahalanobis distance between x and μ_k. Bayes decision rule is, in this case, a quadratic function of the observation x.

2.4.2. *Linear Discriminant Analysis.* To reduce the number of parameters to estimate, one may assume that all classes share the same covariance matrix $\Sigma_k = \Sigma$ for all $k = 1, \ldots, K$. This amounts to supposing that the K classes have the same shape. This assumption yields

$$L_k(x) = \mu_k^t \Sigma^{-1} \mu_k - 2\mu_k^t \Sigma^{-1} x - 2\log(\pi_k) + C.$$

This is the so-called Linear Discriminant Analysis (LDA) method where the associated Bayes decision rule is linear with respect to the observation x.

2.4.3. *GMM in practice.* The estimation of the parameters is simple. We still have $\hat{\pi}_k = n_k/n$ and the maximum likelihood estimators are closed-form: $\hat{\mu}_k = \bar{x}_k$ is the sample mean of $\mathcal{C}_k$, $\hat{\Sigma}_k = V_k$ is the sample covariance matrix of $\mathcal{C}_k$ and $\hat{\Sigma} = n/(n-K)W$ where W is the within-class covariance matrix, see Paragraph 2.1. Let us highlight, that, in high dimension (*i.e.* if p is large), inverting $\hat{\Sigma}_k$ or $\hat{\Sigma}$ may be an issue, see Bergé *et al.* (2012) or the chapter by C. Bouveyron in this book.

When several models are available (for instance LDA and QDA), a natural way to choose one of them is to use a cross-validation procedure to select the model that minimizes a sample based estimate of future miss-classification risk. See Bensmail and Celeux (1996) for an application to the selection of parsimonious Gaussian models.

2.4.4. *Binary Linear Discriminant Analysis.* We consider the particular case where $K = 2$ and $\Sigma_1 = \Sigma_2 = \Sigma$. In such a situation, the score (6) is linear and is given by

$$\begin{aligned} S(x) &= \frac{1}{2}(L_2(x) - L_1(x)) \\ &= x^t \Sigma^{-1}(\mu_1 - \mu_2) - \frac{1}{2}(\mu_1 + \mu_2)^t \Sigma^{-1}(\mu_1 - \mu_2) + \log(\pi_2/\pi_1). \end{aligned} \tag{7}$$

Once the parameters are estimated, the classification rule becomes $\hat{\delta}^\star(x) = 1$ if and only if

$$\begin{aligned} x^t W^{-1}(\bar{x}_1 - \bar{x}_2) &> (\bar{x}_1 + \bar{x}_2)^t W^{-1}(\bar{x}_1 - \bar{x}_2) + \log(n_2/n_1) \\ x^t a_1 &> (\bar{x}_1 + \bar{x}_2)^t W^{-1}(\bar{x}_1 - \bar{x}_2) + \log(n_2/n_1), \end{aligned}$$

where a_1 is the Fisher discriminant axis, see Paragraph 2.2. It thus appears that Binary LDA reduces to comparing the projection $x^t a_1$ of the data x to classify on the Fisher discriminant axis a_1 with a threshold $(\bar{x}_1+\bar{x}_2)^t W^{-1}(\bar{x}_1-\bar{x}_2)+\log(n_2/n_1)$ depending on the geometry of the problem.

2.4.5. *Binary logistic regression.* As previously mentioned, in case of binary classification ($K = 2$), the probability to affect x to the first class is $\mathbb{P}(Z = 1|X = x) = \Psi(S(x))$, see (5). If, moreover, it is assumed that $\Sigma_1 = \Sigma_2 = \Sigma$, then the score $S(x)$ is a linear function of x, see (7). This is the binary LDA method described above. Binary logistic regression directly assumes that there exist $\beta_0 \in \mathbb{R}$ and $\beta \in \mathbb{R}^p$ such that

$$\mathbb{P}(Z = 1|X = x) = \Psi(\beta_0 + \beta^t x)$$

without any other hypothesis. The estimation of β_0 and β is performed by maximizing the conditional likelihood:

$$\prod_{z_i=1} \Psi(\beta_0 + \beta^t x_i) \prod_{z_i=2} (1 - \Psi(\beta_0 + \beta^t x_i)). \tag{8}$$

The advantage of binary logistic regression is to be potentially more general than binary LDA since it does not make any Gaussian assumption. The computation of the estimators is however more difficult since the maximization of (8) requires numerical procedures.

3. Unsupervised classification

Recall that the goal is to split a dataset $x_1, \ldots, x_n$ of n observations in $\mathbb{R}^p$ into K homogeneous groups $\mathcal{C}_1, \ldots, \mathcal{C}_K$. The labels are still denoted by $z_1, \ldots, z_n$ but are not observed. In a first time, the number K of clusters is supposed to be known. The choice of K is addressed in Paragraph 3.3.

3.1. K-means algorithm.

Each cluster $\mathcal{C}_k$ is represented by a vector $\mu_k \in \mathbb{R}^p$ called prototype for all $k = 1, \ldots, K$. The binary numbers $r_{ik} = \mathbb{I}\{z_i = k\}$, $i = 1, \ldots, n$, $k = 1, \ldots, K$ indicating the class membership of the observations are called responsibilities. It is assumed that each observation x_i belongs to one and only one cluster so that $\sum_{k=1}^K r_{ik} = 1$ for all $i = 1, \ldots, n$. K-means algorithm translates the clustering problem as the computation of the vector of prototypes and responsibilities $\theta := (\mu_k, r_{ik})_{i=1,\ldots,n,\ k=1,\ldots,K}$. To this end, the following optimization problem is introduced:

$$\min_{\theta} \sum_{i=1}^{n} \sum_{k=1}^{k} r_{ik} \|x_i - \mu_k\|^2 \tag{9}$$

under the constraints: $r_{ik} \in \{0, 1\}$ and $\sum_{\ell=1}^K r_{i\ell} = 1$ for all $i = 1, \ldots, n$, $k = 1, \ldots, K$. Since this problem has no explicit solution, the principle of K-means algorithm is to perform an alternate minimization, see Hartigan and Wong (1979):

- Minimization with respect to the responsibilities, assuming that the prototypes are known. The solution is given by: $r_{ik} = 1$ if and only if μ_k is the closest prototype of x_i, $i = 1, \ldots, n$, $k = 1, \ldots, K$. This amounts to partitioning the observations according to the Voronoi diagram generated by the prototypes.
- Minimization with respect to the prototypes, assuming that the responsibilities are known. The solution is given by: μ_k is the mean of the x_i affected to the cluster $\mathcal{C}_k$, that is

$$\mu_k = \sum_{i=1}^{n} r_{ik} x_i \Bigg/ \sum_{i=1}^{n} r_{ik} \ , \tag{10}$$

  for all $k = 1, \ldots, K$.

In practice, it is recommended to standardize the data before running the algorithm. It has been proved that each iteration of K-means algorithm decreases the criterion to minimize (9). The algorithm thus converges to a local minimum and the result may depend on the initialization. However, since the algorithm is usually very fast, it is possible to run it multiple times with different starting conditions.

K-means algorithm suffers from some drawbacks. First, the hard assignment of data points x_i to clusters $\mathcal{C}_k$ may be unstable: A small change of a data point can move it to another cluster. Second, the choice of the number of clusters K is difficult. In the following subsubsection, we shall work in the GMM framework. This will allow us to replace hard assignments by "soft" probabilistic assignments.

3.2. **Maximum likelihood in the GMM.** In the unsupervised framework, both the model parameters $\theta_k = (\pi_k,\ \mu_k,\ \Sigma_k)$ and the labels z_{ik}, $i = 1, \ldots, n$, $k = 1, \ldots, K$ are unknown. Denoting the unknown (multidimensional) parameter by $\theta = (\theta_1, \ldots, \theta_K)$, the inference is still based on the log-likelihood

$$\mathcal{L}(\theta, x_1, \ldots, x_n) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k f(x_i, \mu_k, \Sigma_k) \right).$$

Recall that $f(\cdot, \mu, \Sigma)$ is the p-dimensional Gaussian density with mean μ and covariance matrix Σ. Since there is a sum in the logarithm, the maximum likelihood estimator of θ is not explicit. Let us however consider the gradient of the log-likelihood with respect to μ_k, $k = 1, \ldots, K$:

$$\begin{aligned} \nabla_{\mu_k} \mathcal{L} &= -\sum_{i=1}^{n} \frac{\pi_k f(x_i, \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j f(x_i, \mu_j, \Sigma_j)} \Sigma_k^{-1}(x_i - \mu_k) \\ &= -\sum_{i=1}^{n} \mathbb{P}(Z = k | X = x_i) \Sigma_k^{-1}(x_i - \mu_k), \end{aligned}$$

in view of (1) and (2). Annulling the previous gradient yields, for all $k = 1, \ldots, K$,

$$\hat{\mu}_k = \frac{\sum_{i=1}^{n} \mathbb{P}(Z = k | X = x_i)\ x_i}{\sum_{i=1}^{n} \mathbb{P}(Z = k | X = x_i)}. \tag{11}$$

It appears that $\hat{\mu}_k$ is a weighted mean of the observations. Let us precise that $\hat{\mu}_k$ is not a proper estimator of μ_k since the weights *i.e.* the posterior probabilities $\mathbb{P}(Z = k | X = x_i)$ are unknown. Comparing to the K-means formula (10), the binary weights are replaced by real values in the unit interval. Similar calculations yield, for all $k = 1, \ldots, K$:

$$\hat{\Sigma}_k = \frac{\sum_{i=1}^{n} \mathbb{P}(Z = k | X = x_i)\ (x_i - \mu_k)(x_i - \mu_k)^t}{\sum_{i=1}^{n} \mathbb{P}(Z = k | X = x_i)}, \tag{12}$$

with a similar interpretation. Using a Lagrange multiplier technique to take into account the constraint $\sum_{k=1}^{K} \pi_k = 1$ entails:

$$\hat{\pi}_k = \frac{1}{n} \sum_{i=1}^{n} \mathbb{P}(Z = k | X = x_i). \tag{13}$$

The intuitive idea of the Expectation-Maximization (EM) algorithm is to alternate the calculations (11)–(13) with an update of the posterior probabilities according to

$$\mathbb{P}(Z = k | X = x_i) = \frac{\pi_k f(x_i, \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j f(x_i, \mu_j, \Sigma_j)}, \tag{14}$$

see (1) and (2). The EM algorithm is however based on a more general maximization principle and is not limited to Gaussian mixtures, see Dempster *et al.* (1977). In our considered framework, the algorithm can be summarized as follows.

3.2.1. *EM algorithm.*

- *Initialization:* Initial guess for the model parameters.
- *E Step:* Estimation of $t_{ik} := \mathbb{P}(Z = k|X = x_i)$ using

$$\hat{t}_{ik} = \frac{\hat{\pi}_k f(x_i, \hat{\mu}_k, \hat{\Sigma}_k)}{\sum_{j=1}^K \hat{\pi}_j f(x_i, \hat{\mu}_j, \hat{\Sigma}_j)}$$

- *M Step:* Maximization of log-likelihood yielding the previous formulas:

$$\begin{aligned}
\hat{\mu}_k &= \frac{\sum_{i=1}^n \hat{t}_{ik}\, x_i}{\sum_{i=1}^n \hat{t}_{ik}}, \\
\hat{\Sigma}_k &= \frac{\sum_{i=1}^n \hat{t}_{ik}\, (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^t}{\sum_{i=1}^n \hat{t}_{ik}}, \\
\hat{\pi}_k &= \frac{1}{n} \sum_{i=1}^n \hat{t}_{ik}.
\end{aligned}$$

- Repeat until convergence.

It can be shown that the EM algorithm increases the likelihood at each iteration. This property allows some monitoring of the convergence. In practice, the iterations are stopped when the increase of the log-likelihood is behind a small threshold. EM algorithm converges to a local maxima of the likelihood, the initialization is thus an important issue, addressed for instance in Biernacki *et al.* (2003). The most common practice is to initialize the EM algorithm with K-means clustering. Stochastic and classification versions of EM have been developed (G. Celeux and G. Govaert, 1992). Either the *a posteriori* probabilities t_{ik} are used to randomly select the class belongings (random assignment), or are binarized (hard assignment). A simple implementation of the EM algorithm in **R** is provided in Figure 4. In this toy case, we limit ourselves to the situation already considered in Figure 2 and Figure 3 where $K = 2$ and $p = 1$. The algorithm is initialized with $\mu_1^{(0)} = 0$, $\mu_2^{(0)} = 1$, $\pi_1^{(0)} = \pi_2^{(0)} = 1/2$ and $\sigma_1^{(0)} = \sigma_2^{(0)}$ are set to the sample standard-deviation. The number of iterations is fixed by the user. An example of result is depicted in Figure 5. The EM algorithm has been run with 100 iterations on a sample simulated from the model considered in Figure 2 and Figure 3. The estimated parameters are $\hat{\pi}_1 = 0.48$, $\hat{\pi}_2 = 0.52$, $\hat{\mu}_1 = 0.46$, $\hat{\mu}_2 = 1.98$, $\hat{\sigma}_1^2 = 2.94$ and $\hat{\sigma}_2^2 = 0.82$. The true and estimated densities look very similar. Note that the results may depend on the simulated data.

3.2.2. *Links with K-means algorithm.* Let us consider the GMM, with common and spherical covariance matrices *i.e.* $\Sigma_k = \varepsilon I_p$ for all $k = 1, \ldots, K$. Replacing in (14), the *a posteriori* probabilities are given by

$$\begin{aligned}
t_{ik} &= \frac{\pi_k \exp\left(-\frac{1}{2\varepsilon}\|x_i - \mu_k\|^2\right)}{\sum_{j=1}^K \pi_j \exp\left(-\frac{1}{2\varepsilon}\|x_i - \mu_j\|^2\right)} \\
&= \frac{1}{1 + \sum_{j \neq k} \frac{\pi_j}{\pi_k} \exp\left(\frac{1}{2\varepsilon}(\|x_i - \mu_k\|^2 - \|x_i - \mu_j\|^2)\right)}.
\end{aligned}$$

Letting $\varepsilon \to 0$, t_{ik} becomes a binary number: $t_{ik} = \mathbb{I}\{\|x_i - \mu_k\| < \|x_i - \mu_j\|, \text{ for all } j \neq k\}$, and we find back the K-means algorithm. K-means can thus be interpreted as

a particular case of EM. This suggests that K-means is well adapted to mixtures where all the clusters share a common spherical shape.

As an illustration, let us consider the clustering of the Old Faithful geyser dataset. Old Faithful is a geyser located in Yellowstone National Park (USA). It is one of the most predictable geographical features on Earth, erupting every 40 to 100 minutes. This dataset can be downloaded at `http://www.stat.cmu.edu/~larry/all-of-statistics/=data/faithful.dat` and is also included in the `mclust` package, see Fraley and Raftery (2003). This dataset includes $n = 272$ observations described by $p = 2$ variables (duration of the eruption, waiting time between eruptions). In this subsubsection, the number of classes is arbitrarily set to $K = 2$. In Figure 6, it appears that the clustering obtained with the K-means model (equal and diagonal covariance matrices) and the clustering with the QDA model (free covariance matrices) are identical.

3.3. **Selecting the number of clusters.** Various criteria have been introduced to measure the suitability of a model by balancing its goodness of fit and its complexity. Here, we focus on criteria based on a penalization of the log-likelihood (computed for the optimal parameter):

$$IC = \mathcal{L}(\hat{\theta}, x_1, \ldots, x_n) - \nu(K)\varphi(n)$$

where $\hat{\theta}$ is the maximum likelihood estimator, $\nu(K)$ is the number of (free) parameters in the model and $\varphi(n)$ is a function of n. The most famous examples are the Akaike Information Criteria (AIC), see Akaike (1974), where $\varphi(n) = 1$ and the Bayesian Information Criteria (BIC), see Schwarz (1978) where $\varphi(n) = \log(n)/2$. Further details and theoretical developments on AIC and BIC can be found respectively in Aitkin and Rubin (1985) and Kass and Raftery (1995). In practice, it has been remarked that, usually, BIC works better than AIC on mixtures (Bouveyron *et al.*, 2011) but none of them takes into account the classification purpose. Alternative criteria have been introduced, such as ICL (Integrated Completed Likelihood), see Biernacki *et al.* (2000), to overcome this limitation. The principle of ICL is to introduce an additional penalization on overlapping classes in BIC.

As an illustrated, let us consider the clustering of the Old Faithful geyser dataset. In Figure 6, the clustering was achieved assuming $K = 2$. Here, we use the BIC criterion to select K in the set $\{1, \ldots, 5\}$ and to choose between common covariance matrices (LDA model) or free covariance matrices (QDA model). It appears in Figure 7 that the largest value of BIC is obtained with $K = 3$ components associated with the LDA model. The corresponding clustering is also depicted in Figure 7 on a visualization space different from the original one (Figure 6), see the package documentation. Let us highlight that the decision boundaries are linear, see Paragraph 2.4.

4. Conclusion, recent developments

As a conclusion, mixture models are an efficient tool for both supervised and unsupervised classifications. They offer a nice theoretical framework for model selection and for computing classification probabilities. They also encompass geometric methods as particular cases. From the practical point of view, we have seen that they are naturally multiclass and that efficient algorithms are available. It is also possible to deal with missing data thanks to the EM algorithm.

In this chapter, we focused on Gaussian mixture models, but a lot of other models are available in the statistical literature. First, in case of discrete data, multinomial models can be used, see for instance Bouguila *et al.* (2003), Celeux and Govaert (1991) or Goldstein and Dillon (1978). Second, to tackle the case of heavy-tailed or asymmetric data, several extensions of Gaussian models have been recently introduced: Skew normal distribution (Vilca *et al.*, 2014), t-distributions (Andrews and McNicholas, 2012 or Forbes and Wraith, 2014), asymmetric Laplace distribution (Franczak *et al.*, 2014) and skew t-distributions (Lee and McLachlan, 2013 or Wraith and Forbes, 2015). Finally, an extension of the Gaussian mixture model to non quantitative data has been proposed by Bouveyron *et. al.* (2015). The introduction of a kernel function permits to deal with various kind of data including categorical data, functional data or networks.

References

[1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, **19**(6), 716–723, 1974.

[2] M. Aitkin & D. Rubin. Estimation and hypothesis testing in finite mixture models. *Journal of the Royal Statistical Society. Series B*, **47**(1), 67–75, 1985.

[3] J. Andrews & P. McNicholas. Model-based clustering, classification, and discriminant analysis via mixtures of multivariate t-distributions. *Statistics and Computing*, **22**(5), 1021–1029, 2012.

[4] H. Bensmail & G. Celeux. Regularized Gaussian discriminant analysis through eigenvalue decomposition. *Journal of the American statistical Association*, **91**(436), 1743–1748, 1996.

[5] N. Bouguila, D. Ziou & J. Vaillancourt. Novel mixtures based on the Dirichlet distribution: application to data and image classification. *In Machine Learning and Data Mining in Pattern Recognition*, pages 172–181, Springer, 2003.

[6] C. Bouveyron, M. Fauvel & S. Girard. Kernel discriminant analysis and clustering with parsimonious Gaussian process models, *Statistics and Computing*, **25**, 1143–1162, 2015.

[7] L. Bergé, C. Bouveyron & S. Girard. HDclassif: An R package for model-based clustering and discriminant analysis of high-dimensional data, *Journal of Statistical Software*, **46**(6), 1–29, 2012.

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

[9] C. Biernacki, G. Celeux & G. Govaert. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Computational Statistics and Data Analysis*, **41**, 561–575, 2003.

[10] C. Bishop. *Pattern Recognition and Machine Learning*, Springer, 2006.

[11] C. Bouveyron, G. Celeux & S. Girard. Intrinsic dimension estimation by maximum likelihood in isotropic probabilistic PCA. *Pattern Recognition Letters*, **32**(14), 1706–1713, 2011.

[12] G. Celeux & G. Govaert. Clustering criteria for discrete data and latent class models. *Journal of classification*, **8**, 157–176, 1991.

[13] G. Celeux & G. Govaert. A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, **14**(3), 315–332, 1992.

[14] A. Dempster, N. Laird & D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B*, **39**, 1–38, 1977.

[15] F. Forbes & D. Wraith. A new family of multivariate heavy-tailed distributions with variable marginal amounts of tail-weight: application to robust clustering. *Statistics and Computing*, **24**(6), 971–984, 2014.

[16] D. Fraix-Burnet & D. Valls-Gabaud. *Regression methods for astrophysics*, EDP Sciences, 2014.

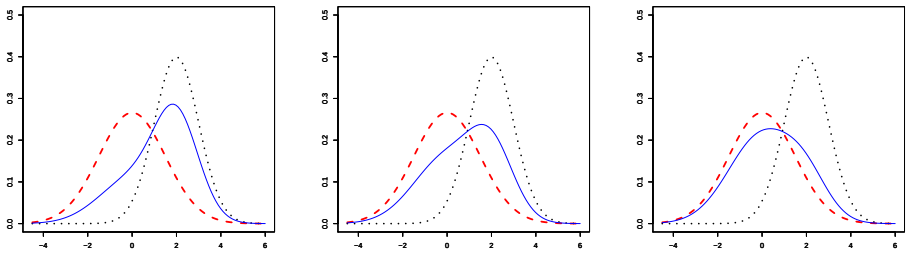
[17] C. Fraley & A. E. Raftery. Enhanced software for model-based clustering, density estimation, and discriminant analysis: MCLUST. *Journal of Classification*, **20**, 263–286, 2003.

[18] B.C. Franczak, R.P. Browne & P.D. McNicholas. Mixtures of shifted asymmetric Laplace distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36** (6), 1149–1157, 2014.

[19] M. Goldstein & W.R. Dillon. *Discrete discriminant analysis*. John Wiley & Sons, New York, 1978.
[20] J. Hartigan & M. Wong. A K-means clustering algorithm. *Applied Statistics*, **28**, 100–108, 1979.
[21] T. Hastie, R. Tibshirani & J. Friedman. *Elements of Statistical Learning*, Springer, 2001.
[22] I. Jolliffe. *Principal component analysis*. John Wiley & Sons, Ltd, 2002.
[23] R. Kass & A. Raftery. Bayes factors. *Journal of the American Statistical Association*, **90**(430), 773–795, 1995.
[24] S. Lee & G. McLachlan. Finite mixtures of multivariate skew t-distributions: some recent and new results. *Statistics and Computing*, **24**(2), 181–202, 2013.
[25] L. van der Maaten, E. Postma, & H. van den Herik. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, **10**, 66–71, 2009.
[26] G. J. McLachlan & D. Peel. *Finite Mixture Models*, Wiley, New York, 2000.
[27] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, **6**(2), 461–464, 1978.
[28] F. Vilca, N. Balakrishnan & C. Zeller. Multivariate skew-normal generalized hyperbolic distribution and its properties. *Journal of Multivariate Analysis*, **128**, 73–85, 2014.
[29] D. Wraith & F. Forbes. Location and scale mixtures of Gaussians with flexible tail behaviour: Properties, inference and application to multivariate clustering. *Computational Statistics and Data Analysis*, **90**, 61–73, 2015.

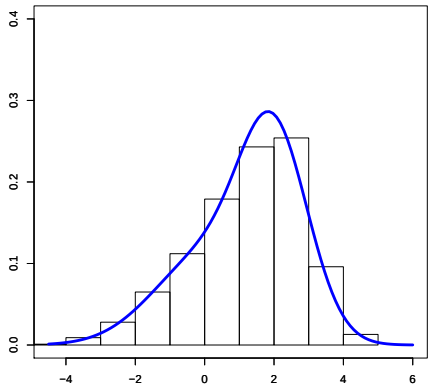
Inria Grenoble Rhône-Alpes & Laboratoire Jean Kuntzmann

Institut Polytechnique de Bordeaux & Inria Bordeaux Sud Ouest & Institut de Mathématiques de Bordeaux



```
# mixture parameters
pi <- 0.4
mu1 <- 0 ; sigma1 <- 1.5
mu2 <- 2 ; sigma2 <- 1
# visualization parameters
left <- -4.5 ; right <- 6 ; top <- 0.5
grd <- seq(from=left,to=right,length=100)
# plots
plot(grd,dnorm(grd,mean=mu1,sd=sigma1),type="l",col=2,
     xlim=c(left,right),ylim=c(0,top),xlab="",ylab="",
     lty="dashed",lwd=4)
par(new=TRUE)
plot(grd,dnorm(grd,mean=mu2,sd=sigma2),type="l",col=1,
     xlim=c(left,right),ylim=c(0,top),xlab="",ylab="",
     lty="dotted",lwd=4)
par(new=TRUE)
plot(grd,pi*dnorm(grd,mean=mu1,sd=sigma1)+
         (1-pi)*dnorm(grd,mean=mu2,sd=sigma2),
     type="l",col=4,xlim=c(left,right),ylim=c(0,top),
     xlab="",ylab="",lwd=2)
```

FIGURE 2. Top: Density of a GMM with $K = 2$ components and $p = 1$ variable. The parameters are given by $\mu_1 = 0$, $\mu_2 = 2$, $\sigma_1 = 1.5$, $\sigma_2 = 2$, $\pi_1 = \pi$ and $\pi_2 = 1 - \pi$. From left to right, the proportion π is varying in $\{0.4, 0.6, 0.8\}$. Red dashed curve: $f(\cdot, \mu_1, \sigma_1^2)$, Black dotted curve: $f(\cdot, \mu_2, \sigma_2^2)$, Blue solid curve: density of the mixture. Bottom: The corresponding **R** code for $\pi = 0.4$.

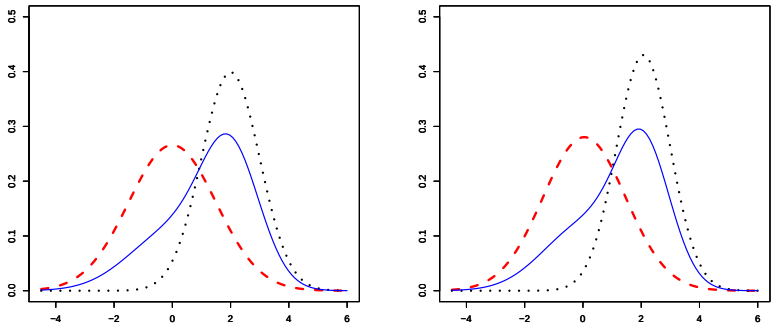



```
# visualization parameters
left <- -4.5 ; right <- 6 ; top <- 0.4
grd <- seq(from=left,to=right,length=100)
# sample size
n <- 1000
# generate a sample from the mixture
generated.smple <- rep(0,n)
for (i in 1:n){
  aux <- runif(1,min=0,max=1)
  if (aux<pi) { # simulate from component 1
    generated.smple[i] <- rnorm(1,mean=mu1,sd=sigma1)}
  else { # simulate from component 2
    generated.smple[i] <- rnorm(1,mean=mu2,sd=sigma2)}
}
# histogram of the sample and true density of the mixture
hist(generated.smple,freq=FALSE,xlim=c(left,right),
     ylim=c(0,top),xlab="",ylab="",main="")
par(new=TRUE)
plot(grd,pi*dnorm(grd,mean=mu1,sd=sigma1)+
     (1-pi)*dnorm(grd,mean=mu2,sd=sigma2),type="l",col=4,
     xlim=c(left,right),ylim=c(0,top),xlab="",ylab="",lwd=2)
```

FIGURE 3. Top: Density of a GMM with $K = 2$ components and $p = 1$ variable (blue) superimposed to the histogram of a dataset of $n = 1000$ observations simulated from the same model (black). The parameters are given by $\mu_1 = 0$, $\mu_2 = 2$, $\sigma_1 = 1.5$, $\sigma_2 = 2$, $\pi_1 = \pi$ and $\pi_2 = 1 - \pi$. Here, the proportion is fixed to $\pi = 0.4$. Bottom: The corresponding **R** code.

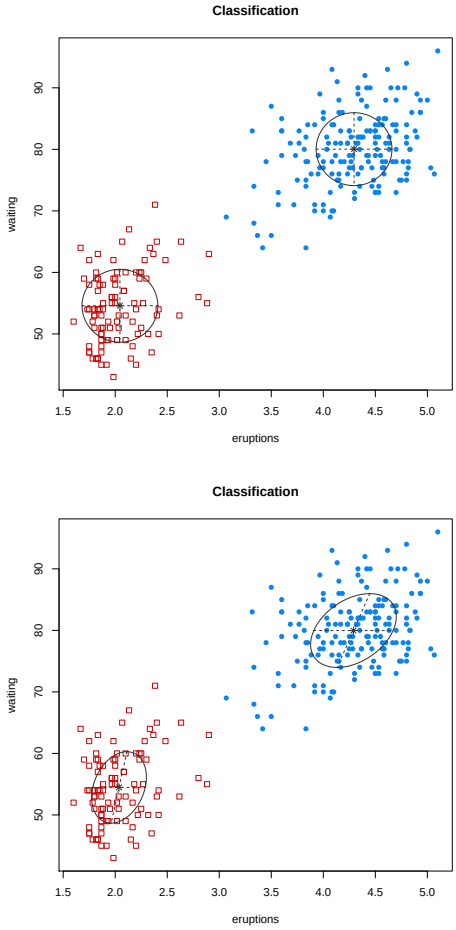
```
EM <- function(smple,nb.iter=10){

  # intialization using the histogram
  mu <- c(0,1)
  vari <- c(var(smple),var(smple))
  prp <- c(0.5,0.5)

  # main loop (the number of iterations is set by the user)
  for (iter in 1:nb.iter){
    # E step
    post1<-prp[1]*dnorm(smple,mean=mu[1],sd=sqrt(vari[1]))
    post2<-prp[2]*dnorm(smple,mean=mu[2],sd=sqrt(vari[2]))
    total<-post1+post2
    post1<-post1/total
    post2<-post2/total
    # M step
    prp[1] <- mean(post1)
    prp[2] <- mean(post2)
    mu[1]<-weighted.mean(smple,post1)
    mu[2]<-weighted.mean(smple,post2)
    vari[1]<-weighted.mean((smple-mu[1])^2,post1)
    vari[2]<-weighted.mean((smple-mu[2])^2,post2)
  }
  # the function returns the estimated proportions, means,
  # variances as well as the posterior probabilities.
  list(prp=prp,mu=mu,vari=vari,post=cbind(post1,post2))
}
```

FIGURE 4. EM algorithm for GMM coded in **R** in the one-dimensional case ($p = 1$) and for two clusters ($K = 2$).



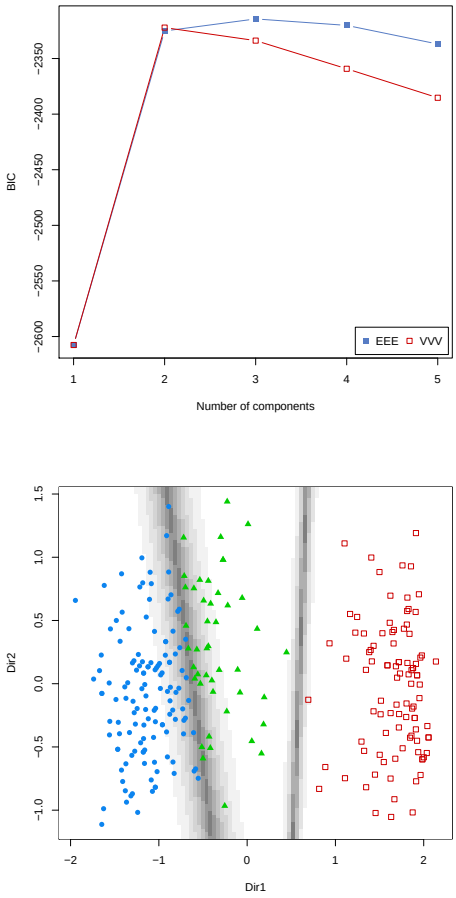
```
# run the EM algorithm with 100 iterations.
res <- EM(generated.smple,nb.iter=100)
# visualization parameters
left <- -4.5 ; right <- 6 ; top <- 0.5
grd <- seq(from=left,to=right,length=100)
# plots
plot(grd,dnorm(grd,mean=res$mu[1],sd=sqrt(res$vari[1])),
     type="l",col=2,xlim=c(left,right),ylim=c(0,top),
     xlab="",ylab="",lty="dashed",lwd=4)
par(new=TRUE)
plot(grd,dnorm(grd,mean=res$mu[2],sd=sqrt(res$vari[2])),
     type="l",col=1,xlim=c(left,right),ylim=c(0,top),
     xlab="",ylab="",lty="dotted",lwd=4)
par(new=TRUE)
plot(grd,res$prp[1]*dnorm(grd,mean=res$mu[1],
     sd=sqrt(res$vari[1]))
       +res$prp[2]*dnorm(grd,mean=res$mu[2],
     sd=sqrt(res$vari[2])),type ="l",col=4,
     xlim=c(left,right),ylim =c(0,top),xlab="",ylab="",
     lwd=2)
```

FIGURE 5. Top left (reproduced from Figure 2): Density of a GMM with $K = 2$ components and $p = 1$ variable. The parameters are given by $\mu_1 = 0$, $\mu_2 = 2$, $\sigma_1 = 1.5$, $\sigma_2 = 2$, $\pi_1 = \pi$ and $\pi_2 = 1 - \pi$ with $\pi = 0.4$. Red dashed curve: $f(\cdot, \mu_1, \sigma_1^2)$, Black dotted curve: $f(\cdot, \mu_2, \sigma_2^2)$, Blue solid curve: density of the mixture. Top right: Estimated densities with the EM algorithm (Figure 4) on the simulated sample (Figure 3). Bottom: The corresponding **R** code for plotting the top right figure.




```
# install and load the mclust package
install.packages("mclust")
library(mclust)
# clustering with 2 groups and K-means model
res1 <- Mclust(faithful,G=2,modelNames="EEI")
# plot the clustering
plot(res1,what="classification")
# clustering with 2 groups and QDA
res2 <-Mclust(faithful,G=2,modelNames="VVV")
# plot the clustering
plot(res2,what="classification")
```

FIGURE 6. Clustering of the Old Faithful geyser dataset. Top: results obtained with diagonal and equal covariance matrices (K-means model encoded as EEI in `mclust`). Center: results obtained with free covariance matrices (referred to as QDA in the supervised framework and encoded as VVV in `mclust`). The density level sets are depicted in grey. Bottom: The corresponding **R** code.




```
# install and load the mclust package
install.packages("mclust")
library(mclust)
# clustering with {1,...,5} groups and LDA, QDA models
res <- Mclust(faithful,G=c(1,2,3,4,5),
              modelNames=c("EEE","VVV"))
# plot the BIC values
plot(res,what="BIC")
# plot the clustering and the decision boundaries
dr <- MclustDR(res)
plot(dr,what="boundaries")
```

FIGURE 7. Top: BIC values computed on the Old Faithful geyser dataset for $K \in \{1,\ldots,5\}$ and for two models: free covariance matrices (referred to as QDA in the supervised framework and encoded as VVV in `mclust`, red curve with □ symbols) and equal covariance matrices (referred to as LDA in the supervised framework and encoded as EEE in `mclust`, blue curve with ■ symbols). The best model here is EEE and $K = 3$ groups. Center: Clustering associated with the best model. The decision boundaries are depicted in grey. Bottom: The corresponding **R** code.