rebmix: An R Package for Normal, Lognormal and Weibull Finite Mixture Models

Marko Nagode
University of Ljubljana

Abstract

The **rebmix** package for R provides functions for random univariate and multivariate finite mixture generation, number of components, component weights and component parameters estimation and plotting of the finite mixtures. It relies on the REBMIX algorithm that requires preprocessing, information criterion and conditionally independent normal, lognormal or Weibull component densities. The rest is accomplished by the algorithm optimizing the component parameters, mixing weights and number of components successively based on the boundary conditions, such as the maximum number of components, total of positive relative deviations, number of classes or nearest neighbours. The algorithm is robust and time efficient and is insensitive to the number of components and random variables. It results in slightly worse estimates than the EM algorithm and can be used either to assess the initial set of the unknown parameters and number of components for the EM algorithm or as a standalone procedure that is a good compromise between the nonparametric and parametric methods to the finite mixture estimation. The datasets analysed are the galaxy, iris, wine, complex 1 and simulated 1.

Keywords: finite mixture, lognormal distribution, normal distribution, parameter estimation, R software, REBMIX algorithm, Weibull distribution.

1. Introduction

Finite mixture models are used increasingly to model the distributions of a wide variety of random phenomena. For the multivariate data of continuous nature, attention is paid to the use of multivariate normal components because of their computational convenience (McLachlan, Peel, Basford, and Adams 1999; Ingrassia and Rocci 2007; Frühwirth-Schnatter 2006). However, in fatigue and reliability analyses, lognormal and Weibull distributions are preferred due to their flexibility and their definition for continuous positive random variables only (Majeske 2003; Sultan, Ismail, and Al-Moisheer 2007; Touw 2009).

The finite mixture models have seen a real boost in popularity over the last decade due to the tremendous increase in available computing power. These models can be applied to data where observations originate from various groups and the group affiliations are not known, and on the other hand to provide approximations for multimodal distributions Leisch (2004). Some of the latest models can be found also in van Dijk, Hoogerheide, and Ardia (2009); Benaglia, Chauveau, Hunter, and Young (2009); Grün and Leisch (2008); Fraley and Raftery (2007); McLachlan and Peel (2000).

The REBMIX algorithm origins in Nagode and Fajdiga (1998). Later on it has evolved

(Nagode and Fajdiga 2000; Nagode, Klemenc, and Fajdiga 2001; Nagode and Fajdiga 2006), but its kernel has remained almost unchanged. Presently (Nagode and Fajdiga 2011b,a) it stands for a robust, time efficient tool that can be used either to assess the initial set of unknown parameters and the number of components for, e.g., the EM algorithm (Bučar, Nagode, and Fajdiga 2004) or as a standalone procedure that is a good compromise between the nonparametric and parametric methods to the finite mixture estimation.

The **rebmix** implementation of REBMIX extends the set of algorithms available for random univariate and multivariate finite mixture generation, number of components, component weights and component parameters estimation and plotting of the finite mixtures in the R language and environment for statistical computing (R Development Core Team 2011).

The outline of the paper is as follows: Section 2 presents the algorithm. Section 3 analyses the performance of the approach by studying the galaxy, iris, wine, complex 1 and simulated 1 datasets. Section 4 lists the conclusions and future work.

2. Algorithm

Let $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ be an observed d dimensional dataset of size n of continuous vector observations $\boldsymbol{y}_j$. Each observation is assumed to follow predictive mixture density

$$f(\boldsymbol{y}|c, \boldsymbol{w}, \boldsymbol{\Theta}) = \sum_{l=1}^{c} w_l f(\boldsymbol{y}|\boldsymbol{\theta}_l) \tag{1}$$

with conditionally independent component densities

$$f(\boldsymbol{y}|\boldsymbol{\theta}_l) = \prod_{i=1}^{d} f(y_i|\boldsymbol{\theta}_{il}) \tag{2}$$

indexed by vector parameter $\boldsymbol{\theta}_l$. The objective of the analysis is the inference about the unknowns: the number c of components, component weights w_l summing to 1 and component parameters $\boldsymbol{\theta}_l$.

2.1. Preprocessing of observations

The algorithm requires the preprocessing of observations. By the histogram approach, the dataset is counted into a finite number of nonoverlapping, equally sized and regularly distributed bins. Assuming that bin means $\boldsymbol{y}_j = (y_{1j}, \ldots, y_{dj})^\top$ are given by

$$y_{ij} = y_{i0} + \text{'An arbitrary integer'} \times h_{ij}, \; i = 1, \ldots, d \tag{3}$$

the fraction of observations k_j for $j = 1, \ldots, k$ falling into volume V_j is counted out, where y_{i0} stands for an arbitrary origin and k depicts the total number of bins. Similarly, if the Parzen window is employed, the fraction of observations falling into V_j centered on observation $\boldsymbol{y}_j$ is obtained. In both cases, the volume is taken to be a hypersquare with the sides of length h_{ij}. This yields $V_j = \prod_{i=1}^{d} h_{ij}$. Moreover, $h_{ij} = h_i$ and $V_j = V$. If the k-nearest neighbour approach is used, the fraction of observations falling into normalized hypersphere $V_j = \pi^{d/2} R_j^d / \Gamma(1 + d/2)$ of radius R_j centered on observation $\boldsymbol{y}_j$ contains $k_j = k$ observations.

The class widths for the histogram and Parzen window

$$h_i = \frac{y_{i\max} - y_{i\min}}{k}$$

depend on the minimum $y_{i\min} = \min y_{ij}$ and maximum $y_{i\max} = \max y_{ij}$ observations. For the histogram preprocessing origin is preset to

$$y_{i0} = y_{i\min} + \frac{h_i}{2}$$

The $k-1$ nearest neighbours are searched around $\boldsymbol{y}_j$ based on the normalized Euclidean distance

$$R = \max\left\{R_{\min}, \sqrt{\sum_{i=1}^{d}\left(\frac{y_{ik} - y_{ij}}{h_i}\right)^2}\right\} \text{ for } k \neq j, \text{ where } h_i = y_{i\max} - y_{i\min}$$

Minimum radius $0 < R_{\min} \leq 1$. It is advisable to keep it very close to zero. The recommended value is 0.001.

2.2. Global mode detection

The global mode coincides with $\boldsymbol{y}_m$, where empirical density f_{lj} takes on maximum value

$$m = \arg\max f_{lj} \rightarrow (\boldsymbol{y}_m, f_{lm}) \tag{4}$$

If observations are binned into the histogram, then

$$f_{lj} = \frac{k_{lj}}{n_l}\frac{1}{V_j},\ j = 1, \ldots, k \tag{5}$$

where frequencies k_{lj} are all set to k_j initially and total number of observations in class l is

$$n_l = \sum_{j=1}^{k} k_{lj}$$

If the Parzen window or k-nearest neighbour approach is applied,

$$f_{lj} = \frac{k_{lj}}{n_l}\frac{k_j}{V_j},\ j = 1, \ldots, n \tag{6}$$

Frequencies k_{lj} are all set to 1 initially, $n_l = \sum_{j=1}^{n} k_{lj}$ and component weight $w_l = n_l/n$. Moreover, the lth component conditional empirical density at the global mode for the histogram approach

$$f_{i|\hat{i}.lm} = \frac{k_{lm}}{\sum_{j=1,\, y_{\hat{i}j}=y_{\hat{i}m}}^{k} k_{lj}}\frac{1}{h_{im}} = \frac{k_{lm}}{k_{i|\hat{i}.lm}}\frac{1}{h_{im}} \tag{7}$$

is required, where index $\hat{i} = 1, \ldots, i-1, i+1, \ldots, d$. If $d = 1$, then $k_{i|\hat{i}.lm} = n_l$ and $f_{i|\hat{i}.lm} = f_{lm}$. For the Parzen window and k-nearest neighbour approach

$$f_{i|\hat{i}.lm} = \frac{k_{lm}}{\sum_{j=1,\, y_{\hat{i}j}=y_{\hat{i}m}}^{n} k_{lj}}\frac{k_m}{h_{im}} = \frac{k_{lm}}{k_{i|\hat{i}.lm}}\frac{k_m}{h_{im}} \tag{8}$$

2.3. Clustering of observations

The clustering of observations is an iterative procedure of identifying the observations belonging to the lth component. The deviations between k_{lj} and the predictive component frequencies for the histogram approach are given by

$$e_{lj} = k_{lj} - n_l f(\boldsymbol{y}_j|\boldsymbol{\theta}_l)V_j \tag{9}$$

However, for the Parzen window and k-nearest neighbour approach

$$e_{lj} = k_{lj} - n_l f(\boldsymbol{y}_j|\boldsymbol{\theta}_l)V_j/k_j \tag{10}$$

To identify the most deviating observations, relative positive deviations $\varepsilon_{lj} = e_{lj}/k_{lj}$ and maximum positive relative deviation $\varepsilon_{l\max}$ are calculated. Total of the positive and negative deviations

$$e_{l\mathrm{p}} = \sum_{j=1,\, e_{lj}>0}^{k} e_{lj} \text{ and } e_{l\mathrm{n}} = \sum_{j=1,\, e_{lj}<0}^{k} \max\{e_{lj}, -r_j\}$$

where r_j stand for the residual frequencies. If index k is replaced by n the equation can be used with the Parzen window and k-nearest neighbour approach, too. Total of the positive relative deviations of the lth component is then

$$D_l = \frac{e_{l\mathrm{p}}}{n_l} \tag{11}$$

where $0 \leq D_l \leq 1$. The observations that inequality $\varepsilon_{lj} > \varepsilon_{l\max}(1 - a_\mathrm{r})$ holds for are not assumed to belong to the lth component and therefore move to the residue. Number of iterations I depends on acceleration rate $0 < a_\mathrm{r} \leq 1$. It is best to keep a_r close to zero. The recommended value is 0.1. On the contrary, the observations where $e_{lj} < 0$ are transferred back to the lth component. The clustering of observations continues with the renewed rough parameter and component weight estimation until

$$D_l \leq \frac{D_\mathrm{min}}{w_l} \tag{12}$$

Constant $0 < D_\mathrm{min} \leq 1$ is optimized by the information criterion. The clustering of observations ends with the enhanced component parameter estimation.

2.4. Rough component parameter estimation

The clustering of observations depends on the rough component parameters. Proper extraction of observations belonging to the lth component is assured by the restraints that prevent the component from its flowing away from the global mode, as at least one component is supposed to be in the vicinity. The first restraint ensures the equivalence of the probability densities at $\boldsymbol{y}_m$

$$f_{lm} = f(\boldsymbol{y} = \boldsymbol{y}_m|\boldsymbol{\theta}_l) \tag{13}$$

The second restraint makes the global mode of component density coincide with $\boldsymbol{y}_m$

$$\frac{\partial f(\boldsymbol{y} = \boldsymbol{y}_m|\boldsymbol{\theta}_l)}{\partial \boldsymbol{y}} = 0 \tag{14}$$

Total number of unknown parameters $\boldsymbol{\theta}_l = (\boldsymbol{\theta}_{1l}, \ldots, \boldsymbol{\theta}_{dl})^\top$ is greater than the available number of restraints (13) and (14). Therefore additional restraints are required. They are obtained from the equivalence of component conditional empirical densities (Nagode and Fajdiga 2006)

$$\varepsilon f_{i|\hat{i}.lm} = f(y_i = y_{im}|\boldsymbol{\theta}_{il}) = f_{i|\hat{i}.l\max},\ i = 1, \ldots, d \tag{15}$$

Allowing for the independence of components (2), the left hand side of restraint (13) can be rewritten as

$$f_{lm} = \prod_{i=1}^{d} \varepsilon f_{i|\hat{i}.lm}$$

where

$$\varepsilon = \min\left\{1, \left(\frac{f_{lm}}{\prod_{i=1}^{d} f_{i|\hat{i}.lm}}\right)^{\frac{1}{d}}\right\} \tag{16}$$

The left hand side of restraint (15) is thus multiplied by ε to satisfy restraint (13), where (14) and (15) stand for the rigid restraints resulting in rough normal component parameters

$$\mu_{il} = y_{im} \text{ and } \sigma_{il} = \frac{1}{\sqrt{2\pi}\varepsilon f_{i|\hat{i}.lm}} \tag{17}$$

For lognormal and Weibull parametric families see Nagode and Fajdiga (2011b,a), where also loose restraints are introduced. The rigid restraints become loose if y_{im} and $f_{i|\hat{i}.lm}$ of (17) are supposed to be bounded by

$$y_{im} - ah_{im} \leq y_{im} \leq y_{im} + ah_{im} \text{ and } f_{i|\hat{i}.l\min} \leq f_{i|\hat{i}.lm} \leq f_{i|\hat{i}.l\max} \tag{18}$$

Constant a is one for the histogram approach, except for the distributions with $y_i \geq 0$ and $y_{im} < h_{im}$, where $a = y_{im}/h_{im}$. For the Parzen window and k-nearest neighbour $a = y_{im}/2h_{im}$ for the distributions with $y_i \geq 0$ and $y_{im} < h_{im}/2$, otherwise $a = 1/2$. The observations at $f_{i|\hat{i}.l\min}$ are supposed to follow a uniform distribution

$$f_{i|\hat{i}.l\min} = \frac{1}{y_{i|\hat{i}.l\max} - y_{i|\hat{i}.l\min}}$$

where $y_{i|\hat{i}.l\max} = \max y_{i|\hat{i}.lm}$ and $y_{i|\hat{i}.l\min} = \min y_{i|\hat{i}.lm}$. Optimal y_{im} and $f_{i|\hat{i}.lm}$ are obtained by minimizing the maximum relative positive deviation

$$\min_{j=1,\ldots,k \text{ or } n|\varepsilon_{lj}>0,\ 0.001<F(y_{ij}|\boldsymbol{\theta}_{il})<0.999} \varepsilon_{lj} \rightarrow (y_{im}, f_{i|\hat{i}.l\mathrm{m}})$$

as explained thoroughly by Nagode and Fajdiga (2011b,a). The loose restraints prevent superfluous component occurrence if their modes collide considerably.

2.5. Enhanced component parameter estimation

Maximum likelihood is applied to get enhanced component parameters. When the histogram is applied, enhanced normal component parameters are given by

$$\mu_{il} = \frac{1}{n_l}\sum_{j=1}^{k} k_{lj} y_{ij} \text{ and } \sigma_{il}^2 = \frac{1}{n_l}\sum_{j=1}^{k} k_{lj} y_{ij}^2 - \mu_{il}^2 \tag{19}$$

Index k should be replaced by n if the Parzen window or k-nearest neighbour approach is used.

2.6. Component mean and variance calculation

Component means and variances of the normal distribution are calculated to enable the classification of the remaining observations

$$m_{il} = \mu_{il} \text{ and } V_{il} = \sigma_{il}^2 + \mu_{il}^2 \tag{20}$$

2.7. Bayes classification of the remaining observations

With the increase of the number of components, the number n_l of the remaining observations decreases. When the component weight attains the minimum weight

$$w_l \leq w_{\min} = 2lD_{\min} \tag{21}$$

The classification of the remaining observations is accomplished by the Bayes decision rule (Duda and Hart 1973)

$$\begin{gathered} l = \arg\max w_l f(\boldsymbol{y}_j|\boldsymbol{\theta}_l) \\ w_l = w_l + \frac{k_{lj}}{n},\ m_{il} = m_{il} + \frac{k_{lj}(y_{ij} - m_{il})}{nw_l} \text{ and } V_{il} = V_{il} + \frac{k_{lj}(y_{ij}^2 - V_{il})}{nw_l} \end{gathered} \tag{22}$$

where k_{lj} is added to the lth class and the component weight, the component mean as well as the component variance are recalculated (Bishop 1995). Once all k bin means or all n observations are processed, the mixture parameters are gained by inverting (20).

2.8. Algorithm flow

The REBMIX is an iterative numerical procedure listed in Algorithm 1. It requires nine input parameters, whereby the last three should advisably be fixed. It consists of three main loops: the inner 9 → 37, the middle 6 → 41 and the outer loop 4 → 47. The numbers are line indices. In line 2 the observations are preprocessed, as described in Section 2.1. In line 3, constants $D_{\min}$, information criterion IC_{opt} and frequencies k_{lj} are initiated. Next, the outer loop begins. Line 5 presumes that the mixture consists of one component, then the number r of observations to separate is set to n and n_l to n. If ratio n_l/n is greater than the minimum weight introduced in Section 2.7, the middle loop enters. Otherwise, the finite mixture parameter estimation for $k \in K$ is completed.

In lines 7 and 8, global mode argument m is detected as explained in Section 2.2, component weight w_l is calculated and frequencies r_j are all set to zero. If iteration number $I \leq I_{\max}$, the inner loop enters, otherwise in line 38 the component mean and variance are calculated (in Section 2.6). Next, number of components c is set to l, number of observations r is decreased by n_l, l is incremented, number r of the remaining observations joins n_l, residue frequencies r_j are all moved to k_{lj}, and the Stop criterion is determined.

The inner loop is divided into three sections. In line 10 the component parameters are estimated roughly (in Section 2.4). In the second section 11 → 23, total of positive relative

deviations D_l and maximum relative deviation $\varepsilon_{l\max}$ are calculated. The number of iterations depends on acceleration rate a_r. In the third section 24 $\rightarrow$ 35, the maximum and negative deviations are transferred between frequencies k_{lj} and residue r_j. This way deviations e_{lj} are reduced gradually. The negative value of e_{lj} can never be higher than residue value r_j. If this is not true, deviation e_{lj} is corrected, as listed in line 19. When the condition in line 24 is not fulfilled, the enhanced component parameter estimation is carried out (in Section 2.5) and the inner loop ends.

The enhanced component parameter estimation may fail. In this instance, the component parameters are reset to the state just before the failure occurred. In line 42 the remaining observations are classified by the Bayes decision rule, as depicted in Section 2.7. Further on, the information criterion, e.g., Akaike (1974)

$$\mathrm{IC} = -2\log L(c, \boldsymbol{w}, \boldsymbol{\Theta}) + 2M \tag{23}$$

is calculated, whereas the number of free parameters for the univariate normal, lognormal or Weibull mixture can be written as

$$M = 2c + c - 1 \tag{24}$$

The log likelihood function for the binned observations is given by

$$\log L(c, \boldsymbol{w}, \boldsymbol{\Theta}) = \sum_{j=1}^{k} k_j \log f(\boldsymbol{y}_j | c, \boldsymbol{w}, \boldsymbol{\Theta}) \tag{25}$$

Otherwise,

$$\log L(c, \boldsymbol{w}, \boldsymbol{\Theta}) = \sum_{j=1}^{n} \log f(\boldsymbol{y}_j | c, \boldsymbol{w}, \boldsymbol{\Theta}) \tag{26}$$

This way global optimum IC_opt corresponding to the optimal number c_opt of components, weights $\boldsymbol{w}_\mathrm{opt}$ and parameters $\boldsymbol{\Theta}_\mathrm{opt}$ can always be found. In line 46, D_min is decreased in such a way that total of positive relative deviations

$$D = cD_\mathrm{min}^\mathrm{old} = (c+1)D_\mathrm{min}^\mathrm{new}$$

for c and $c+1$ components is preserved. When line 47 is fulfilled, the procedure stops. If index k in Algorithm 1 is replaced by n and line 15 is replaced by (10) the algorithm, presented for the histogram approach, can also be used with the Parzen window and k-nearest neighbour.

3. Examples

To illustrate the use of the REBMIX algorithm, two univariate and three multivariate samples are considered. The **rebmix** is loaded and the prompt before starting new page is set to `TRUE`.

```
R> library("rebmix")
R> devAskNewPage(ask = TRUE)
```

Algorithm 1 REBMIX

Require: Preprocessing, D, $c_{\max}$, Information criterion, Parametric family, K, $R_{\min}$, a_{r} and Restraints.

Ensure: Preprocessing is one of histogram, Parzen window or k-nearest neighbour, $0 < D \leq 1$, $c_{\max} \in \mathbb{N}$, Information criterion is one of AIC, AIC3, AIC4, AICc, BIC, CAIC, HQC, MDL2, MDL5, AWE, CLC, ICL, PC or ICL-BIC, Parametric family is one of normal, lognormal or Weibull, $K \subset \mathbb{N}$, $R_{\min} = 0.001$, $a_{\mathrm{r}} = 0.1$ and Restraints are loose.

for all k such that $k \in K$ **do**
  Preprocessing of observations
  $D_{\min} \leftarrow 0.025$, $\mathrm{IC}_{\mathrm{opt}} \leftarrow \infty$, $k_{lj} \leftarrow k_j$ for $j = 1$ to k
  repeat
    $l \leftarrow 1$, $r \leftarrow n$, $n_l \leftarrow n$
    while $n_l/n > 2lD_{\min}$ **do**
      Global mode detection
      $I \leftarrow 1$, $w_l \leftarrow n_l/n$, $r_j \leftarrow 0$ for $j = 1$ to k
      while $I \leq I_{\max}$ **do**
        Rough component parameter estimation
        $e_{l\mathrm{p}} \leftarrow 0$, $e_{l\mathrm{n}} \leftarrow 0$, $e_{l\max} \leftarrow 0$
        for $j = 1$ **to** k **do**
          $e_{lj} \leftarrow 0$, $\varepsilon_{lj} \leftarrow 0$
          if $k_{lj} > 0$ **or** $r_j > 0$ **then**
            $e_{lj} \leftarrow k_{lj} - n_l f(\boldsymbol{y}_j|\boldsymbol{\theta}_l)V_j$
            if $e_{lj} > 0$ **then**
              $\varepsilon_{lj} \leftarrow e_{lj}/k_{lj}$, $\varepsilon_{l\max} \leftarrow \max\{\varepsilon_{l\max}, \varepsilon_{lj}\}$, $e_{l\mathrm{p}} \leftarrow e_{l\mathrm{p}} + e_{lj}$
            else
              $e_{lj} \leftarrow \max\{e_{lj}, -r_j\}$, $e_{l\mathrm{n}} \leftarrow e_{l\mathrm{n}} - e_{lj}$
            end if
          end if
        end for
        $D_l \leftarrow e_{l\mathrm{p}}/n_l$, $\varepsilon_{l\max} \leftarrow \varepsilon_{l\max}(1 - a_{\mathrm{r}})$
        if $D_l > D_{\min}/w_l$ **then**
          for all j such that $1 \leq j \leq k$ **and** $\varepsilon_{lj} > \varepsilon_{l\max}$ **do**
            $k_{lj} \leftarrow k_{lj} - e_{lj}$, $r_j \leftarrow r_j + e_{lj}$, $n_l \leftarrow n_l - e_{lj}$
          end for
          $e_{l\mathrm{p}} \leftarrow e_{l\mathrm{p}}/D_l - n_l$, $f \leftarrow e_{l\mathrm{p}}/e_{l\mathrm{n}}$ if $e_{l\mathrm{n}} > e_{l\mathrm{p}}$ otherwise $f \leftarrow 1$
          for all j such that $1 \leq j \leq k$ **and** $e_{lj} < 0$ **do**
            $e_{lj} \leftarrow fe_{lj}$, $k_{lj} \leftarrow k_{lj} - e_{lj}$, $r_j \leftarrow r_j + e_{lj}$, $n_l \leftarrow n_l - e_{lj}$
          end for
          $w_l \leftarrow n_l/n$
        else
          Enhanced component parameter estimation, **break**
        end if
        $I \leftarrow I + 1$
      end while
      Component mean and variance calculation
      $c \leftarrow l$, $r \leftarrow r - n_l$, $l \leftarrow l + 1$, $n_l \leftarrow r$, $k_{lj} \leftarrow r_j$ for $j = 1$ to k
      Stop $\leftarrow c \geq k$ **or** $c \geq c_{\max}$ **or** $cD_{\min} < D$, **break** if Stop = **true**
    end while
    Bayes classification of the remaining observations as well as log likelihood and information criterion calculation
    if $\mathrm{IC} < \mathrm{IC}_{\mathrm{opt}}$ **then**
      $\log L \rightarrow \log L_{\mathrm{opt}}$, $\mathrm{IC} \rightarrow \mathrm{IC}_{\mathrm{opt}}$, $c \rightarrow c_{\mathrm{opt}}$, $\boldsymbol{w} \rightarrow \boldsymbol{w}_{\mathrm{opt}}$, $\boldsymbol{\Theta} \rightarrow \boldsymbol{\Theta}_{\mathrm{opt}}$
    end if
    $D_{\min} \leftarrow cD_{\min}/(c + 1)$
  until Stop = **true**
end for

3.1. Galaxy dataset

The dataset analysed in Roeder (1990) contains the measurements of the velocities of 82 galaxies diverging away from our own galaxy. The multimodality of the velocities may indicate the presence of super clusters of galaxies surrounded by large voids, each mode representing a cluster as it moves away at its own speed (Roeder 1990, gives more background). Richardson and Green (1997) concluded from their approach that the number of components ranged from 5 to 7, while McLachlan and Peel (1997) provided the support for six components. Stephens (2000) reported that three components were optimal for the mixture of normal and four for the mixture of t distributions.

The galaxy dataset is loaded and written as tab delimited ASCII file in `galaxy.txt`.

```
R> data("galaxy")
R> write.table(galaxy, file = "galaxy.txt", sep = "\t",
+      eol = "\n", row.names = FALSE, col.names = FALSE)
```

The `REBMIX` object and `Table` are initialized.

```
R> REBMIX <- array(list(NULL), c(3, 3, 3))
R> Table <- NULL
```

Total of positive relative deviations `D` is set to 0.0025, maximum number of components `cmax` to 12. The influence of the Akaike (Akaike 1974) information criterion AIC, the Bayesian (Schwarz 1978) information criterion BIC and the classification likelihood criterion CLC (see Biernacki and Govaert 1997) for normal, lognormal and Weibull parametric families and the three preprocessing types on predictive number of components c is studied. The optimal number of classes and nearest neighbours are searched within broad utmost limits `K`.

```
R> Preprocessing <- c("histogram", "Parzen window", "k-nearest neighbour")
R> InformationCriterion <- c("AIC", "BIC", "CLC")
R> pdf <- c("normal", "lognormal", "Weibull")
R> K <- list(7:20, 7:20, 2:10)
```

See `help("REBMIX")` in **rebmix** for details about specifying arguments for the REBMIX algorithm. For Table 1 to be filled in function `REBMIX` is called $3 \times 3 \times 3 = 27$ times.

```
R> for (i in 1:3) {
+      for (j in 1:3) {
+          for (k in 1:3) {
+              REBMIX[[i, j, k]] <- REBMIX(Dataset = "galaxy.txt",
+                  Preprocessing = Preprocessing[k], D = 0.0025,
+                  cmax = 12, InformationCriterion = InformationCriterion[j],
+                  pdf = pdf[i], K = K[[k]])
+              if (is.null(Table))
+                  Table <- REBMIX[[i, j, k]]$summary
+              else Table <- merge(Table, REBMIX[[i, j,
+                  k]]$summary, all = TRUE, sort = FALSE)
+          }
+      }
+  }
```

It returns an object of class `REBMIX`. Data frame `w` contains c component weights w_l summing to 1, `Theta` stands for a $3 \times d \times c$ data frame. The first, fourth, etc. rows contain c parametric family types `pdfi`, one of normal, lognormal or Weibull. The second, fifth, etc. rows contain c component parameters `theta1.i`, one of μ_{il} for normal and lognormal distributions or θ_{il} for Weibull distribution. The third, sixth, etc. rows contain c component parameters `theta2.i`. One of σ_{il} for normal and lognormal distributions or β_{il} for Weibull distribution. In the `summary` data frame additional information about dataset, preprocessing, D, $c_{\max}$, information criterion type, $R_{\max}$, a_{r}, restraints type, optimal c, optimal k, y_{i0}, optimal h_i, calculation time t_{c} in ms, information criterion IC and log likelihood $\log L$ is stored.

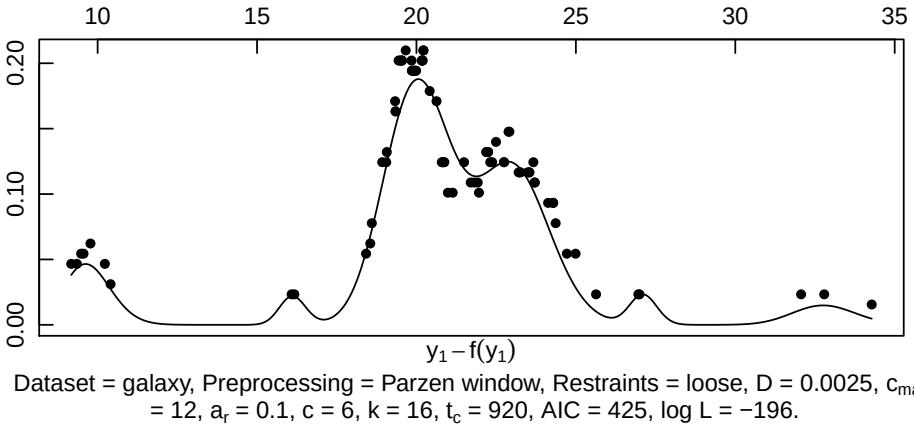

Figure 1: Galaxy dataset. Empirical densities (circles) and predictive lognormal mixture density (solid line).

The `plot` method delivers a fitted finite mixture with the legend in Figure 1. For the details about specifying arguments for the `plot` method see `help("plot.REBMIX")`. The maximum log likelihood resulting in 6 components is obtained for the lognormal parametric family, the Parzen window preprocessing and the AIC (see Figure 1), whereas most frequently four components appear in Table 1. Thus the **rebmix** leads to the number of components similar to Stephens (2000). The two spurious components reported about by McLachlan and Peel (1997) can be identified by the algorithm, too. For the particular dataset only the AIC is appropriate. It gives 3 to 6 components. The BIC and CLC have turned out to be inappropriate.

Preprocessing	Information criterion	Normal				Lognormal				Weibull			
		c	k	IC	$\log L$	c	k	IC	$\log L$	c	k	IC	$\log L$
histogram		5	20	430	-201	3	15	427	-205	4	19	428	-203
Parzen window	AIC	4	15	435	-206	6	16	425	-196	3	20	456	-220
k-nearest neighbour		4	7	451	-215	4	7	451	-214	4	10	457	-217
histogram		3	15	455	-210	2	19	443	-210	4	19	455	-203
Parzen window	BIC	4	15	461	-206	4	16	459	-205	3	20	476	-220
k-nearest neighbour		4	7	478	-215	4	7	477	-214	2	10	482	-230
histogram		2	19	434	-217	2	19	421	-210	4	19	440	-203
Parzen window	CLC	7	15	422	-197	6	16	416	-196	1	7	482	-241
k-nearest neighbour		1	6	481	-240	1	8	510	-255	1	6	483	-241

Table 1: Number of component classes and nearest neighbours for galaxy dataset.

3.2. Iris dataset

The well known set of iris data, as collected originally by Anderson (1935) and first analysed by Fisher (1936), is considered here. It is available at Asuncion and Newman (2007) consisting of the measurements of the length and width of both sepals and petals of 50 plants for each of the three types of iris species setosa, versicolor and virginica.

The iris dataset is loaded, split into three subsets for the three species and written as tab delimited ASCII files without the `Species` column in `iris.txt`, `iris1.txt`, `iris2.txt` and `iris3.txt`.

```
R> data("iris")
R> iriscolnames <- !(colnames(iris) %in% "Species")
R> iris1 <- iris[iris$Species == "setosa", iriscolnames]
R> iris2 <- iris[iris$Species == "versicolor", iriscolnames]
R> iris3 <- iris[iris$Species == "virginica", iriscolnames]
R> iris <- iris[, iriscolnames]
R> write.table(iris, file = "iris.txt", sep = "\t", eol = "\n",
+      row.names = FALSE, col.names = FALSE)
R> write.table(iris1, file = "iris1.txt", sep = "\t", eol = "\n",
+      row.names = FALSE, col.names = FALSE)
R> write.table(iris2, file = "iris2.txt", sep = "\t", eol = "\n",
+      row.names = FALSE, col.names = FALSE)
R> write.table(iris3, file = "iris3.txt", sep = "\t", eol = "\n",
+      row.names = FALSE, col.names = FALSE)
```

The `REBMIX` object and `Table` are reinitialized.

```
R> REBMIX <- array(list(NULL), c(6, 3))
R> Table <- NULL
```

The three preprocessing types and six selection criteria AIC, AWE (Banfield and Raftery 1993), BIC, CLC, integrated classification likelihood criterion ICL as proposed by Biernacki, Celeux, and Govaert (1998) implemented with $\alpha = 0.5$ and its approximation ICL-BIC for the normal parametric family are compared. The optimal number of classes and nearest neighbours are searched within broad utmost limits `K`.

```
R> Preprocessing <- c("histogram", "Parzen window", "k-nearest neighbour")
R> InformationCriterion <- c("AIC", "AWE", "BIC", "CLC",
+      "ICL", "ICL-BIC")
R> K <- list(6:25, 6:25, 3:13)
```

Argument `Dataset` points to the ASCII files. The `REBMIX` function is called $6 \times 3 = 18$ times. The number of components is assessed for the set as well as for the three subsets. The results of the analysis are listed in Table 2.

```
R> for (i in 1:6) {
+      for (j in 1:3) {
+          REBMIX[[i, j]] <- REBMIX(Dataset = c("iris.txt",
```

```
+             "iris1.txt", "iris2.txt", "iris3.txt"), Preprocessing = Preprocessing[j],
+             InformationCriterion = InformationCriterion[i],
+             pdf = rep("normal", 4), K = K[[j]])
+         if (is.null(Table))
+             Table <- REBMIX[[i, j]]$summary
+         else Table <- merge(Table, REBMIX[[i, j]]$summary,
+             all = TRUE, sort = FALSE)
+     }
+ }
```

Preprocessing	Information criterion	c	k	IC	$\log L$	c for species setosa	versicolor	virginica
histogram		11	25	508	-156	13	5	10
Parzen window	AIC	14	25	545	-147	8	5	13
k-nearest neighbour		4	7	653	-292	2	2	2
histogram		3	17	993	-323	1	1	1
Parzen window	AWE	2	12	1004	-391	1	1	1
k-nearest neighbour		4	7	1074	-292	1	1	1
histogram		5	18	745	-262	2	3	2
Parzen window	BIC	4	12	741	-283	1	3	4
k-nearest neighbour		4	7	759	-292	2	2	2
histogram		15	25	294	-127	13	14	13
Parzen window	CLC	15	25	330	-142	14	13	13
k-nearest neighbour		4	7	618	-292	5	2	3
histogram		5	18	779	-262	1	3	2
Parzen window	ICL	4	12	778	-283	1	3	4
k-nearest neighbour		4	7	792	-292	1	2	2
histogram		5	18	781	-262	1	3	2
Parzen window	ICL-BIC	4	12	779	-283	1	3	4
k-nearest neighbour		4	7	793	-292	1	2	2

Table 2: Number of component classes and nearest neighbours for iris dataset.

It can be concluded that AIC and CLC overestimate the number of components for the set (left part of the table) and for the three subsets significantly. Only the AWE for the histogram preprocessing recognizes three components for the set and one component for each subset. The BIC recognizes too many components for the subsets.

However, according to the log likelihood, ICL and ICL-BIC provide the best results for the histogram preprocessing. Interestingly, the number of components of the set equals 5 for the histogram, which is in accordance with Wilson (1982), who suggested that both, the versicolor and virginica species should be split into two subspecies although the analysis by McLachlan and Peel (2000) using maximum likelihood methods suggests that this is not justified for the virginica subset. Also, Stephens (2000) reported that the superfluous components might appear to model the lack of normality in the subset, rather than interpretable groups. The subset results show that the setosa subset is represented by a single component. The numbers of predictive components for versicolor and virginica are 3 and 2, respectively. The `plot` method delivers Figure 2.

3.3. Wine dataset

Next, the results of a wine recognition problem are considered. The set consists of 178

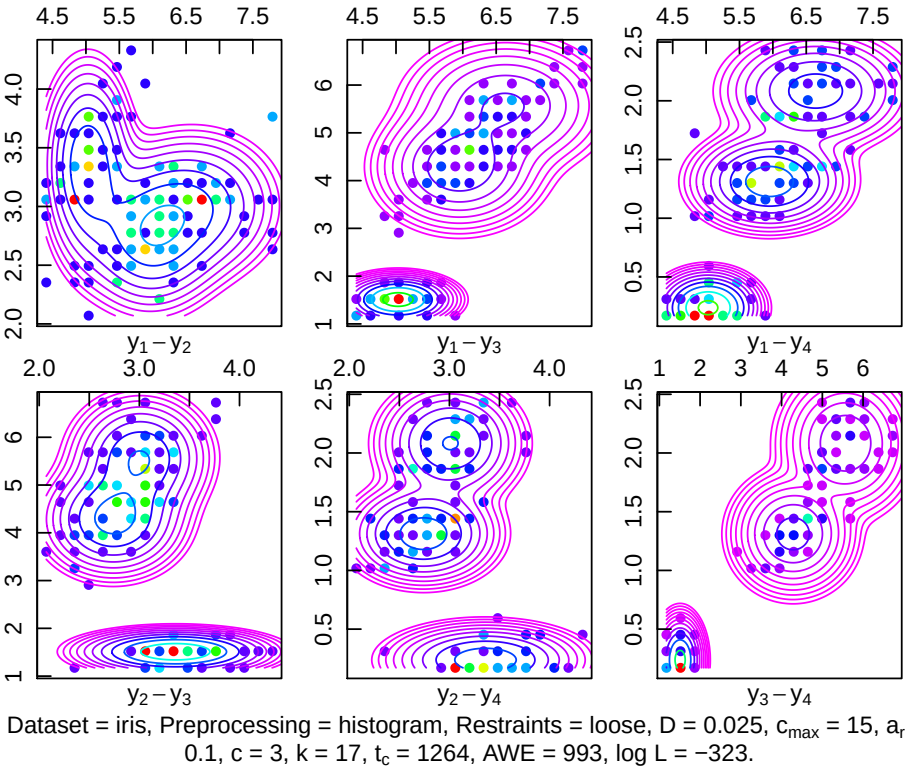

Figure 2: Iris dataset. Empirical densities (circles) and predictive multivariate marginal normal mixture densities (contour lines).

13 dimensional exemplars that are a set of chemical analysis of three types of wine (Asuncion and Newman 2007).

The standalone replication script `rebmix.R` delivers the R sample code. The results of the analysis are plotted in Figure 3 and listed in Table 3. The AIC and CLC overestimate the number of components for the set (left part of the table) and for the subsets and are thus not applicable. The AWE, BIC, ICL and ICL-BIC recognize three components for the set and one component for each subset for the histogram and Parzen window preprocessing. In a classification context, this is a well posed problem with well behaved class structures (see also Roberts, Everson, and Rezek 2000).

3.4. Complex 1 dataset

Next, 15 component univariate normal mixture is generated and the probability density is estimated.

```
R> n <- c(998, 263, 1086, 487, 213, 1076, 232, 784, 840,
+       461, 773, 24, 811, 1091, 861)
R> Theta <- rbind(pdf = "normal", theta1 = c(688.4, 265.1,
+       30.8, 934, 561.6, 854.9, 883.7, 758.3, 189.3, 919.3,
+       98, 143, 202.5, 628, 977), theta2 = c(12.4, 14.6,
+       14.8, 8.4, 11.7, 9.2, 6.3, 10.2, 9.5, 8.1, 14.7,
```

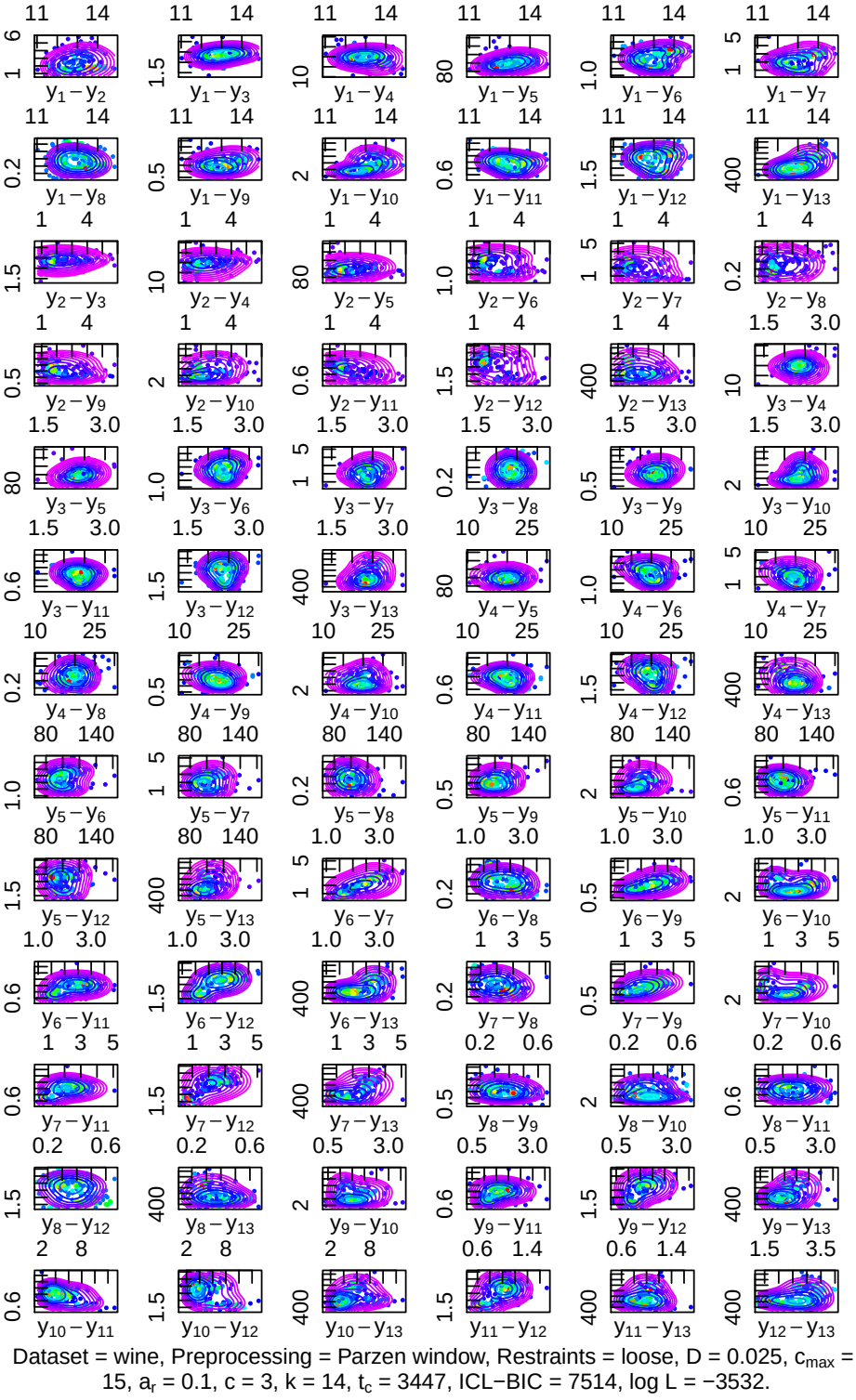

Figure 3: Wine dataset. Empirical densities (circles) and predictive multivariate marginal normal mixture densities (contour lines).

Preprocessing	Information criterion	c	k	IC	$\log L$	c for cultivar 1	2	3
histogram		12	11	6815	-3084	14	15	13
Parzen window	AIC	10	12	6826	-3144	14	15	12
k-nearest neighbour		7	11	7074	-3349	1	1	1
histogram		3	15	8078	-3487	1	1	1
Parzen window	AWE	3	14	8169	-3532	1	1	1
k-nearest neighbour		1	8	8374	-4013	1	1	1
histogram		3	15	7388	-3487	1	1	1
Parzen window	BIC	3	14	7478	-3532	1	1	1
k-nearest neighbour		4	6	7631	-3538	1	1	1
histogram		15	8	6076	-3017	14	15	13
Parzen window	CLC	15	16	6214	-3071	14	15	13
k-nearest neighbour		8	11	6691	-3326	3	11	7
histogram		3	15	7423	-3487	1	1	1
Parzen window	ICL	3	14	7514	-3532	1	1	1
k-nearest neighbour		4	6	7673	-3538	1	1	1
histogram		3	15	7423	-3487	1	1	1
Parzen window	ICL-BIC	3	14	7514	-3532	1	1	1
k-nearest neighbour		4	6	7674	-3538	1	1	1

Table 3: Number of component classes and nearest neighbours for wine dataset.

```
+      11.7, 7.4, 10.1, 14.6))
R> RNGMIX <- RNGMIX(Dataset = "complex1.txt", n = n, Theta = Theta)

RNGMIX Version 2.2.1
Dataset = complex1.txt

R> REBMIX <- REBMIX(Dataset = "complex1.txt", Preprocessing = "histogram",
+      D = 0.0025, cmax = 30, InformationCriterion = "BIC",
+      pdf = "normal", K = seq(14, 200, 4))

REBMIX Version 2.2.1
Dataset = complex1.txt

R> REBMIX$w[[1]]

    comp1   comp2   comp3   comp4   comp5   comp6   comp7   comp8
w 0.13861 0.09292 0.10889 0.10028 0.07798 0.10169 0.09841 0.08014
    comp9  comp10  comp11  comp12  comp13  comp14  comp15  comp16
w 0.07841 0.02560 0.00342 0.01806 0.01547 0.01024 0.00518 0.02143
   comp17
w 0.02327

R> REBMIX$Theta[[1]]

        comp1  comp2  comp3  comp4  comp5  comp6  comp7  comp8  comp9
pdf    normal normal normal normal normal normal normal normal normal
theta1  193.6  856.9  628.2  688.9  758.9  927.7   32.2  979.0  100.0
```

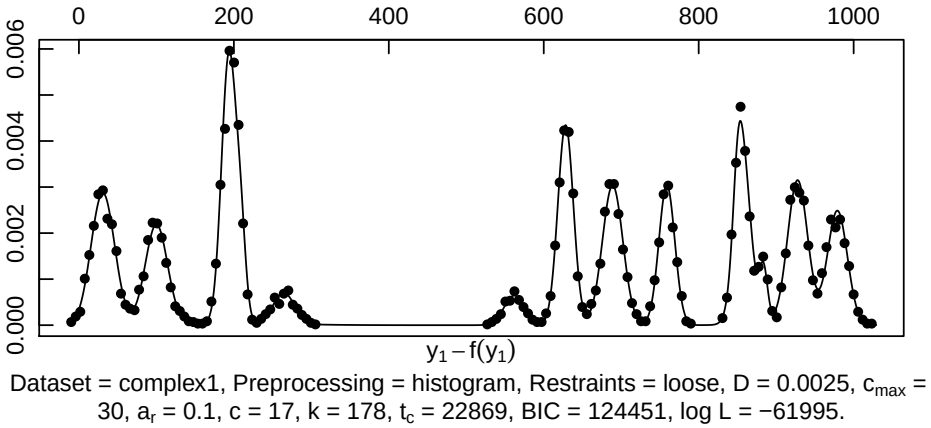

Figure 4: Complex 1 dataset. Empirical densities (circles) and predictive normal mixture density (solid line).

```
theta2   9.48  10.56   9.99  13.17  10.23  12.94  14.65  12.85  14.33
       comp10 comp11 comp12 comp13 comp14 comp15 comp16 comp17
pdf    normal normal normal normal normal normal normal normal
theta1  209.1  883.9  850.6  883.6   18.8  220.7  562.1  264.4
theta2   6.43  30.94   5.92   5.27  12.23  68.68  13.26  13.62
```

```
R> REBMIX$summary
```

```
   Dataset Preprocessing      D cmax InformationCriterion  ar
1 complex1     histogram 0.0025   30                 BIC 0.1
  Restraints  c   k    y0    h    tc     IC   logL
1      loose 17 178 -9.95 5.84 22869 124451 -61995
```

Random seed $r_{\text{seed}} = -1$, the number of classes ranges from 14 (Sturges 1926) to 200 corresponding to the RootN rule, the maximum number of components is set to 30 and the information criterion to BIC. Total number of observations $n = 10000$. Consequently, the histogram preprocessing is applied. See `help("RNGMIX")` in **rebmix** for details about specifying arguments for the random univariate or multivariate finite mixture generation.

From Figure 4 it can be noted that it is possible to restore the mixture of 15 well separated components. Total of positive relative deviations $D = 0.025$ is mostly appropriate. However, if the components with a low probability of occurrence are expected, it should decrease. The complex 1 dataset requires $D = 0.0025$ to get at the minimum BIC. The calculation time increases if D decreases. The optimal BIC is observed by a larger number of iterations.

3.5. Simulated 1 dataset

Set 1 consists of $n = 625$ four dimensional observations obtained by generating samples separately from each of five normal distributions. The component sample sizes, means and covariance matrices, which are those adopted in Bozdogan (1993) and Celeux and Soromenho (1996), are displayed below

$$\begin{array}{lll}
\boldsymbol{\mu}_1 = (10, 12, 10, 12)^\top & \boldsymbol{\Sigma}_1 = \boldsymbol{I}_p & n_1 = 75 \\
\boldsymbol{\mu}_2 = (8.5, 10.5, 8.5, 10.5)^\top & \boldsymbol{\Sigma}_2 = \boldsymbol{I}_p & n_2 = 100 \\
\boldsymbol{\mu}_3 = (12, 14, 12, 14)^\top & \boldsymbol{\Sigma}_3 = \boldsymbol{I}_p & n_3 = 125 \\
\boldsymbol{\mu}_4 = (13, 15, 7, 9)^\top & \boldsymbol{\Sigma}_4 = 4\boldsymbol{I}_p & n_4 = 150 \\
\boldsymbol{\mu}_5 = (7, 9, 13, 15)^\top & \boldsymbol{\Sigma}_5 = 9\boldsymbol{I}_p & n_5 = 175
\end{array}$$

The optimal $c = 5$ component normal mixture model with diagonal component covariance matrices is fitted (McLachlan and Ng 2000; McLachlan and Peel 2000) by using the EMMIX algorithm McLachlan *et al.* (1999). It results in BIC = 11479.

The EMMIX algorithm recognizes five components as optimal regardless of the selection criterion. Ten random starts are performed to initialize the EM algorithm. The solution corresponding to the largest local maximum of the log likelihood located is taken as the MLE after the elimination of local maximizers considered to be spurious on the basis of the relevant sizes of the fitted generalized component variances.

Next, 100 samples are generated with random seeds r_{seed} ranging from -1 to -100.

```
R> n <- c(75, 100, 125, 150, 175)
R> Theta <- rbind(rep("normal", 5), c(10, 8.5, 12, 13, 7),
+      c(1, 1, 1, 2, 3), rep("normal", 5), c(12, 10.5, 14,
+          15, 9), c(1, 1, 1, 2, 3), rep("normal", 5), c(10,
+          8.5, 12, 7, 13), c(1, 1, 1, 2, 3), rep("normal",
+          5), c(12, 10.5, 14, 9, 15), c(1, 1, 1, 2, 3))
R> RNGMIX <- RNGMIX(Dataset = paste("Simulated1_", 1:100,
+      ".txt", sep = ""), n = n, Theta = Theta)
```

In total, 100 finite mixture estimations are performed for the histogram preprocessing and BIC.

```
R> REBMIX <- REBMIX(paste("Simulated1_", 1:100, ".txt",
+      sep = ""), Preprocessing = "histogram", InformationCriterion = "BIC",
+      pdf = rep("normal", 4), K = seq(10, 28, 2))
```

The results are as follows:

```
R> c <- REBMIX$summary$c
R> IC <- REBMIX$summary$IC
R> summary(c)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   3.00    5.00    6.00    6.04    7.00   13.00
```

```
R> summary(IC, digits = 5)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  11474   11722   11879   11895   12062   12292
```

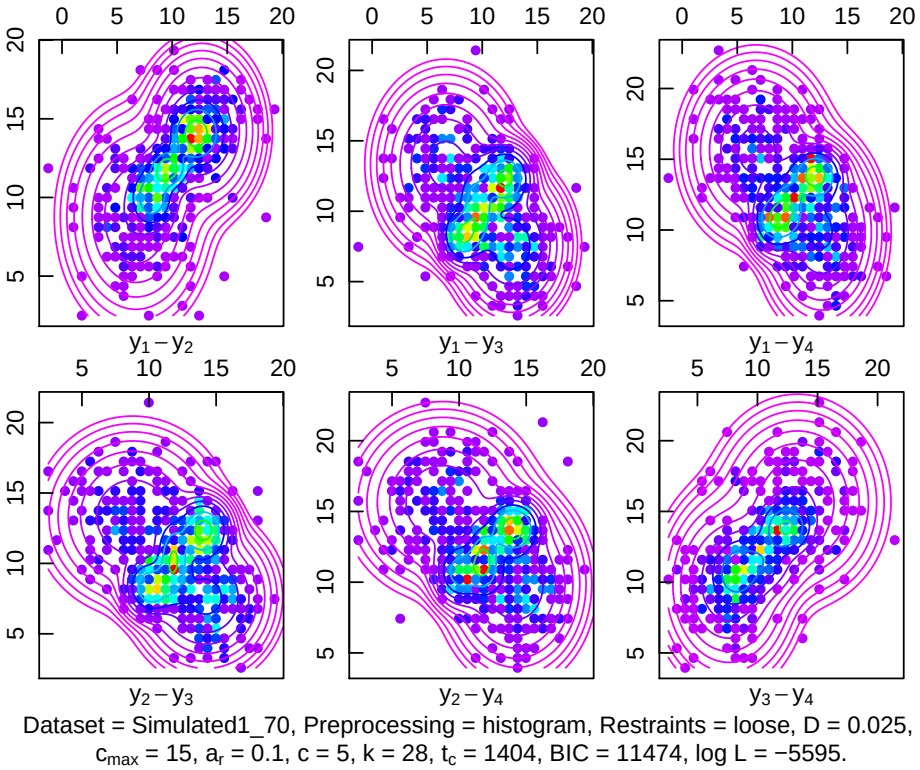

Figure 5: Simulated 1 dataset. Empirical densities (circles) and predictive multivariate marginal normal mixture densities (contour lines).

The minimum BIC = 11474 corresponds to the 70th sample in Figure 5. The BIC predicts 6.04 components on average, where probability P of identifying exactly $c = 5$ components equals 0.3. The fastest histogram preprocessing results in the highest probability of identifying the true number of components and in the most suitable average number of components c for the simulated 1 dataset. The Parzen window and k-nearest neighbour are therefore left out here. The REBMIX approaches the EMMIX regarding the information criterion only at the lower limit. However, it is insensitive to spurious local maximizers, robust and fast especially if the optimal number of classes or the nearest neighbours can be guessed at least approximately.

4. Conclusions and future work

The article presents the REBMIX algorithm and the **rebmix** package. The galaxy, iris, wine, complex 1 and simulated 1 datasets are studied on the x64 architecture. By applying the **tikzDevice** package (Sharpsteen and Bracken 2010), LaTeX plots with legends can be obtained. The REBMIX algorithm leads to slightly worse estimates than the EM algorithm and can be used to assess the initial set of the unknown parameters and number of components for the EM algorithm or as a standalone procedure that is a good compromise between the nonparametric and parametric methods to the finite mixture estimation.

Its major superiorities are robustness and time efficiency especially with the histogram and Parzen window preprocessing for all sample sizes. The k-nearest neighbour is more suitable

for smaller samples. Its advantages are more stressed for complex mixtures composed of numerous components.

There are several possibilities to further decrease the calculation time, which have been left for the future. The number of components affects the computational time, but it does not contribute to the numerical instability of the algorithm. The REBMIX is being extended to mixed categorical variables. The binomial parametric family is already available in the attached C source code and is to be validated. The pseudo code in Nagode and Fajdiga (2011b) differs slightly from Algorithm 1. The effect of the differences on the predictive finite mixtures is negligible and improve the calculation speed considerably. Potentially, the REBMIX can be used for pattern recognition and as a neural network.

References

Akaike H (1974). "A New Look at the Statistical Model Identification." *IEEE Transactions on Automatic Control*, **19**, 716–723.

Anderson E (1935). "The Irises of the Gaspé Peninsula." *Bulletin of the American Iris Society*, **59**, 2–5.

Asuncion A, Newman DJ (2007). "UCI Machine Learning Repository." URL http://www.ics.uci.edu/~mlearn/MLRepository.html.

Banfield JD, Raftery AE (1993). "Model-Based Gaussian and Non-Gaussian Clustering." *Biometrics*, **49**, 803–821.

Benaglia T, Chauveau D, Hunter DR, Young DS (2009). "**mixtools**: An R Package for Analyzing Finite Mixture Models." *Journal of Statistical Software*, **32**, 1–29.

Biernacki C, Celeux G, Govaert G (1998). "Assessing a Mixture Model for Clustering with the Integrated Classification Likelihood." *Technical Report 3521*, INRIA, Rhône-Alpes.

Biernacki C, Govaert G (1997). "Using the Classification Likelihood to Choose the Number of Clusters." *Computing Science and Statistics*, **29**, 451–457.

Bishop CM (1995). *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford.

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 O Opitz, B Lausen, R Klar (eds.), *Information and Classification*, pp. 40–54. Springer-Verlag, Heidelberg.

Bučar T, Nagode M, Fajdiga M (2004). "Reliability Approximation Using Finite Weibull Mixture Distributions." *Reliability Engineering & System Safety*, **84**, 241–251.

Celeux H, Soromenho G (1996). "An Entropy Criterion for Assessing the Number of Clusters in a Mixture Model." *Journal of Classification*, **13**, 195–212.

Duda RO, Hart PE (1973). *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York.

Fisher RA (1936). "The Use of Multiple Measurements in Taxonomic Problems." *Annals of Eugenics*, **7**, 179–188.

Fraley C, Raftery AE (2007). "Model-based Methods of Classification: Using the **mclust** Software in Chemometrics." *Journal of Statistical Software*, **18**, 1–13.

Frühwirth-Schnatter S (2006). *Finite Mixture and Markov Switching Models*. Springer-Verlag Series in Statistics, New York.

Grün B, Leisch F (2008). "**FlexMix** Version 2: Finite Mixtures with Concomitant Variables and Varying and Constant Parameters." *Journal of Statistical Software*, **28**, 1–35.

Ingrassia S, Rocci R (2007). "Constrained Monotone EM Algorithms for Finite Mixture of Multivariate Gaussians." *Computational Statistics & Data Analysis*, **51**, 5339–5351.

Leisch F (2004). "**FlexMix**: A General Framework for Finite Mixture Models and Latent Class Regression in R." *Journal of Statistical Software*, **11**, 1–18.

Majeske KD (2003). "A Mixture Model for Automobile Warranty Data." *Reliability Engineering & System Safety*, **81**, 71–77.

McLachlan GJ, Ng SK (2000). "A Comparison of some Information Criteria for the Number of Components in a Mixture Model." *Technical report*, Department of Mathematics, University of Queensland, Brisbane.

McLachlan GJ, Peel D (1997). "Contribution to the Discussion of Paper by S. Richardson and P.J. Green." *Journal of the Royal Statistical Society B*, **59**, 779–780.

McLachlan GJ, Peel D (2000). *Finite Mixture Models*. John Wiley & Sons, New York.

McLachlan GJ, Peel D, Basford KE, Adams P (1999). "Fitting of Mixtures of Normal and t-Components." *Journal of Statistical Software*, **4**(2).

Nagode M, Fajdiga M (1998). "A General Multi-Modal Probability Density Function Suitable for the Rainflow Ranges of Stationary Random Processes." *International Journal of Fatigue*, **20**, 211–223.

Nagode M, Fajdiga M (2000). "An Improved Algorithm for Parameter Estimation Suitable for Mixed Weibull Distributions." *International Journal of Fatigue*, **22**, 75–80.

Nagode M, Fajdiga M (2006). "An Alternative Perspective on the Mixture Estimation Problem." *Reliability Engineering & System Safety*, **91**, 388–397.

Nagode M, Fajdiga M (2011a). "The REBMIX Algorithm for the Multivariate Finite Mixture Estimation." *Communications in Statistics - Theory and Methods*, **40**(11), 2022–2034.

Nagode M, Fajdiga M (2011b). "The REBMIX Algorithm for the Univariate Finite Mixture Estimation." *Communications in Statistics - Theory and Methods*, **40**(5), 876–892.

Nagode M, Klemenc J, Fajdiga M (2001). "Parametric Modelling and Scatter Prediction of Rainflow Matrices." *International Journal of Fatigue*, **23**, 525–532.

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Richardson S, Green PJ (1997). "On Bayesian Analysis of Mixtures with an Unknown Number of Components." *Journal of the Royal Statistical Society B*, **59**, 731–792.

Roberts SJ, Everson R, Rezek I (2000). "Maximum Certainty Data Partitioning." *Pattern Recognition*, **33**, 833–839.

Roeder K (1990). "Density Estimation with Confidence Sets Exemplified by Superclusters and Voids in the Galaxies." *Journal of American Statistical Association*, **85**, 617–624.

Schwarz G (1978). "Estimating the Dimension of the Model." *The Annals of Statistics*, **6**, 461–464.

Sharpsteen C, Bracken C (2010). ***tikzDevice**: A Device for R Graphics Output in PGF/TikZ Format*. R package version 0.5.3, URL http://CRAN.R-project.org/package=tikzDevice.

Stephens M (2000). "Bayesian Analysis of Mixture Models with an Unknown Number of Components - An Alternative to Reversible Jump Methods." *The Annals of Statistics*, **28**, 40–74.

Sturges HA (1926). "The Choice of a Class Interval." *Journal of American Statistical Association*, **21**, 65–66.

Sultan KS, Ismail MA, Al-Moisheer AS (2007). "Mixture of Two Inverse Weibull Distributions: Properties and Estimation." *Computational Statistics & Data Analysis*, **51**, 5377–5387.

Touw AE (2009). "Bayesian Estimation of Mixed Weibull Distributions." *Reliability Engineering & System Safety*, **94**, 463–473.

van Dijk HK, Hoogerheide LF, Ardia D (2009). "Adaptive Mixture of Student-t Distributions as a Flexible Candidate Distribution for Efficient Simulation: The R Package **AdMit**." *Journal of Statistical Software*, **29**, 1–31.

Wilson SR (1982). "Sound and Exploratory Data Analysis." In *Compstat 1982, Proceedings Computational Statistics*, pp. 447–450. Physica-Verlag, Vienna.

Affiliation:

Marko Nagode
Faculty of Mechanical Engineering
Aškerčeva 6
1000 Ljubljana, Slovenia
E-mail: Marko.Nagode@fs.uni-lj.si
URL: http://www.fs.uni-lj.si/