rebmix: An R Package for Continuous and Discrete 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, bootstrapping and plotting of the finite mixtures. It relies on the REBMIX algorithm that requires preprocessing, information criterion and conditionally independent normal, lognormal, Weibull, gamma, binomial, Poisson or Dirac 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 can be used either to assess the initial set of the unknown parameters and number of components for, e.g., 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, complex 2 and simulated 1.

Keywords: continuous variable, discrete variable, finite mixture, parameter estimation, R software, REBMIX algorithm.

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) and avoids the drawbacks of

the EM algorithm:

- The EM algorithm converges to a local maximum of the likelihood function very quickly.
- There are often several other promising local optimal solutions in the vicinity of the solutions obtained from methods that provide good initial guesses of the solution.
- Model selection criteria usually assumes that the global optimal solution of the log-likelihood function can be obtained. However, achieving this is computationally intractable.
- Some regions in the search space do not contain any promising solutions. The promising and non-promising regions co-exist, and it often becomes challenging to avoid wasting computational resources to search in non-promising regions.

reported in Reddy and Rajaratnam (2010) by updating the number of components, component weights and component parameters sequentially and not simultaneously (see also Celeux, Chrétien, Forbes, and Mkhadri 2001). Later on the REBMIX has evolved (Nagode and Fajdiga 2000; Nagode, Klemenc, and Fajdiga 2001; Nagode and Fajdiga 2006, 2011b,a), but its kernel has remained almost unchanged. The paper extends it to discrete variables by adding binomial, Poisson and Dirac parametric families. Gamma parametric family is added as well.

REBMIX 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, bootstrapping and plotting of the finite mixtures in the R language and environment for statistical computing (R Development Core Team 2011). The **rebmix** package has been published on the Comprehensive R Archive Network and is available at `http://CRAN.R-project.org/package=rebmix`.

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, complex 2 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 or discrete 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 components can currently belong to either normal

$$f(y_i|\boldsymbol{\theta}_{il}) = \frac{1}{\sqrt{2\pi}\sigma_{il}} \exp\left\{-\frac{1}{2}\frac{(y_i-\mu_{il})^2}{\sigma_{il}^2}\right\},$$

lognormal

$$f(y_i|\boldsymbol{\theta}_{il}) = \frac{1}{\sqrt{2\pi}\sigma_{il}y_i} \exp\left\{-\frac{1}{2}\frac{(\log(y_i)-\mu_{il})^2}{\sigma_{il}^2}\right\},$$

Weibull

$$f(y_i|\boldsymbol{\theta}_{il}) = \frac{\beta_{il}}{\theta_{il}}\left(\frac{y_i}{\theta_{il}}\right)^{\beta_{il}-1} \exp\left\{-\left(\frac{y_i}{\theta_{il}}\right)^{\beta_{il}}\right\},$$

gamma

$$f(y_i|\boldsymbol{\theta}_{il}) = \frac{1}{\Gamma[\beta_{il}]y_i}\left(\frac{y_i}{\theta_{il}}\right)^{\beta_{il}} \exp\left\{-\frac{y_i}{\theta_{il}}\right\},$$

binomial

$$f(y_i|\boldsymbol{\theta}_{il}) = \binom{\theta_{il}}{y_i} p_{il}^{y_i}(1-p_{il})^{\theta_{il}-y_i},$$

Poisson

$$f(y_i|\boldsymbol{\theta}_{il}) = \frac{e^{-\theta_{il}}\theta_{il}^{y_i}}{y_i!}$$

or Dirac

$$f(y_i|\boldsymbol{\theta}_{il}) = \begin{cases} 1 & y_i = \theta_{il} \\ 0 & \text{otherwise} \end{cases}$$

parametric family types. The objective of the analysis is the inference about the number c of components, component weights w_l summing to 1 and component parameters $\boldsymbol{\theta}_l$.

The REBMIX algorithm is an iterative numerical procedure relying on the suppositions:

- It is always possible to assign empirical densities to an arbitrary dataset.
- Based on the empirical densities, global mode position can be identified.
- Once the global mode position and its empirical density are known, rough component parameters of the predictive component density can be estimated.
- Based on the rough component parameters, the dataset can be clustered successively into the classes linked to the predictive component densities and the residue.
- The number c of components equals the number of the classes.
- Enhanced component parameters and the component weights can be assessed for all classes.
- The residue can be distributed between the existing components by the Bayes decision rule and the parameters of the finite mixture can be fine-tuned.

Sections 2.1 to 2.7 give the theoretical backgrounds for the algorithm, while Section 2.8 lists and explains its flow.

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 $\bar{\boldsymbol{y}}_j = (\bar{y}_{1j}, \ldots, \bar{y}_{dj})^\top$ are given by

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

the fraction of observations k_j for $j = 1, \ldots, v$ falling into volume V_j is counted out, where $\bar{y}_{i0}$ stands for an arbitrary origin and v depicts the 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 and continuous parametric families

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

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

$$\bar{y}_{i0} = y_{i\min} + \frac{h_i}{2}.$$

However, discrete parametric families require $h_i = 1$ and $\bar{y}_{i0} = y_{i\min}$. The $k - 1$ nearest neighbours are searched around $\boldsymbol{y}_j$ based on the normalized Euclidean distance

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

If $N \geq k$ nearest neighbours coincide, then R is the distance to the nearest non-coincident neighbour multiplied by $(k/(N+1))^{1/d}$.

2.2. Global mode detection

Argument m at which empirical density f_{lj}

$$m = \arg\max_j f_{lj} \tag{4}$$

attains its maximum determines the global mode. If observations are binned into the histogram, then

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

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

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

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

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

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

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

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 positive and negative deviations

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

where r_j stand for the residual frequencies. If index v is replaced by n the equation can be used with the Parzen window and k-nearest neighbour approach, too. Total of 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 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 equivalence of component conditional empirical densities (Nagode and Fajdiga 2006) at $\hat{\boldsymbol{y}}_m = \bar{\boldsymbol{y}}_m$ for the histogram or at $\hat{\boldsymbol{y}}_m = \boldsymbol{y}_m$ for the Parzen window and k-nearest neighbour results in

$$\varepsilon f_{i|\hat{i}.lm} = f(y_i = \hat{y}_{im}|\boldsymbol{\theta}_{il}) = f_{i|\hat{i}.l\mathrm{max}},\ i = 1, \ldots, d. \tag{13}$$

Restraint (13) is sufficient for single parameter component densities, such as for Dirac and exponential. Allowing for the independence of components (2) it yields

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

where

$$\varepsilon = \min\left\{1, \left(\frac{f_{lm}}{\prod_{i=1}^{d} f_{i|\hat{i}.lm}}\right)^{\frac{1}{d}}\right\}. \tag{14}$$

On the other hand, for Rayleigh, Poisson or binomial distribution with known θ_{il} it is assumed

$$\frac{\partial f(y_i = \hat{y}_{im}|\boldsymbol{\theta}_{il})}{\partial y_i} = 0,\ i = 1, \ldots, d. \tag{15}$$

The rough component parameters for single parameter distributions are thus gained from (13) or (15). For two parameter normal, lognormal, Weibull or gamma distribution Lagrange multiplier

$$\Lambda(\boldsymbol{\theta}_{il}, \lambda_{il}) = -\int_{-\infty}^{+\infty} f(y_i|\boldsymbol{\theta}_{il}) \log(f(y_i|\boldsymbol{\theta}_{il}))dy_i + \lambda_{il} \log(f(y_i = \hat{y}_{im}|\boldsymbol{\theta}_{il})/f_{i|\hat{i}.l\mathrm{max}}) \tag{16}$$

provides a strategy for entropy maximization subject to logarithm of (13). The rough component parameters for two parameter distributions are then a solution of

$$\nabla_{\boldsymbol{\theta}_{il},\lambda_{il}} \Lambda(\boldsymbol{\theta}_{il}, \lambda_{il}) = 0,\ i = 1, \ldots, d. \tag{17}$$

Constrained entropy (16) maximization enables rough Weibull and gamma parameter estimation for shape parameter $\beta_{il} > 0$ and not only for $\beta_{il} > 1$ as in Nagode and Fajdiga (2011b,a). Rough normal component parameters are given by

$$\mu_{il} = \hat{y}_{im} \text{ and } \sigma_{il} = \frac{1}{\sqrt{2\pi} f_{i|\hat{i}.l\max}}. \tag{18}$$

Similarly, rough lognormal

$$f(\lambda_{il}) = \frac{\lambda_{il}-1}{\lambda_{il}} + \log(\lambda_{il}(\lambda_{il}-1)) + 2\log(\sqrt{2\pi} f_{i|\hat{i}.l\max}\hat{y}_{im}) = 0,$$
$$\mu_{il} = \lambda_{il} - 1 + \log(\hat{y}_{im}) \text{ and } \sigma_{il} = \sqrt{\lambda_{il}(\lambda_{il}-1)}, \tag{19}$$

Weibull

$$f(\alpha_{il}) = \frac{\alpha_{il}-1}{\lambda_{il}} e^{\frac{1}{\alpha_{il}}} - f_{i|\hat{i}.l\max}\hat{y}_{im} e = 0,\ \lambda_{il} = \frac{\alpha_{il}}{\beta_{il}},$$
$$\beta_{il} = \alpha_{il} + \gamma + \log\left(\frac{\alpha_{il}-1}{\alpha_{il}}\right),\ \theta_{il} = \hat{y}_{im}\left(\frac{\alpha_{il}}{\alpha_{il}-1}\right)^{\frac{1}{\beta_{il}}} \text{ and } \beta_{il} > 0, \tag{20}$$

gamma

$$f(\alpha_{il}) = \frac{1}{2}\log(\beta_{il}) + \beta_{il}\left(\log\left(\frac{\alpha_{il}-1}{\alpha_{il}}\right) + \frac{1}{\alpha_{il}}\right) - \log(\sqrt{2\pi} f_{i|\hat{i}.l\max}\hat{y}_{im}) = 0,$$
$$\beta_{il} = \frac{\gamma(1+\alpha_{il})}{\gamma - 1 - \alpha_{il}\log\left(\frac{\alpha_{il}-1}{\alpha_{il}}\right)},\ \lambda_{il} = \frac{\alpha_{il}}{\beta_{il}},\ \theta_{il} = \frac{\hat{y}_{im}\lambda_{il}}{\alpha_{il}-1} \text{ and } \beta_{il} > 0, \tag{21}$$

binomial

$$p_{il} = \begin{cases} 1 - f_{i|\hat{i}.l\max}^{1/\theta_{il}} & \hat{y}_{im} = 0 \\ f_{i|\hat{i}.l\max}^{1/\theta_{il}} & \hat{y}_{im} = \theta_{il} \\ \hat{y}_{im}/\theta_{il} & \text{otherwise,} \end{cases} \tag{22}$$

rough Poisson

$$\theta_{il} = \begin{cases} -\log(f_{i|\hat{i}.l\max}) & \hat{y}_{im} = 0 \\ \hat{y}_{im} & \text{otherwise} \end{cases} \tag{23}$$

and rough Dirac

$$\theta_{il} = \hat{y}_{im} \tag{24}$$

component parameters are derived, where γ is the Euler-Mascheroni constant. When deriving (21) $\Gamma[\beta_{il}]$ is approximated by the Stirling's formula and digamma function by $\psi(\beta_{il}) = \log(\beta_{il}) - \gamma/\beta_{il}$. Rough binomial parameter $\theta_{il} = \theta_i$ is fixed and equals the number of categories minus one. The rigid restraints become loose if $\hat{y}_{im}$ and $f_{i|\hat{i}.l\max}$ of (18) to (24) are supposed to be bounded by

$$\hat{y}_{im} - ah_{im} \le \hat{y}_{im} \le \hat{y}_{im} + ah_{im} \text{ and } f_{i|\hat{i}.l\min} \le f_{i|\hat{i}.lm} \le f_{i|\hat{i}.l\max}. \tag{25}$$

Constant a is one for the histogram approach, except for the distributions with $y_i \ge 0$ and $\hat{y}_{im} < h_{im}$, where $a = \hat{y}_{im}/h_{im}$. For the Parzen window and k-nearest neighbour $a = \hat{y}_{im}/2h_{im}$

for the distributions with $y_i \geq 0$ and $\hat{y}_{im} < h_{im}/2$, 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 $\hat{y}_{im}$ and $f_{i|\hat{i}.lm}$ are obtained by minimizing the maximum relative positive deviation

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

as explained thoroughly by Nagode and Fajdiga (2011b,a). The loose restraints do no affect the Dirac parameter as here $f_{i|\hat{i}.l\min} = f_{i|\hat{i}.l\max}$. 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

$$\mu_{il} = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\hat{y}_{ij} \text{ and } \sigma_{il}^2 = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\hat{y}_{ij}^2 - \mu_{il}^2. \tag{26}$$

Likewise, enhanced lognormal

$$\mu_{il} = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\log(\hat{y}_{ij}) \text{ and } \sigma_{il}^2 = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\log(\hat{y}_{ij})^2 - \mu_{il}^2, \tag{27}$$

Weibull

$$\theta_{il}^{\beta_{il}} = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\hat{y}_{ij}^{\beta_{il}} \text{ and } f(\beta_{il}) = \frac{1}{\beta_{il}} + \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\log(\hat{y}_{ij}) - \frac{\sum_{j=1}^{v} k_{lj}\hat{y}_{ij}^{\beta_{il}}\log(\hat{y}_{ij})}{\sum_{j=1}^{v} k_{lj}\hat{y}_{ij}^{\beta_{il}}} = 0, \tag{28}$$

gamma

$$\theta_{il} = \frac{1}{\beta_{il} n_l}\sum_{j=1}^{v} k_{lj}\hat{y}_{ij} \text{ and } f(\beta_{il}) = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\log(\hat{y}_{ij}) - \log(\theta_{il}) - \frac{\Gamma'[\beta_{il}]}{\Gamma[\beta_{il}]} = 0, \tag{29}$$

binomial

$$p_{il} = \frac{1}{n_l\theta_{il}}\sum_{j=1}^{v} k_{lj}\hat{y}_{ij}, \tag{30}$$

Poisson

$$\theta_{il} = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\hat{y}_{ij} \tag{31}$$

and Dirac component parameters

$$\theta_{il} = \hat{y}_{im} \tag{32}$$

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

2.6. First and second moment calculation

The first and second moment of the normal

$$m_{il} = \mu_{il} \text{ and } V_{il} = \sigma_{il}^2 + \mu_{il}^2, \tag{33}$$

lognormal

$$m_{il} = e^{\mu_{il} + \frac{\sigma_{il}^2}{2}} \text{ and } V_{il} = e^{2\mu_{il} + 2\sigma_{il}^2}, \tag{34}$$

Weibull

$$m_{il} = \theta_{il}\Gamma\left[1 + \frac{1}{\beta_{il}}\right] \text{ and } V_{il} = \theta_{il}^2\Gamma\left[1 + \frac{2}{\beta_{il}}\right], \tag{35}$$

gamma

$$m_{il} = \theta_{il}\beta_{il} \text{ and } V_{il} = \theta_{il}^2\beta_{il}(1 + \beta_{il}) \tag{36}$$

and the first moment of binomial

$$m_{il} = \theta_{il}p_{il}, \tag{37}$$

Poisson

$$m_{il} = \theta_{il} \tag{38}$$

and Dirac

$$m_{il} = \theta_{il} \tag{39}$$

distributions are calculated to enable the classification of the remaining observations.

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

$$w_l \leq w_{\min} = 2D_{\min}((l-1)b + 1) \tag{40}$$

it is assumed that remaining observations k_{lj} belong to the existing classes and do not form the new ones. Minimum weight multiplier $0 \leq b \leq 1$. The default value is 1. If it decreases, c increases and the predictive mixture density tends to overfit the dataset. Set $b = 0$ for the Dirac parametric family type to obtain an exact fit of the empirical density. The classification of the remaining observations is accomplished by the Bayes decision rule (Duda and Hart 1973)

$$\begin{gathered} l = \arg\max_l w_l f(\boldsymbol{y}_j|\boldsymbol{\theta}_l) \\ w_l = w_l + \frac{k_{lj}}{n},\ m_{il} = m_{il} + \frac{k_{lj}(y_{ij} - m_{il})}{nw_l} \text{ and } V_{il} = V_{il} + \frac{k_{lj}(y_{ij}^2 - V_{il})}{nw_l}, \end{gathered} \tag{41}$$

where k_{lj} is added to the lth class and the component weight and both moments are recalculated (Bishop 1995). Once all v bin means or all n observations are processed, the predictive mixture parameters are gained by inverting (33) to (39).

2.8. Algorithm flow

The REBMIX is listed in Algorithm 1. It requires fifteen arguments, whereby depending on the parametric families five or six of them are mandatory, the rest is optional. It consists of three main loops: the inner 9 $\rightarrow$ 37, the middle 6 $\rightarrow$ 41 and the outer loop 4 $\rightarrow$ 47. The numbers are line indices. In line 2 the observations are preprocessed as described in Section 2.1. In line 3, counter I_1, constant $D_{\min}$ 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 $v \in K$ is completed.

In lines 7 and 8, global mode argument m is detected as explained in Section 2.2, counter I_2 is initiated, component weight w_l is calculated and frequencies r_j are all set to zero. If $I_2 \leq I_{\max}$, the inner loop enters, otherwise in line 38 the first and second moments are calculated (see 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 (see Section 2.4). In the second section 11 $\rightarrow$ 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 (see 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, information criterion, e.g., Akaike (1974)

$$\mathrm{IC} = -2 \log L(c, \boldsymbol{w}, \boldsymbol{\Theta}) + 2M \tag{42}$$

is calculated, whereas the number of free parameters for the normal, lognormal, Weibull and gamma mixtures can be written as

$$M = 2cd + c - 1. \tag{43}$$

The binomial, Poisson and Dirac mixtures require $M = cd + c - 1$. The log likelihood function for the binned observations is given by

$$\log L(c, \boldsymbol{w}, \boldsymbol{\Theta}) = \sum_{j=1}^{v} k_j \log f(\bar{\boldsymbol{y}}_j | c, \boldsymbol{w}, \boldsymbol{\Theta}). \tag{44}$$

Otherwise,

$$\log L(c, \boldsymbol{w}, \boldsymbol{\Theta}) = \sum_{j=1}^{n} \log f(\boldsymbol{y}_j | c, \boldsymbol{w}, \boldsymbol{\Theta}). \tag{45}$$

Finally, total of positive relative deviations for the histogram

$$D = \sum_{j=1}^{v} \left\langle \frac{k_j}{n} - f(\bar{\boldsymbol{y}}_j|c, \boldsymbol{w}, \boldsymbol{\Theta})V_j \right\rangle, \tag{46}$$

Parzen window or k-nearest neighbour

$$D = \sum_{j=1}^{n} \left\langle \frac{1}{n} - \frac{f(\boldsymbol{y}_j|c, \boldsymbol{w}, \boldsymbol{\Theta})V_j}{k_j} \right\rangle \tag{47}$$

is calculated, where $\langle x \rangle = x$ if $x > 0$ and $\langle x \rangle = 0$ if $x \leq 0$. This way global optimum $\mathrm{IC}_{\mathrm{opt}}$ 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, the Stop criterion is redetermined and $D_{\min}$ is decreased in such a way that total of positive relative deviations

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

for c and $c+1$ components is preserved. When line 47 is fulfilled, the procedure stops. If index v 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 four 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)
```

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 moving 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, the `galaxyest` object is initialized and function `REBMIX` is called for normal, lognormal and Weibull parametric families. Maximum number of components $c_{\max}$ is set to 8. The influence of the Akaike (Akaike 1974) information criterion AIC and the Bayesian (Schwarz 1978) information criterion BIC for the histogram and Parzen window preprocessing on predictive number of components c is studied. The optimal number of classes are searched within broad utmost limits K. Argument b is set to 0 as components with a low probability of occurrence may occur.

[1] Mandatory argument.

Algorithm 1 REBMIX

Require: `Dataset`[1], `Preprocessing`[1], `D`, `cmax`, `Criterion`, `Variables`[1], `pdf`[1], `Theta1`[1], `Theta2`, `K`[1], `ymin`, `ymax`, `b`, `ar` and `Restraints`.

Ensure: `Dataset` contains datasets, `Preprocessing` is one of `"histogram"`, `"Parzen window"` or `"k-nearest neighbour"`, $0 \leq$ `D` ≤ 1, `cmax` $\in \mathbb{N}$, `Criterion` is one of `"AIC"`, `"AIC3"`, `"AIC4"`, `"AICc"`, `"BIC"`, `"CAIC"`, `"HQC"`, `"MDL2"`, `"MDL5"`, `"AWE"`, `"CLC"`, `"ICL"`, `"PC"`, `"ICL-BIC"`, `"D"` or `"SSE"`, `Variables` are `"continuous"` or `"discrete"`, `pdf` is one of `"normal"`, `"lognormal"`, `"Weibull"`, `"gamma"`, `"binomial"`, `"Poisson"` or `"Dirac"`, `Theta1` may contain initial binomial parameters, `Theta2` is inactive, `K` $\subset \mathbb{N}$, `ymin` and `ymax` may contain minimum and maximum observations, $0 \leq$ `b` ≤ 1, $0 <$ `ar` ≤ 1 and `Restraints` are `"loose"` or `"rigid"`.

for all v such that $v \in$ `K` **do**
  Preprocessing of observations
  $I_1 \leftarrow 1$, $D_{\min} \leftarrow 0.25$, $k_{lj} \leftarrow k_j$ for $j = 1$ to v
  repeat
    $l \leftarrow 1$, $r \leftarrow n$, $n_l \leftarrow n$
    while $n_l/n > 2D_{\min}((l-1)\mathtt{b}+1)$ **do**
      Global mode detection
      $I_2 \leftarrow 1$, $w_l \leftarrow n_l/n$, $r_j \leftarrow 0$ for $j = 1$ to v
      while $I_2 \leq I_{\max}$ **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** v **do**
          $e_{lj} \leftarrow 0$, $\varepsilon_{lj} \leftarrow 0$
          if $k_{lj} > 0$ **or** $r_j > 0$ **then**
            $e_{lj} \leftarrow k_{lj} - n_l f(\bar{\boldsymbol{y}}_j|\boldsymbol{\theta}_l)V_j$
            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
        $D_l \leftarrow e_{l\mathrm{p}}/n_l$, $\varepsilon_{l\max} \leftarrow \varepsilon_{l\max}(1 - \mathtt{ar})$
        if $D_l > D_{\min}/w_l$ **then**
          for all j such that $1 \leq j \leq v$ **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 $1 \leq j \leq v$ **and** $e_{lj} < 0$ **do**
            $e_{lj} \leftarrow f e_{lj}$, $k_{lj} \leftarrow k_{lj} - e_{lj}$, $r_j \leftarrow r_j + e_{lj}$, $n_l \leftarrow n_l - e_{lj}$
          end for
          $w_l \leftarrow n_l/n$
        else
          Enhanced component parameter estimation, **break**
        end if
        $I_2 \leftarrow I_2 + 1$
      end while
      First and second moment 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 v
      Stop $\leftarrow c \geq v$ **or** $c \geq$ `cmax`, **break** if Stop = **true**
    end while
    Bayes classification of the remaining observations, log likelihood $\log L$, information criterion IC and total of positive relative deviations D 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
    Stop $\leftarrow$ Stop **or** $D \leq$ `D` **or** $I_1 \geq I_{\max}$, $D_{\min} \leftarrow cD_{\min}/(c+1)$, $I_1 \leftarrow I_1 + 1$
  until Stop = **true**
end for

```
R> data("galaxy")
R> galaxyest <- list(normal = NULL, lognormal = NULL, Weibull = NULL)
R> pdf <- c("normal", "lognormal", "Weibull")
R> for (i in 1:3) {
+      galaxyest[[i]] <- REBMIX(Dataset = list(galaxy = galaxy),
+          Preprocessing = c("histogram", "Parzen window"),
+          cmax = 8, Criterion = c("AIC", "BIC"), Variables = "continuous",
+          pdf = pdf[i], K = 7:20, b = 0)
+  }
```

See `help("REBMIX")` in **rebmix** for details about specifying arguments for the function returning an object of class `REBMIX`. List of data frames `w` contains component weights w_l summing to 1, `Theta` stands for a list of data frames containing parametric family types `pdfi`. One of `"normal"`, `"lognormal"`, `"Weibull"`, `"gamma"`, `"binomial"`, `"Poisson"` or `"Dirac"`. Component parameters `theta1.i` follow the parametric family types. One of μ_{il} for normal and lognormal distributions and θ_{il} for Weibull, gamma, binomial, Poisson and Dirac distributions. Component parameters `theta2.i` follow `theta1.i`. One of σ_{il} for normal and lognormal distributions, β_{il} for Weibull and gamma distributions and p_{il} for binomial distribution. Character vector `Variables` contains types of variables. One of `"continuous"` or `"discrete"`. In the `summary` data frame additional information about dataset, preprocessing, D, $c_{\max}$, information criterion type, b, a_{r}, restraints type, optimal c, optimal k, $\bar{y}_{i0}$, optimal h_i, information criterion IC and log likelihood $\log L$ is stored. Position `pos` in the `summary` data frame at which log likelihood $\log L$ attains its maximum is available, too.

The summary output for normal, lognormal and Weibull mixture may be obtained using the `summary` method.

```
R> summary(galaxyest$normal)

  Dataset Preprocessing Criterion c v/k IC  logL M
1 galaxy  histogram     AIC       5 15  423 -197 14
2 galaxy  histogram     BIC       3 19  442 -204 8
3 galaxy  Parzen window AIC       4 16  430 -204 11
4 galaxy  Parzen window BIC       4 16  456 -204 11
Maximum logL = -197 at pos = 1.

R> summary(galaxyest$lognormal)

  Dataset Preprocessing Criterion c v/k IC  logL M
1 galaxy  histogram     AIC       5 15  424 -198 14
2 galaxy  histogram     BIC       3 15  450 -207 8
3 galaxy  Parzen window AIC       4 16  428 -203 11
4 galaxy  Parzen window BIC       4 16  455 -203 11
Maximum logL = -198 at pos = 1.

R> summary(galaxyest$Weibull)
```

```
  Dataset Preprocessing Criterion c v/k IC  logL M
1 galaxy  histogram     AIC       6 18  427 -196 17
2 galaxy  histogram     BIC       4 17  460 -206 11
3 galaxy  Parzen window AIC       8 10  457 -206 23
4 galaxy  Parzen window BIC       2 7   482 -230 5
Maximum logL = -196 at pos = 1.
```

The minimum information criterion and the maximum log likelihood are observed for the histogram preprocessing, whereas the maximum log likelihood resulting in 6 components coincides with the Weibull parametric family type, the histogram preprocessing and the AIC. Most frequently four components appear. The **rebmix** leads thus to the number of components similar to Stephens (2000). The two spurious components reported about by McLachlan and Peel (1997) can not be identified by the algorithm. For the particular dataset the AIC is favorable. It gives 4 to 6 components.

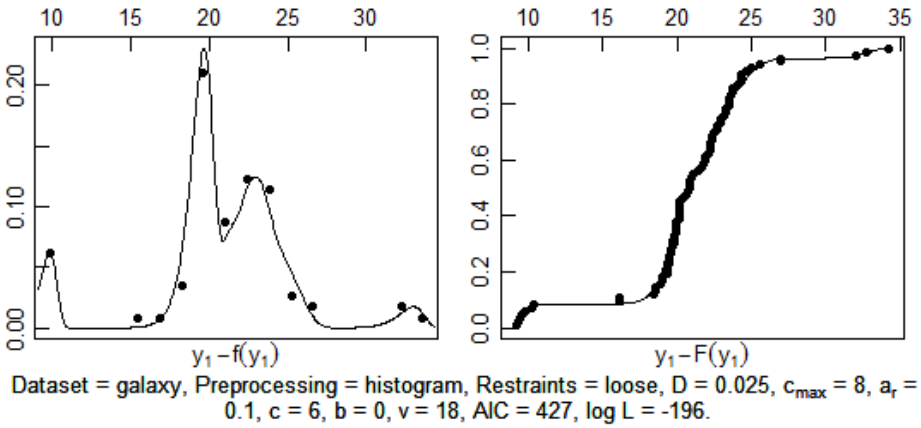

Figure 1: Galaxy dataset. Empirical density and distribution function (circles) and predictive Weibull mixture density and distribution function (solid line).

```
R> plot(galaxyest$Weibull, pos = 1, what = c("den", "dis"),
+       ncol = 2, npts = 1000)
```

The `plot` method delivers a fitted finite mixture with the legend in Figure 1. The corresponding predictive Weibull mixture parameters are given by the `coef` method.

```
R> coef(galaxyest$Weibull, pos = 1)
```

```
       comp1   comp2   comp3   comp4   comp5   comp6
w      0.367   0.259   0.228   0.0854  0.0372  0.0237
pdf    Weibull Weibull Weibull Weibull Weibull Weibull
theta1 19.7    22.7    24.4    9.9     33      21
theta2 30.6    20.4    16.5    19.3    42.2    41
```

For the details about specifying arguments for the `plot` and `coef` methods see `help("plot.REBMIX")` and `help("coef.REBMIX")`, respectively.

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 and the `Species` column is removed.

```
R> data("iris")
R> iris <- iris[, !(colnames(iris) %in% "Species")]
```

The three preprocessing types and six selection criteria AIC, AWE (Banfield and Raftery 1993), BIC, classification likelihood criterion CLC (Biernacki and Govaert 1997), 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 multivariate normal parametric family type are compared. The optimal number of classes and nearest neighbours are searched within broad utmost limits `K`.

```
R> irisest <- REBMIX(Dataset = list(iris = iris), Preprocessing = c("histogram",
+      "Parzen window", "k-nearest neighbour"), Criterion = c("AIC",
+      "AWE", "BIC", "CLC", "ICL", "ICL-BIC"), Variables = rep("continuous",
+      4), pdf = rep("normal", 4), K = list(6:25, 6:25,
+      3:13))
```

The number of components is assessed for the set. The results of the analysis are summarized below.

```
R> summary(irisest)
```

```
   Dataset Preprocessing       Criterion c  v/k IC   logL M
1  iris    histogram           AIC       13 25  517  -143 116
2  iris    histogram           AWE       3  19  992  -317 26
3  iris    histogram           BIC       5  17  749  -264 44
4  iris    histogram           CLC       15 25  316  -137 134
5  iris    histogram           ICL       5  17  772  -264 44
6  iris    histogram           ICL-BIC   5  17  774  -264 44
7  iris    Parzen window       AIC       13 19  563  -166 116
8  iris    Parzen window       AWE       3  22  997  -323 26
9  iris    Parzen window       BIC       5  12  769  -274 44
10 iris    Parzen window       CLC       14 25  357  -159 125
11 iris    Parzen window       ICL       3  22  788  -323 26
12 iris    Parzen window       ICL-BIC   3  22  788  -323 26
13 iris    k-nearest neighbour AIC       5  3   661  -286 44
14 iris    k-nearest neighbour AWE       3  3   1106 -376 26
15 iris    k-nearest neighbour BIC       5  3   793  -286 44
16 iris    k-nearest neighbour CLC       5  3   616  -287 44
17 iris    k-nearest neighbour ICL       5  3   834  -287 44
18 iris    k-nearest neighbour ICL-BIC   5  3   837  -287 44
Maximum logL = -137 at pos = 4.
```

It can be concluded that AIC and CLC overestimate the number of components for the histogram and Parzen window significantly. Hence, maximum $\log L$ does not necessarily lead to accurate predictions. In all other cases either 3 or 5 components are predicted, 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 `plot` method delivers Figure 2.

```
R> plot(irisest, pos = 5, what = c("den", "IC", "logL",
+       "D"), nrow = 3, ncol = 3, npts = 1000)
```

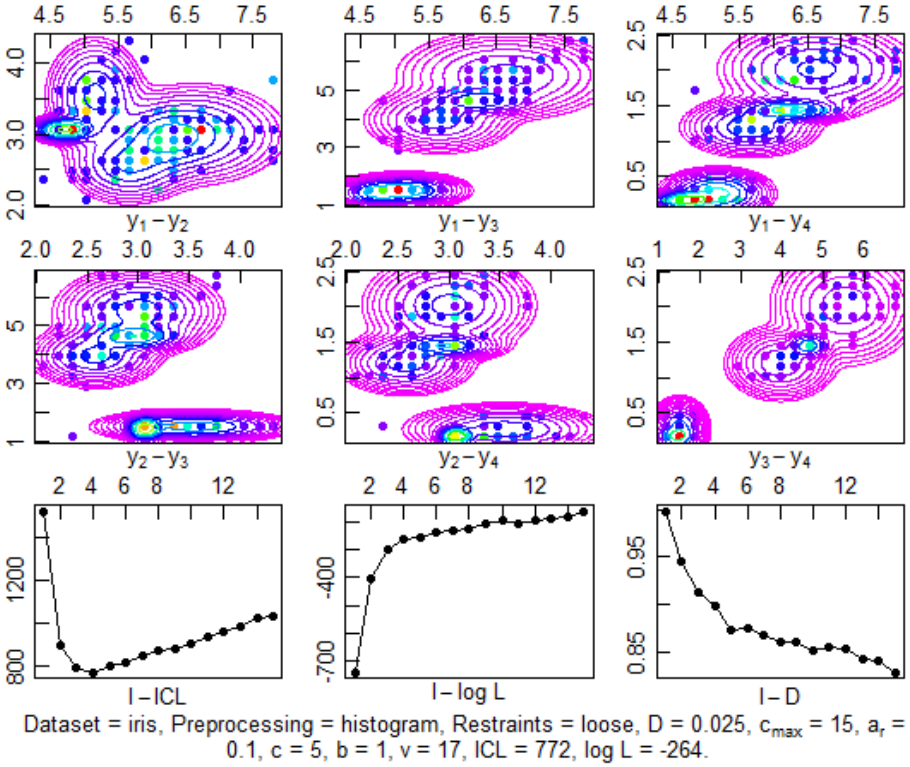

Figure 2: Iris dataset. Empirical densities (circles), predictive multivariate marginal normal mixture densities (contour lines) and progress charts.

3.3. Wine dataset

Next, the results of a wine recognition problem are considered. The set consists of 178 13 dimensional exemplars that are a set of chemical analysis of three types of wine (Asuncion and Newman 2007).

The AIC and CLC overestimate the number of components and are thus not applicable. The AWE, BIC, ICL and ICL-BIC recognize three components 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).

```
R> data("wine")
R> wine <- wine[, !(colnames(wine) %in% "Cultivar")]
R> wineest <- REBMIX(Dataset = list(wine = wine), Preprocessing = c("histogram",
+      "Parzen window"), Criterion = c("AIC", "AWE", "BIC",
+      "CLC", "ICL", "ICL-BIC"), Variables = rep("continuous",
+      13), pdf = rep("normal", 13), K = 8:27)

R> summary(wineest)

   Dataset Preprocessing Criterion c  v/k IC   logL  M
1  wine    histogram     AIC       15 8   6845 -3019 404
2  wine    histogram     AWE       3  9   8277 -3589 80
3  wine    histogram     BIC       3  9   7593 -3589 80
4  wine    histogram     CLC       15 8   6090 -3019 404
5  wine    histogram     ICL       3  9   7622 -3589 80
6  wine    histogram     ICL-BIC   3  9   7622 -3589 80
7  wine    Parzen window AIC       15 8   6594 -2893 404
8  wine    Parzen window AWE       3  17  8368 -3629 80
9  wine    Parzen window BIC       3  17  7673 -3629 80
10 wine    Parzen window CLC       15 8   5839 -2893 404
11 wine    Parzen window ICL       3  17  7714 -3629 80
12 wine    Parzen window ICL-BIC   3  17  7714 -3629 80
Maximum logL = -2893 at pos =  7.
Maximum logL = -2893 at pos = 10.
```

3.4. Complex 1 dataset

Next, a 15 component univariate normal mixture is generated by calling the `RNGMIX` function. It demands character vector `Dataset` containing list names of data frames that datasets are written in, random seed `rseed`, vector `n` containing number of observations in classes n_l and a matrix containing c parametric family types `pdfi`. One of `"normal"`, `"lognormal"`, `"Weibull"`, `"gamma"`, `"binomial"`, `"Poisson"` or `"Dirac"`. Component parameters `theta1.i` follow the parametric family types. One of μ_{il} for normal and lognormal distributions and θ_{il} for Weibull, gamma, binomial, Poisson and Dirac distributions. Component parameters `theta2.i` follow `theta1.i`. One of σ_{il} for normal and lognormal distributions, β_{il} for Weibull and gamma distributions and p_{il} for binomial distribution.

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

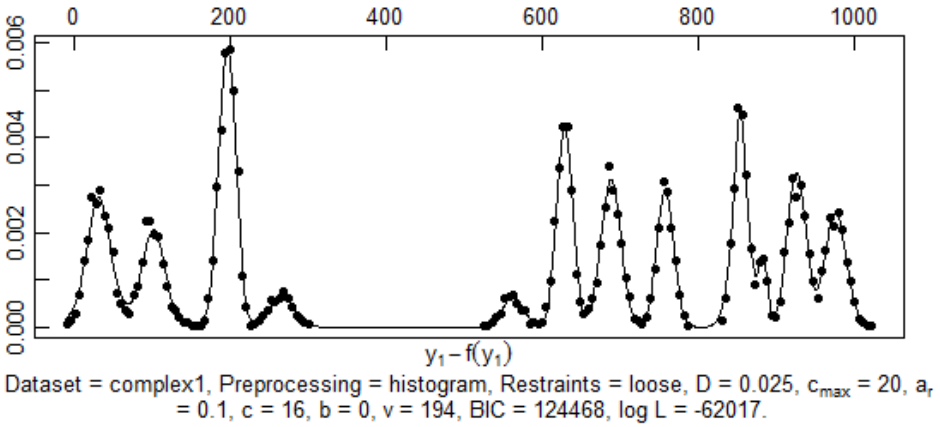

Figure 3: Complex 1 dataset. Empirical density (circles) and predictive **rebmix** normal mixture density in black solid line.

The `complex1$Dataset` holds a list of data frames of size $n \times d$. See `help("RNGMIX")` in **rebmix** for details. The preprocessing is set to histogram, maximum number of components to 20 and information criterion to BIC. The number of classes ranges from 14 (Sturges 1926) to 200 corresponding to the RootN rule and function `REBMIX` is called for the normal parametric family type.

```
R> time <- system.time(complex1est1 <- REBMIX(Dataset = complex1$Dataset,
+      Preprocessing = "histogram", cmax = 20, Criterion = "BIC",
+      Variables = "continuous", pdf = "normal", K = seq(14,
+          200, 4), b = 0))
```

As the true number of components is supposed to be unknown, the predictive normal mixture parameters are estimated for $2 \leq \mathtt{k} \leq 20$ as in the case of the REBMIX algorithm. The `REBMIX` function yields $\log L = -62017$ at 16 components in 4.44 s. To plot the **rebmix** mixture in Figure 3 the `plot` method is called.

```
R> plot(complex1est1, npts = 1000)
```

3.5. Complex 2 dataset

A multivariate mixed continuous-discrete 5 component mixture is generated here by calling the `RNGMIX` function.

```
R> n <- c(390, 110, 300, 70, 130)
R> Theta <- rbind(pdf1 = rep("lognormal", 5), theta1.1 = c(0.8,
+      1.3, 3.4, 2.7, 4.3), theta2.1 = c(0.5, 0.7, 0.2,
+      0.4, 0.1), pdf2 = rep("Poisson", 5), theta1.2 = c(10,
+      7.3, 1.7, 3.3, 5), pdf3 = rep("binomial", 5), theta1.3 = c(10,
+      10, 10, 10, 10), theta2.3 = c(0.9, 0.7, 0.5, 0.3,
+      0.1), pdf4 = rep("Weibull", 5), theta1.4 = c(20,
+      45, 60, 90, 120), theta2.4 = c(2, 3.1, 6.3, 2.5,
```

```
+      7))
R> complex2 <- RNGMIX(Dataset = "complex2", n = n, Theta = Theta)
```

The preprocessing is set to histogram, maximum number of components to 8 and information criterion to BIC. The number of classes ranges from 10 (Sturges 1926) to 64 corresponding to the RootN rule and function `REBMIX` is called for the multivariate lognormal-Poisson-binomial-Weibull parametric family type. Let initial component parameter `Theta1` for the binomial parametric family type be known and be set to 10. The others do not require `Theta1`. Therefore they equal `NA`. Minimum weight multiplier `b` = 0.

```
R> time <- system.time(complex2est <- REBMIX(Dataset = complex2$Dataset,
+      Preprocessing = "histogram", cmax = 8, Criterion = "BIC",
+      Variables = c("continuous", "discrete", "discrete",
+          "continuous"), pdf = c("lognormal", "Poisson",
+          "binomial", "Weibull"), Theta1 = c(NA, NA, 10,
+          NA), K = seq(10, 64, 1), b = 0))
```

The `REBMIX` function yields $\log L = -11754$ at 6 components in 3.42 s. To plot the **rebmix** mixture in Figure 4 the `plot` method is called.

```
R> plot(complex2est, what = c("dens", "marg", "IC", "D"),
+      nrow = 4, ncol = 3, npts = 1000)
```

By calling the `boot.REBMIX` method `B` bootstrap samples of length `n` are generated for the `x` object of class `REBMIX` at position `pos`, where bootstrap `Bootstrap` can be one of default `"parametric"` or `"nonparametric"`. Arguments `replace` and `prob` affect the nonparametric bootstrap only, see `help("sample")` and McLachlan and Peel (1997) for details about replacement and weighted bootstrap.

```
R> complex2boot <- boot.REBMIX(x = complex2est, pos = 1,
+      Bootstrap = "p", B = 100, n = NULL, replace = TRUE,
+      prob = NULL)

R> complex2boot

$c
  [1] 8 7 7 8 7 8 8 8 8 7 7 7 8 8 8 7 6 7 7 8 8 8 6 7 2 8 7 8 8 8 8 8
 [33] 8 7 2 7 8 7 8 8 2 8 8 7 8 8 8 8 8 6 8 6 8 8 5 8 7 8 8 6 7 6 8 5
 [65] 8 8 8 8 7 8 7 6 8 8 6 8 7 8 8 7 8 8 8 6 8 7 7 7 8 8 8 7 7 8 8 8
 [97] 8 5 8 8

$c.mode
[1] 8

$c.prob
[1] 0.59
```

```
$c.se
[1] 1.22

$theta1.1.se
[1] 0.466 0.966 0.995 0.946 1.133 1.166 0.997 0.888

$theta1.2.se
[1] 1.95 4.06 4.01 3.47 3.74 3.55 2.38 2.19

$theta1.3.se
[1] 0 0 0 0 0 0 0 0

$theta1.4.se
[1] 10.1 19.8 27.4 26.4 35.8 38.0 35.4 28.5

$theta2.1.se
[1] 0.257 0.503 0.495 0.387 0.341 0.378 0.316 0.354

$theta2.3.se
[1] 0.105 0.272 0.288 0.256 0.303 0.305 0.222 0.194

$theta2.4.se
[1] 1.35 2.97 3.83 4.99 3.71 2.82 4.49 4.47

$w.se
[1] 0.0559 0.0634 0.0738 0.0684 0.0417 0.0457 0.0479 0.0341

$c.cv
[1] 0.167

$theta1.1.cv
[1] 0.318 0.418 0.392 0.339 0.404 0.364 0.319 0.313

$theta1.2.cv
[1] 0.206 0.607 0.693 0.729 0.626 0.604 0.467 0.478

$theta1.3.cv
[1] 0 0 0 0 0 0 0 0

$theta1.4.cv
[1] 0.412 0.490 0.626 0.471 0.597 0.481 0.430 0.387

$theta2.1.cv
[1] 0.254 0.746 0.813 0.859 0.805 0.929 0.745 0.770

$theta2.3.cv
[1] 0.121 0.409 0.490 0.478 0.552 0.771 0.562 0.419
```

```
$theta2.4.cv
[1] 0.474 0.517 0.610 0.651 0.463 0.380 0.552 0.529

$w.cv
[1] 0.191 0.358 0.590 0.636 0.625 0.559 0.594 0.501

attr(,"class")
[1] "boot.REBMIX"
```

The `complex2boot` object of class `boot.REBMIX` holds a data frame `c` containing numbers c of components for B bootstrap samples, standard error `c.se`, coefficient of variation `c.cv`, mode `c.mode` and mode probability `c.prob` of the numbers of components. Component weights `w`, component parameters `theta1.i` and `theta2.i`, standard errors `w.se`, `theta1.i.se` and `theta2.i.se` and coefficients of variation `w.cv`, `theta1.i.cv` and `theta2.i.cv` for those bootstrap samples for which c equals mode c_{m} are returned, too. See `help("boot.REBMIX")` in **rebmix** for details.

3.6. Simulated 1 dataset

Dataset 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> simulated1 <- RNGMIX(Dataset = paste("Simulated1_", 1:100,
+      sep = ""), n = n, Theta = Theta)
```

In total, 100 finite mixture estimations are performed by calling the `REBMIX` function.

```
R> time <- system.time(simulated1est1 <- REBMIX(simulated1$Dataset,
+      Preprocessing = "histogram", Criterion = "BIC", Variables = rep("continuous",
+          4), pdf = rep("normal", 4), K = seq(10, 28, 2),
+      b = 0))
R> c <- as.numeric(simulated1est1$summary$c)
R> IC <- as.numeric(simulated1est1$summary$IC)
```

The results are as follows:

```
R> summary(c)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   4.00    5.00    5.00    5.44    6.00    9.00

R> summary(IC, digits = 5)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  11646   11825   11890   11889   11953   12154
```

The `REBMIX` function predicts 5.44 components on average in 30.5 s, where probability P of identifying exactly $c = 5$ components equals 0.35. 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.

4. Conclusions and future work

The article presents the REBMIX algorithm and the **rebmix** package. The galaxy, iris, wine, complex 1, complex 2 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 can be used to assess the initial set of the unknown parameters and number of components for, e.g., 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 number of components affects the computational time, but it does not contribute to the numerical instability of the algorithm. Its major superiorities are robustness and time efficiency especially with the histogram preprocessing for all sample sizes. The Parzen window and k-nearest neighbour are more suitable for smaller samples. Its advantages are more stressed for complex mixtures composed of numerous components. The `predict` method that enables class membership prediction is already available in the **rebmix** package and has been validated. See `help("predict.list")` for details. The REBMIX can thus also be used for pattern recognition.

There are several possibilities to further improve the algorithm that have been left for the future. The **rebmix** could be extended to other parametric family types including the multivariate normal ones with full covariance matrices (Nagode *et al.* 2001).

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 G, Chrétien S, Forbes F, Mkhadri A (2001). "A Component-Wise EM Algorithm for Mixtures." *Journal of Computational and Graphical Statistics*, **10**, 697–712.

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.

Reddy CK, Rajaratnam B (2010). "Learning Mixture Models via Component-Wise Parameter Smoothing." *Computational Statistics and Data Analysis*, **54**, 732–749.

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/

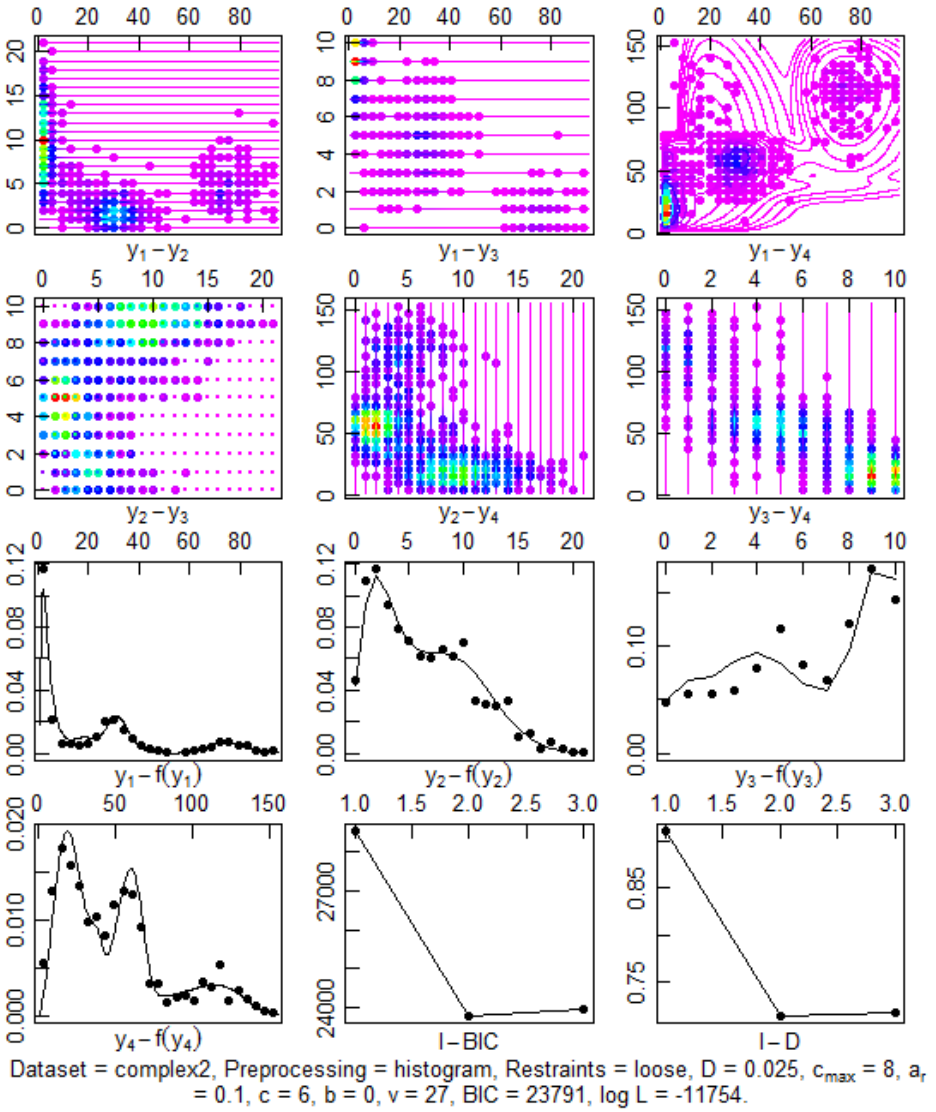

Figure 4: Complex 2 dataset. Empirical densities (coloured large circles), predictive multivariate marginal lognormal-Poisson-binomial-Weibull mixture densities (coloured lines and small circles), empirical densities (circles), predictive univariate marginal lognormal, Poisson, binomial and Weibull mixture densities and progress charts (solid line).