Finite Mixture Modeling via REBMIX

Marko Nagode*[1]

*[1]University of Ljubljana, Faculty of Mechanical Engineering, Aškerčeva 6, 1000 Ljubljana, Slovenia
marko.nagode@fs.uni-lj.si

***Abstract*- The REBMIX algorithm for fitting finite mixture models implemented in R package rebmix is presented. It provides functions for random univariate and multivariate finite mixture generation, the number of components, component weights and component parameter estimation, bootstrapping and the plotting of finite mixtures. It requires preprocessing of observations, information criterion and conditionally independent normal, lognormal, Weibull, gamma, binomial, Poisson or Dirac component densities. The algorithm optimizes the component parameters, mixing weights and number of components successively based on boundary conditions, such as the maximum number of components and number of bins or nearest neighbours. The algorithm is robust, time efficient and can be used either to assess an initial set of unknown parameters and number of components, e.g., for the EM algorithm, or as a standalone algorithm providing a good compromise between parametric and nonparametric methods of finite mixture estimation. Univariate and multivariate datasets are analysed for validation purposes**.

Keywords-*Continuous Variable; Discrete Variable; Mixture Estimation; R Package*

I.INTRODUCTION

Finite mixture models are used increasingly to model the distributions of a wide variety of random phenomena. For multivariate data of a continuous nature, attention is paid to the use of multivariate normal components because of their computational convenience [1, 2, 3]. However, in fatigue and reliability analysis, lognormal and Weibull distributions are preferred due to their flexibility and applicability to continuous positive random variables only [4, 5, 6, 7].

The finite mixture models have seen a real boost in popularity over the last two decades due to a tremendous increase in available computing power. These models can be applied to datasets, where observations originate from categories whose affiliations are not known, or it can provide approximations for multimodal distributions [8]. Some of the latest models can be found in [9, 10, 11, 12, 13, 14, 15, 16, 17].

The REBMIX algorithm addresses the problem of finite mixture estimation numerically. It originates in [18] and is based on the assumption that the problem of finite mixture estimation can be broken into multiple problems of parameter estimation for basic parametric family types. Here the governing equations preexist and originate in the maximum likelihood. REBMIX is a sequential algorithm that avoids the following drawbacks of the simultaneous EM algorithm [19, 20]:

- 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 criterion 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.

Over time REBMIX has evolved [21, 22, 23, 24, 25]. This paper extends it by adding gamma, binomial, Poisson and Dirac parametric families. The reader should expect the derivation of a numerical algorithm, which combines basic statistical techniques in a closed unit and not a concise statistical derivation of the governing equations. The latter does not seem to be feasible.

REBMIX has been compared to the recognizable FlexMix [26] algorithm in [7]. The comparison shows that REBMIX is a robust, time efficient tool that can be used either to assess an initial set of unknown parameters and the number of components for other algorithms [27, 28] or as a standalone procedure that acts as a good compromise between the parametric and nonparametric methods of finite mixture estimation. The **rebmix** implementation of REBMIX [29] extends the set of algorithms available for random univariate and multivariate finite mixture generation, number of components, component weights and component parameter estimation, bootstrapping and plotting of the finite mixtures in the `R` language and environment for statistical computing [30].

II.ALGORITHM

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

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

with conditionally independent component densities

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

indexed by the parameter vector $\mathbf{\theta}_l = (\mathbf{\theta}_{1l}, \ldots, \mathbf{\theta}_{dl})^\top$. The components can currently belong to either normal

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

lognormal

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

Weibull

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

gamma

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

binomial

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

Poisson

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

or Dirac

$$f(y_i|\mathbf{\theta}_{il}) = \left\{ \begin{array}{ll} 1 & y_i = \theta_{il} \\ 0 & \text{otherwise} \end{array} \right.$$

parametric family types. The object of the analysis is to obtain the number of components c, component weights w_l summing to 1 and component parameters $\mathbf{\theta}_l$. The REBMIX algorithm is an iterative numerical procedure which does not rely on rigorous statistical analysis. Instead it makes good use of the suppositions:

- It is always possible to assign empirical densities to an arbitrary dataset.
- Based on the empirical densities, a 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 of components c equals the number of classes.
- Component parameters and the component weights can be enhanced for all classes.
- The unassigned observations can be distributed between the existing components by the Bayes decision rule and the parameters of the finite mixture can be fine-tuned.

The idea is thus to assign the component densities one after another to the empirical mixture density. At the same time the observations belonging to the component densities are extracted from the dataset. By increasing the number of components, the number of observations in the dataset decreases. When the dataset attains a lower limit, it is assumed that the unassigned observations can be distributed between the existing components. Sections A. to G. give the theoretical backgrounds for the algorithm, while Section H. lists and explains its flow.

A.Preprocessing of Observations

The algorithm requires the preprocessing of observations. Using a histogram, the dataset is counted into a finite number of nonoverlapping, equally sized and regularly distributed bins. Assuming that the mean of a bin $\bar{\mathbf{y}}_j = (\bar{y}_{1j}, \ldots, \bar{y}_{dj})^\top$ is 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 the 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 $\mathbf{y}_j$ is obtained. In both cases, the volume is taken to be a hypersquare with sides of length h_{ij}. This yields $V_j = \prod_{i=1}^{d} h_{ij}$. Moreover, for both approaches bin widths $h_{ij} = h_i$ and volumes $V_j = V$ are kept constant. If the k-nearest neighbour is used, the fraction of observations falling into a normalized hypersphere $V_j = \pi^{d/2} R_j^d / \Gamma[1 + d/2]$ of radius R_j centered on observation $\mathbf{y}_j$ contains a constant number $k_j = k$ of observations. The bin 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 and continuous parametric families, the origin is preset to

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

Discrete parametric families require $h_i = 1$ and $\bar{y}_{i0} = y_{i\min}$. The kth $- 1$ nearest neighbour $\mathbf{y}_{\hat{j}}$ is searched around $\mathbf{y}_j$ based on the normalized Euclidean distance

$$R_j = \sqrt{\sum_{i=1}^{d} \left(\frac{y_{i\hat{j}} - y_{ij}}{y_{i\max} - y_{i\min}} \right)^2} \text{ for } \hat{j} \neq j \text{ and } h_{ij} = 2R_j(y_{i\max} - y_{i\min}).$$

If $N \geq k$ nearest neighbours coincide, then the normalized Euclidean distance R_j to the first nearest non-coincident neighbour $\mathbf{y}_{\hat{j}}$ is multiplied by $(k/(N+1))^{1/d}$. Infinite empirical density estimations are thus prevented.

B.Global Mode Detection

The argument m at which the empirical density f_{lj} attains its maximum

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

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 initially set to k_j and the number of observations in class l is

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

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

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

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

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

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

$$f_{i|\hat{\imath}.lm} = \frac{k_{lm}}{\sum_{j=1,\ |y_{\hat{\imath}j} - y_{\hat{\imath}m}| \leq h_{\hat{\imath}m}/2}^{n} k_{lj}} \frac{k_m}{h_{im}} = \frac{k_{lm}}{k_{i|\hat{\imath}.lm}} \frac{k_m}{h_{im}}. \tag{8}$$

C.Clustering of Observations

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

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

For the Parzen window and k-nearest neighbour

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

To identify those observations which deviate the most, relative positive deviations $\varepsilon_{lj} = e_{lj}/k_{lj}$ and a maximum positive relative deviation $\varepsilon_{l\max}$ are calculated. The 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 represents the frequency of unassigned observations. If index v is replaced by n, the equation can also be used with the Parzen window and k-nearest neighbour. The 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 for which the inequality $\varepsilon_{lj} > \varepsilon_{l\max}(1 - a_\mathrm{r})$ holds are not assumed to belong to the lth component and are therefore moved to the residue. The number of iterations depends on the acceleration rate $0 < a_\mathrm{r} \leq 1$, whereby it is best to keep a_r close to zero. The recommended value is 0.1. On the contrary, those observations for which $e_{lj} < 0$ are transferred back to the lth component. The clustering of observations continues with the renewed rough parameter and component weight estimation until

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

The constant $0 < D_{\min} \leq 1$ is optimized by one of the information criteria listed under argument `Criterion` in Fig. 1. The clustering of observations ends with the enhanced component parameter estimation.

D.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 prevents a component from flowing away from the global mode as at least one component is supposed to be in its vicinity. This yields

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

where $\hat{\mathbf{y}}_m = \bar{\mathbf{y}}_m$ for the histogram and $\hat{\mathbf{y}}_m = \mathbf{y}_m$ for the Parzen window and k-nearest neighbour. Restraint (13) is insufficient if $d > 1$ even for single parameter component densities, such as Dirac and exponential parametric families. Allowing for the independence of components in equation (2), equation (13) yields

$$\prod_{i=1}^{d} f(y_i = \hat{y}_{im}|\boldsymbol{\theta}_{il}) = f_{lm} = \prod_{i=1}^{d} \varepsilon f_{i|\hat{\imath}.lm}, \tag{14}$$

from where the required restraints

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

can be derived. In addition, from known f_{lm} and $f_{i|\hat{\imath}.lm}$, it follows that

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

where the upper limit of ε is set to 1. For Rayleigh, Poisson or binomial distributions with known θ_{il} it is assumed

$$\frac{\partial f(y_i = \hat{y}_{im}|\boldsymbol{\theta}_{il})}{\partial y_i} = 0,\ i = 1, \ldots, d. \tag{17}$$

The rough component parameters for single parameter distributions are thus gained from restraints (15) or (17). The restraint chosen should ensure that the predictive component density at $\hat{y}_{im}$ attains its maximum. For two parameter normal, lognormal, Weibull or gamma distributions, the Lagrange multiplier

$$\Lambda(\boldsymbol{\theta}_{il}, \lambda_{il}) = -\int_{-\infty}^{+\infty} f(y_i|\boldsymbol{\theta}_{il}) \log(f(y_i|\boldsymbol{\theta}_{il}))dy_i + \lambda_{il} \log(f(y_i = \hat{y}_{im}|\boldsymbol{\theta}_{il})/f_{i|\hat{\imath}.l\max}) \tag{18}$$

provides a strategy for entropy maximization subject to the logarithm of (15). The rough component parameters for two parameter distributions are then a solution of

$$\nabla_{\boldsymbol{\theta}_{il}, \lambda_{il}} \Lambda(\boldsymbol{\theta}_{il}, \lambda_{il}) = 0, \ i = 1, \ldots, d. \tag{19}$$

Constrained entropy (18) maximization enables rough Weibull and gamma parameter estimation for shape parameter $\beta_{il} > 0$ and not only for $\beta_{il} > 1$ as in [24, 25]. Rough normal component parameters are given by

$$\mu_{il} = \hat{y}_{im} \text{ and } \sigma_{il} = \frac{1}{\sqrt{2\pi} f_{i|\hat{\imath}.l\max}}. \tag{20}$$

Similarly, rough lognormal

$$f(\lambda_{il}) = \frac{\lambda_{il} - 1}{\lambda_{il}} + \log(\lambda_{il}(\lambda_{il} - 1)) + 2\log(\sqrt{2\pi} f_{i|\hat{\imath}.l\max} \hat{y}_{im}) = 0,$$
$$\mu_{il} = \lambda_{il} - 1 + \log(\hat{y}_{im}) \text{ and } \sigma_{il} = \sqrt{\lambda_{il}(\lambda_{il} - 1)}, \tag{21}$$

Weibull

$$f(\alpha_{il}) = \frac{\alpha_{il} - 1}{\lambda_{il}} e^{\frac{1}{\alpha_{il}}} - f_{i|\hat{\imath}.l\max} \hat{y}_{im} e = 0, \ \lambda_{il} = \frac{\alpha_{il}}{\beta_{il}},$$
$$\beta_{il} = \alpha_{il} + \gamma + \log\left(\frac{\alpha_{il} - 1}{\alpha_{il}}\right), \theta_{il} = \hat{y}_{im} \left(\frac{\alpha_{il}}{\alpha_{il} - 1}\right)^{\frac{1}{\beta_{il}}} \text{ and } \beta_{il} > 0, \tag{22}$$

gamma

$$f(\alpha_{il}) = \frac{1}{2}\log(\beta_{il}) + \beta_{il}\left(\log\left(\frac{\alpha_{il} - 1}{\alpha_{il}}\right) + \frac{1}{\alpha_{il}}\right) - \log(\sqrt{2\pi} f_{i|\hat{\imath}.l\max} \hat{y}_{im}) = 0,$$
$$\beta_{il} = \frac{\gamma(1 + \alpha_{il})}{\gamma - 1 - \alpha_{il}\log\left(\frac{\alpha_{il} - 1}{\alpha_{il}}\right)}, \lambda_{il} = \frac{\alpha_{il}}{\beta_{il}}, \theta_{il} = \frac{\hat{y}_{im}\lambda_{il}}{\alpha_{il} - 1} \text{ and } \beta_{il} > 0, \tag{23}$$

binomial

$$p_{il} = \begin{cases} 1 - f_{i|\hat{\imath}.l\max}^{1/\theta_{il}} & \hat{y}_{im} = 0 \\ f_{i|\hat{\imath}.l\max}^{1/\theta_{il}} & \hat{y}_{im} = \theta_{il} \\ \hat{y}_{im}/\theta_{il} & \text{otherwise,} \end{cases} \tag{24}$$

rough Poisson

$$\theta_{il} = \begin{cases} -\log(f_{i|\hat{\imath}.l\max}) & \hat{y}_{im} = 0 \\ \hat{y}_{im} & \text{otherwise} \end{cases} \tag{25}$$

and rough Dirac

$$\theta_{il} = \hat{y}_{im} \tag{26}$$

component parameters are derived, where γ is the Euler-Mascheroni constant. When deriving (23) $\Gamma[\beta_{il}]$ is approximated by the Stirling's formula and digamma function by $\psi(\beta_{il}) = \log(\beta_{il}) - \gamma/\beta_{il}$, the rough binomial parameter $\theta_{il} = \theta_i$ is fixed and equals the number of categories minus one.

Such rigid restraints result in poor component parameter estimation if the modes of several component densities coincide. The loose restraints introduced in [24] improve component parameter estimation and offer further evolution opportunities. The rigid restraints become loose if $f_{i|\hat{\imath}.l\max}$ in equations (20) to (26) is replaced by $f_{i|\hat{\imath}.lm}$, where

$$0 \le f_{i|\hat{\imath}.lm} \le f_{i|\hat{\imath}.l\max}. \tag{27}$$

Instead of minimizing the maximum relative positive deviation [24], the simpler root finding of the total of relative deviations is used here to attain the optimal $f_{i|\hat{\imath}.lm}$. For the histogram total of relative deviations

$$D_{i|\hat{\imath}.lm} = 1 - \sum_{j=1,\ \bar{y}_{\hat{\imath}j} = \bar{y}_{\hat{\imath}m}}^{v} f(y_i = \bar{y}_{ij}|\boldsymbol{\theta}_{il}) h_{ij} \tag{28}$$

equals the fraction of observations falling into regions on the y_i axis with zero empirical probability. If $D_{i|\hat{\imath}.lm}$ is close to zero, e.g., 0.002, then observations not contributing significantly to the lth component should not affect the loose component parameter estimation. This yields

$$\sum_{j=1,\ \bar{y}_{\hat{\imath}j}=\bar{y}_{\hat{\imath}m}}^{v} f(\bar{y}_{ij}|\theta_{il})h_{ij} = 0.998 \tag{29}$$

Equation (29) can be solved for optimal $f_{i|\hat{\imath}.lm}$ by the bisection root finding method. If the root does not exist, then $f_{i|\hat{\imath}.lm} = f_{i|\hat{\imath}.l\max}$. For the Parzen window and k-nearest neighbour the root of

$$\sum_{j=1,\ |y_{\hat{\imath}j}-y_{\hat{\imath}m}|\leq h_{\hat{\imath}m}/2}^{n} f(y_{ij}|\theta_{il})h_{ij}/k_j = 0.998 \tag{30}$$

is searched for optimal $f_{i|\hat{\imath}.lm}$. The Dirac parameter θ_{il} of (26) does not require $f_{i|\hat{\imath}.lm}$ optimization.

E.Enhanced Component Parameter Estimation

Maximum likelihood is employed to obtain enhanced component parameters. For the histogram, enhanced normal component parameters are given by

$$\mu_{il} = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\hat{y}_{ij} \text{ and } \sigma_{il}^2 = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\hat{y}_{ij}^2 - \mu_{il}^2. \tag{31}$$

Likewise, enhanced lognormal

$$\mu_{il} = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\log(\hat{y}_{ij}) \text{ and } \sigma_{il}^2 = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\log(\hat{y}_{ij})^2 - \mu_{il}^2, \tag{32}$$

Weibull

$$\theta_{il}^{\beta_{il}} = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\hat{y}_{ij}^{\beta_{il}} \text{ and } f(\beta_{il}) = \frac{1}{\beta_{il}} + \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\log(\hat{y}_{ij}) - \frac{\sum_{j=1}^{v} k_{lj}\hat{y}_{ij}^{\beta_{il}}\log(\hat{y}_{ij})}{\sum_{j=1}^{v} k_{lj}\hat{y}_{ij}^{\beta_{il}}} = 0, \tag{33}$$

gamma

$$\theta_{il} = \frac{1}{\beta_{il}n_l}\sum_{j=1}^{v} k_{lj}\hat{y}_{ij} \text{ and } f(\beta_{il}) = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\log(\hat{y}_{ij}) - \log(\theta_{il}) - \frac{\Gamma'[\beta_{il}]}{\Gamma[\beta_{il}]} = 0, \tag{34}$$

binomial

$$p_{il} = \frac{1}{n_l\theta_{il}}\sum_{j=1}^{v} k_{lj}\hat{y}_{ij}, \tag{35}$$

Poisson

$$\theta_{il} = \frac{1}{n_l}\sum_{j=1}^{v} k_{lj}\hat{y}_{ij} \tag{36}$$

and Dirac component parameters

$$\theta_{il} = \hat{y}_{im} \tag{37}$$

are estimated. Index v is replaced by n for the Parzen window or k-nearest neighbour.

F.First and Second Moment Calculation

The first and second moment of the normal

$$m_{il} = \mu_{il} \text{ and } V_{il} = \sigma_{il}^2 + \mu_{il}^2, \tag{38}$$

lognormal

$$m_{il} = e^{\mu_{il}+\frac{\sigma_{il}^2}{2}} \text{ and } V_{il} = e^{2\mu_{il}+2\sigma_{il}^2}, \tag{39}$$

Weibull

$$m_{il} = \theta_{il}\Gamma\left[1+\frac{1}{\beta_{il}}\right] \text{ and } V_{il} = \theta_{il}^2\Gamma\left[1+\frac{2}{\beta_{il}}\right], \tag{40}$$

gamma

$$m_{il} = \theta_{il}\beta_{il} \text{ and } V_{il} = \theta_{il}^2\beta_{il}(1 + \beta_{il}) \tag{41}$$

and the first moment of binomial

$$m_{il} = \theta_{il}p_{il}, \tag{42}$$

Poisson

$$m_{il} = \theta_{il} \tag{43}$$

and Dirac

$$m_{il} = \theta_{il} \tag{44}$$

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

G.Bayes Classification of the Unassigned Observations

With an increase in the number of components, the number n_l of unassigned observations decreases

$$\sum_{k=1}^{l-1} w_k + \frac{n_l}{n} = 1. \tag{45}$$

The upper limit $l - 1$ equals the assigned number of components c and l the index of an additional component that has to be checked whether or not it should be assigned to the finite mixture. The total of positive relative deviations D is then given by

$$D = \sum_{k=1}^{l-1} w_k D_k + \frac{n_l}{n} D_l,$$

where n_l/n and D_l stand for the weight of the unassigned observations and the total of positive relative deviations of the unassigned observations, respectively. As $e_{l\mathrm{p}}$ in (11) for any unassigned component equals n_l, D_l equals 1. This yields

$$D = \sum_{k=1}^{l-1} w_k \frac{e_{k\mathrm{p}}}{n_k} + \frac{n_l}{n}. \tag{46}$$

From equations (12) and (46) it follows that

$$D \leq D_{\min}(l - 1) + \frac{n_l}{n}. \tag{47}$$

It is reasonable to stop assigning new components when the weight of the unassigned observations is less than or equal to

$$\frac{n_l}{n} \leq D_{\min}(l - 1). \tag{48}$$

Unassigned observations k_{lj} are then assumed to belong to the existing classes and do not form new ones. The classification of unassigned observations is accomplished by the Bayes decision rule [31]

$$l = \arg\max_l w_l f(\mathbf{y}_j|\boldsymbol{\theta}_l)$$

$$w_l = w_l + \frac{k_{lj}}{n}, m_{il} = m_{il} + \frac{k_{lj}(y_{ij} - m_{il})}{n w_l} \text{ and } V_{il} = V_{il} + \frac{k_{lj}(y_{ij}^2 - V_{il})}{n w_l}, \tag{49}$$

where k_{lj} is added to the lth class and the component weight and both moments are recalculated [32]. Once all v bin means or all n observations are processed, the predictive mixture parameters are gained by inverting equations (38) to (44), inclusive.

H.Algorithm Flow

REBMIX is listed in Fig. 1. It requires fourteen arguments and, depending on the parametric families, five or six of them are mandatory whilst the rest are optional. It consists of four main loops: the inner 9 → 37, the middle 6 → 41, the outer 4 → 47 and the outmost loop 1 → 48. The numbers are line indices.

In line 1 the outmost loop begins. It runs over an ordered set of numbers `K` of bins v for the histogram and the Parzen window or numbers of nearest neighbours k for the k-nearest neighbour. If absolute distance between the number at which IC attains its optimum and the closest two numbers in `K` is greater than 1, set `K` is extended by the golden section search method. In line 2 the

[1] Mandatory argument.

```
Require: Dataset$^1$, Preprocessing$^1$, cmax, Criterion, Variables$^1$, pdf$^1$, Theta1$^1$, Theta2, K$^1$, y0, ymin,
    ymax, ar and Restraints.
Ensure: Dataset contains datasets, Preprocessing is one of "histogram", "Parzen window" or "k-nearest
    neighbour", 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 $\subset \mathbb{N}$, y0 may contain origins, ymin
    and ymax may contain minimum and maximum observations, $0 <$ ar $\leq 1$ and Restraints are "loose" or "rigid".
for all K do
   Preprocessing of observations
   $I_1 \leftarrow 1$, $D_{\min} \leftarrow 0.25$, $k_{lj} \leftarrow k_j$ for $j = 1$ to $v$
   while $I_1 \leq I_{\max}$ do
      $l \leftarrow 1$, $r \leftarrow n$, $n_l \leftarrow n$
      while $n_l/n > D_{\min}(l-1)$ 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
                  $e_{lj} \leftarrow k_{lj} - n_l f(\bar{\mathbf{y}}_j|\boldsymbol{\theta}_l)V_j$
                  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 unassigned observations, log likelihood $\log L$, information criterion IC and total of positive
      relative deviations $D$ calculation
      if IC $<$ IC$_{\mathrm{opt}}$ then
         $\log L \rightarrow \log L_{\mathrm{opt}}$, IC $\rightarrow$ IC$_{\mathrm{opt}}$, $c \rightarrow c_{\mathrm{opt}}$, $\mathbf{w} \rightarrow \mathbf{w}_{\mathrm{opt}}$, $\boldsymbol{\Theta} \rightarrow \boldsymbol{\Theta}_{\mathrm{opt}}$
      end if
      break if Stop $=$ true, $D_{\min} \leftarrow cD_{\min}/(c+1)$, $I_1 \leftarrow I_1 + 1$
   end while
end for
```

Fig. 1 REBMIX algorithm

observations are preprocessed as described in Section A.. In line 3, counter I_1, constant $D_{\min}$ and frequencies k_{lj} are initiated. Next, the outer loop begins. Line 5 assumes the mixture consists of one component, the number of observations r to separate is set to n and n_l to n. If inequality (48) in line 6 is not fulfilled, the middle loop begins.

In lines 7 and 8, the global mode argument m is detected as explained in Section B., counter I_2 is initiated, component weight w_l is calculated and frequencies r_j are set to zero. If $I_2 \leq I_{\max}$, the inner loop begins, otherwise in line 38 the first and the second moments are calculated (see Section F.). In line 39 the number of components c is set to l, the number of observations r is decreased by n_l, l is incremented, the number of the unassigned observations r joins n_l, and the frequencies of the unassigned observations r_j are moved to k_{lj}. If the number of components c in line 40 is greater or equal than v or `cmax`, Stop is set to true and the middle loop ends.

The inner loop is divided into three sections. In line 10 the component parameters are roughly estimated (see Section D.). In the second section 11 $\rightarrow$ 23, deviations e_{lj}, the total of positive deviations $e_{l\mathrm{p}}$, the total of negative deviations $e_{l\mathrm{n}}$, the total of positive relative deviations D_l and maximum relative deviation $\varepsilon_{l\max}$ are calculated. The number of iterations depends on the acceleration rate a_{r}. In the third section 24 $\rightarrow$ 35, the maximum and negative deviations are transferred between frequencies k_{lj} and r_j. This way deviations e_{lj} are reduced gradually. The negative value of e_{lj} can never be higher than 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 E.) and the inner loop ends.

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 unassigned observations are classified by the Bayes decision rule as depicted in Section G.. Further on, the information criterion, e.g., [33]

$$\mathrm{IC} = -2 \log L(c, \mathbf{w}, \mathbf{\Theta}) + 2M \tag{50}$$

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

$$M = 2cd + c - 1. \tag{51}$$

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

$$\log L(c, \mathbf{w}, \mathbf{\Theta}) = \sum_{j=1}^{v} k_j \log f(\bar{\mathbf{y}}_j | c, \mathbf{w}, \mathbf{\Theta}). \tag{52}$$

Otherwise

$$\log L(c, \mathbf{w}, \mathbf{\Theta}) = \sum_{j=1}^{n} \log f(\mathbf{y}_j | c, \mathbf{w}, \mathbf{\Theta}). \tag{53}$$

Finally, the total of positive relative deviations for the histogram

$$D = \sum_{j=1}^{v} \left\langle \frac{k_j}{n} - f(\bar{\mathbf{y}}_j | c, \mathbf{w}, \mathbf{\Theta}) V_j \right\rangle, \tag{54}$$

Parzen window or k-nearest neighbour

$$D = \sum_{j=1}^{n} \left\langle \frac{1}{n} - \frac{f(\mathbf{y}_j | c, \mathbf{w}, \mathbf{\Theta}) V_j}{k_j} \right\rangle \tag{55}$$

is calculated, where $\langle x \rangle = x$ if $x > 0$ and $\langle x \rangle = 0$ if $x \leq 0$. This way the optimum $\mathrm{IC}_{\mathrm{opt}}$ corresponding to the optimal number of components c_{opt}, weights $\mathbf{w}_{\mathrm{opt}}$ and parameters $\mathbf{\Theta}_{\mathrm{opt}}$ is reached. The outer loop ends if the Stop criterion in line 46 is true. Otherwise $D_{\min}$ is decreased in such a way that the total of positive relative deviations

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

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

III.APPLICATION

A comparison between the EM algorithm and the REBMIX algorithm for time efficiency and accuracy has already been studied in [22, 23]. Therefore, the latest results [7] obtained by comparing the EM algorithm implemented in FlexMix [26] to REBMIX are only summarized here. Two cases have been investigated. In case 1, a dataset of size $n = 10000$ generated from the three component multivariate Weibull-normal mixture density has been considered. The FlexMix algorithm attained an AIC $= 223736$ with three components in 149.42s, whereas the REBMIX algorithm resulted in an AIC $= 223999$ with four components in 1.53s. In case 2, the Weibull-normal mixture density has been applied to a real dataset, where the number of components was not known. The FlexMix algorithm has shown some convergence problems and resulted in five components and an AIC $= 562113$ in 1801.52s, whereas the REBMIX algorithm resulted in nine components and an AIC $= 559637$ in 7.09s. The latest study has confirmed our previous experience with the EM algorithm. The REBMIX algorithm is time efficient and robust. The main advantage of the EM algorithm is its accuracy if convergence problems are avoided by a carefully selecting the initial set of unknown parameters. Since the efficiency and accuracy of the REBMIX algorithm have already been dealt with in [22, 23, 7], the rest of this section stresses other capabilities of the REBMIX algorithm by studying univariate and multivariate datasets.

A. Univariate Normal Datasets

As the family of finite normal mixtures is very flexible, it was used by [34] to represent a wide variety of density shapes in their analytical study of the mean integrated squared error of the kernel density estimator. To demonstrate that the family of normal mixtures is very broad, 15 carefully chosen examples were given. The first five represent different types of problems that can arise for unimodal densities. The rest of the densities are multimodal. Densities 6 to 9 are mildly multimodal and might be estimated fairly well with a dataset of a moderate size. The remaining densities are strongly multimodal and are very hard to recover in full with moderate dataset size. However, they are still worth studying because the issue of just how much can be recovered is important.

To find out the extent to which **rebmix** has improved, datasets of sizes 100, 1000 and 10000 are generated for these 15 finite normal mixtures under identical conditions as in [24]. For each dataset size and each density from 1 to 15, 100 datasets are generated. See `help("RNGMIX")`, `help("REBMIX")` and `help("kseq")` in **rebmix** for details.

The `REBMIX` function is applied to the three dataset sizes and the 15 finite normal mixtures. The preprocessing is set to histogram, the maximum number of components to 20 and the information criterion to BIC. The number of bins ranges from `Sturges` [35] to `Log10` corresponding to the Log_{10} rule or to the RootN rule. One or two values appear in Tables 1 and 2. Where two values are shown, they are separated by a slash sign. If the calculations are performed for `K` ranging from `Sturges:Log10`, then the values in the tables are on the left side of the slash sign. If `K` ranges from `Sturges:RootN`, then the values are on the right side of the slash sign. To speed up computation, `K` is set to `kseq(from = Sturges, to = RootN, f = 0.05)` for datasets of size 10000. If there is no slash, `K = Sturges:Log10`. The origin `y0 = 0.0`, minimum `ymin` and maximum `ymax` observations are set to the outmost values of 100 datasets attained by the `RNGMIX` function.

The true numbers of components c from which the datasets are generated for the 15 densities, predictive mean numbers of components μ_c and the corresponding standard deviations σ_c are shown in Table 1 and are compared to Table 2 in [24]. In testing the null hypothesis, that the predictive mean number of components μ_c with the corresponding standard deviation σ_c equals the

TABLE 1 TRUE NUMBER OF COMPONENTS c AND PREDICTIVE MEAN NUMBER OF COMPONENTS μ_c WITH THE CORRESPONDING STANDARD DEVIATION σ_c FOR UNIVARIATE NORMAL DATASETS

			$n = 100$		$n = 1000$		$n = 10000$	
Density		c	μ_c	σ_c	μ_c	σ_c	μ_c	σ_c
1	Gaussian	1	1.0	0.00	1.0	0.00	1.0	0.24
2	Skewed unimodal	3	1.2	0.36	2.1	0.29	4.4/2.4	1.53/0.84
3	Strongly skewed	8	2.5	0.56	7.3	1.09	16.2	1.74
4	Kurtotic unimodal	2	2.0	0.10	2.1	0.22	2.0	0.00
5	Outlier	2	2.2	0.55	2.5	0.50	2.2	0.49
6	Bimodal	2	1.6	0.50	2.0	0.00	2.2	0.66
7	Separated bimodal	2	2.0	0.10	2.0	0.20	2.4/2.0	0.69/0.00
8	Skewed bimodal	2	1.5	0.50	2.2	0.36	4.0	1.23
9	Trimodal	3	1.8	0.42	2.4/2.5	0.48/0.50	4.2/3.0	1.36/0.20
10	Claw	6	1.1	0.27	3.4/5.8	1.64/1.55	9.5	1.33
11	Double claw	9	1.6	0.50	2.0	0.10	2.1	0.62
12	Asymmetric claw	6	1.1	0.27	4.3	1.94	10.1	2.28
13	Asymmetric double claw	8	1.7	0.45	2.0	0.17	5.0	3.08
14	Smooth comb	6	3.2	0.55	5.1	0.96	10.0	2.15
15	Discrete comb	6	3.9	0.32	5.0	0.83	11.0	1.32

true number of components c, one sample t-test

$$t = \frac{\mu_c - c}{\sigma_c / \sqrt{n_c}} \tag{56}$$

is used. Let statistics t calculated from Table 2 [24] be t^{old} and let those from Table 1 be t^{new}. If $t^{\text{new}} < t^{\text{old}}$, an improvement denoted by + in Table 2 is observed. If $t^{\text{new}} = t^{\text{old}}$, notation 0 means unchanged. If $t^{\text{new}} > t^{\text{old}}$, a deterioration denoted by – is observed.

Table 2 shows that for $n = 100$ an improvement of 20% is achieved. For $n = 1000$ an improvement of 27% is achieved. However, for $n = 10000$ a worsening by 7% is observed. This means that for the three dataset sizes a 13% change of the old state is observed. If `RootN` is chosen for maximum `K`, no improvement is observed for $n = 100$ as the value of `K` for $n = 100$ for both Log_{10} and RootN rules is identical. For $n = 1000$ and 10000 an improvement of 53% and 20% is observed, respectively. This means that for the three dataset sizes the old state is improved by 31% on average.

The gaussian density can be recovered by the REBMIX algorithm. When the unimodal density is skewed, μ_c tends towards $c = 3$ for higher n. The strongly skewed density can not be recovered satisfactorily even for $n = 10000$. These kind of densities point out one of the limitations of the REBMIX algorithm which will be difficult to suppress. The kurtotic unimodal, outlier, bimodal and separated bimodal densities can also be recovered however the skewed bimodal density is slightly overfitted for $n = 10000$. The trimodal density can be better recovered for higher n. The claw density consists of one broad density with $w_1 = 1/2$ and five superimposed narrow densities with $w_2 = \ldots = w_6 = 1/10$. It can be recovered for $n = 1000$, however overfitting occurs for high n. This suggests further improvements of the loose restraints are required. The double claw density consists of a bimodal density with $w_1 = w_2 = 49/100$ and seven spikes with $w_3 = \ldots = w_9 = 1/350$. Insensitivity of the REBMIX to these spikes can be treated as beneficial. A similar conclusion holds for the asymmetric double claw density where three narrow densities with component weights of $7/300$ can be recovered completely for the highest n. The asymmetric double claw density can be better recovered for higher n values. Further improvements of the loose restraints could also lead to improvements in this area. The smooth comb and discrete comb densities can be modelled if n is high enough although overfitting occurs. The optimal number of bins should preferably be searched within the broadest limits of `K`. The `rebmix.univariate.normal.R` script file that generates the results is available in the **rebmix** package `demo` subdirectory. To run it enter `demo("rebmix.univariate.normal.R","rebmix")` and press return.

TABLE 2 THREE-STATE LOGIC COMPARISON OF THE t STATISTICS FOR UNIVARIATE NORMAL DATASETS

Density		$n = 100$	$n = 1000$	$n = 10000$
1	Gaussian	0	0	0
2	Skewed unimodal	–	+	–/+
3	Strongly skewed	+	+	–
4	Kurtotic unimodal	+	–	0
5	Outlier	–	–	+
6	Bimodal	+	0	+
7	Separated bimodal	0	0	–/0
8	Skewed bimodal	+	+	+
9	Trimodal	–	–/+	–/0
10	Claw	+	–/+	–
11	Double claw	+	+	+
12	Asymmetric claw	+	+	+
13	Asymmetric double claw	–	+	+
14	Smooth comb	+	+	–
15	Discrete comb	–	+	–
		20%	27%/53%	-7%/20%

B.Galaxy Dataset

The dataset analysed in [36] 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 [36, gives more background]. Richardson and Green [37] concluded from their approach that the number of components ranged from 5 to 7, while [38] provided the support for six components. Stephens [39] reported that three components were optimal for the mixture of normal and four for the mixture of t distributions.

The `REBMIX` function in **rebmix** is entered for normal, lognormal, Weibull and gamma parametric families. The maximum number of components `cmax` is set to 10. The influence of the Akaike [33] information criterion AIC and the Bayesian [40] information criterion BIC for the histogram and Parzen window preprocessing on predictive number of components c is studied. The optimal number of bins is searched within the broadest limits for `K = 7:20`. The minimum information criterion and the maximum log likelihood in Table 3 are observed for the histogram preprocessing. The maximum log likelihood resulting in 6 components coincides with the lognormal and gamma parametric family types and the AIC. Most frequently six and four components appear. For this particular dataset the AIC is favorable giving 4 to 6 components. The `plot` method delivers a fitted finite mixture, complete with legend, in Fig. 2. The corresponding predictive lognormal mixture parameters in Table 4 are given

TABLE 3 PREDICTIVE NUMBER OF COMPONENTS c, NUMBER OF BINS v, INFORMATION CRITERION IC AND LOG LIKELIHOOD $\log L$ FOR GALAXY DATASET

Parametric family	Preprocessing	Criterion	c	v	IC	$\log L$
normal	histogram	AIC	6	20	424	-195
	histogram	BIC	4	19	453	-202
	Parzen window	AIC	6	18	431	-199
	Parzen window	BIC	4	16	459	-205
lognormal	histogram	AIC	6	20	422	-194
	histogram	BIC	3	19	442	-203
	Parzen window	AIC	6	19	436	-201
	Parzen window	BIC	3	8	461	-213
Weibull	histogram	AIC	5	15	428	-200
	histogram	BIC	4	17	458	-205
	Parzen window	AIC	4	11	435	-207
	Parzen window	BIC	4	11	462	-207
gamma	histogram	AIC	6	20	423	-194
	histogram	BIC	3	19	444	-204
	Parzen window	AIC	6	18	432	-199
	Parzen window	BIC	4	14	462	-207

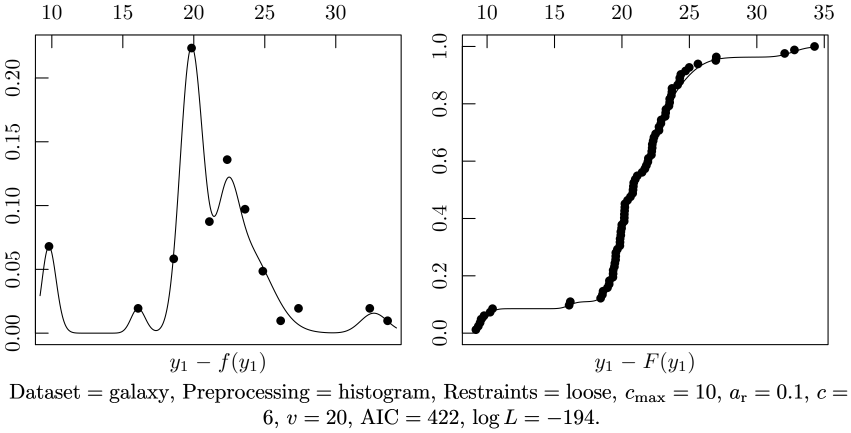

Fig. 2 Empirical density and distribution function (circles) and predictive lognormal mixture density and distribution function (solid line) for galaxy dataset

TABLE 4 COMPONENT WEIGHTS AND LOGNORMAL COMPONENT PARAMETERS FOR GALAXY DATASET

l	1	2	3	4	5	6
w_l	0.456	0.154	0.243	0.0854	0.0372	0.0244
μ_l	2.99	3.11	3.18	2.28	3.49	2.78
σ_l	0.041	0.034	0.0648	0.0511	0.029	0.0312

by the `coef` method. For the details on specifying arguments for the `plot` and `coef` methods see `help("plot.REBMIX")` and `help("coef.REBMIX")`, respectively. The `rebmix.galaxy.R` script file is made available via `demo("rebmix.galaxy","rebmix")`.

C.Mixed Continuous-discrete Dataset

A multivariate mixed continuous-discrete 3 component mixture is generated here by calling the `RNGMIX` function. The `REBMIX` function is called for the multivariate lognormal-Poisson-binomial-Weibull parametric family type. To plot the **rebmix** mixture in Fig. 3 the `plot` method is called. By calling the `boot.REBMIX` function `B` bootstrap datasets of length `n` are generated for the `x` object of class `REBMIX` at position `pos`, where bootstrap `Bootstrap` can be one of the default `"parametric"` or `"nonparametric"` options. Arguments `replace` and `prob` affect the nonparametric bootstrap only, see `help("sample")` and [38] for details regarding replacement and weighted bootstrap.

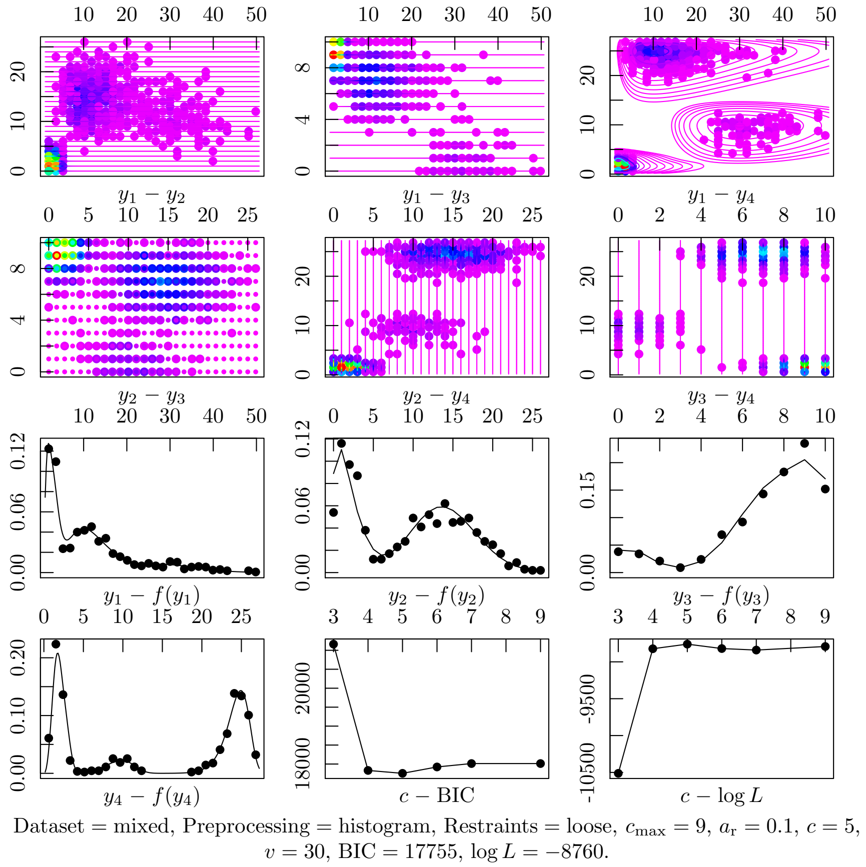

Fig. 3 Empirical densities (large coloured circles), predictive multivariate marginal lognormal-Poisson-binomial-Weibull mixture densities (coloured lines and small circles), empirical densities (circles), predictive univariate marginal lognormal, Poisson, binomial and Weibull mixture densities and progress charts (solid line)

The `mixedboot` object of class `boot.REBMIX` holds a data frame `c = {9, 8, 6, ..., 7, 5, 5, 5, 5}` containing numbers of components c for `B = 100` bootstrap datasets, standard error `c.se = 1.54`, coefficient of variation `c.cv = 0.259`, mode `c.mode = 5` and mode probability `c.prob = 0.31` 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 datasets for which `c = 5` are also returned. See `help("boot.REBMIX")` in **rebmix** for details. The `demo("rebmix.mixed.continuous.discrete", "rebmix")` makes the section reproducible.

IV.CONCLUSIONS

The article presents the REBMIX algorithm and the **rebmix** package. The datasets are studied on the x64 architecture. By applying the **tikzDevice** package [41], plots with legends are obtained. The REBMIX algorithm can be used to assess an initial set of unknown parameters and number of components for, e.g., the EM algorithm or as a standalone procedure that is a good compromise between the parametric and nonparametric methods of finite mixture estimation. Its major advantages are robustness and time efficiency. Its advantages are more apparent if mixtures are composed of a larger number of components. On the other hand, it requires the preprocessing of observations, which is a drawback, especially where smaller datasets are concerned. The restoration of overlapping components is not always adequate either. However, further improvements of the loose restraints seem to be feasible. The **rebmix** package can be broadened to other parametric family types, including those with conditionally dependent component densities, e.g., multivariate normal mixtures with full covariance matrices. Currently multivariate normal mixtures with diagonal covariance matrices are available. In time further improvements will see an extensibility of the REBMIX algorithm whereby users should have the ability to provide their own parametric family types for the rapid prototyping of new finite mixture models. The `RCLSMIX` function that enables class membership prediction is available in the **rebmix** package. See `help("RCLSMIX")` for details. The REBMIX is thus also intended to be used for pattern recognition.

REFERENCES

[1] Geoff McLachlan and David Peel. The emmix algorithm for the fitting of normal and t-components. *Journal of Statistical Software*, 4(2):1–14, 1999.

[2] Salvatore Ingrassiaa and Roberto Roccib. Constrained monotone em algorithms for finite mixture of multivariate gaussians. *Computational Statistics & Data Analysis*, 51(11):5339–5351, 2007.

[3] Sylvia Frühwirth-Schnatter. *Finite Mixture and Markov Switching Models*. Springer-Verlag Series in Statistics, New York, 2006.

[4] Karl D Majeske. A mixture model for automobile warranty data. *Reliability Engineering & System Safety*, 81(1):71–77, 2003.

[5] Khalaf S Sultan, Moshira A Ismail, and Asmaa S Al-Moisheer. Mixture of two inverse weibull distributions: Properties and estimation. *Computational Statistics & Data Analysis*, 51(11):5377–5387, 2007.

[6] Anduin E Touw. Bayesian estimation of mixed weibull distributions. *Reliability Engineering & System Safety*, 94(2):463–473, 2009.

[7] Mitja Franko and Marko Nagode. Probability density function of the equivalent stress amplitude using statistical transformation. *Reliability Engineering & System Safety*, 134:118–125, 2015.

[8] Friedrich Leisch. Flexmix: A general framework for finite mixture models and latent class regression in r. *Journal of Statistical Software*, 11(8):1–18, 2004.

[9] Geoff McLachlan and Sharon X Lee. Emmixuskew: An r package for fitting mixtures of multivariate skew t distributions via the em algorithm. *Journal of Statistical Software*, 55(12):1–22, 2013.

[10] Marie Chavent, Vanessa Kuentz-Simonet, Benot Liquet, and Jrme Saracco. Clustofvar: An r package for the clustering of variables. *Journal of Statistical Software*, 50(13):1–16, 2012.

[11] Bettina Grün, Ioannis Kosmidis, and Achim Zeileis. Extended beta regression in r: Shaken, stirred, mixed, and partitioned. *Journal of Statistical Software*, 48(11):1–25, 2012.

[12] Volodymyr Melnykov, Wei-Chen Chen, and Ranjan Maitra. Mixsim: An r package for simulating data to study performance of clustering algorithms. *Journal of Statistical Software*, 51(12):1–25, 2012.

[13] David Ardia, Lennart F Hoogerheide, and Herman K van Dijk. Adaptive mixture of student-t distributions as a flexible candidate distribution for efficient simulation: The r package admit. *Journal of Statistical Software*, 29(3):1–31, 2009.

[14] Tatiana Benaglia, Didier Chauveau, David R Hunter, and Derek S Young. mixtools: An r package for analyzing finite mixture models. *Journal of Statistical Software*, 32(6):1–29, 2009.

[15] Bettina Grün and Friedrich Leisch. Flexmix version 2: Finite mixtures with concomitant variables and varying and constant parameters. *Journal of Statistical Software*, 28(4):1–35, 2008.

[16] Chris Fraley and Adrian Raftery. Model-based methods of classification: Using the mclust software in chemometrics. *Journal of Statistical Software*, 18(6):1–13, 2007.

[17] Geoff McLachlan and David Peel. *Finite Mixture Models*. John Wiley & Sons, New York, 2000.

[18] Marko Nagode and Matija Fajdiga. A general multi-modal probability density function suitable for the rainflow ranges of stationary random processes. *International Journal of Fatigue*, 20(3):211–223, 1998.

[19] Chandan K Reddya and Bala Rajaratnam. Learning mixture models via component-wise parameter smoothing. *Computational Statistics and Data Analysis*, 54(3):732–749, 2010.

[20] Gilles Celeux, Stphane Chrétien, and Florence Forbes. A component-wise em algorithm for mixtures. *Journal of Computational and Graphical Statistics*, 10(4):697–712, 2001.

[21] Marko Nagode and Matija Fajdiga. An improved algorithm for parameter estimation suitable for mixed weibull distributions. *International Journal of Fatigue*, 22(1):75–80, 2000.

[22] Marko Nagode, Jernej Klemenc, and Matija Fajdiga. Parametric modelling and scatter prediction of rainflow matrices. *International Journal of Fatigue*, 23(6):525–532, 2001.

[23] Marko Nagode and Matija Fajdiga. An alternative perspective on the mixture estimation problem. *Reliability Engineering & System Safety*, 91(4):388–397, 2006.

[24] Marko Nagode and Matija Fajdiga. The rebmix algorithm for the univariate finite mixture estimation. *Communications in Statistics - Theory and Methods*, 40(5):876–892, 2011.

[25] Marko Nagode and Matija Fajdiga. The rebmix algorithm for the multivariate finite mixture estimation. *Communications in Statistics - Theory and Methods*, 40(11):2022–2034, 2011.

[26] Bettina Grün and Friedrich Leisch. *flexmix: Flexible Mixture Modeling*, 2015. R package version 2.3-13.

[27] Mengjie Chen, Murat Gunel, and Hongyu Zhao. Somatica: Identifying, characterizing and quantifying somatic copy number aberrations from cancer genome sequencing data. *PLOS ONE*, 8(11):1–10, 2013.

[28] Tomaž Bučar, Marko Nagode, and Matija Fajdiga. Reliability approximation using finite weibull mixture distributions. *Reliability Engineering & System Safety*, 84(3):241–251, 2004.

[29] Marko Nagode. *rebmix: Rough - Enhanced - Bayesian Finite Mixture Modeling*. A, 2015. R package version 2.7.1.

[30] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015.

[31] Richard O Duda and Peter E Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973.

[32] Christopher M Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

[33] Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(51):716–723, 1974.

[34] J S Marron and M P Wand. Exact mean integrated squared error. *The Annals of Statistics*, 20(2):712–736, 1992.

[35] Herbert A Sturges. The choice of a class interval. *Journal of American Statistical Association*, 21(153):65–66, 1926.

[36] Kathryn Roeder. Density estimation with confidence sets exemplified by superclusters and voids in the galaxies. *Journal of American Statistical Association*, 85(411):617–624, 1990.

[37] Sylvia Richardson and Peter J Green. On bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society B*, 59(4):731–792, 1997.

[38] Geoff McLachlan and David Peel. Contribution to the discussion of paper by s. richardson and p.j. green. *Journal of the Royal Statistical Society B*, 59(4):779–780, 1997.

[39] Matthew Stephens. Bayesian analysis of mixture models with an unknown number of components - an alternative to reversible jump methods. *The Annals of Statistics*, 28(1):40–74, 2000.

[40] Gideon Schwarz. Estimating the dimension of the model. *The Annals of Statistics*, 6(2):461–464, 1978.

[41] Charlie Sharpsteen, Cameron Bracken, Kirill Müller, and Yihui Xie. *tikzDevice: R Graphics Output in LaTeX Format*, 2015. R package version 0.8.1.