rebmix: The Rebmix Package

Marko Nagode

August 13, 2015

Abstract

The **rebmix** package 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.

1 Introduction

To cite the REBMIX algorithm please refer to (Nagode and Fajdiga, 2011a,b; Nagode, 2015). For theoretical backgrounds please upload `http://doi.org/10.5963/JAO0302001`.

2 Examples

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

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

2.1 Gamma datasets

Three gamma mixtures are considered (Wiper et al., 2001). The first has four well-separated components with means 2, 4, 6 and 8, respectively

$$\begin{array}{lll} \theta_1 = 1/100 & \beta_1 = 200 & n_1 = 100 \\ \theta_2 = 1/100 & \beta_2 = 400 & n_2 = 100 \\ \theta_3 = 1/100 & \beta_3 = 600 & n_3 = 100 \\ \theta_4 = 1/100 & \beta_4 = 800 & n_4 = 100. \end{array}$$

The second has equal means but different variances and weights

$$\begin{array}{lll} \theta_1 = 1/27 & \beta_1 = 9 & n_1 = 40 \\ \theta_2 = 1/270 & \beta_2 = 90 & n_2 = 360. \end{array}$$

The third is a mixture of a rather diffuse component with mean 6 and two lower weighted components with smaller variances and means of 2 and 10, respectively

$$\begin{array}{lll} \theta_1 = 1/20 & \beta_1 = 40 & n_1 = 80 \\ \theta_2 = 1 & \beta_2 = 6 & n_2 = 240 \\ \theta_3 = 1/20 & \beta_3 = 200 & n_3 = 80. \end{array}$$

The gamma mixtures are 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(100, 100, 100, 100)
R> Theta <- rbind(pdf = "gamma", theta1 = c(1/100, 1/100, 1/100,
+      1/100), theta2 = c(200, 400, 600, 800))
R> gamma1 <- RNGMIX(Dataset = "gamma1", n = n, Theta = Theta)
R> n <- c(40, 360)
R> Theta <- rbind(pdf = "gamma", theta1 = c(1/27, 1/270), theta2 = c(9,
+      90))
R> gamma2 <- RNGMIX(Dataset = "gamma2", n = n, Theta = Theta)
R> n <- c(80, 240, 80)
R> Theta <- rbind(pdf = "gamma", theta1 = c(1/20, 1, 1/20), theta2 = c(40,
+      6, 200))
R> gamma3 <- RNGMIX(Dataset = "gamma3 ", n = n, Theta = Theta)
```

The `gamma1$Dataset`, `gamma2$Dataset` and `gamma3$Dataset` hold 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 8 and information criterion to AIC or BIC. The number of classes ranges from 30 to 80 and function `REBMIX` is called for the gamma parametric family type.

```
R> gamma1est <- REBMIX(Dataset = gamma1$Dataset, Preprocessing = "histogram",
+      cmax = 8, Criterion = c("AIC", "BIC"), Variables = "continuous",
+      pdf = "gamma", K = 30:80)
R> gamma2est <- REBMIX(Dataset = gamma2$Dataset, Preprocessing = "histogram",
+      cmax = 8, Criterion = "BIC", Variables = "continuous", pdf = "gamma",
+      K = 30:80)
R> gamma3est <- REBMIX(Dataset = gamma3$Dataset, Preprocessing = "histogram",
+      cmax = 8, Criterion = "BIC", Variables = "continuous", pdf = "gamma",
+      K = 30:80)
```

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, 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. See `help("summary.REBMIX")` for details.

```
R> summary(gamma1est)

  Dataset Preprocessing Criterion c v/k IC   logL M
1 gamma1  histogram     AIC       6 59  1011 -488 17
2 gamma1  histogram     BIC       4 54  1067 -501 11
Maximum logL = -488 at pos = 1.
```

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

```
R> plot(gamma2est, pos = 1, what = c("den", "dis"), ncol = 2, npts = 1000)
```

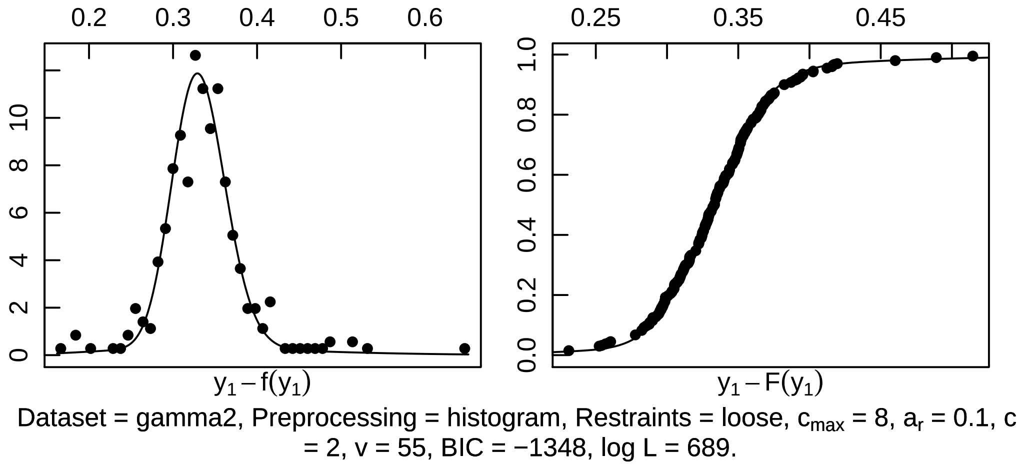

Figure 1: Gamma 2 dataset. Empirical density (circles) and predictive gamma mixture density in black solid line.

```
R> coef(gamma2est)

       comp1   comp2
w      0.911   0.0893
pdf    gamma   gamma
theta1 0.00299 0.0453
theta2 111     8.1
```

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

By calling the boot.REBMIX method 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 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> gamma3boot <- boot.REBMIX(x = gamma3est, pos = 1, Bootstrap = "p",
+       B = 10, n = NULL, replace = TRUE, prob = NULL)

R> gamma3boot

$c
 [1] 3 3 3 3 3 3 3 3 3 3

$c.mode
[1] 3

$c.prob
[1] 1

$c.se
[1] 0
```

```
$theta1.se
[1] 0.0899 0.1236 0.1565

$theta2.se
[1]  78.1 133.4  88.5

$w.se
[1] 0.0408 0.0513 0.0574

$c.cv
[1] 0

$theta1.cv
[1] 0.500 1.033 0.619

$theta2.cv
[1] 2.028 0.797 1.378

$w.cv
[1] 0.118 0.163 0.169

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

The `gamma3boot` object of class `boot.REBMIX` holds a data frame `c` containing numbers c of components for B bootstrap datasets, 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 datasets for which c equals mode c_{m} are returned, too. See `help("boot.REBMIX")` in **rebmix** for details.

```
R> summary(gamma3boot)

           comp1 comp2 comp3
w.cv       0.118 0.163 0.169
theta1.cv 0.5   1.03  0.619
theta2.cv 2.03  0.797 1.38
Mode probability = 1 at c = 3 components.
```

2.2 Poisson dataset

Dataset consists of $n = 600$ two dimensional observations obtained by generating data points separately from each of three Poisson distributions. The component dataset sizes and parameters, which are those studied in Ma et al. (2009), are displayed below

$$\begin{array}{ll} \boldsymbol{\theta}_1 = (3, 2)^\top & n_1 = 200 \\ \boldsymbol{\theta}_2 = (9, 10)^\top & n_2 = 200 \\ \boldsymbol{\theta}_3 = (15, 16)^\top & n_3 = 200 \end{array}$$

For the dataset Ma et al. (2009) conduct 100 experiments by selecting different initial values of the mixing proportions. In all the cases, the adaptive gradient BYY learning algorithm leads to the correct model selection, i.e., finally allocating the correct number of Poissons for the dataset. In the meantime, it also results in an estimate for each parameter in the original or true Poisson mixture which generated the dataset. As the dataset of Ma et al. (2009) can not exactly be reproduced, 100 datasets are generated with random seeds r_{seed} ranging from -1 to -100.

```
R> n <- c(200, 200, 200)
R> Theta <- rbind(rep("Poisson", 3), c(3, 9, 15), rep("Poisson",
+      3), c(2, 10, 16))
R> poisson <- RNGMIX(Dataset = paste("Poisson_", 1:100, sep = ""),
+      n = n, Theta = Theta)
```

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

```
R> poissonest <- REBMIX(Dataset = poisson$Dataset, Preprocessing = "histogram",
+      cmax = 6, Criterion = "MDL5", Variables = rep("discrete",
+          2), pdf = rep("Poisson", 2), K = 1)
R> c <- as.numeric(poissonest$summary$c)
R> IC <- as.numeric(poissonest$summary$IC)
```

The results are as follows:

```
R> summary(c)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2.00    3.00    3.00    2.98    3.00    5.00

R> summary(IC, digits = 5)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   6927    7066    7120    7130    7184    7350
```

The REBMIX function predicts 2.98 components on average, where probability of identifying exactly $c = 3$ components equals 0.65. To plot the mixture in Figure 2 the plot method is called.

3 Summary

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

J. Ma, J. Liu, and Z. Ren. Parameter estimation of poisson mixture with automated model selection through byy harmony learning. *Pattern Recognition*, 42(11):2659–2670, 2009. doi: 10.1016/j.patcog.2009.03.029.

G. McLachlan and D. 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. URL http://www.jstor.org/stable/2985194.

M. Nagode. Finite mixture modeling via rebmix. *Journal of Algorithms and Optimization*, 3(2):14–28, 2015. doi: 10.5963/JAO0302001.

M. Nagode and M. Fajdiga. The rebmix algorithm for the univariate finite mixture estimation. *Communications in Statistics - Theory and Methods*, 40(5):876–892, 2011a. doi: 10.1080/03610920903480890.

M. Nagode and M. Fajdiga. The rebmix algorithm for the multivariate finite mixture estimation. *Communications in Statistics - Theory and Methods*, 40(11):2022–2034, 2011b. doi: 10.1080/03610921003725788.

M. Wiper, D. R. Insua, and F. Ruggeri. Mixtures of gamma distributions with applications. *Journal of Computational and Graphical Statistics*, 10(3):440–454, 2001. URL http://www.jstor.org/stable/1391098.

```
R> plot(poissonest, pos = 58, what = c("dens", "marg", "IC", "D",
+      "logL"), nrow = 2, ncol = 3, npts = 1000)
```

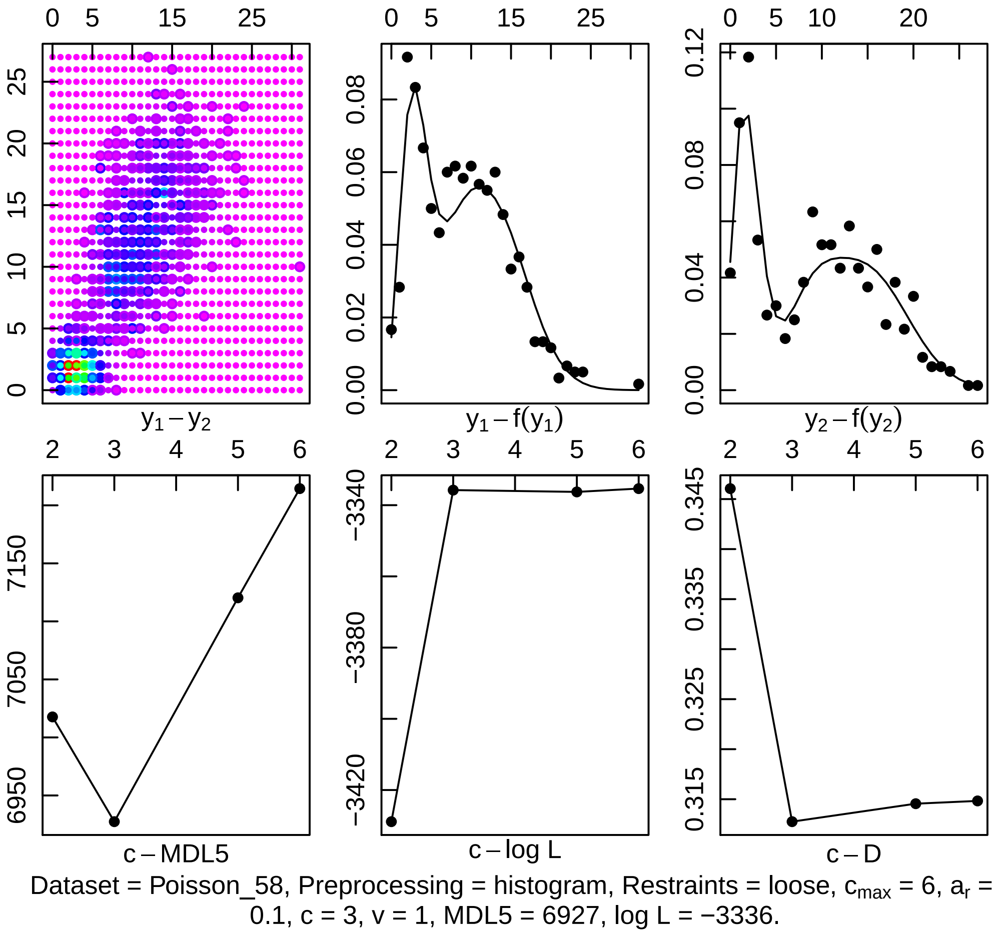

Figure 2: Poisson dataset. Empirical densities (coloured large circles), predictive multivariate Poisson-Poisson mixture density (coloured small circles), empirical densities (circles), predictive univariate marginal Poisson mixture densities and progress charts (solid line).

Marko Nagode
University of Ljubljana
Faculty of Mechanical Engineering
Aškerčeva 6
1000 Ljubljana
Slovenia
Marko.Nagode@fs.uni-lj.si.