Rmixmod: A MIXture MODelling R package

Rémi Lebret

Rmixmod: A MIXture MODelling R package

R. Lebret[a,1] and **S. Iovleff**[a,2] and **F. Langrognet**[b]

[a]Laboratoire de mathématiques Paul Painlevé
U.M.R. 8524 - CNRS - Université Lille 1 - INRIA Lille Nord-Europe - MODAL Team
Cité Scientifique - 59655 Villeneuve d'Ascq Cedex - FRANCE
[1] remi.lebret@math.univ-lille1.fr
[2] serge.iovleff@math.univ-lille1.fr

[b]Laboratoire de mathématiques de Besançon
U.M.R. 6623 - CNRS - Université de Franche-Comté
16 route de Gray - 25030 Besançon - FRANCE
florent.langrognet@univ-fcomte.fr

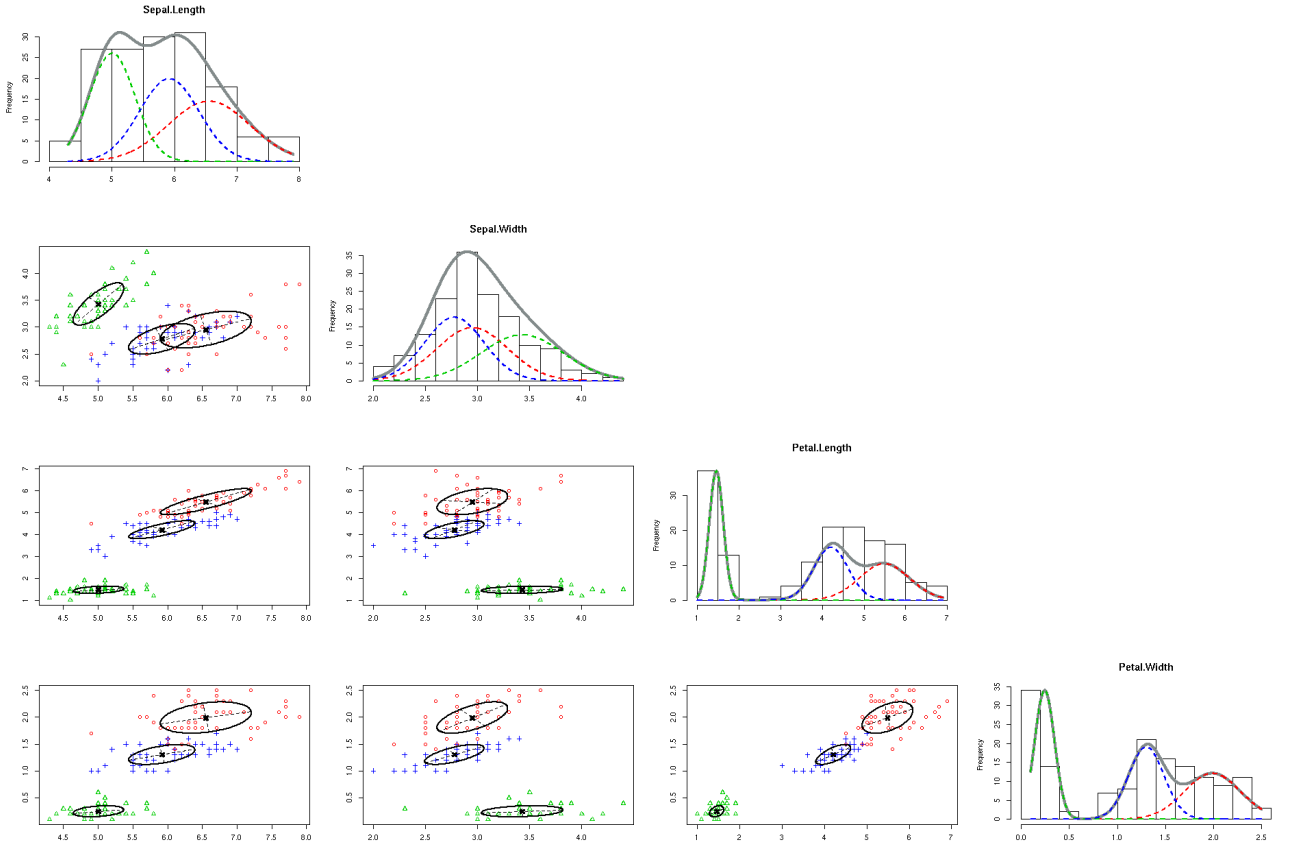
Keywords: model-based clustering, discriminant analysis, visualization, C++, R

Abstract: Mixmod [1] is a well-established software for fitting a mixture model of multivariate Gaussian or multinomial components to a given data set with either a clustering, a density estimation or a discriminant analysis point of view. It is written in C++ and its core library has been interfaced with Scilab and Matlab. It lacked an interface with R. The Rmixmod package provides a bridge between the C++ core library of Mixmod and the R statistical computing environment. Both cluster analysis and discriminant analysis can be now performed using Rmixmod. Many options are available to speficy the models and the strategy to run. Rmixmod is dealing with 28 multivariate Gaussian mixture models for quantitative data and 10 multivariate multinomial mixture models for qualitative data. Estimation of the mixture parameters is performed via the EM, the SEM or the CEM algorithms. These three algorithms can be chained and initialized in several different ways which leads to obtain original fitting strategies. Different model selection criteria are proposed according to the modelling purpose. User-friendly outputs and graphs allow for a good visualisation of the results. Rmixmod is available on CRAN.

An example of clustering in a quantitative case: The outputs and graphs of Rmixmod are illustrated on the well-known iris flower data set. `iris` is a data frame with 150 cases (rows) and 5 variables (columns) named `Sepal.Length`, `Sepal.Width`, `Petal.Length`, `Petal.Width`, and `Species`. The first fourth variables are quantitative and the `Species` variable is qualitative with 3 modalities. Hence, it is natural to fit a three component Gaussian mixture to this data set to retrieve the true partition. That can be done with the function `mixmodCluster()`:

```
# load Rmixmod package into R environment
R> library(Rmixmod)

# run a cluster analysis on the four quantitative variables of iris with three
# clusters, all the Gaussian models, the BIC and ICL model selection criteria
R> xem <- mixmodCluster(iris[1:4], 3, models=mixmodGaussianModel(), criterion=c("BIC","ICL"))

# show a summary of the best model containing the estimated parameters, the likelihood
# and the criteria values (here the output has been truncated)
R> summary(xem)
*************************************************************
* Number of samples     =  150
* Problem dimension     =  4
```

```
*************************************************************
* Number of cluster =  3
*         Criterion =  BIC(553.4052) ICL(557.6575)
*        Model Type =  Gaussian_p_Lk_Dk_A_Dk
*        Parameters =  list by cluster
*        Cluster  1 :
              Proportion =  0.3333
                   Means =  6.5516 2.9510 5.4909 1.9904
               Variances = |     0.4282     0.1078     0.3310     0.0630 |
                           |     0.1078     0.1155     0.0879     0.0606 |
                           |     0.3310     0.0879     0.3585     0.0831 |
                           |     0.0630     0.0606     0.0831     0.0847 |
*        Cluster  2 :
[ ... ]
*        Cluster  3 :
[ ... ]
*    Log-likelihood =  -186.5112
*************************************************************

# show the partition returned by the mixmodCluster() function
R> xem["partition"]
  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [38] 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 1 3 1 3
 [75] 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1
[149] 1 1

# the plot() function has been redefined to get on the same graph:
# - a 1D representation with densities and data
# - a 2D representation with isodensities, data points and partition
R> plot(xem)
```



Reference

[1] Biernacki C., Celeux G., Govaert G., Langrognet F., (2006). Model-Based Cluster and Discriminant Analysis with the MIXMOD Software. *Computational Statistics and Data Analysis*, vol. 51/2, pp. 587-600.