Rmixmod: A MIXture MODelling R package

Rémi Lebret[1,2], Serge Iovleff[1], Florent Langrognet[3]

[1]**Laboratoire de mathématiques Paul Painlevé** - U.M.R. 8524 - CNRS - Université Lille 1 - INRIA Lille Nord-Europe - MODAL Team
[2]**Laboratoire Heudiasyc** - U.M.R. 7253 - CNRS - Université de Technologie Compiègne
[3]**Laboratoire de mathématiques de Besançon** - U.M.R. 6623 - CNRS - Université de Franche-Comté

Overview

- Mixmod is a software for modelling **quantitative/qualitative data** written in C++ (**www.mixmod.org**)
- Rmixmod provides a bridge between the C++ core library and the R statistical computing environment
- Both **cluster analysis** and **discriminant analysis** can be performed
- Many options are available to speficy the models and the strategy to run
- Package implementing **S4 objects**
- **Rmixmod is available on CRAN**

Models, algorithms and criteria are fully described in the paper of the reference section.

Mixture Models

Mixture probability density function (pdf) f is a weighted sum of K components densities :

$$f(\mathbf{x}_i|\theta) = \sum_{k=1}^{K} p_k h(\mathbf{x}_i|\lambda_k)$$

- $h(.|\lambda_k)$ denotes a d-dimensional distribution parameterized by λ_k
- p_k are the mixing proportions
- λ_k are the component of the distribution

Multivariate Gaussian mixture models

In the **quantitative case**, h is the density of a Gaussian distribution with mean μ_k and variance matrix Σ_k

$\Rightarrow \quad \lambda_k = (\mu_k, \Sigma_k)$

28 Gaussian models based on the eigenvalue decomposition of the variances matrices are available. They depend on constraints on the variance matrix: same variance matrix between clusters, spherical variance matrix, etc.

► Gaussian models are computed with the `mixmodGaussianModel()` function

Multivariate multinomial mixture models

In the **qualitative case**, h is a multinomial distribution with a center $\mathbf{a}_k^j$ and the dispersion ε_k^j around this center for the jth variable of the kth component.

$\Rightarrow \quad \lambda_k = (\mathbf{a}_k, \varepsilon_k)$

10 multinomial models are available. ε_k^j can be independent of the variable j, independent of the component k or independent of both the variable j and the component k.

► Multinomial models are computed with the `mixmodMultinomialModel()` function

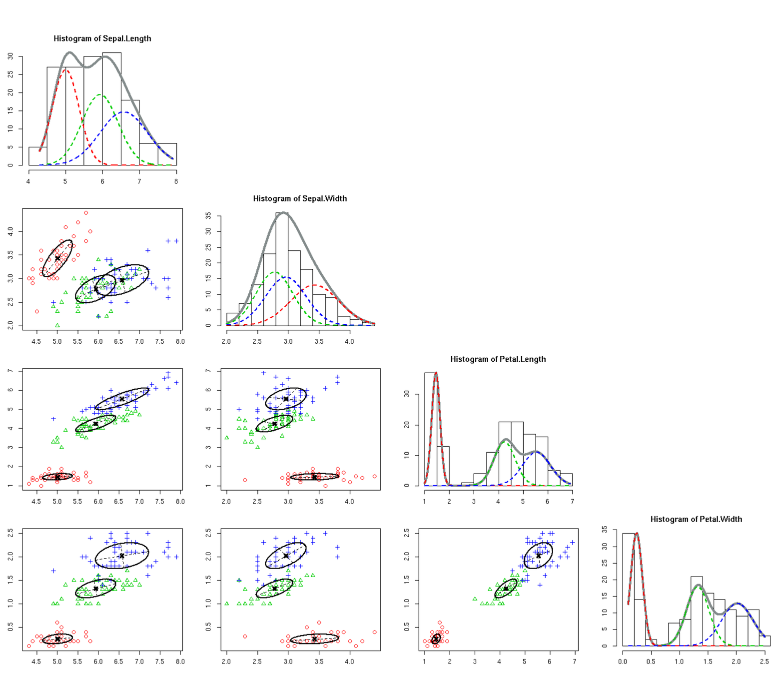
Visualization

`print()`, `show()`, `summarize()`, `plot()` and `hist()` functions have been redefined to visualize results of analyses.

Example in a quantitative case

`iris` is a data frame with 150 cases and 5 variables named `Sepal.Length`, `Sepal.Width`, `Petal.Length`, `Petal.Width`, and `Species`. Variables are quantitative except `Species` which is qualitative with 3 modalities.

R> plot(xem1)



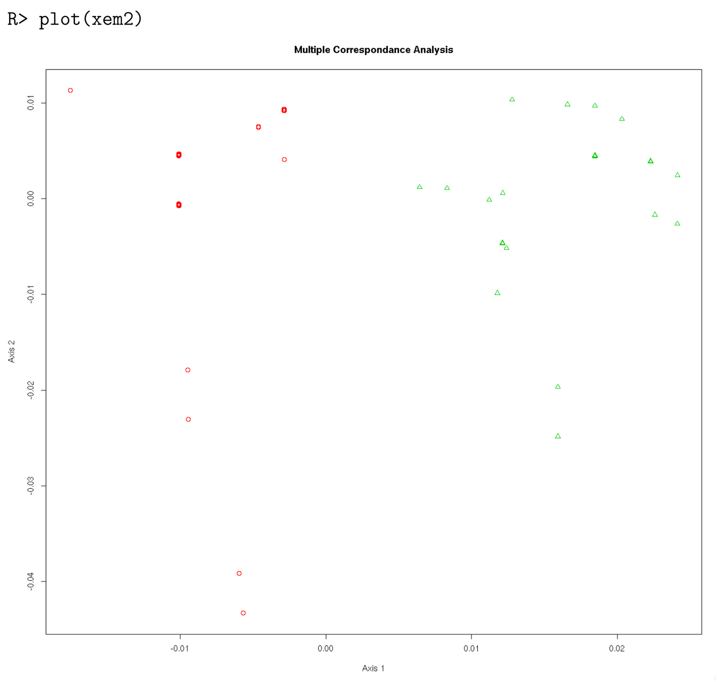
On the diagonal, 1D representation with densities and data
On lower triangular, 2D representation with isodensities, data points and labels

```
R> summary(xem1)
*******************************************************************
* Number of samples = 150
* Problem dimension = 4
*******************************************************************
*       Number of cluster = 3
*              Model Type = Gaussian_p_Lk_Dk_A_Dk
*               Criterion = BIC(554.9853) ICL(560.8033) NEC(0.0303)
*              Parameters = list by cluster
*                  Cluster 1 :
*                        Proportion = 0.3333
*                             Means = 5.0060 3.4280 1.4620 0.2460
*                         Variances = | 0.1308 0.1065 0.0189 0.0114 |
*                                     | 0.1065 0.1520 0.0116 0.0097 |
*                                     | 0.0189 0.0116 0.0283 0.0058 |
*                                     | 0.0114 0.0097 0.0058 0.0108 |
*                  Cluster 2 :
*                        Proportion = 0.3333
*                             Means = 5.9347 2.7776 4.2469 1.3204
*                         Variances = | 0.2388 0.0766 0.1627 0.0493 |
*                                     | 0.0766 0.0877 0.0796 0.0372 |
*                                     | 0.1627 0.0796 0.1900 0.0602 |
*                                     | 0.0493 0.0372 0.0602 0.0391 |
*                  Cluster 3 :
*                        Proportion = 0.3333
*                             Means = 6.5765 2.9627 5.5392 2.0176
*                         Variances = | 0.4191 0.1044 0.3089 0.0501 |
*                                     | 0.1044 0.1075 0.0748 0.0544 |
*                                     | 0.3089 0.0748 0.3219 0.0614 |
*                                     | 0.0501 0.0544 0.0614 0.0763 |
*          Log-likelihood = -187.3012
*******************************************************************
```

Example in a qualitative case

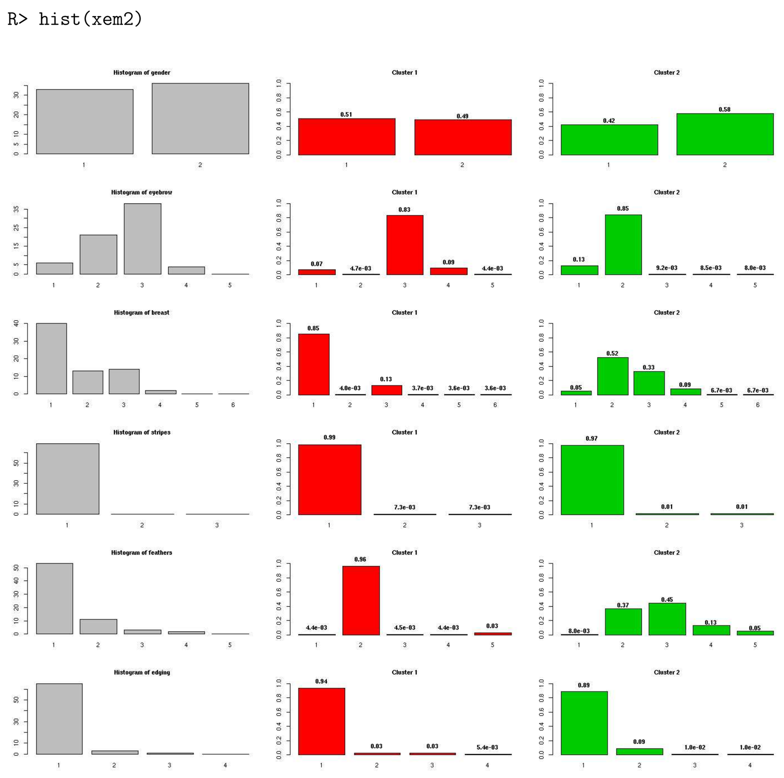
`birds` dataset contains details on the morphology of 69 birds (puffins). Each individual (bird) is described by six qualitative variables. One variable for the gender and five variables giving a morphological description of the birds.

R> plot(xem2)



Data and labels in the first MCA 2D-space

R> hist(xem2)



Estimation and Selection

- Estimation of the mixture parameters is considered through **Maximum Likelihood** via the following algorithms:
  - the **EM** (*Expectation Maximization*)
  - the **SEM** (*Stochastic EM*)
  - the **CEM** (*Clustering EM*)
- Algorithms can be chained to obtain **original fitting strategies** (e.g. CEM then EM with results of CEM)
- Many **initialization strategies** combining those algorithms are possible

► Strategies are defined with the `mixmodStrategy()` function

- The models and the number of clusters can be chosen by **different criteria**:
  - **BIC** (*Bayesian Information Criterion*)
  - **ICL** (*Integrated Completed Likelihood*, a classification version of BIC)
  - **NEC** (*Entropy Criterion*)
  - **CV** (*Cross Validation*)

Cluster Analysis

Discovering a group structure in a $n \times d$ data matrix $\mathbf{x} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ where $\mathbf{x}_i$ is an individual in $\mathbb{R}^d$.

$\Rightarrow$ Result is a partition of $\mathbf{x}$ into K groups defined with the labels $\mathbf{z} = \{\mathbf{z}_1, ..., \mathbf{z}_n\}$, with $\mathbf{z}_i = (z_{i1}, \ldots, z_{iK})$, $z_{ik} = 1$ or 0 according to $\mathbf{x}_i$ is assigned to the kth group or not

► Cluster analysis is computed with the `mixmodCluster(data,nbCluster)` function

Arguments
- `data`: a data matrix $\mathbf{x}$
- `nbCluster`: a list of K groups
- optional arguments set with default values: `strategy` (S4 object), `models` (S4 object), `criterion`, …

Return values
- `bestResult`: a S4 object containing results of the best model (estimated p_k, λ_k, partition, etc)
- `results`: a list of S4 objects containing results of all models.

Examples

```
# cluster analysis of iris with a list of cluster (from 2 to 8 clusters), all the Gaussian
  models, the BIC, ICL and NEC model selection criteria and an original strategy
R> xem1 <- mixmodCluster(iris[1:4], 2:8, models=mixmodGaussianModel(), criterion=c("BIC","
 ICL","NEC"), strategy=mixmodStrategy(name=c("SEM","EM"), initMethod="random"))

# cluster analysis of birds with 2 clusters
R> xem2 <- mixmodCluster(birds, 2)
```

Discriminant Analysis

Labels $\mathbf{z}$ are known.

$\Rightarrow$ Estimate the group $\mathbf{z}_{n+1}$ of any new individual $\mathbf{x}_{n+1}$ of $\mathbb{R}^d$ with unknown label.

Discriminant analysis in Rmixmod is divided into two steps:

1. Learning step

Obtaining a classification rule from the training observations

► Learning is computed with the `mixmodLearn(data,knownPartition)` function

Arguments
- `data`: a data matrix $\mathbf{x}$
- `knownPartition`: vector containing the known labels $\mathbf{z}$
- optional arguments set with default values: `models`, `criterion`, …

Return values
- `bestResult`: a S4 object containing results of the best model (estimated p_k, λ_k, partition, etc)
- `results`: a list of S4 objects containing results of all models.

Example

```
# start by extract 10 observations from iris dataset
R> remaining.obs <- sample(1:nrow(iris),10)
# then run the learning step without those 10 observations
# use of the variable Species as the partition
R> learn <- mixmodLearn(iris[-remaining.obs,1:4], iris$Species[-remaining.obs])
```

2. Prediction step

Assigning remaining observations to one of the groups

► Prediction is computed with the `mixmodPredict(data,classificationRule)` function

Arguments
- `data`: a data matrix $\mathbf{x}$
- `classificationRule`: vector containing the known labels $\mathbf{z}$

Return values
- `partition`: vector containing the predicted partition
- `proba`: a matrix containing probabilities of the prediction

Example

```
# prediction of the 10 remaining observations with the classification rule obtained in the
  learning step
R> prediction <- mixmodPredict(iris[iris.partition,1:4], learn["bestResult"])
```

Reference

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