mixedClust: an R package for mixed data classiication, clustering and co-clustering

Margot Selosse, Julien Jacques, Christophe Biernacki

mixedClust: an R package for mixed data classification, clustering and co-clustering

Package functionalities

The package provides model-based algorithm for clustering, co-clustering and classification with mixed-type data.
Principal functions are:

```
mixedClust # to perform clustering
mixedCoclust # to perform co-clustering
mixedClassif # to perform classification, in a parsimonious way or not
predictions # use the result from mixedClassif for predictions
```

Notations

- $\boldsymbol{x}$: N rows and $J = J_1 + \ldots + J_d + \ldots + J_D$ columns
- $\boldsymbol{x}$ composed of several matrices : $\boldsymbol{x^1}, \ldots, \boldsymbol{x^D}$
- $\boldsymbol{x^d}$: $N \times J_d$ matrix
- x^d is made of variables from one of 5 different types: **Continuous**, **Nominal**, **Ordinal**, **Integer** or **Functional**.
- In unsupervised methods: G clusters in line, $H_1 \ldots H_D$ clusters in column

$$\boldsymbol{x} = \left[\left[\quad \boldsymbol{x}^1 \quad \right] \ldots \left[\quad \boldsymbol{x}^D \quad \right]\right], \boldsymbol{x}^d = (x_{ij}^d)_{1 \leq i \leq N;\, 1 \leq j \leq J_d}.$$

Models (for $D = 2$)

Legend : -Observed partitions -Latent partitions

- Clustering:

$$p(\boldsymbol{x}; \Theta) = \sum_{v \in V} p(v; \Theta) \times p(\boldsymbol{x^1} | v; \Theta) p(\boldsymbol{x^2} | v; \Theta)$$

- Co-clustering:

$$p(\boldsymbol{x}; \Theta) = \sum_{v, w^1, w^2} p(v; \Theta) p(w^1; \Theta) p(w^2; \Theta) \times p(\boldsymbol{x^1} | v, w^1; \Theta) p(\boldsymbol{x^2} | v, w^2; \Theta)$$

- Classification without parsimony:

$$p(\boldsymbol{x}; \Theta) = p(v; \Theta) \times p(\boldsymbol{x^1} | v; \Theta) p(\boldsymbol{x^2} | v; \Theta)$$

- Classification with parsimony (obtained by clustering the features):

$$p(\boldsymbol{x}; \Theta) = \sum_{w^1, w^2} p(v; \Theta) p(w^1; \Theta) p(w^2; \Theta) \times p(\boldsymbol{x^1} | v, w^1; \Theta) p(\boldsymbol{x^2} | v, w^2; \Theta)$$

The parameters we want to estimate are:

$\Theta = \left(\alpha_{gh}^d, \gamma_g, \rho_h^d\right)_{1 \leq h \leq H_d \text{ and } 1 \leq d \leq D}$

- α_{gh}^d: parameters of distribution of g^{th} row-cluster and h^{th} column-cluster of $\boldsymbol{x^d}$. **It will depend on the type of $\boldsymbol{x^d}$.**
- γ_g: mixing proportion of g^{th} row-cluster
- ρ_h^d: mixing proportion of h^{th} column cluster for $\boldsymbol{x}^d$

Inference

EM and BIC not tractable in co-clustering, due to the double missing structure.
Consequently, we use:

- Stochastic EM algorithm, with a Gibbs sampler for the latent variables simulation
- ICL-BIC criterion for model selection

Results for classification on real dataset

Dataset

- Trauma-survey: 823 persons answered to 88 psychological questions about anxiety, depression, anger and possibly traumatizing life events. 307 of them were diagnosed with trauma, and 516 were declared not traumatized.
- $\boldsymbol{x}^1$: Categorical data from 17 questions about traumatizing life events.
- $\boldsymbol{x}^2$: Ordinal data from 71 questions about anger, depression and anxiety.
- 2/3 of the dataset was used to train the model. The last 1/3 was then used for prediction.

Results

	precision	recall	specificity
not parsimonious	0.75	0.80	0.83
(H_1, H_2) = (3,8)	0.78	0.88	0.85
(H_1, H_2) = (2,5)	**0.82**	**0.92**	**0.88**

Table: Precision, recall and specificity for different kc.

On classification with parsimony:

- Better results are obtained on predictions when we introduce parsimony than when we don't.
- Parsimony training result gives less parameters, which makes easier the interpretation.

Features clusters for parsimonious classification

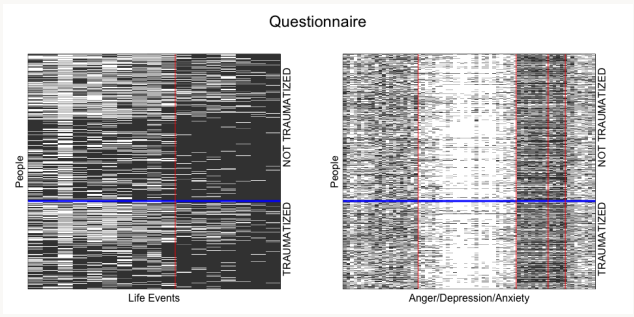

Figure: Patients classification and features clusters. Categorical answers about life events on the left. Ordinal answers about Anger/Anxiety/Depression on the right.

R code

```
######## Defining the dataset properties ######
dlist = c(0,17) # defining where each type begins in the complete
    dataset
distrib = c("Multinomial","Bos") # defining the distribution types

######## defining the SEM-Gibbs algorithm configuration ########
nbSEM = 250 # total number of iterations
nbSEMburn = 200 # burn-in period
nbindmini = 10 # minimum number of elements in one block
init = "kmeans" # initialization type

######## defining the number of clusters ########
kr = 2 # Two classes : Traumatized/Not Traumatized
kc = c(2,5) # Introducing parsimony (put to 0 for no parsimony)

######## running the classification function ########
classif = mixedClassif(M.train, y.train, dlist, kr = kr, kc = kc, init,
     distrib, nbSEM, nbSEMburn,nbindmini)

prediction <- predictions(classif, M.test)

######## printing predicted labels ########
prediction@zr_topredict;
```

References

[1] V. Robert. Classification et détection de signaux de pharmacovigilance dans des bases de données de grandes dimensions. PhD Thesis, Université Paris Sud, 2017.

[2] G. Govaert, M. Nadif. Co-Clustering. *John Wiley & Sons*, 2013.

[3] C. Keribin, G. Govaert, and G. Celeux. Estimation d'un modèle à blocs latents par l'algorithme SEM. *42èmes Journées de Statistique*, 2010.

M. Selosse[1], J. Jacques[1], C. Biernacki[2]
[1]: Université de Lyon, Lyon 2, ERIC EA 3083 ,[2]: Université Lille 1, Inria