mixedClust: an R package for mixed data classification, clustering and co-clustering
Margot Selosse, Julien Jacques, Christophe Biernacki

To cite this version:
Margot Selosse, Julien Jacques, Christophe Biernacki. mixedClust: an R package for mixed data classification, clustering and co-clustering. 25th Summer Session Working Group on Model-Based Clustering, Jul 2018, Ann Arbor, United States. <hal-01949171>
mixedClust: an R package for mixed data classification, clustering and co-clustering

Package functionalities

The package provides model-based algorithms 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

- $x$: $N$ rows and $J_1 + \ldots + J_d$ columns
- $x$ composed of several matrices: $x^1, \ldots, x^D$
- $x^d$: $N \times J_d$ 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

$$x = \begin{bmatrix} x^1 \ldots x^D \end{bmatrix}, \quad x^d = (x^d_{ij})_{i \in [1,N], j \in [1,J_d]}$$

Models (for $D = 2$)

Legend: - Observed partitions - Latent partitions

- Clustering:
  $$p(x; \Theta) = \sum_{z \in Z} p(z; \Theta) \times p(x^1 | z; \Theta)p(x^2 | z; \Theta)$$

- Co-clustering:
  $$p(x; \Theta) = \sum_{i, w, w'} p(i; \Theta)p(w; \Theta)p(w'; \Theta) \times p(x^1 | i, w, w'; \Theta)p(x^2 | i, w, w'; \Theta)$$

- Classification without parsimony:
  $$p(x; \Theta) = p(i; \Theta) \times p(x^1 | i; \Theta)p(x^2 | i; \Theta)$$

- Classification with parsimony (obtained by clustering the features):
  $$p(x; \Theta) = \sum_{i, w, w'} p(i; \Theta)p(w; \Theta)p(w'; \Theta) \times p(x^1 | i, w, w'; \Theta)p(x^2 | i, w, w'; \Theta)$$

The parameters we want to estimate are:

$$\Theta = (\gamma^{d}_{hi}, \rho^{d}_{hi}, \rho^{d}_{i}, \alpha^{d}_{hi}, \mu_{1}, \mu_{2})_{h_i \leq H_d \wedge 1 \leq d \leq D}$$

- $\gamma^{d}_{hi}$: parameters of distribution of $g^h$ row-cluster and $h^D$ column-cluster of $x^d$. It will depend on the type of $x^d$.
- $\alpha^{d}_{hi}$: mixing proportion of $g^h$ row-cluster
- $\rho^{d}_{hi}$: mixing proportion of $h^h$ column-cluster for $x^d$

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 and possibly traumatizing life events. 307 of them were diagnosed with trauma, and 516 were declared not traumatized.
- $x^1$: Categorical data from 17 questions about traumatizing life events.
- $x^2$: 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

<table>
<thead>
<tr>
<th>not parsimonious</th>
<th>precision</th>
<th>recall</th>
<th>specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(H_1, H_2) = (1,3)$</td>
<td>0.75</td>
<td>0.80</td>
<td>0.83</td>
</tr>
<tr>
<td>$(H_1, H_2) = (2,5)$</td>
<td>0.82</td>
<td>0.92</td>
<td>0.88</td>
</tr>
</tbody>
</table>

Table: Precision, recall and specificity for different $k_c$.

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/Anger/Depression on the right.

R code

```R
# ####### Defining the dataset properties #######
dist = c("Multinomial", "Bos") # defining the distribution types
distrib = c(1, 1) # defining where each type begins in the complete dataset

# ####### defining the EM-Gibbs algorithm configuration #######
distrib = c(1, 1) # total number of iterations
nsEM = 200 # burn-in period
nbBlock = 10 # minimum number of elements in one block
init = "kmeans" # initialization type

# ####### defining the number of clusters #######
kr = 2 # Two classes: Traumatized/Not Traumatized

# ####### running the classification function #######
classify = mixedClassif (x_train, y_train, dist = dist, kr = kr, kc = kc, init = init)

# ####### printing predicted labels #######
predictions = predict(classify, x_test)
```

References


M. Selosse¹, J. Jacques¹, C. Biernacki²
¹: Université de Lyon, Lyon 2, ERIC EA 3083, ²: Université Lille 1, Inria