Rankcluster: An R package for clustering multivariate partial rankings

Julien Jacques, Quentin Grimonprez, Christophe Biernacki

Rankcluster: An R Package For Clustering Multivariate Partial Rankings

by Julien Jacques, Quentin Grimonprez and Christophe Biernacki

Abstract **Rankcluster** is the first **R** package proposing both modeling and clustering tools for ranking data, potentially multivariate and partial. Ranking data are modeled by the Insertion Sorting Rank (ISR) model, which is a meaningful model parametrized by a central ranking and a dispersion parameter. A conditional independence assumption allows to take into account multivariate rankings, and clustering is performed by the mean of mixtures of multivariate ISR model. The clusters' parameters (central rankings and dispersion parameters) help the practitioners in the interpretation of the clustering. Moreover, the **Rankcluster** package provides an estimation of the missing ranking positions when rankings are partial. After an overview of the mixture of multivariate ISR model, the **Rankcluster** package is described and its use is illustrated through two real datasets analysis.

Introduction

Ranking data occur when a number of subjects are asked to rank a list of objects according to their personal preference order. Such data are of great interest in human activities involving preferences, attitudes or choices like Psychology, Sociology, Politics, Marketing, *etc.* For instance, the voting system *single transferable vote*, occurring in Ireland, Australia and New Zealand, is based on preferential voting (Gormley and Murphy, 2008). In a lot of applications, the study of ranking data discloses heterogeneity, due for instance to different political meanings, different human preferences, *etc.* A clustering analysis is then a useful tool for statistical exploration of such datasets, by characterizing the datasets by the mean of a reduced number of clusters of similar ranking data. In the present work, the clusters will be characterized by a median ranking (location parameter) and by a dispersion parameter. The interpretation of these two parameters for each cluster will produce a comprehensive and synthetic analysis of the whole dataset.

In the present paper two studies in sociology and in sport are carried out. In the sociology study, several students were asked to rank five words according to strength of association (least to most associated) with the target word "Idea": Thought, Play, Theory, Dream and Attention. The cluster analysis of the resulting rankings from this study allows to characterize the population of students by exhibiting three clusters of students, that can be interpreted (thanks to the parameters specific to each cluster) as follows: scientific students, literary-minded students and a third group of students in an intermediate position between scientific and too literary-minded. The second study focuses on the performance of English football clubs during the last two decades, and more particularly those of the "Big Four" clubs: Manchester United, Liverpool, Arsenal and Chelsea. For this, the rankings of these four clubs at the English championship (Premier League) and according to the UEFA coefficients are analyzed. The cluster analysis exhibits two clusters, relatively similar, reflecting a certain stability in the hierarchy between these clubs, except for the position of Chelsea within the hierarchy: it appears that Chelsea obtains better results in the second half of the period 1993-2013, what can be probably explained by the acquisition of the club by a Russian investor in 2003.

Recently, Jacques and Biernacki (2012) proposed a model-based clustering algorithm in order to analyse and explore such ranking data. This algorithm is able to take into account multivariate rankings (when several rankings are simultaneously observed, as for instance in the sport application in which rankings at the Premier League and rankings according to the UEFA coefficient are observed simultaneously each year) with potential partial rankings (when a subject did not rank all the objects). To the best of our knowledge, this is the only clustering algorithm for ranking data with a so wide application scope. This algorithm is based on an extension of the Insertion Sorting Rank (ISR) model (Biernacki and Jacques, 2013) for ranking data, which is a meaningful and effective model obtained by modeling the ranking generating process assumed to be a sorting algorithm. The ISR model is parametrized by a location parameter (the modal ranking) and a dispersion parameter. The heterogeneity of the rank population is modeled by a mixture of ISR whereas conditional independence assumption allows the extension to multivariate rankings. Maximum likelihood estimation is performed through a SEM-Gibbs algorithm, in which partial rankings are considered as missing data, what allows to simulate them during the estimation process.

This algorithm has been implemented in C++ and is available through the **Rankcluster** package for R, available on the CRAN website and presented in depth in the sequel of this paper.

The paper is organized as follows: the next section briefly presents the clustering algorithm proposed in Jacques and Biernacki (2012). Then, the following sections describe the existing R packages dedicated to ranking data, and the functionalities of the **Rankcluster** package. The **Rankcluster** package is then illustrated through the cluster analysis of two datasets: 1. Fligner and Verducci's *words* univariate dataset without any missing ranking positions (Fligner and Verducci, 1986), and 2. the rankings of the Big Four of English Football (Manchester United, Liverpool, Arsenal and Chelsea) according to the Premier League results and to their UEFA coefficients between 1993 and 2013 (bivariate dataset with missing ranking positions). The last section concludes the work.

Overview of the model-based clustering algorithm

This section gives an overview of the model-based clustering algorithm for multivariate partial rankings proposed in Jacques and Biernacki (2012). It relies on the univariate ISR model that we introduce first.

The univariate ISR model

Rank data arise when judges or subjects are asked to rank several objects $\mathcal{O}_1, \ldots, \mathcal{O}_m$ according to a given order of preference. The resulting ranking can be designed by its *ordering* representation $x = (x^1, \ldots, x^m) \in \mathcal{P}_m$ which signifies that Object $\mathcal{O}_{x^h}$ is the hth ($h = 1, \ldots, m$), where $\mathcal{P}_m$ is the set of the permutations of the first m integers, or by its *ranking* representation $x^{-1} = (x_1^{-1}, \ldots, x_m^{-1})$, which contains the ranks assigned to the objects and means that $\mathcal{O}_i$ is in x_i^{-1}th position ($i = 1, \ldots, m$). In the **Rankcluster** package, the ranking representation is used, but users working with the ordering representation can use a function implemented in the package in order to convert their data from one representation to another one.

Based on the assumption that a rank datum is the result of a sorting algorithm based on paired comparisons, and that the judge who ranks the objects uses the insertion sort because of its optimality properties (minimum number of paired comparisons), Biernacki and Jacques (2013) stated the following so-called ISR model:

$$\mathrm{p}(x; \mu, \pi) = \frac{1}{m!} \sum_{y \in \mathcal{P}_m} \mathrm{p}(x|y; \mu, \pi) = \frac{1}{m!} \sum_{y \in \mathcal{P}_m} \pi^{G(x,y,\mu)} (1-\pi)^{A(x,y)-G(x,y,\mu)}, \tag{1}$$

where

- $\mu \in \mathcal{P}_m$, the modal ranking, is a *location parameter*. Its *opposite* ranking $\bar{\mu}$ ($\bar{\mu} = \mu \circ \bar{e}$ with $\bar{e} = (m, \ldots, 1)$) is the rank of smallest probability,
- $\pi \in [\frac{1}{2}, 1]$, which is the probability of good paired comparison in the sorting algorithm (*good* means result in accordance with μ), is a *scale parameter*: the distribution is uniform when $\pi = \frac{1}{2}$ and the mode μ of the distribution is uniformly more pronounced when π grows, being a Dirac in μ when $\pi = 1$,
- the sum over $y \in \mathcal{P}_m$ corresponds to all the possible initial presentation orders of the objects to rank (with identical prior probabilities equal to $1/m!$ because they are unknown),
- $G(x, y, \mu)$ is equal to the number of good paired comparisons during the sorting process leading to return x when the presentation order is y,
- $A(x, y)$ corresponds to the total number of paired comparisons (good or wrong).

The accurate definitions of $G(x, y, \mu)$ and $A(x, y)$ can be found in Biernacki and Jacques (2013).

Mixture of multivariate ISR

Let now redefine $\mathbf{x} = (x^1, \ldots, x^p) \in \mathcal{P}_{m_1} \times \ldots \times \mathcal{P}_{m_p}$ as a *multivariate* rank, in which $x^j = (x^{j1}, \ldots, x^{jm_j})$ is a rank of m_j objects ($1 \leq j \leq p$).

The population of multivariate ranks is assumed to be composed of K groups in proportions p_k ($p_k \in [0, 1]$ and $\sum_{k=1}^K p_k = 1$). Given a group k, the p components $x^1, \ldots, x^p$ of the multivariate rank datum $\mathbf{x}$ are assumed to be sampled from independent ISR distributions with corresponding modal rankings $\mu_k^1, \ldots, \mu_k^p$ (each $\mu_k^j \in \mathcal{P}_{m_j}$) and good paired comparison probabilities $\pi_k^1, \ldots, \pi_k^p \in [\frac{1}{2}, 1]$.

The unconditional probability of a rank $\mathbf{x}$ is then

$$\mathrm{p}(\mathbf{x}; \boldsymbol{\theta}) = \sum_{k=1}^K p_k \prod_{j=1}^p \mathrm{p}(x^j; \mu_k^j, \pi_k^j), \tag{2}$$

where $\boldsymbol{\theta} = (\pi_k^j, \mu_k^j, p_k)_{k=1,\ldots,K, j=1,\ldots,p}$ and $\mathrm{p}(x^j; \mu_k^j, \pi_k^j)$ is defined by (1).

Each component x^j of $\mathbf{x}$ can be full or partial. Frequently, the objects in the top positions will be ranked and the missing ones will be at the end of the ranking, but our model does not impose such situation and is able to work with partial ranking whatever are the positions of the missing data (see details in Jacques and Biernacki (2012)).

Estimation algorithm

Parameter estimation is performed thanks to maximum likelihood. However, maximum likelihood estimation is not straightforward since several missing data occur: the cluster memberships z_i of the observations, the presentation orders $\mathbf{y}_i$ ($\mathbf{y}_i = (y_i^1, \ldots, y_i^p)$ is the presentation order for the ith observation) and the unobserved ranking positions, denoted by $\hat{\mathbf{x}}_i$ (for partial rankings). In such a situation, a convenient way to maximize the likelihood is to consider an EM algorithm (Dempster et al., 1977). This algorithm relies on the completed-data log-likelihood, and proceeds in iterating an E step, in which the conditional expectation of the completed-data log-likelihood is computed, and a M step, in which the model parameters are estimated by maximizing the conditional expectation computed in the E step. Unfortunately, the EM algorithm is tractable only for univariate full rankings with moderate m ($m \leq 7$), respectively for mathematical and numerical reasons. In particular, when partial rankings occur, the E step is intractable since the completed-data log-likelihood is not linear for all three types of missing data (refer to Jacques and Biernacki (2012) for its expression). A SEM-Gibbs approach is then proposed in Jacques and Biernacki (2012) to overcome these problems.

The fundamental idea of this algorithm is to reduce the computational complexity that is present in both E and M steps of EM by removing all explicit and extensive use of the conditional expectations of any product of missing data. First, it relies on the SEM algorithm (Geman and Geman, 1984; Celeux and Diebolt, 1985) which generates the latent variables at a so-called stochastic step (S step) from the conditional probabilities computed at the E step. Then these latent variables are directly used in the M step. Second, the advantage of the SEM-Gibbs algorithm in comparison with the basic SEM ones relies on the fact that the latent variables are generated without calculating conditional probabilities at the E step, thanks to a Gibbs algorithm. Refer to Jacques and Biernacki (2012) for more details.

Existing R packages for ranking data

To the best of our knowledge, there exists only two packages dedicated to the analysis of ranking data, available on the CRAN website, but their functionalities are significantly limited in comparison to our package **Rankcluster**, as we discuss now:

- **pmr** package for R (Lee and Yu, 2013): provide some descriptive statistics and modeling tools using classical rank data models for full univariate ranking data: Luce models, distance-based models, and rank-ordered logit (refer to Marden (1995) for a description of these models). Visualization of ranking data using polytopes is also available for less than four objects to rank ($m \leq 4$).
- **RMallow** package for R: suppose to perform clustering of univariate ranking data using mixture of Mallows model (Murphy and Martin (2003)).

Rankcluster proposes modeling and clustering tools on the basis of the mixture of multivariate ISR presented in the previous section. Comparing to the existing packages, **Rankcluster** is the only package taking into account multivariate and partial ranking data.

Overview of the Rankcluster functions

This section presents briefly the functions of the **Rankcluster** package. For more details, refer to the help of the functions.

The main function: rankclust()

The `rankclust()` function performs modeling or cluster analysis of a dataset of rankings. For cluster analysis, the number of clusters can be specified by the user, or selected by the BIC (Schwarz, 1978) (or ICL (Biernacki et al., 2000)) criterion in a list of possible numbers provided by the user. By default, the number of clusters is equal to one, and consequently modeling of the dataset is performed (*i.e.*

a single homogeneous multivariate ISR model is fitted to the data). Main inputs and outputs are described below.

This function has only one mandatory argument, `data`, which is a matrix composed of the n observed ranks in their ranking representation. For **univariate rankings** the number of columns of `data` is m (default value of argument `m`). For **multivariate rankings**, `data` has $m_1 + \ldots + m_p$ columns: the first m_1 columns contain x^1 (first dimension), the columns $m_1 + 1$ to $m_1 + m_2$ contain x^2 (second dimension), and so on. In this case, the argument `m` must be filled with the vector of size $(m_1, \ldots, m_j)$. Several parameters allow also to set up the different tuning parameters (iteration numbers) used in the SEM-Gibbs estimation. Refer to Jacques and Biernacki (2012) and to `rankclust()` help for more details (section 2.5 gives also some examples of iteration numbers choices). The option `run` allows to set the number of initializations of the SEM-Gibbs algorithm (1 by default). In the case of multiple initializations, the best solution according to the approximated log-likelihood is retained.

The `rankclust()` function returns an instance of the `ResultTab` class. Its attributes contain 5 slots, among which `results` which is a list containing all $k = 1, \ldots, K$ classes, summarized in 18 slots, among which the main ones are:

- `proportion`: a K-vector of proportions $p_1, \ldots, p_K$.
- `pi`: a $K \times p$-matrix composed of the scale parameters π_k^j ($1 \leq k \leq K$ and $1 \leq j \leq p$).
- `mu`: a matrix with K lines and $m_1 + \ldots + m_p$ columns in which line k is composed of the location parameters $(\mu_k^1, \ldots, \mu_k^p)$ of cluster k.
- `ll, bic, icl`: values of the log-likelihood, BIC criterion and ICL criterion.
- `tik`: a $n \times K$-matrix containing the estimation of the conditional probabilities for the observed ranks to belong to each cluster.
- `partition`: a n-vector containing the partition estimation resulting from the clustering.
- `partialRank`: a matrix containing the full rankings, estimated using the within cluster ISR parameters when the ranking is partial.
- `distanceProp, distancePi, distanceMu`: distances between the final estimation and the current value at each iteration of the SEM-Gibbs algorithm (except the burning phase) for respectively: proportions p_k, scale parameters π_k^j, location parameters μ_k^j.
- `distanceZ`: a vector of size `Qsem-Bsem` containing the rand index (Rand, 1971) between the final estimated partition and the current value at each iteration of the SEM-Gibbs algorithm (except the burning phase).

If `res` is variable name of the result of `rankclust()`, each slot can be reached by `res[k]@slotname`, where `k` is the cluster index and `slotname` is the name of the selected slot (`proportion`, `pi` ...). For the slots `ll, bic, icl`, `res["slotname"]` returns a vector of size K containing the values of the slot for each cluster index.

Companion functions

In addition to the main function, `rankclust()`, several companion functions are available in **Rankcluster**:

- `convertRank()`: converts ranking representation x^{-1} of a rank to its ordering representation x, and *vice-versa* since $x \circ x^{-1} = x^{-1} \circ x$.
- `distCayley(), distHamming(), distKendall(), distSpearman()`: compute usual distances between rankings (refer to Marden (1995)) for either ranking or ordering representation.
- `frequence()`: transforms a raw dataset composed of a matrix of ranks (one rank per line, with possibly equal lines if the same rank is observed several times) into a matrix rank/frequency containing in line each different observed ranks and one additional last column with the frequency of observation of these ranks. Conversely, `unfrequence()` transforms a rank/frequency dataset in a raw dataset, as requested in input argument of `rankclust()`.
- `khi2()`: performs a chi-squared goodness-of-fit tests and return the p-value of the test (refer to Biernacki and Jacques (2013) for details).
- `kullback()`: estimates the Kullback-Leibler divergence between two ISR models.
- `simulISR()`: simulates a univariate and unimodal dataset of full rankings according to the ISR model.
- `probability()`: computes the probability of a ranking (multivariate or not) according to the ISR model.

Rankcluster through examples

This section illustrates the use of the `rankclust()` function on two real datasets. The first one, `words`, is a well-known dataset in ranking study, due to Fligner and Verducci (1986), which consists of words associations by students. The second one, `big4` consists of the rankings of the Big Four of English Football (Manchester United, Liverpool, Arsenal and Chelsea) according to the Premier League results and to their UEFA coefficients between 1993 and 2013.

The `words` dataset

This dataset was collected under the auspices of the Graduate Record Examination Board (Fligner and Verducci, 1986). A sample of 98 college students were asked to rank five words according to strength of association (least to most associated) with the target word "Idea": 1 = Thought, 2 = Play, 3 = Theory, 4 = Dream and 5 = Attention.

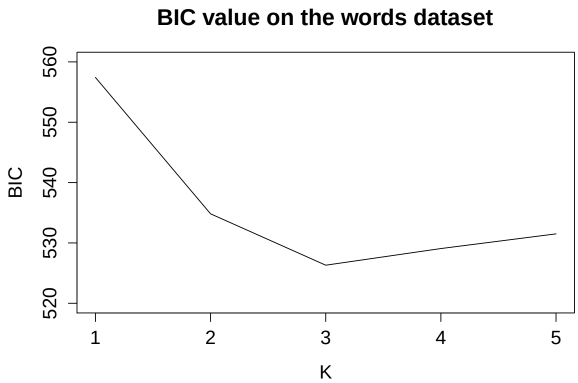

Figure 1: Value of the BIC criterion with mixture of `ISR` for the `words` dataset.

First we start by installing and loading the **Rankcluster** package and then loading the `words` dataset:

```
R> install.packages("Rankcluster")
R> library(Rankcluster)
R> data(words)
```

Using the `rankclust()` function, a clustering with respectively 1 to 5 clusters is estimated:

```
R> res=rankclust(words$data,m=words$m,K=1:5,Qsem=1000,Bsem=100,Ql=500,Bl=50,maxTry=20,run=10)
```

The number of SEM-Gibbs iterations (`Qsem`) has been set to 1000, with a burning phase of 100 iterations (`Bsem`). For likelihood approximation the numbers of iterations (`Ql` and `Bl`) have been divided by two. Option `maxTry=20` allows to restart the estimation algorithm in the limit of 20 times if one cluster becomes empty (frequent for $K = 5$). Finally, the SEM-Gibbs algorithm is initialized 5 times (`run=5`), and the best solution (according to the approximated likelihood) is retained. Computing time on a laptop with 2.80GHz CPU is about 3 minutes (7 seconds per run et per number of clusters). The reader who wants to test more quickly the package can reduced the number of runs, using for instance:

```
R> res=rankclust(words$data,m=words$m,K=1:5,Qsem=200,Bsem=20,Ql=200,Bl=20,maxTry=20)
```

The values of the BIC criterion, reached by `res["bic"]` and plotted on Figure 1, tend to select three clusters.

The parameter estimation for $K = 3$ are given below for proportions p_k, scales π_k and modes μ_k:

```
> res[3]@proportion
[1] 0.3061224 0.4918367 0.2020408
> res[3]@pi
         dim 1
cl 1 0.9060649
cl 2 0.9416822
```

```
cl 3 0.8642753
> res[3]@mu
     dim 1
cl 1     5 1 3 4 2
cl 2     5 1 4 3 2
cl 3     5 2 4 3 1
```

The words Thought is the most associated with Idea for all clusters. Regarding the rankings of the four other words can suggest an interesting interpretation of the clusters. Indeed, the first cluster, composed of about 30% of the students, is characterized by the following modal ranking: Play, Attention, Theory, Dream, Thought. Students of this cluster are probably literary-minded students, rankings the word Dream just after Thought. Students of the second cluster (about half of total students) are probably more scientific since they rank the word Theory just after Thought, and so before the word Dream: Play, Attention, Dream, Theory, Thought. This cluster is also the most homogeneous, with a high scale parameter value (low dispersion): $\pi_2 \simeq 0.94$. Finally, the last cluster is characterized by the following mode: Attention, Play, Dream, Theory, Thought. The only difference in the modal ranking with the scientific students is the preference of Play rather than Attention. This cluster, which is the smallest (20% of the students), can be qualified as intermediary cluster, probably composed of a set of students not too scientific or too literary-minded, as evidenced by the smallest of the three scale parameter values ($\pi_3 \simeq 0.86$).

The `big4` dataset

In the two last decades, the English football has been dominated by four clubs, forming the "Big Four": Manchester United, Liverpool, Arsenal and Chelsea. In this application, we analyse the rankings of these four teams at the English championship (Premier League) and their rankings according to the UEFA coefficients. This coefficient is an European statistic on football teams based on the results of European football competitions and used for ranking and seeding teams in international competitions. The `big4` dataset, available in the package, is composed of Premier League rankings and UEFA rankings from 1993 to 2013, in ranking notation (club "1" is Manchester United, "2" is Liverpool, "3" is Arsenal and "4" is Chelsea). In 2001 Arsenal and Chelsea had the same UEFA coefficient and then are tied for the first ranking dimension. With **Rankcluster**, one way to take into account such tied in ranking data is to consider the corresponding ranking positions as missing: the UEFA ranking becomes then $(1, 0, 0, 2)$ for 2001, what means that Manchester United is the first, Liverpool is the last, and the two intermediate positions are for Arsenal and Chelsea in an unknown order.

First, the `big4` dataset is loaded:

```
R> data(big4)
```

Then, the number of clusters is estimated thanks to the BIC criterion. For this, clustering for 1 to 3 clusters is estimated with the `rankclust()` function. It should be noticed that for 3 clusters, the algorithm has to be launched several times since it often converges to a solution with one empty clusters. The values of the BIC criterion are plotted on Figure 2, and tend to select two groups (which confirms that with 3 clusters the estimation algorithm often converges to one empty cluster).

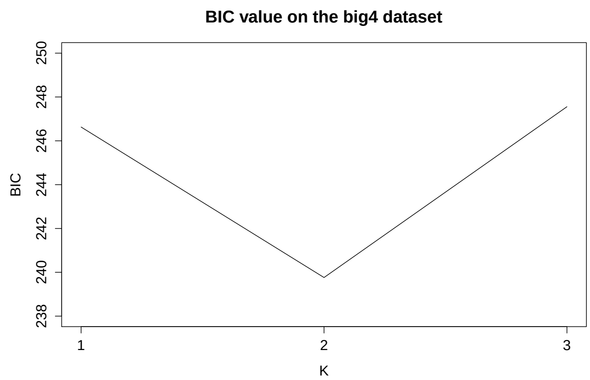

Figure 2: Values of the BIC criterion with mixture of ISR for the `big4` dataset.

The clustering with $K = 2$ is obtained in about 25 seconds on a laptop 2.80GHz CPU:

```
R> res=rankclust(big4$data,m=big4$m,K=2,Qsem=1000,Bsem=100,Ql=500,Bl=50,maxTry=20,run=5)
```

The printed outputs for $K = 2$ are given below: value of the log-likelihood (ll), values of BIC and ICL criteria, estimation of the proportions p_k's, the modes μ_k^j's, the scales π_k^j's, the estimated partition and finally the conditional probability of the observations to belong to each cluster (t_{ik}).

```
R> res[2]
******************************************************************
Number of clusters: 2
******************************************************************
ll= -108.5782
bic = 244.5571
icl = 253.5601
proportion: 0.3854034 0.6145966
mu:
    dim1       dim2
cl1    1 3 2 4    1 3 2 4
cl2    1 3 4 2    1 4 3 2
pi:
         dim1      dim2
cl1 0.9698771 0.7759781
cl2 0.6945405 0.7707101
partition:
[1] 2 1 1 1 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2
tik:
[,1] [,2]
[1,] 6.280426e-01 0.371957427
[2,] 9.933149e-01 0.006685094
[3,] 9.565052e-01 0.043494814
[4,] 9.923445e-01 0.007655450
[5,] 6.609592e-01 0.339040841
[6,] 9.916482e-01 0.008351815
[7,] 1.215925e-03 0.998784075
[8,] 3.500066e-01 0.649993363
[9,] 3.441535e-02 0.965584647
[10,] 4.063712e-01 0.593628767
[11,] 9.589745e-01 0.041025518
[12,] 9.751644e-01 0.024835551
[13,] 6.806917e-02 0.931930833
[14,] 3.175113e-03 0.996824887
[15,] 1.263591e-03 0.998736409
[16,] 7.713598e-06 0.999992286
[17,] 9.115424e-04 0.999088458
[18,] 1.734860e-01 0.826513968
[19,] 1.605939e-04 0.999839406
[20,] 7.425989e-02 0.925740107
[21,] 2.171738e-02 0.978282624
******************************************************************
```

The estimated clustering exhibits two groups, the second one being larger than the first one ($p_1 \simeq 0.39$ and $p_2 \simeq 0.61$). The values of the modes for each cluster and dimension leads to two interesting remarks. First, the ranking in each dimension is very similar in both clusters: exactly the same for cluster 1 and just one transposition in the last two positions for cluster 2. This means that the performance of the clubs at the Premier League is highly correlated with their UEFA rankings, which is related to the results of the clubs in the European competitions over the previous five seasons. This first comment shows a certain inertia in the performance of the clubs. Secondly, the distinction between the two clusters is essentially due to the position of the club "4", Chelsea: indeed, in the first cluster Chelsea is the last in both rankings, but it is in second position in the second cluster. Moreover, in the partition, we find that cluster 2 is mainly present in the second half of the period 1993-2013 (see for instance the conditional probabilities of cluster membership on Figure 3). This rise of Chelsea's results can be explained by the acquisition of the club by a Russian investor (Abramovich) in 2003, who brought great players in the club.

In addition to this information, the `summary()` function gives an overview of the partition by printing the five ranks of highest probability and the five ranks of highest entropy for each cluster:

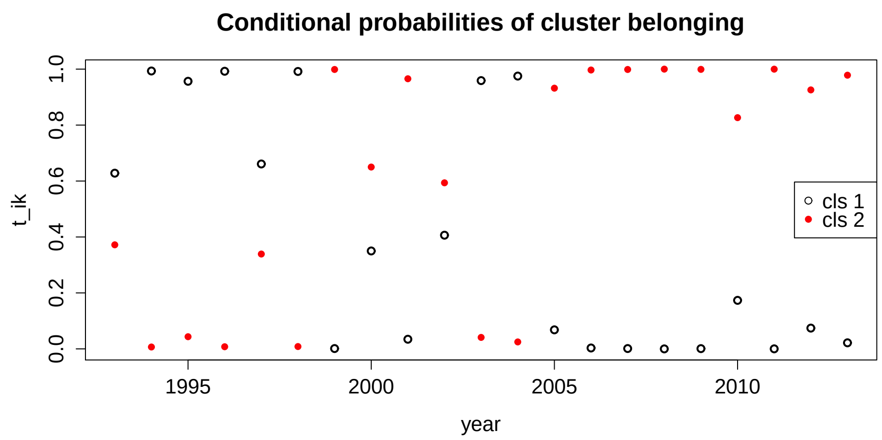

Figure 3: Conditional probabilities for each observation (year) to belong to cluster 1 (black circle) and 2 (red filled circle).

```
R> summary(res)
```

The ranks of highest probability are the best representatives of the cluster, whereas the ranks of highest entropy are those for which their membership to the cluster are the less obvious. Notice that the full list of the cluster member with their probability and entropy are available through the slots `probability` and `entropy`. Table 1 gives an example of these outputs for cluster 2. The rankings of 2011 are the most representative of the cluster, and the five most representatives of the cluster correspond to rankings after 2003. Similarly, the four observations whose membership to cluster 2 is the most questionable correspond to observations before 2003. This information confirms the previous analysis indicating that cluster 2 is due to the rise of Chelsea in the first positions.

year	UEFA	Prem. League	probability
2011	(1,3,4,2)	(1,4,3,2)	2.367e-04
2008	(4,2,3,1)	(1,4,3,2)	6.862e-05
2013	(2,4,3,1)	(1,4,3,2)	3.529e-05
2009	(3,2,4,1)	(1,2,4,3)	3.097e-05
2005	(1,2,3,4)	(3,4,2,1)	2.151e-05
year	UEFA	Prem. League	entropy
2002	(1,4,2,3)	(3,2,1,4)	0.6755106
1993	(1,2,3,4)	(1,2,3,4)	0.6599892
2000	(1,4,3,2)	(1,3,2,4)	0.6474507
1997	(1,2,3,4)	(1,3,2,4)	0.6403972
2010	(1,0,0,4)	(2,4,3,1)	0.4613709

Table 1: Rankings with the highest entropies and probabilities in the second cluster.

The `summary()` function prints also the estimated full ranking for each partial ranking. For instance, in 2001 Arsenal and Chelsea had the same UEFA coefficient, and when asking to our model to differentiate these two teams, Arsenal is ranked before Chelsea, what is not surprising as we already remarked that the results of Chelsea were generally among the worst of the Big Four before 2003.

Finally, the variability of estimation of the model parameters can be achieved by the mean of the distances between the final estimation and the current value at each step of the SEM-Gibbs algorithm (refer to Jacques and Biernacki (2012) for accurate definitions of these distances). These distances are available in the slots `distanceProp`, `distancePi`, `distanceMu` of the output `res[2]`. The standard deviation of these distances can be used for instance as an indicator of estimation variability. For instance, the standard deviation of the Kendall distance (see Marden (1995)) between the final estimation of the modes and its current value at each step of the SEM-Gibbs algorithm is for cluster 2: 0.52 for the UEFA coefficients rankings and 0.43 for Premier League rankings. Similarly, the standard deviation of the scales estimations is for cluster 2 about 0.002 for both the UEFA coefficients and the Premier League rankings. Let note that these variabilities are relatively small, due to the low overlapping of the two clusters (scale coefficients are quite high). In a similar way, the slot `distanceZ` illustrates the convergence of the SEM-Gibbs algorithm by given the rand index (Rand, 1971) between the final partition and the current partition at each SEM-Gibbs iteration (Figure 4).

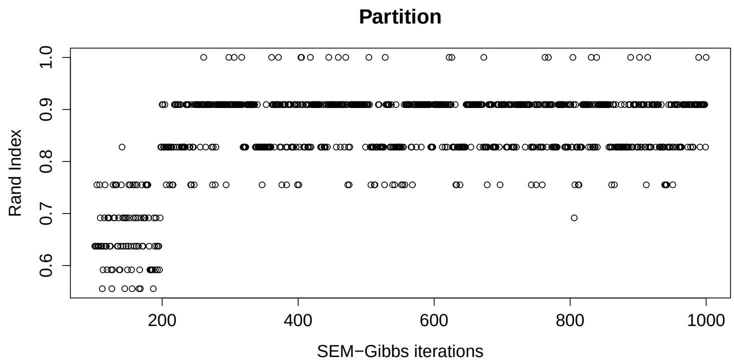

Figure 4: Evolution of the partition along with the SEM-Gibbs iterations: values of the Rand index between current and final partitions.

Conclusion

Rankcluster is the first R package dedicated to ranking data, allowing modeling and cluster analysis for multivariate partial ranking data. Available on the CRAN website, this package is simple of use with its main function, `rankclust()`, having only one mandatory argument, the ranking dataset. By default a single homogeneous multivariate ISR model is fitted to the data, and mentioning the number of desired clusters leads to perform a cluster analysis, with selection of the number of clusters if several numbers are given. The analysis of two real datasets presented in this paper allows to illustrate the possibilities of the package, and also constitutes a user guide for the practitioners.

Bibliography

C. Biernacki and J. Jacques. A generative model for rank data based on sorting algorithm. *Computational Statistics & Data Analysis*, 58:162–176, 2013. [p]

C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):719–725, 2000. [p]

G. Celeux and J. Diebolt. The SEM algorithm: A probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quarterly*, 2(1):73–82, 1985. [p]

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B. Methodological*, 39(1):1–38, 1977. With discussion. [p]

M. Fligner and J. Verducci. Distance based ranking models. *Journal of the Royal Statistical Society. Series B. Methodological*, 48(3):359–369, 1986. [p]

A. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Matching Intelligence*, 6:721–741, 1984. [p]

I. Gormley and T. Murphy. A mixture of experts model for rank data with applications in election studies. *Annals of Applied Statistics*, 2(4):1452–1477, 2008. [p]

J. Jacques and C. Biernacki. Model-based clustering for multivariate partial ranking data. Technical Report 8113, Inria Research Report, 2012. [p]

P. Lee and P. Yu. An R package for analyzing and modeling ranking data. *BMC Medical Research Methodology*, 13(65):1–11, 2013. [p]

J. Marden. *Analyzing and modeling rank data*, volume 64 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London, 1995. [p]

T. Murphy and D. Martin. Mixtures of distance-based models for ranking data. *Comput. Statist. Data Anal.*, 41(3-4):645–655, 2003. [p]

W. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 55:846–850, 1971. [p]

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978. [p]

Julien Jacques
MODAL team (Inria) & Laboratoire Paul Painlevé (UMR CNRS 8524)
University Lille I
Cité Scientifique, 59655 Villeneuve d'Ascq cedex
France julien.jacques@polytech-lille.fr

Quentin Grimonprez
MODAL team
Inria Lille-Nord Europe
40 avenue du Halley, 59650 Villeneuve d'Ascq
France quentin.grimonprez@inria.fr

Christophe Biernacki
MODAL team (Inria) & Laboratoire Paul Painlevé (UMR CNRS 8524)
University Lille I
Cité Scientifique, 59655 Villeneuve d'Ascq cedex
France christophe.biernacki@math.univ-lille1.fr