Unsupervised learning

4: Multidimensional Scaling & Unfolding

Antonio D'Ambrosio

Outline

1. MDS
2. MDS by hand
3. Metric vs non-metric MDS
4. STRESS
5. Classical Scaling
6. MDS algorithms
7. Unfolding

Introduction I

Unsupervised learning is often performed as part of an exploratory data analysis.

It can be hard to assess the results obtained from unsupervised learning methods: we have not a "supervisor". In unsupervised learning, there is no way to check our work because we don't know the true answer.

There are several models and methods dealing with unsupervised learning: factor analysis, principal component analysis, correspondence analysis, multidimensional scaling, cluster analysis....

Introduction II

We will focus our attention on Multidimensional Scaling and Unfolding.

Multidimensional Scaling

Introduction I

Multidimensional Scaling (MDS) generally is a technique used to visualize the level of (dis)similarity of individual cases or stimuly of a data set.
It is generraly used to

- visualizing proximity data
- uncovering latent dimensions of judgment
- testing structural hypotheses

Visualizing proximity data I

From Borg, Groenen and Mair, 2013: The figure shows a case from industrial psychology. Its 27 points represent 25 items and two indexes from an employee survey in an international IT company (Liu et al. 2004). Two examples for the items are: "All in all, I am satisfied with my pay", and "I like my work", both employing a Likert-type response scale ranging from "fully agree" to "fully disagree." The two indexes are scale values that summarize the employees' responses to a number of items that focus on their affective commitment to the company and on their general job satisfaction, respectively. The distance between two points represents the correlation of the respective variables.

Visualizing proximity data II

Fig. 2.1 MDS representation of the intercorrelations of 25 items and 2 indexes of an employee survey in an international IT company. The grayed area around organizational commitment contains likely drivers of commitment

Uncovering latent dimensions of judgment I

Why does Julia look like Mike's daughter?
How come that a Porsche appears to be more similar to a Ferrari than to a Cadillac?
To explain such judgments or perceptions, distance models offer themselves as natural candidates. In such models, the various objects are first conceived as points in a psychological space that is spanned by the subjective attributes of the objects. The distances among the points then serve to generate overall impressions of greater or smaller similarity.

Uncovering latent dimensions of judgment II

Wish (1971) wanted to know the attributes that people use when judging the similarity of different countries. He conducted an experiment where 18 students were asked to rate each pair of 12 different countries on their overall similarity. For these ratings, an answer scale from "extremely dissimilar" (1) to "extremely similar" (9) was offered to the respondents. No explanation was given on what was meant by "similar".

Uncovering latent dimensions of judgment III

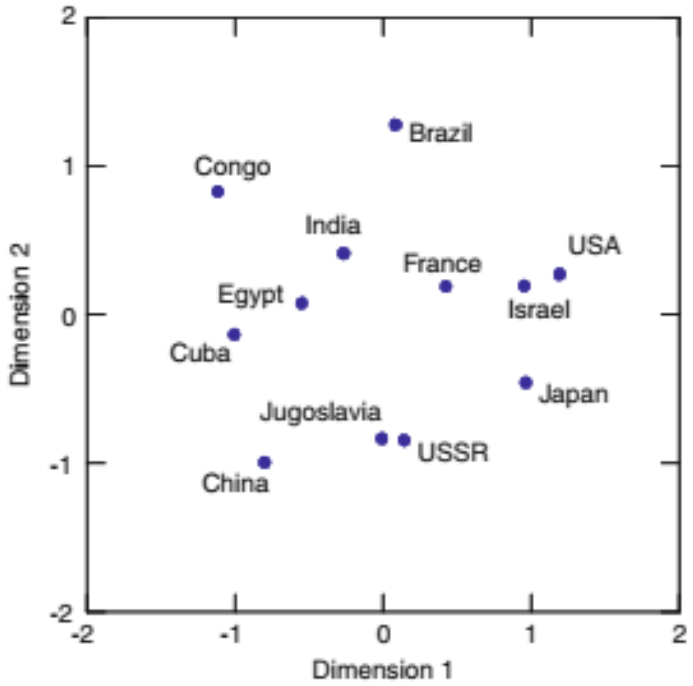
From data...

Table 2.1 Mean similarity ratings for 12 countries (Wish 1971)

Country		1	2	3	4	5	6	7	8	9	10	11	
Brazil	1	–											
Congo	2	4.83	–										
Cuba	3	5.28	4.56	–									
Egypt	4	3.44	5.00	5.17	–								
France	5	4.72	4.00	4.11	4.78	–							
India	6	4.50	4.83	4.00	5.83	3.44	–						
Israel	7	3.83	3.33	3.61	4.67	4.00	4.11	–					
Japan	8	3.50	3.39	2.94	3.83	4.22	4.50	4.83	–				
China	9	2.39	4.00	5.50	4.39	3.67	4.11	3.00	4.17	–			
USSR	10	3.06	3.39	5.44	4.39	5.06	4.50	4.17	4.61	5.72	–		
USA	11	5.39	2.39	3.17	3.33	5.94	4.28	5.94	6.06	2.56	5.00	–	
Jugoslavia	12	3.17	3.50	5.11	4.28	4.72	4.00	4.44	4.28	5.06	6.67	3.56	–

Uncovering latent dimensions of judgment IV

... to visualization: how interpret latent dimensions?



Testing structural hypotheses I

A frequent application of MDS is using it to test structural hypotheses. For example,suppose that persons are asked to solve several test items, which can be classified on the basis of their content into different categories of two design factors (facets). Some test items require the testee to solve computational problems with numbers and numerical operations (N). Other items ask for geometrical solutions (G). The data in the small example are the intercorrelations of eight intelligence test items. The items are coded in terms of the facets "Format = N(umerical), G(eometrical)" and "Requirement = A(pply), I(nfer)".

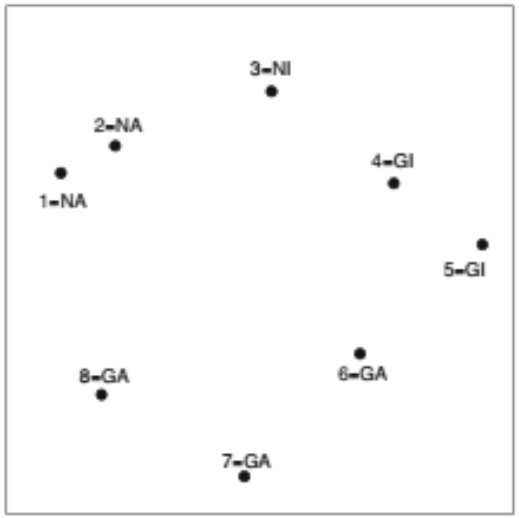
Testing structural hypotheses II

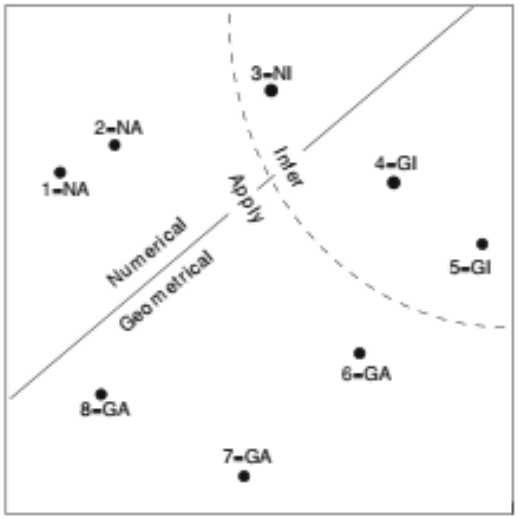
Table 2.4 Intercorrelations of eight intelligence test items, together with codings on two facets

Format	Requirement	Item	1	2	3	4	5	6	7	8
N	A	1	1.00	0.67	0.40	0.19	0.12	0.25	0.26	0.39
N	A	2	0.67	1.00	0.50	0.26	0.20	0.28	0.26	0.38
N	I	3	0.40	0.50	1.00	0.52	0.39	0.31	0.18	0.24
G	I	4	0.19	0.26	0.52	1.00	0.55	0.49	0.25	0.22
G	I	5	0.12	0.20	0.39	0.55	1.00	0.46	0.29	0.14
G	A	6	0.25	0.28	0.31	0.49	0.46	1.00	0.42	0.38
G	A	7	0.26	0.26	0.18	0.25	0.29	0.42	1.00	0.40
G	A	8	0.39	0.38	0.24	0.22	0.14	0.38	0.40	1.00

Do Format and Requirement surface in some way in the MDS solution?.

Testing structural hypotheses III





MDS I

MDS has been predominantly used as a tool for analyzing *proximity data* of all kinds (e.g., correlations, similarity ratings, co-occurrence data). It is a technique for analysis of data that are (dis)similarities observed on a set of objects.

MDS models such data as distances among points in a (reduced) geometrical space.

MDS II

Imagine to have m points defined in a space of dimension p. Goal of MDS is find a space $\mathbf{X}$ of dimension r ($r < p$) that represents the coordinates of the m points.

From $\mathbf{X}$ is possible compute the distance matrix among the m points in the r-dimensional space. The goal is find a configuration such that the distances d_{ij} in the reduced space are as close as possible to the raw distances-dissimilarities in the full p-dimensional space.

MDS by hand I

Suppose to observe the distances between 10 cities:

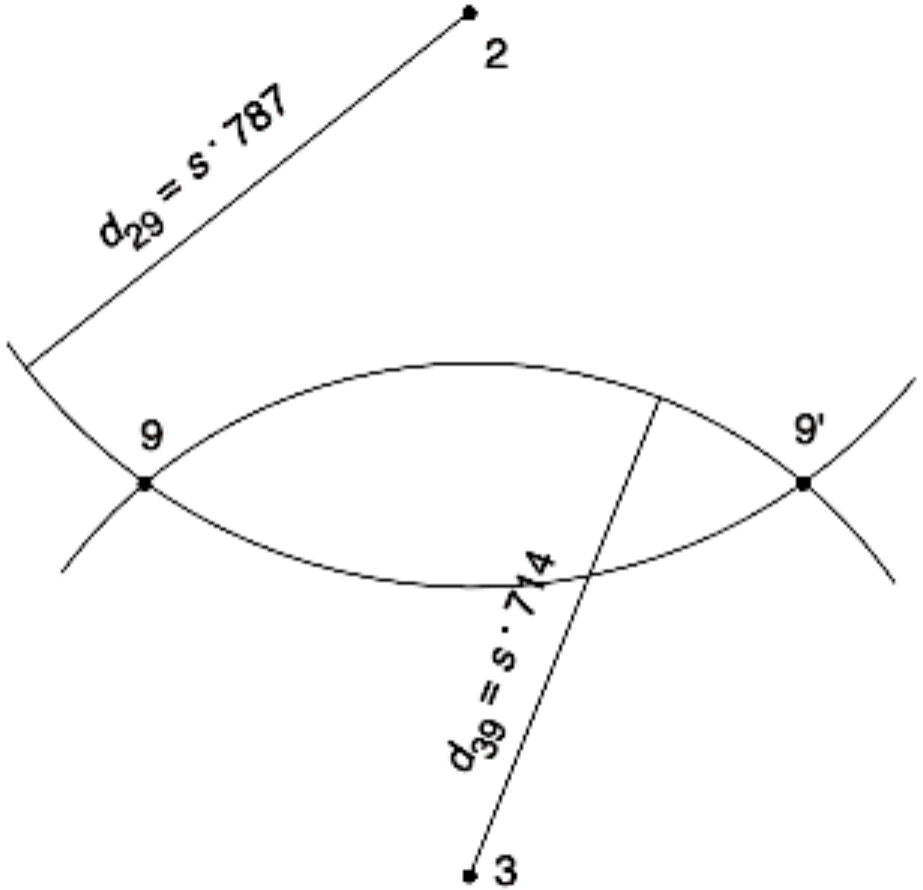
	1	2	3	4	5	6	7	8	9	10
1	0	569	667	530	141	140	357	396	570	190
2	569	0	1212	1043	617	446	325	423	787	648
3	667	1212	0	201	596	768	923	882	714	714
4	530	1043	201	0	431	608	740	690	516	622
5	141	617	596	431	0	177	340	337	436	320
6	140	446	768	608	177	0	218	272	519	302
7	357	325	923	740	340	218	0	114	472	514
8	396	423	882	690	337	272	114	0	364	573
9	569	787	714	516	436	519	472	364	0	755
10	190	648	714	622	320	302	514	573	755	0

MDS by hand II

Identify the maximum distance in the table, here $d23 = 1212$. Place two points on a piece of paper such that their distance is proportional to $d23$. Choose a scale factor s so that the reconstructed map has a convenient overall size. For example, we want the largest distance in the map to be equal to 5 cm, then $s = 0.004125$. All values are then multiplied by s, which leaves invariant the proportions or ratios of the data. Draw a line segment with a length $s \times 1212 = 5$.

MDS by hand III

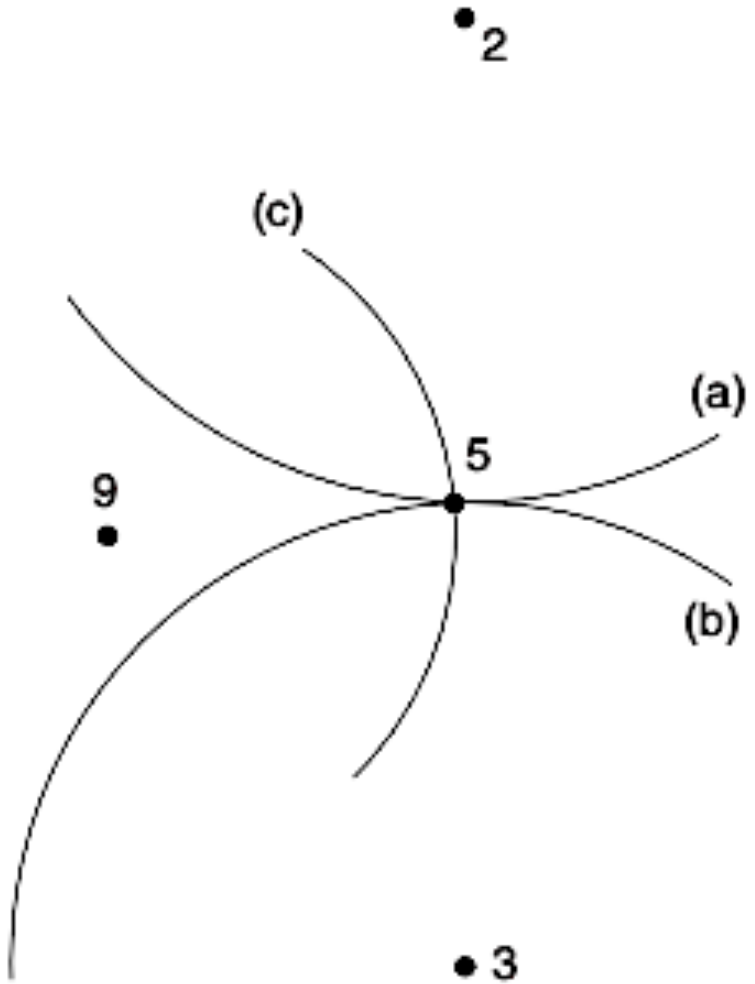
Assume now that we consider city 9: we know that $d29 = 787$, hence point 9 must lie anywhere on the circle with radius $s \times 787$ cm around point 2. On the other hand, $d39 = 714$, so point 9 also must lie on the circle with radius $s \times 714$ city 3. We have two possible solutions, between points 9 and 9' we choose the first.



MDS by hand IV

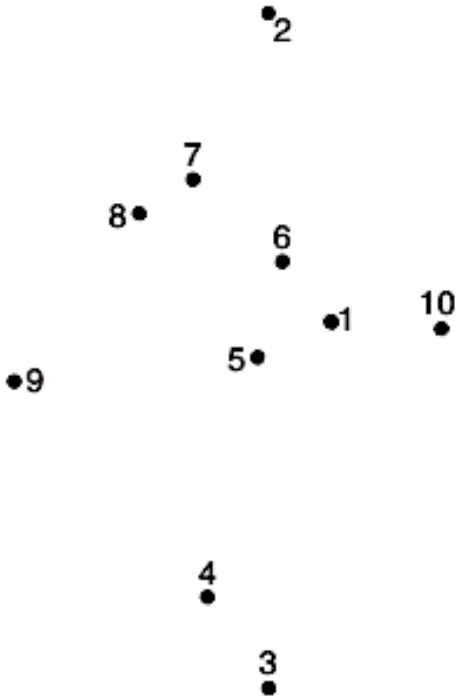
Pick now, say, city 5. Its position must lie on

- a) the circle around point 2 with radius $s \times 617(d25)$;
- b) the circle around point 3 with radius $s \times 596(d35)$;
- c) the circle around point 9 with radius $s \times 436(d95)$.



MDS by hand V

And so on, until all points have been placed



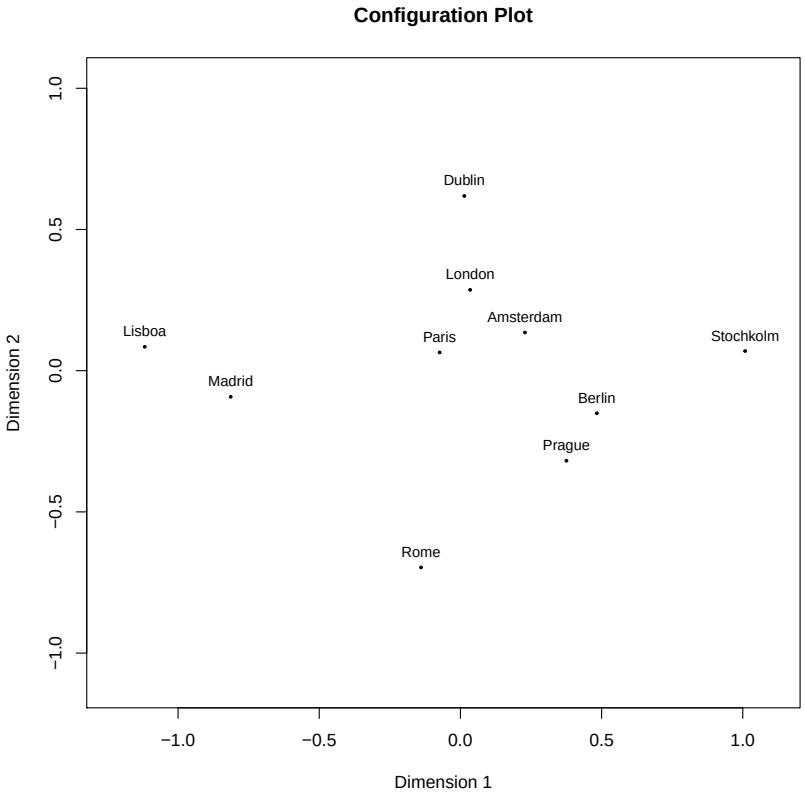
A classical example: European Cities I

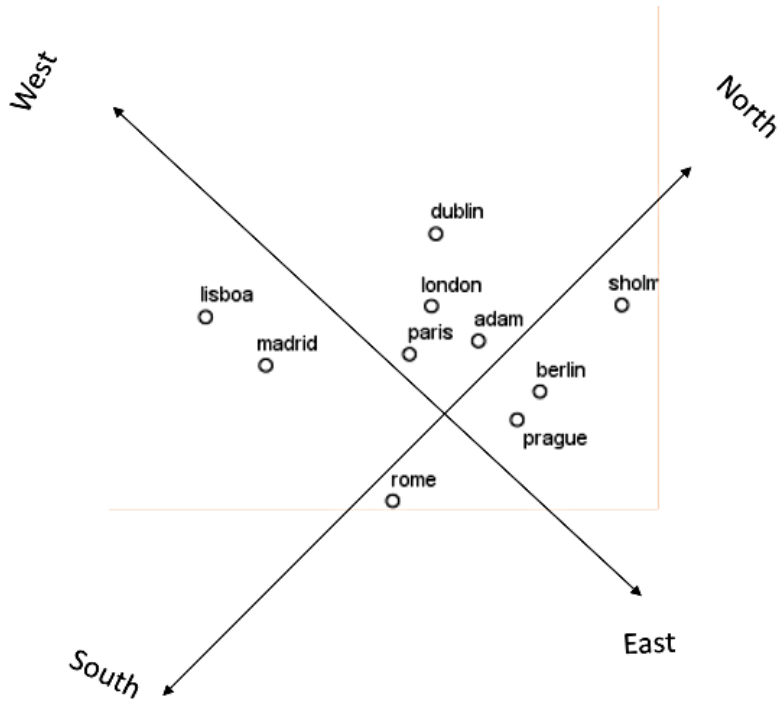
	London	Stockholm	Lisboa	Madrid	Paris	Amsterdam	Berlin	Prague	Rome	Dublin
London	0.00	569.00	667.00	530.00	141.00	140.00	357.00	396.00	569.00	190.00
Stockholm	569.00	0.00	1212.00	1043.00	617.00	446.00	325.00	423.00	787.00	648.00
Lisboa	667.00	1212.00	0.00	201.00	596.00	768.00	923.00	882.00	714.00	714.00
Madrid	530.00	1043.00	201.00	0.00	431.00	608.00	740.00	690.00	516.00	622.00
Paris	141.00	617.00	596.00	431.00	0.00	177.00	340.00	337.00	436.00	320.00
Amsterdam	140.00	446.00	768.00	608.00	177.00	0.00	218.00	272.00	519.00	302.00
Berlin	357.00	325.00	923.00	740.00	340.00	218.00	0.00	114.00	472.00	514.00
Prague	396.00	423.00	882.00	690.00	337.00	272.00	114.00	0.00	364.00	573.00
Rome	569.00	787.00	714.00	516.00	436.00	519.00	472.00	364.00	0.00	755.00
Dublin	190.00	648.00	714.00	622.00	320.00	302.00	514.00	573.00	755.00	0.00

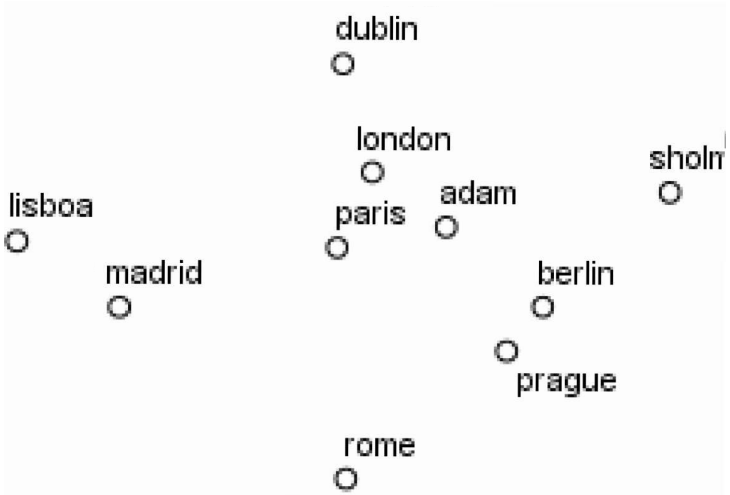
A classical example: European Cities II

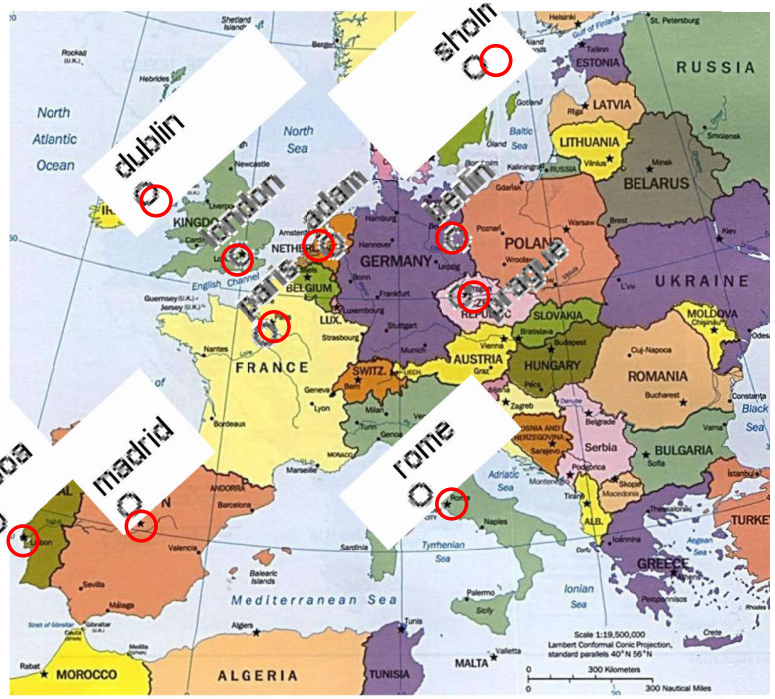
```
> X=as.data.frame(EUCITIES)
> row.names(X)=colnames(X)
> require(smacof)
> mdssol=mds(as.dist(X),type="ordinal")
> mdssol$conf[,2]=mdssol$conf[,2]*-1 #why?
```

A classical example: European Cities III






dublin
london
sholm
lisboa
paris
adam
madrid
berlin
prague
rome


sholm
dublin
london
adam
berlin
paris
prague
lisboa
madrid
rome
RUSSIA
ESTONIA
LATVIA
LITHUANIA
BELARUS
North Atlantic Ocean
North Sea
Baltic Sea
KINGDOM
NETHERL.
GERMANY
POLAND
BELGIUM
LUX.
FRANCE
SWITZ.
AUSTRIA
SLOVAKIA
HUNGARY
UKRAINE
MOLDOVA
ROMANIA
Serbia
BULGARIA
GREECE
TURKE
Mediterranean Sea
Adriatic Sea
Ionian Sea
Tyrrhenian Sea
MOROCCO
ALGERIA
TUNISIA
MALTA

Metric versus non-metric I

A MDS model is a proposition such that, given some proximities (p_{ij}) and a transformation of the proximities ($f(p_{ij})$), these become as close as possible to the distances among the points of the configuration $\mathbf{X}$

$$f(p_{ij}) = d_{ij}(\mathbf{X}).$$

The function p can be completely specified, or it can be assumed to belong to some class of functions. The choice of f specifies the MDS model.

Metric versus non-metric II

A main difference of various MDS models is the scale level that the models assume for the proximities. The most popular MDS model in research publications using MDS has been ordinal MDS, also known as non-metric MDS.

Mon-metric MDS builds on the premise that the proximities p_{ij} are on an ordinal scale: only their ranks are taken as reliable and valid information. The task of ordinal MDS is to generate a r-dimensional configuration $\mathbf{X}$ so that the distances over X are ordered as closely as possible as the proximities.

In other words, in ordinal MDS is required that

$$p_{ij} < p_{kl} \rightarrow d_{ij}(\mathbf{X}) < d_{kl}(\mathbf{X})$$

Metric versus non-metric III

A second class of MDS models is called metric MDS. Such models specify an analytic function for f.

$f(p_{ij}) =$

- absolute MDS: p_{ij};
- ratio MDS: $b \times p_{ij}$, with $b > 0$;
- interval MDS: $a + b \times p_{ij}$, with $a \geq 0$, $b \geq 0$;
- spline: polynomial sum of p_{ij}.

STRESS I

The loss function to be minimized, as well as the main badness of fit measure to be evaluated, is called STRESS. The single error is

$$e_{ij}^2 = [f(p_{ij}) - d_{ij}(\mathbf{X})]^2.$$

Summing over all the possible pairs a measure called *raw STRESS* is obtained:

$$\sigma_r(\mathbf{X}) = \sum_{i>j} e_{ij}^2.$$

STRESS II

The raw STRESS is generally normalized in this way:

$$\sigma_1(\mathbf{X}) = \frac{\sigma_r}{\sum_{i<j} d_{ij}^2(\mathbf{X})}.$$

A measure for evaluating the MDS solution is the Kruskal's STRESS-1:

$$\text{STRESS-1} = \sqrt{\sigma_1(\mathbf{X})}$$

MDS algorithms: classical scaling

Classical scaling:

- Compute the squared dissimilarities matrix $\mathbf{\Delta}^2$;
- Apply the double centering operation to the matrix:

$$\mathbf{D}_{\mathbf{\Delta}} = -\frac{1}{2}\mathbf{J}\mathbf{\Delta}^2\mathbf{J},$$

  with $\mathbf{J} = \mathbf{I}n^{-1}\mathbf{1}\mathbf{1}'$;
- compute eigenvalues ($\mathbf{\Lambda}$) and eigenvectors ($\mathbf{Q}$) of the matrix $\mathbf{D}_{\mathbf{\Delta}}$;
- The coordinates are given by $\mathbf{X} = \mathbf{Q}_+\sqrt{\mathbf{\Lambda}_+}$, in which the subscript $+$ indicates the first r positive eigenvalues ordered in descending order.

…Classical scaling (Torgerson-Gower Scaling)…

1. Compute the squared dissimilarities matrix:

$$\Delta = \begin{bmatrix} 0 & 4{,}05 & 8{,}25 & 5{,}57 \\ 4{,}05 & 0 & 2{,}54 & 2{,}69 \\ 8{,}25 & 2{,}54 & 0 & 2{,}11 \\ 5{,}57 & 2{,}69 & 2{,}11 & 0 \end{bmatrix} \qquad \Delta^{(2)} = \begin{bmatrix} 0 & 16{,}403 & 68{,}063 & 31{,}025 \\ 16{,}403 & 0 & 6{,}4516 & 7{,}2361 \\ 68{,}063 & 6{,}4516 & 0 & 4{,}4521 \\ 31{,}025 & 7{,}2361 & 4{,}4521 & 0 \end{bmatrix}$$

1.1 Compute the matrix J:

$$J = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \frac{1}{4}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0{,}75 & -0{,}25 & -0{,}25 & -0{,}25 \\ -0{,}25 & 0{,}75 & -0{,}25 & -0{,}25 \\ -0{,}25 & -0{,}25 & 0{,}75 & -0{,}25 \\ -0{,}25 & -0{,}25 & -0{,}25 & 0{,}75 \end{bmatrix}$$

2. Apply double centering:

$$B = -\frac{1}{2}\begin{bmatrix} 0{,}75 & -0{,}25 & -0{,}25 & -0{,}25 \\ -0{,}25 & 0{,}75 & -0{,}25 & -0{,}25 \\ -0{,}25 & -0{,}25 & 0{,}75 & -0{,}25 \\ -0{,}25 & -0{,}25 & -0{,}25 & 0{,}75 \end{bmatrix}\begin{bmatrix} 0 & 16{,}40 & 68{,}06 & 31{,}02 \\ 16{,}40 & 0 & 6{,}45 & 7{,}23 \\ 68{,}06 & 6{,}45 & 0 & 4{,}45 \\ 31{,}02 & 7{,}23 & 4{,}45 & 0 \end{bmatrix}\begin{bmatrix} 0{,}75 & -0{,}25 & -0{,}25 & -0{,}25 \\ -0{,}25 & 0{,}75 & -0{,}25 & -0{,}25 \\ -0{,}25 & -0{,}25 & 0{,}75 & -0{,}25 \\ -0{,}25 & -0{,}25 & -0{,}25 & 0{,}75 \end{bmatrix} =$$

$$= \begin{bmatrix} 20{,}521 & 1{,}6444 & -18{,}076 & -4{,}0889 \\ 1{,}6444 & -0{,}82931 & 2{,}0544 & -2{,}8695 \\ -18{,}076 & 2{,}0544 & 11{,}39 & 4{,}632 \\ -4{,}0889 & -2{,}8695 & 4{,}632 & 2{,}3264 \end{bmatrix}$$

…Classical scaling (Torgerson-Gower Scaling)…

3. Compute the eigendecomposition:

$\Lambda =$

35,713	0	0	0
0	-5,5704	0	0
0	0	-2,68E-15	0
0	0	0	3,2653

$Q =$

-0,77325	-0,38786	-0,5	0,040543
-0,014787	0,60978	-0,5	-0,61478
0,60775	-0,58694	-0,5	-0,19013
0,18029	0,36502	-0,5	0,76437

3. 1 Select eigenvalues and eigenvectors

$\sqrt{\Lambda_+} =$

5,976035	0
0	1,807014

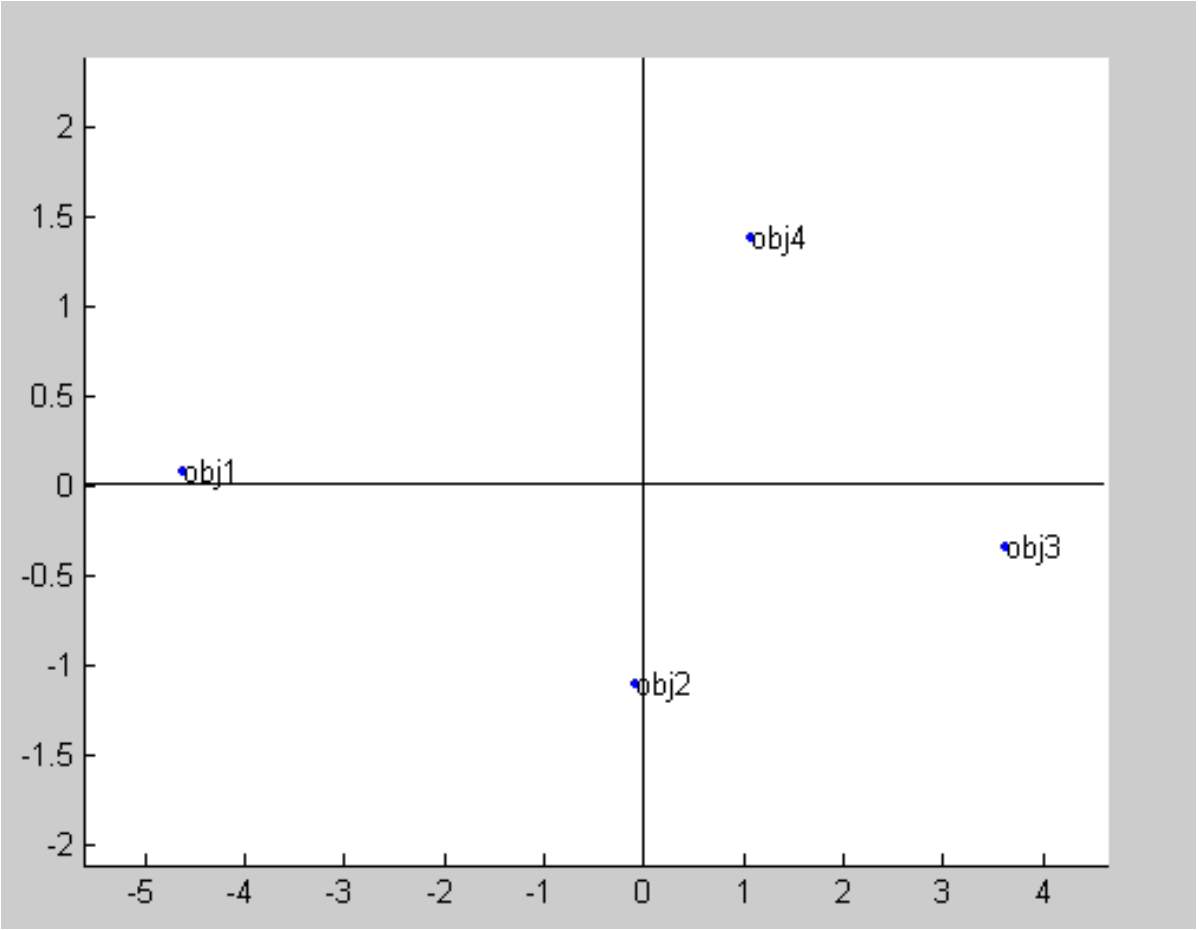
$Q_+ =$

-0,77325	0,040543
-0,014787	-0,61478
0,60775	-0,19013
0,18029	0,76437

4. Find the coordinates:

$$X = Q_+\sqrt{\Lambda_+} = \begin{bmatrix} -0,77 & 0,04 \\ -0,01 & -0,61 \\ 0,60 & -0,19 \\ 0,18 & 0,76 \end{bmatrix} \begin{bmatrix} 5,97 & 0 \\ 0 & 1,81 \end{bmatrix} =$$

-4,621	0,073262
-0,088367	-1,1109
3,6319	-0,34357
1,0774	1,3812


obj4
obj1
obj3
obj2
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-5
-4
-3
-2
-1
0
1
2
3
4

Gradient-based MDS I

In the following, some algorithms for *unweighted* MDS are presented. Often (always?) the classical scaling algorithm is the starting configuration for other algorithms. If the STRESS is minimized, a popular one is the gradient-based MDS (implemented, for example, in MatLab and STATISTICA).
Let the STRESS be defined as

$$\begin{aligned}\sigma &= \sum_{i<j} (\delta_{ij} - d_{ij}(\mathbf{X}))^2 \\ &= \sum_{i<j} \delta_{ij}^2 + \sum_{i<j} d_{ij}^2(\mathbf{X}) - 2\sum_{i<j} \delta_{ij} d_{ij}(\mathbf{X})\end{aligned}$$

Gradient-based MDS II

The second term $\sum_{i<j} d_{ij}^2(\mathbf{X})$ can be written as $trace(\mathbf{XVX}^T)$, where $\mathbf{V}$ is a matrix with elements $v_{ij} = -1$ if $i \neq j$ and $N-1$ otherwise and $\mathbf{X}$ is the $N \times m$ matrix of coordinates.

The third term can be written as $\sum_{i<j} \delta_{ij} d_{ij}(\mathbf{X}) = trace(\mathbf{X}^T \mathbf{B}(X)\mathbf{X})$, where $\mathbf{B}(X)$ is a matrix with elements $b_{ij}(X) = \frac{-d_{ij}(\mathbf{X})}{d_{ij}}$ if $i \neq j$ and $d_{ij} \neq 0$, 0 if $i \neq j$ and $d_{ij} = 0$, and $-\sum_{i \neq j} b_{ij}$.

Gradient-based MDS III

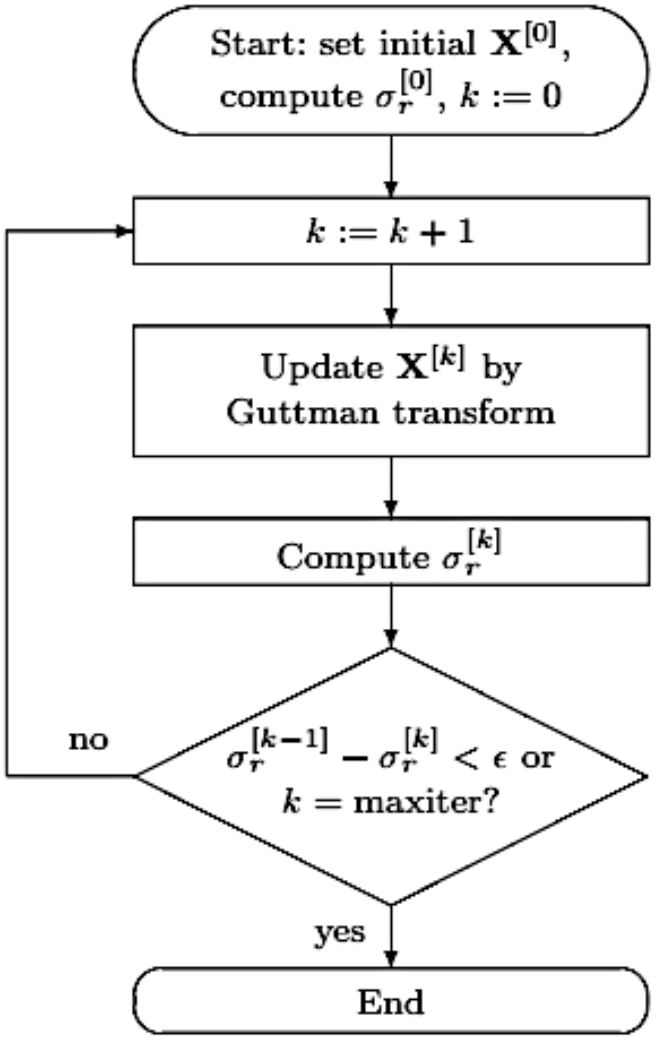
Hence, as the gradient of the STRESS is equal to $\nabla\sigma = 2\mathbf{V}\mathbf{X} - 2\mathbf{B}(X)\mathbf{X}$, gradient methods for minimizing the so-called *Least Squares STRESS* work by updating $\mathbf{X}$ in this way:

$$\mathbf{X}^{k+1} = \mathbf{X}^{k} - 2\alpha^{k}\left(\mathbf{V}\mathbf{X}^{k} - \mathbf{B}(X^{k})\mathbf{X}^{k}\right)$$

SMACOF I

The SMACOF (Scaling by Majorizing A COmplicated Function) algorithm is implemented in SPSS (as PROXSCAL mds) and in R. It works as in the following flow chart

SMACOF II


Start: set initial $\mathbf{X}^{[0]}$, compute $\sigma_r^{[0]}$, $k := 0$
$k := k + 1$
Update $\mathbf{X}^{[k]}$ by Guttman transform
Compute $\sigma_r^{[k]}$
$\sigma_r^{[k-1]} - \sigma_r^{[k]} < \epsilon$ or $k =$ maxiter?
no
yes
End

SMACOF III

By Cauchy-Schwartz inequality we can write

$$\sum_{k=1}^{m}(x_i^{(k)}-x_j^{(k)})(q_i^{(k)}-q_j^{(k)}) \leq \left(\sum_{k=1}^{m}(x_i^{(k)}-x_j^{(k)})^2\right)^{1/2}\left(\sum_{k=1}^{m}(q_i^{(k)}-q_j^{(k)})^2\right)^{1/2},$$

hence the third term in STRESS function can be written as $trace(\mathbf{X}^T\mathbf{B}(X)\mathbf{X}) = trace(\mathbf{X}^T\mathbf{B}(Q)\mathbf{Q})$.
The step $\mathbf{X}^{k+1} = N^{-1}\mathbf{B}(Q)\mathbf{Q}$ is called "Gutman transform". The SMACOF algorithm ensures the use of quadratic function to be minimized. It is generally faster than gradient methods.

Unfolding

Introduction I

"*The unfolding model is a model for preferential choice. It assumes that different individuals perceive various objects of choice in the same way but differ with respect to what they consider an ideal combination of the objects' attributes. In unfolding, the data are usually preference scores (such as rank-orders of preference) of different individuals for a set of choice objects. These data can be conceived as proximities between the elements of two sets, individuals and choice objects. Technically, unfolding can be seen as a special case of MDS where the within-sets proximities are missing*" (Borg and Groenen, 2005, pag. 293)

Introduction II

Unfolding can be seen as a special case of Multidimensional Scaling (MDS).

Unfolding: loss function

The most accepted formulation of the problem in terms of a badness-of-fit function is given in a least squares framework by minimization of the STRESS function (?):

$$\sigma^2(A, B, \hat{\Delta}) = \sum_{i=1}^{m}\sum_{j=1}^{n}(\hat{\delta}_{ij} - d_{ij})^2, \text{ where:}$$

- $\hat{\Delta}$ is a $m \times n$ matrix in which each entry $\hat{\delta}_{ij}$ represents the disparity or monotonically transformed dissimilarity between the *ith*-subject and the *jth*-object;
- $d_{ij} = d_{ij}(A, B)$ represents the distance, usually Euclidean, between the corresponding points in dimension K, $i = 1, \ldots, m$, $j = 1 \ldots, n$.

Unfolding I

Unfolding is typically performed when the are preference data.

Individuals are represented as *ideal points* in the MDS space so that the distances from each ideal point to the object points correspond to the preference scores.

Two blocks of dissimilarities are missing

Unfolding II

		Parties				Persons				
		A	B	C	D	1	2	3	4	5
Parties	A	-	-	-	-	1	4	4	4	3
	B	-	-	-	-	3	2	1	1	2
	C	-	-	-	-	2	3	2	3	1
	D	-	-	-	-	4	1	3	2	4
Persons	1	1	3	2	4	-	-	-	-	-
	2	4	2	3	1	-	-	-	-	-
	3	4	1	2	3	-	-	-	-	-
	4	4	1	3	2	-	-	-	-	-
	5	3	2	1	4	-	-	-	-	-

Fictitious preference values of 5 persons for 4 parties

table from Borg, Groenen and Mair, 2013

Unfolding III

There are both metric and non-metric unfolding models.

Most of the time (quite always?) one is interested in non-metric unfolding. The used transformation is usually the *monotone regression*.

There are two approaches to *ties*:

1. Primary approach: allow ties to be broken;
2. Secondary approach: do not allow ties to be broken.

Unfolding IV

From a geometric point-of-view, unfolding can easily lead to unstable solutions.

This is so because the model rests on data that constrain only a sub-set of the distances, namely the distances among ideal points and object points, but not the distances among ideal points and also not the distances among object points.

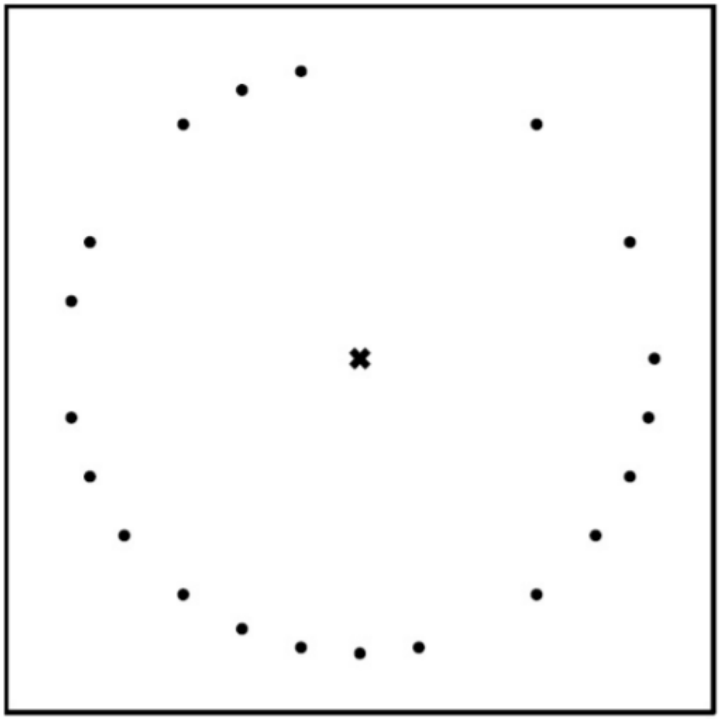
Moreover, in real data, object points and ideal points are often not thoroughly mixed. In the Unfolding solution single points can be moved around arbitrarily in ample solution regions without affecting the Stress at all.

Problems with unfolding I

A common problem is the risk to obtain *degenerate solutions*. Such undesirable solutions are sometimes easily recognized, for example, if the person points are all located on a circular arc, while the object points are lumped together in the center of the circle.

In this case the Unfolding solution is quite perfect (i.e., $\mathrm{STRESS} \approx 0$), but the graphical solution is absolutely uninformative.

Problems with unfolding II



Problems with Unfolding: degenerate solutions I

Strategies to avoid degenerate solutions:

- Modify the STRESS function (i.e., STRESS-2, obtained by dividing σ_r by $\sum_{i<j}(d_{ij}(\mathbf{X}) - \bar{d})^2$
- ALSCAL minimizes the so-called *S-STRESS*:

$$\sigma^2 = \sum_{i<j}(\hat{\delta}^2_i j - d^2_{ij})^2$$

Problems with Unfolding: degenerate solutions II

- Penalization of the loss function (Implemented in SPSS-PREFSCAL. Today it is also in R, package SMACOF);

$$\sigma_p = \sigma^2(A, B, \hat{\Delta})^{\lambda} \left(1 + \frac{w}{v^2(\hat{\Delta})}\right), \text{ where:}$$

  - λ and w are the two penalty parameters fixed by the user;
  - $v^2(\hat{\Delta})$ denotes the (squared) coefficient of variation for the elements of $\hat{\Delta}$.
- "Augmenting" the data matrix.

Problems with Unfolding: degenerate solutions III

The SMACOF algorithm for rectangular tables (namely, the Unfolding) is implemented in both R (smacof package) and SPSS.
ALSCAL is implemented in SPSS
Note that the *smacof* package provides both metric (ratio transformation with coefficient equal to 1) and non-metric unfolding (PRESCAL algorithm).

Problems with Unfolding: degenerate solutions IV

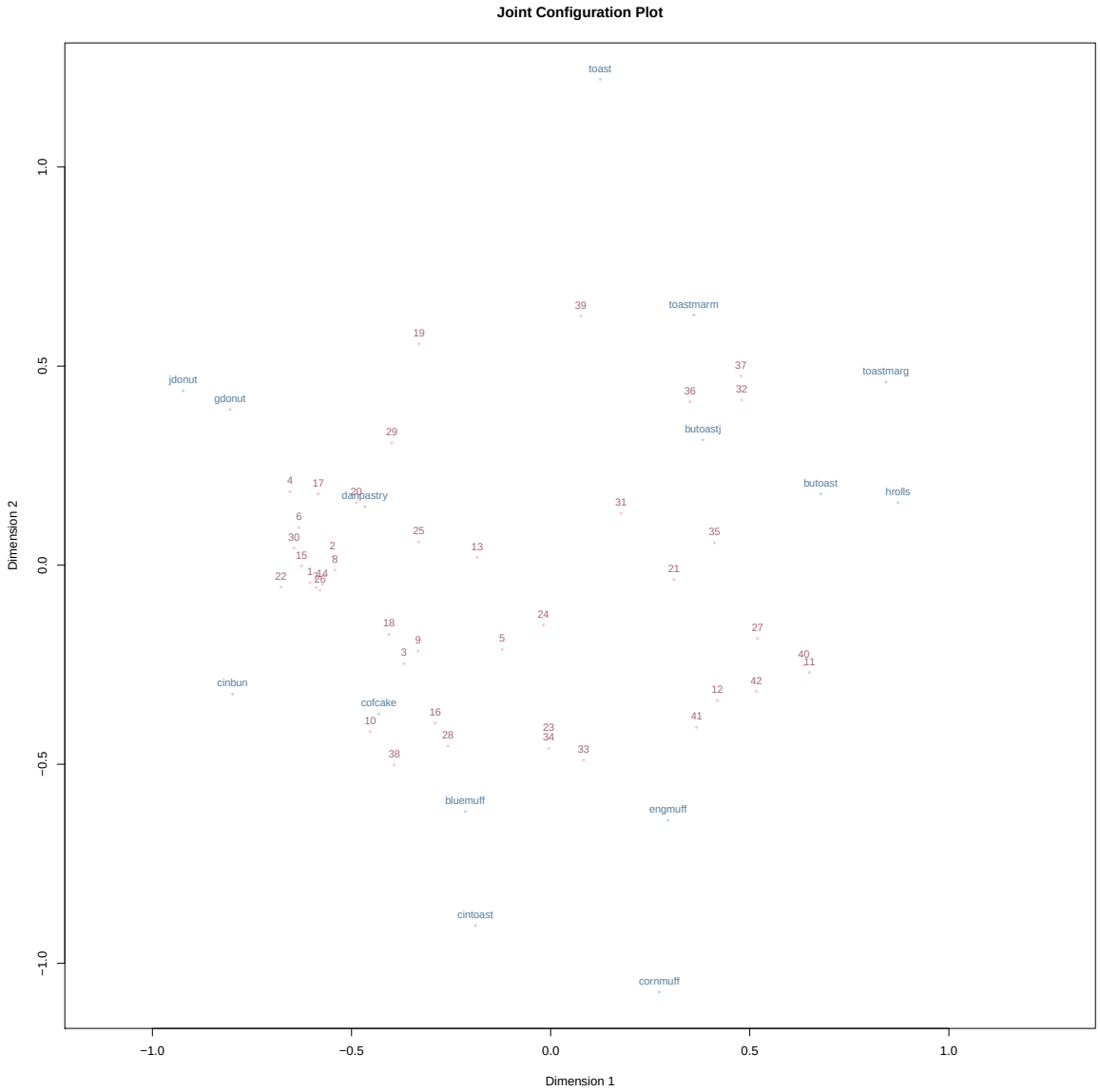
Augmenting the data matrix allows, very often, to use any "standard" MDS algorithm in order to perform unfolding. New Augmented data matrix based on Kemeny distance will be available soon

Unfolding (and, generally, MDS) is closely related to cluster analysis.

Unfolding: breakfast data

	toast	butoast	engmuff	jdonut	cintoast	bluemuff	hrolls	toastmarm	butoastj	toastmarg	cinbun	danpastry	gdonut	cofcake	cornmuff
1	13	12	7	3	5	4	8	11	10	15	2	1	6	9	14
2	15	11	6	3	10	5	14	8	9	12	7	1	4	2	13
3	15	10	12	14	3	2	9	8	7	11	1	6	4	5	13
4	6	14	11	3	7	8	12	10	9	15	4	1	2	5	13
5	15	9	6	14	13	2	12	8	7	10	11	1	4	3	5
6	9	11	14	4	7	6	15	10	8	12	5	2	3	1	13
7	9	14	5	6	8	4	13	11	12	15	7	2	1	3	10
8	15	10	12	6	9	2	13	8	7	11	3	1	5	4	14
9	15	12	2	4	5	8	10	11	3	13	7	9	6	1	14
10	15	13	10	7	6	4	9	12	11	14	5	2	8	1	3
11	9	2	4	15	8	5	1	10	6	7	11	13	14	12	3
12	11	1	2	15	12	3	4	8	7	14	10	9	13	5	6
13	12	1	14	4	5	6	11	13	2	15	10	3	9	8	7
14	13	11	14	5	4	12	10	8	7	15	3	2	6	1	9
15	12	11	8	1	4	7	14	10	9	13	5	2	6	3	15
16	15	12	4	14	5	3	11	9	7	13	6	8	1	2	10
17	7	10	8	3	13	6	15	12	11	9	5	1	4	2	14
18	7	12	6	4	10	1	15	9	8	13	5	3	14	2	11
19	2	9	8	5	15	12	7	10	6	11	1	3	4	13	14
20	10	11	15	6	9	4	14	2	13	12	8	1	3	7	5
21	12	1	2	10	3	15	5	6	4	13	7	11	8	9	14
22	14	12	10	1	11	5	15	8	7	13	2	6	4	3	9
23	14	6	1	13	2	5	15	8	4	12	7	10	9	3	11
24	10	11	9	15	5	6	12	1	3	13	8	2	14	4	7
25	15	8	7	5	9	10	13	3	11	6	2	1	12	4	14
26	15	13	8	5	10	7	14	12	11	6	4	1	3	2	9
27	11	3	6	14	1	7	9	4	2	5	10	15	13	12	8
28	6	15	3	11	8	2	13	9	10	14	5	7	12	1	4
29	15	7	10	2	12	9	13	8	5	6	11	1	3	4	14
30	15	10	7	2	9	6	14	12	8	11	5	3	1	4	13
31	11	4	9	10	15	8	6	5	1	13	14	2	12	3	7
32	9	3	10	13	14	11	1	2	4	5	15	6	7	8	12
33	15	8	1	11	10	2	4	13	14	9	6	5	12	3	7
34	15	8	3	11	10	2	4	13	14	9	6	5	12	1	7
35	15	6	10	14	12	8	2	4	3	5	11	1	13	7	9
36	12	2	13	11	9	15	3	1	4	5	6	8	10	7	14
37	5	1	6	11	12	10	7	4	3	2	13	9	8	14	15
38	15	11	7	13	4	6	9	14	8	12	1	10	3	2	5
39	6	1	12	5	15	9	2	7	11	3	8	10	4	14	13
40	14	1	5	15	4	6	3	8	9	2	12	11	13	10	7
41	10	3	2	14	9	1	8	12	13	4	11	5	15	6	7
42	13	3	1	14	4	10	5	15	6	2	11	7	12	8	9

```
> res=unfolding(breakfast,type="ordinal"
+                    ,omega=0.5,lambda=1)
> res$stress
[1] 0.2947333
```


Joint Configuration Plot
toast
toastmarm
toastmarg
jdonut
gdonut
butoastj
butoast
hrolls
dampastry
cinbun
cofcake
bluemuff
engmuff
cintoast
cornmuff
Dimension 1
Dimension 2
−1.0
−0.5
0.0
0.5
1.0

Unfolding by augmenting data matrix: Spearman ρ distance

Van Deun, Heiser and Delbeke proposed to augment the data matrix in order to obtain and entire dissimilarity matrix to be analyzed with a standard MDS program.

They used the Spearman ρ distance for rankings

Recall: Spearman distance

Let A and B be two rankings of n elements. A distance defined as

$$d(A, B) = \sum_{i=1}^{n} |A_i - B_i|^p$$

for some $0 < p < \infty$ is a set of spatial distances for rankings.

If $p = 1$ the distance d is known as *Spearman's footrule* distance.

If $p = 2$, d is known as *Spearman* distance, and it is the distance measure related to the well-known Spearman's ρ.

Unfolding by augmenting data matrix: Spearman ρ distance (cont'd) I

Let $\mathbf{X}$ a $m \times n$ data matrix in which there are m individuals and n items to be ranked.

- Add n additional rows to $\mathbf{X}$. These rows represent the items by tied rankings in this way: $\pi_{kk'} = 1$ if $k = k'$, $\pi_{kk'} = 1 + n/2$ if $k \neq k'$, $k, k' = 1, \ldots, n$;
- center the data with respect to $\mathbf{c} = [(n+1)/2]\mathbf{1}$;
- Scale the object vectors by multiplying the additional rows by $((n+1)^{1/2})/3^{1/2}$;

Unfolding by augmenting data matrix: Spearman ρ distance (cont'd) II

- derive the $(m+n) \times (m+n)$ dissimilarity matrix by computing the squared Euclidean distance between the centered and scaled rank vectors (the length is set equal to $(\sqrt{n}\sqrt{(n^2-1)})/\sqrt{12}$);
- perform a non-metric MDS

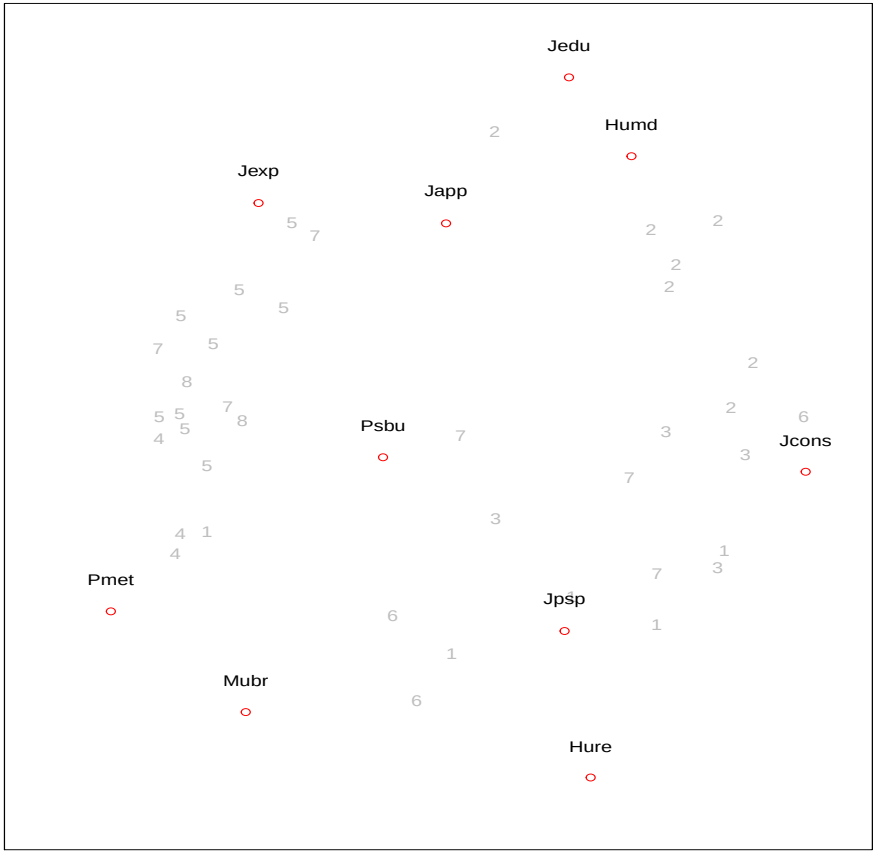
```
> VanDeunTransf = function(X){
+   #Given a rectangualr matrix X of dimension n x m
+   #containing preferences, transform the matrix
+   #into a dissimilarity matrix (n+m) x (n+m)
+   #according the approach by Van Deun,
+   #Heiser and Delbeke (2007)
+   #
+   #Output: Delta, to be use as input in MDS;
+   #     Interactions, for evaluation purposes
+
+   n=dim(X)[1]
+   m=dim(X)[2]
+   nr=row.names(X)
+   if (is.null(nr)){nr=seq(1:n)}
+   nc=colnames(X)
+
```

```
+    C=matrix((m+1)/2,(n+m),m) #center of
+                              #geometrical space
+
+    # step 1: create the objects vectors
+    obj = matrix(0,m,m)
+    for (j in 1:m){
+      obj[j,j]=1
+      obj[j, setdiff(c(1:m),j)]=1+m/2
+    }
+
+    X=rbind(as.matrix(X),obj) #add the n row to X
+    row.names(X)=c(nr,nc)
+
+    #step 2: center the matric with trespect to C
+    Xc=X-C
+
```

```
+   #step 3: Scale the object vectors
+   scalefact=sqrt((m+1))/sqrt(3)
+
+   Xc[(n+1):(n+m),] =
+     Xc[(n+1):(n+m),]*scalefact
+
+   #step4: compute the squared
+   #Euclidean distance among
+   #the rows of the matrix
+
+   XcS=matrix(0,(n+m),m)
+   scalefact2=sqrt((m*(2*m+1)*(m+1))/6)
+
+   for (j in 1:(n+m)){
+     XcS[j,]=scalefact2*Xc[j,]/
+       norm(matrix(Xc[j,]),type="f")
```

```
+    }
+    rownames(XcS)=rownames(X)
+
+    Delta = dist(XcS)^2
+    D=as.matrix(Delta)
+    return(list(Delta=D,
+                Interactions=D[(1:n),(n+1):(n+m)]))
+ }
```

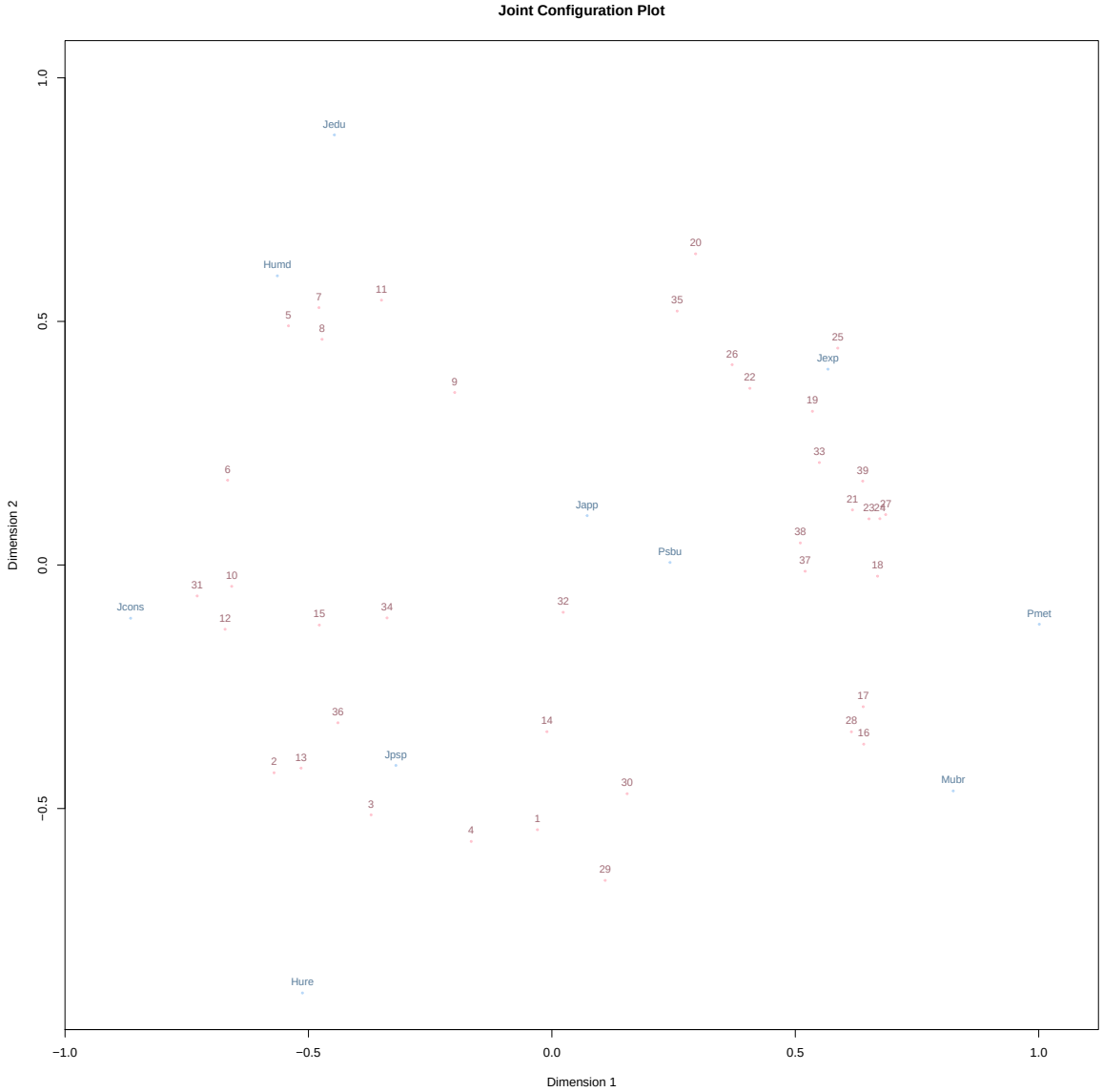
Augmented unfolding solution: Roskam data



Smacof unfolding solution: Roskam data I

```
> load("Roskam.rd")
> res2=unfolding(RoskamGifi,type="ordinal"
+                ,omega=0.5,lambda=1)
> res2$stress
[1] 0.2239048
```

Smacof unfolding solution: Roskam data II



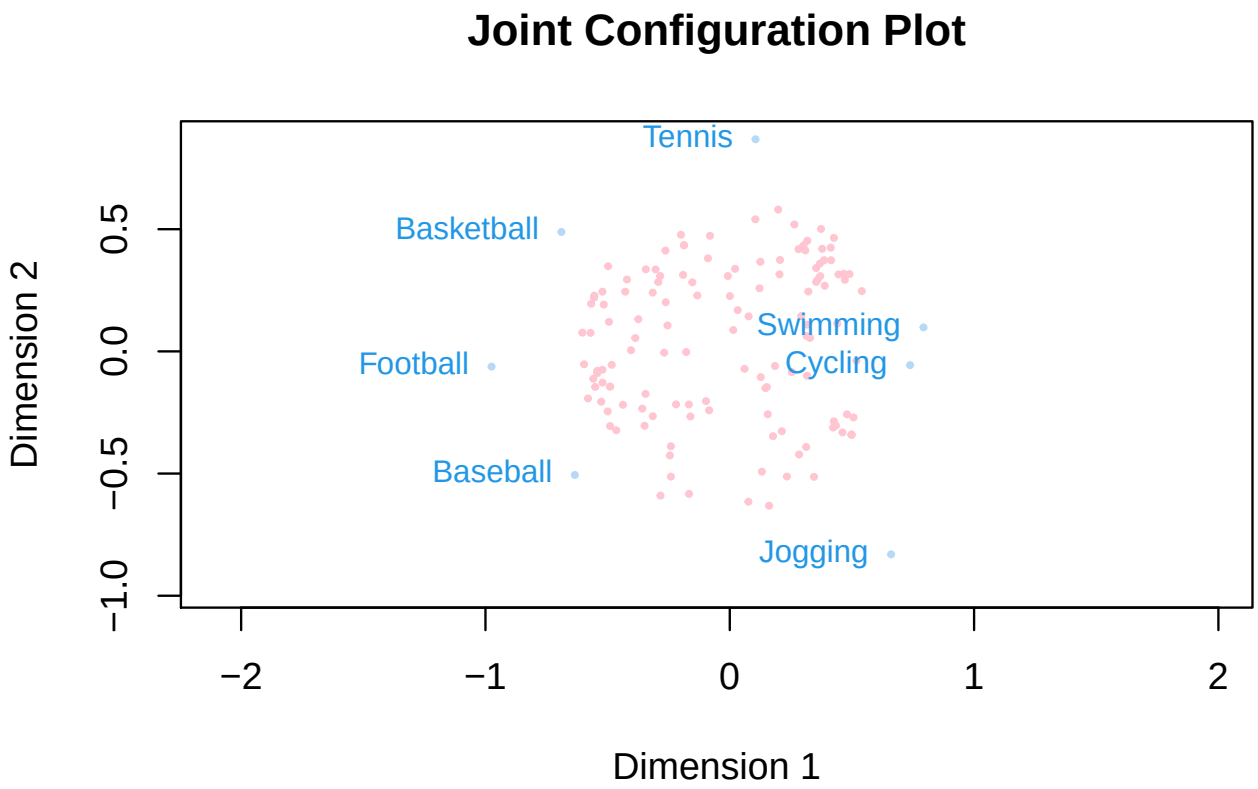
Clustering and Unfolding I

Some example on the "sports" data set: *130 students at the University of Illinois ranked seven sports according to their preference (Baseball, Football, Basketball, Tennis, Cycling, Swimming, Jogging).*

```
> require(ConsRank)
> data(sports)
> unfsol2=prefscal(sports,lambda=0.1,omega=1)
```

What we can conclude from the Unfolding solution?

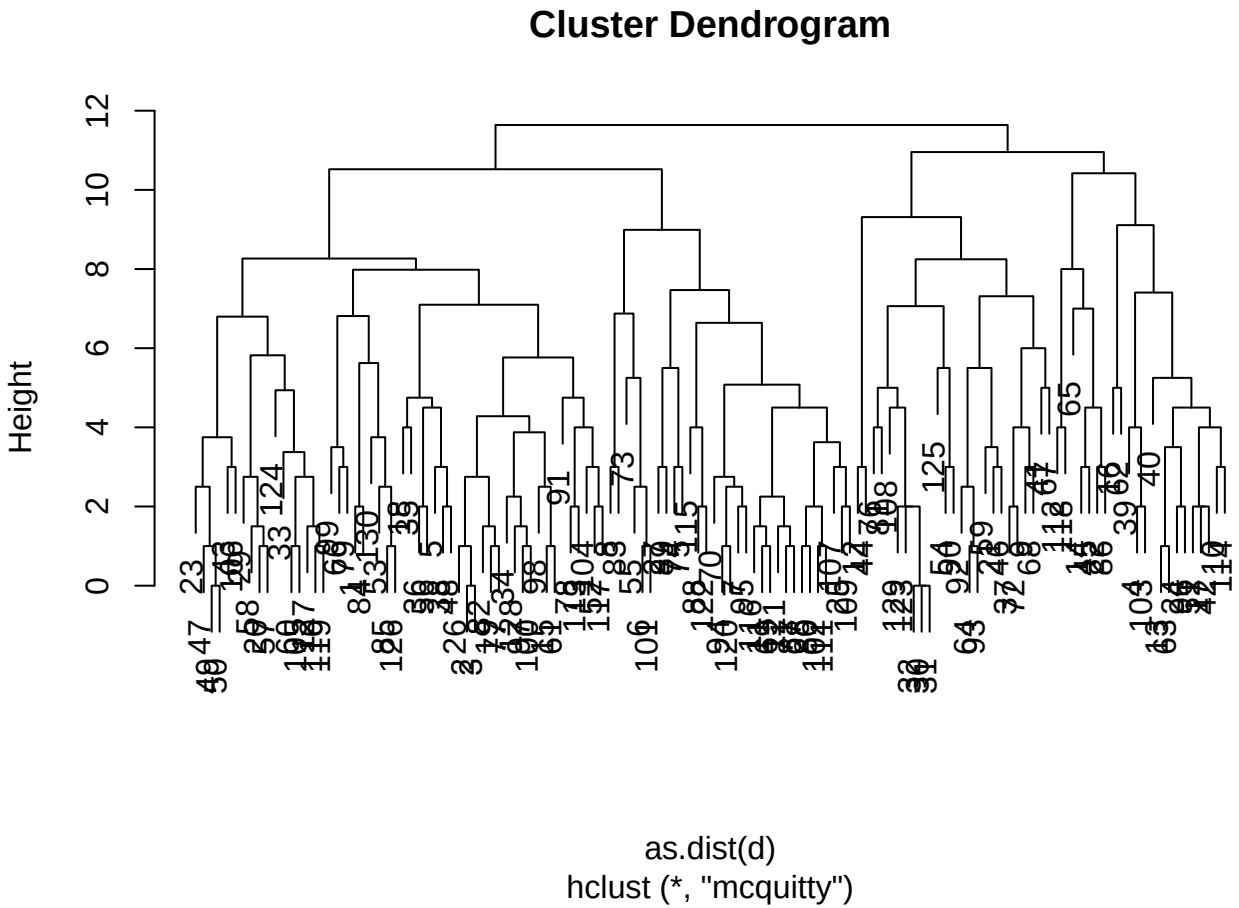
Clustering and Unfolding II



Hierarchical cluster analysis

```
> #1. compute Kemeny distance
> d=kemenyd(sports)
> #2. Convert Kemeny distance into Kendall distance
> d=d/2
> #3. hierarchical clustering with the
>   # weighted average method
> hc=hclust(as.dist(d),method="mcquitty")
```

Let's give a look at the *dendrogram*


Cluster Dendrogram
Height
0
2
4
6
8
10
12
as.dist(d)
hclust (*, "mcquitty")

```
> #1. cut te dendrogram with 4 clusters
> chc=cutree(hc,k=4)
> #2. compute the median ranking within
> #   each cluster
> CR1=consrank(sports[which(chc==1),],ps=FALSE)
> CR2=consrank(sports[which(chc==2),],ps=FALSE)
> CR3=consrank(sports[which(chc==3),],ps=FALSE)
> CR4=consrank(sports[which(chc==4),],ps=FALSE)
> # now let's see the clusters composition
```

Cluster #	Ordering	Averaged τ_X	Size
1	Baseball Football Basketball Tennis Cycling Swimming Jogging	0.556	52
2	Tennis Swimming Cycling Basketball Jogging Football Baseball	0.398	22
3	Swimming Cycling Basketball Tennis Baseball Football Jogging	0.588	31
4	Jogging Swimming Cycling Tennis Baseball Basketball Football	0.523	25

Recall: Kemeny distance

Let A and B two rankings of n elements. Transform the two rankings into two transformed vectors A^* and B^* of dimension $n(n-1)/2$ in which, for each pair i, j, $i < j$:

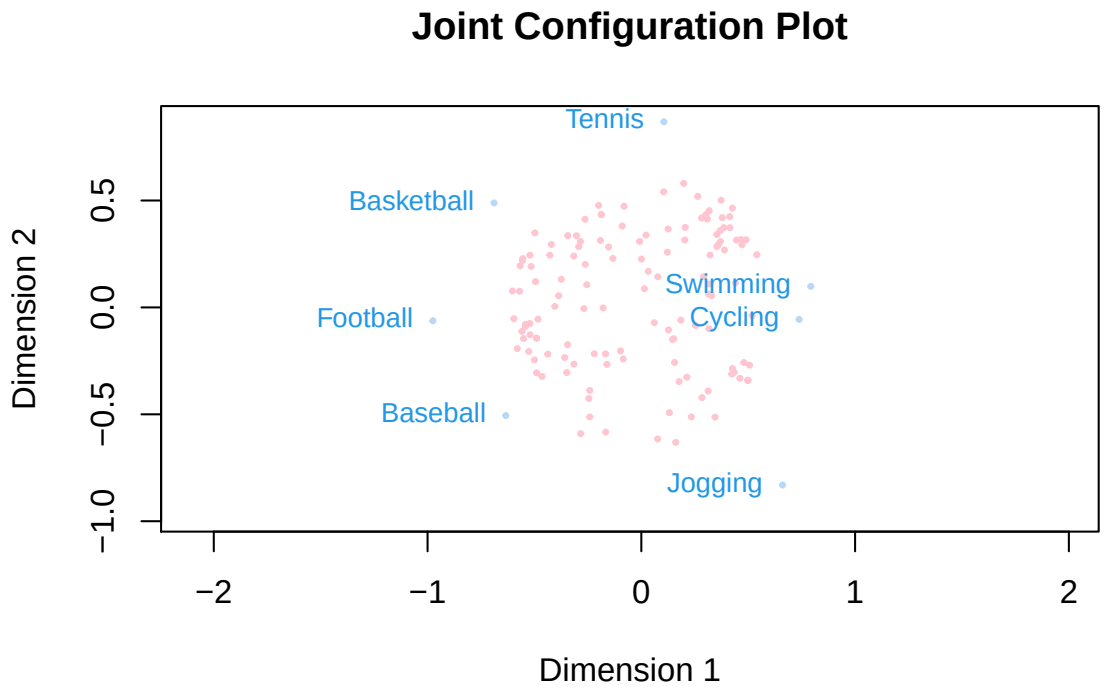
$$
\begin{aligned}
&= 1 \text{ if the i-th item is preferred over the j-th item} \\
A^*(B^*) &= -1 \text{ if the j-th item is preferred over the i-th item} \\
&= 0 \text{ if they are in a tie.}
\end{aligned}
$$

Then, the *Kemeny* distance is defined as

$$d(A, B) = \sum_{i=1}^{n(n-1)/2} |A_i^* - B_i^*|.$$

Clustering and Unfolding

Cluster #	Ordering	Averaged τ_X	Size
1	Baseball Football Basketball Tennis Cycling Swimming Jogging	0.556	52
2	Tennis Swimming Cycling Basketball Jogging Football Baseball	0.398	22
3	Swimming Cycling Basketball Tennis Baseball Football Jogging	0.588	31
4	Jogging Swimming Cycling Tennis Baseball Basketball Football	0.523	25



Bibliography I

Cox, T., and Cox, M. (2000). *Multidimensional Scaling* (2nd ed.). Chapman and Hall/CRC.

Borg, I., and Groenen, P. J. F. (2005). *Modern multidimensional scaling: Theory and applications* (2nd ed.). Springer.

Borg, I., Groenen, P. J. F., and Mair, P. (2013). *Applied multidimensional scaling*. Springer.

Bibliography II

Van Deun, K., Heiser, W. J., and Delbeke, L. (2007). Multidimensional unfolding by nonmetric multidimensional scaling of Spearman distances in the extended permutation polytope. *Multivariate Behavioral Research*, 42(1), 103-132.

D'Ambrosio, A., Vera J.F. & Heiser, W.J. (2021). Avoiding degeneracies in ordinal Unfolding using Kemeny-equivalent dissimilarities for two-way two-mode preference rank data. *Multivariate Behavioral Research*, `https://doi.org/10.1080/00273171.2021.1899892`.