Data visualization

Multidimensional scaling

David Hoksza
http://siret.ms.mff.cuni.cz/hoksza

MDS outline

- Multidimensional scaling (MDS) is a **group of methods** allowing one to **represent (dis)similarities** among pairs of objects **as distances** between points of a **low-dimensional space**

- MDS enables

  - to take **quantifiable relationships between objects** in any space and **embed** those objects into **low-dimensional space** so that the **distances** in the target space **approximate** the original **relationships** as close as possible

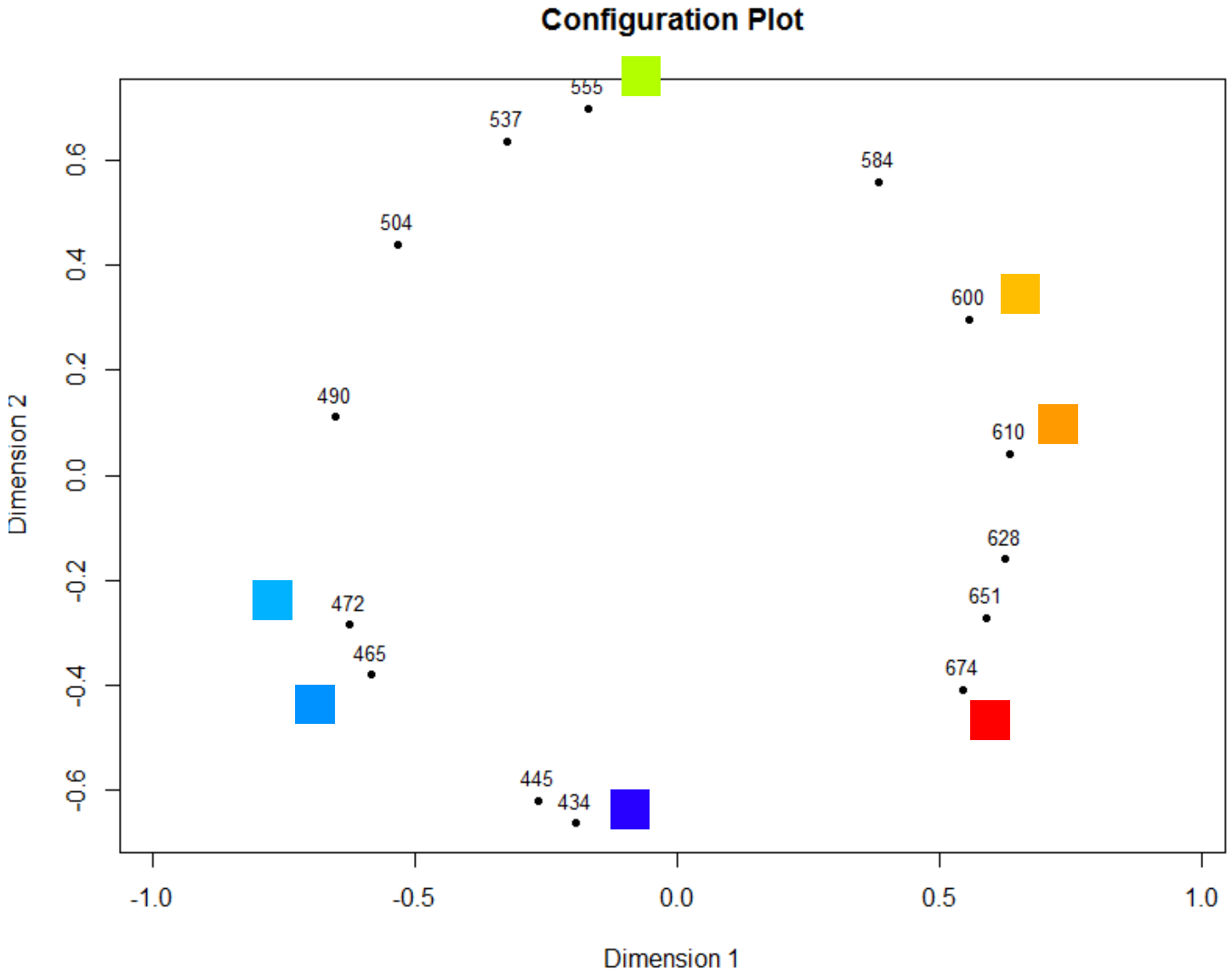
  - to **display the structure** of distance-like data as a geometrical picture

Color similarity perception (1)

- First use of MDS was in psychometrics [Ekman, 1954]
  - Perception of **similarities** of all pairs of 14 different colors by 31 subject
    - Similarities were ranked 0-4 and normalized into the [0;1] interval

	434	445	465	472	490	504	537	555	584	600	610	628	651
445	0.86												
465	0.42	0.50											
472	0.42	0.44	0.81										
490	0.18	0.22	0.47	0.54									
504	0.06	0.09	0.17	0.25	0.61								
537	0.07	0.07	0.10	0.10	0.31	0.62							
555	0.04	0.07	0.08	0.09	0.26	0.45	0.73						
584	0.02	0.02	0.02	0.02	0.07	0.14	0.22	0.33					
600	0.07	0.04	0.01	0.01	0.02	0.08	0.14	0.19	0.58				
610	0.09	0.07	0.02	0.00	0.02	0.02	0.05	0.04	0.37	0.74			
628	0.12	0.11	0.01	0.01	0.01	0.02	0.02	0.03	0.27	0.50	0.76		
651	0.13	0.13	0.05	0.02	0.02	0.02	0.02	0.02	0.20	0.41	0.62	0.85	
674	0.16	0.14	0.03	0.04	0.00	0.01	0.00	0.02	0.23	0.28	0.55	0.68	0.76

Color similarity perception (2)



Proximities data collection

- Dissimilarity (distance) vs similarity → **proximity**

- **Direct** collection of proximities
  - The proximities are outputs of the measurement (e.g., the color data)
  - Collected data are (almost) immediately ready for analysis

- **Derived** proximities
  - The original data do not have a direct relation, but using an appropriate measure the proximities can be derived (e.g., correlation of features or Euclidian distance between coordinates)

Interest in objects themselves

Interest in representation

Examples of input proximity matrices

- **Aggregate proximity** matrix
  - Identification of objects similarity based on respondents preferences (pile sort task)

- **Correlation** matrix
  - Puts objects with high positive correlations near each other, and object with strong negative correlations far apart

- **Flow** matrix
  - E.g., information about the number of transaction between corporations in a given time frame → MDS would reveal corporations or clusters of corporations which trade more often between each other

Idea behind MDS

- There are two basic approaches for MDS

Classical (metric) MDS	Non-metric MDS
• Assume that the dissimilarities are distances and then find coordinates that explain them → **projection** of the distances into coordinates • Linear projection based on the distance matrix	• Optimize the **individual distances** → works on the level of individual distances

Metric vs nonmetric MDS decision

- Input are dissimilarities close to Euclidean distances and there is a believe that a linear transform will suffice to map it into p-dimensional space → **metric MDS**

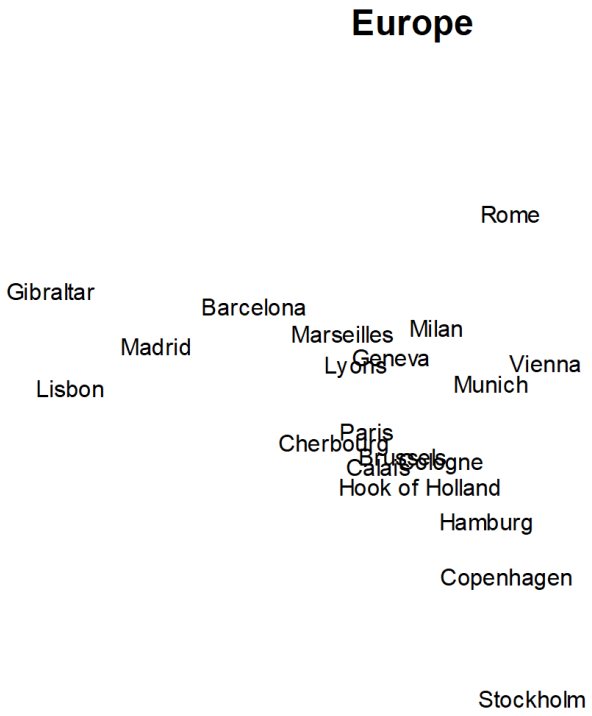
- Linear transform is not enough → monotonic transform → **nonmetric MDS**

Metric MDS

(Classical MDS, Classical scaling, Torgerson scaling, Principal Component Analysis)

Road distances

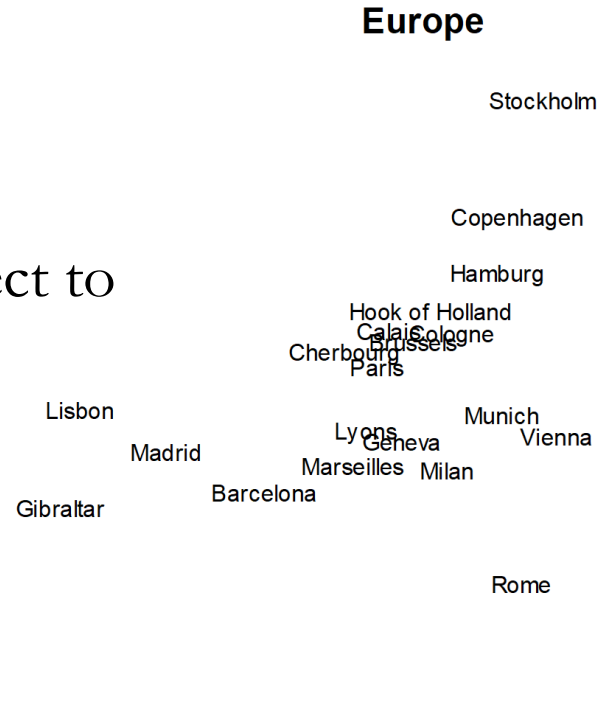
- R dataset *eurodist* - road distance between 21 European cities (almost Euclidean, but not quite)



Ehm….

Invariancy with respect to

- Shift
- Rotation
- Reflection



Classical MDS algorithm outline

- Provides an **analytical** solution, i.e. no iterative optimization required

- Two-step procedure
  - **Input**: **Euclidean distances** between n objects
  - **Output**: **Positions of the objects** up to rotation, reflection, shift

1. Compute scalar product matrix B from the input (metric) distances
2. Compute positions from B
   - Singular value decomposition

- The input is the **matrix of pair-wise Euclidean distances $\boldsymbol{D(X)}$** for X
  - Suppose we knew X (the coordinates we are searching for), then

$$D_{ij}^2(X) = (x_i - x_j)^T(x_i - x_j) = < x_i, x_i > -2x_i^T x_j + < x_j, x_j >$$

$$\boldsymbol{D^2(X)} = m1^T - 2X^T X + 1m^T$$
$$\boldsymbol{= m1^T + 1m^T - 2B}$$

$m^T = [< x_1, x_1 >, \dots < x_n, x_n >]$

- Let's multiply both sides by $-\frac{1}{2}$ and by centering matrix $\boldsymbol{C = I - n^{-1}11^T}$ from both sides → **double centering**

$$-\frac{1}{2}CD^2C = -\frac{1}{2}Cm1^TC - \frac{1}{2}C1m^TC + CBC$$
$$-\frac{1}{2}CD^2C = -\frac{1}{2}Cm0^T - \frac{1}{2}0m^TC + CBC$$
$$\boldsymbol{-\frac{1}{2}CD^2C = CBC = B}$$

Distances do not change under translation → we can assume X as column-centered (column means = 0)

- Now we can **extract MDS coordinates from $\boldsymbol{B}$** by factorizing it

$$\boldsymbol{XX^T = B = Q\Lambda Q^T = (Q\Lambda^{1/2})(\Lambda^{1/2}Q^T)}$$

Positive and symmetric

Classical MDS algorithm steps

1. Compute matrix of squared dissimilarities Δ^2
2. Apply double centering $B_\Delta = -\frac{1}{2} C \Delta^2 \mathrm{C}$
3. Compute eigendecomposition $B_\Delta = \mathrm{Q} \Lambda Q^T$
4. Let's denote matrix of non-zero eigenvalues as Λ_+ with eigenvectors matrix Q_+. Then the coordinate matrix is given by $Q_+ \Lambda_+^{1/2}$

If Δ is a Euclidean distance matrix, then classical MDS finds the coordinates up to a rotation.

Goodness of fit

- To get a low-dimensional representation, we keep m eigenvectors (out of n non-null ones) corresponding to the largest eigenvalues

$$\text{GOF} = \frac{\sum_{i=1}^{m} \lambda_{+i}}{\sum_{i=1}^{n} \lambda_{+i}}$$

- This minimizes

$$\sum_{i,j=1}^{n} \left(\delta_{ij} - d_{ij}(X)\right)^2$$

PCA vs metric MDS

- In PCA one is given a set of objects and their attributes, while in metric MDS the input is the mutual distances of the objects

- If we use **Euclidean distances**, then **PCA yields the same results as metric MDS**

  - **PCA** searches for **eigenvectors** of the covariance matrix $\boldsymbol{X^T X}$ while **MDS** searches for **eigenvectors** of the squared distance matrix $\boldsymbol{XX^T}$

  - Eigenvalues $X^T X$ = eigenvalues XX^T

Nonmetric MDS

Motivation for non-metric MDS

- In metric MDS, there is an implicit assumption that there is a true configuration in m dimensions, i.e., that Δ is a distance matrix.
  - The **proximities** do **not always** behave **like distances** (especially true for human perception-based data)

- Often all we have is **ranking** → non-metric MDS ensures that the **distances in the mapping will respect the ranking**

- The data are not exactly distances, but they are "distance-like" → the goal is to find positions which best approximate the actual distances

Error of the MDS configuration

- Let us have $\boldsymbol{n}$ **objects** with **dissimilarity** denoted as $\boldsymbol{\delta_{ij}}$ for each pair of objects (the between object relationships need to be transformed to dissimilarities)
- $\boldsymbol{X}$ denotes a **configuration** of the n points in m-dimensional space expressed as an $n \times m$ matrix
- $\boldsymbol{d_{ij}(X)} = \left[\sum_{a=1}^{m}\left(x_{ia} - x_{ja}\right)^2\right]^{1/2}$ denotes the Euclidean distance between any two points

- The **total error** of an MDS configuration X is defined as

$$\boldsymbol{\sigma^2\,(X)} = \sum_{i<j}\left(\boldsymbol{d_{ij}(X) - \delta_{ij}}\right)^2$$

Raw stress

- **Raw stress** is a weighted version of the total error

$$\sigma_r^2(X) = \sum_{i<j} w_{ij}\left(d_{ij}(X) - \delta_{ij}\right)^2$$

- The weights can be used to deal with missing values ($w_{ij} = 0$ if δ_{ij} is missing)

Other stress measures

- **Normalized raw Stress**

$$\sigma_n^2(X) = \frac{\sum_{i<j} w_{ij}\left(d_{ij}(X) - \delta_{ij}\right)^2}{\sum_{i<j} w_{ij}\,\delta_{ij}^2}$$

  - σ_n^2 is independent of the scale and the number of dissimilarities

- **Kruskal's stress-1**

$$\sigma_1(X) = \left(\frac{\sum_{i<j} w_{ij}\left(d_{ij}(X) - \delta_{ij}\right)^2}{\sum_{i<j} w_{ij}\,\delta_{ij}^2}\right)^{1/2}$$

Stress considerations

Stress	Goodness of fit
> 0.20	Poor
0.10	Fair
0.05	Good
0.025	Excellent
0.00	Perfect

Kruskal, J. B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika, 29, 1-27

- There exists a guideline for σ_1
- With growing number of dimensions the stress can't increase
- The stress is aggregation over all the pairs of objects → either the stress is spread over all pairs more or less equally or some pairs exhibit high stress
  - Large distances tend to show lower value of stress → **larger patterns (clusters) tend to stay visible**

Nonmetric MDS algorithm

INPUT: Relation matrix $\Delta\ (n \times n)$ of the n input objects

1. Initial configuration - project the objects to arbitrary points in m-dimensional space.
2. Compute the stress of the configuration X. The smaller the value, the greater the correspondence.
3. Adjust coordinates of each point in the direction that minimizes the stress.
4. Repeat steps 2 through 3 until convergence.

Iterative majorization (1)

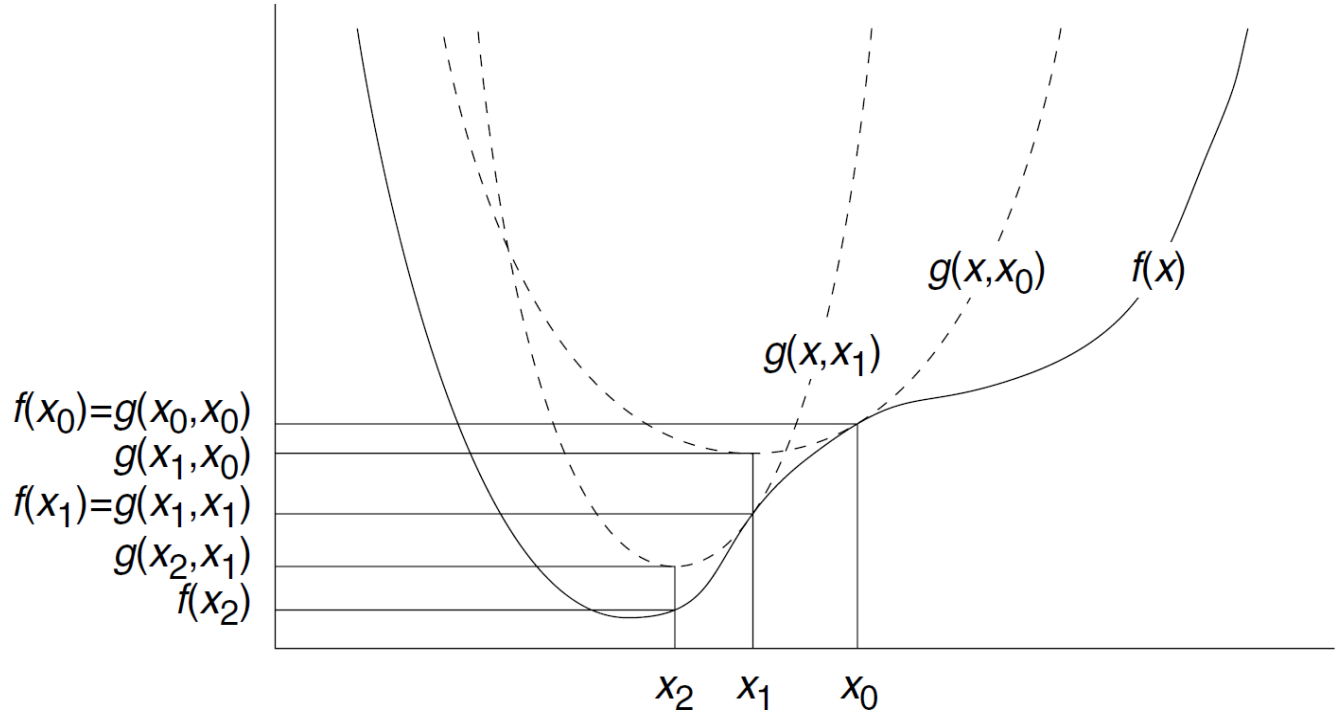
- Method of finding minimum of a complex function $f(x)$ by iteratively replacing it with a simple auxiliary function $g(x, z)$ which majorizes f

- **Requirements** of $\boldsymbol{g(x, z)}$ to be a **majorizing function** of $f(x)$
  - $f(x) \leq g(x, z)$
  - g must touch the surface of f at the supporting point $z \rightarrow f(z) = g(z, z)$
  - $g(x, z)$ should be simpler then $f(x)$, e.g. quadratic

Iterative majorization (2)

- Let minimum of $g(x, z)$ over x by obtained in x^*, then

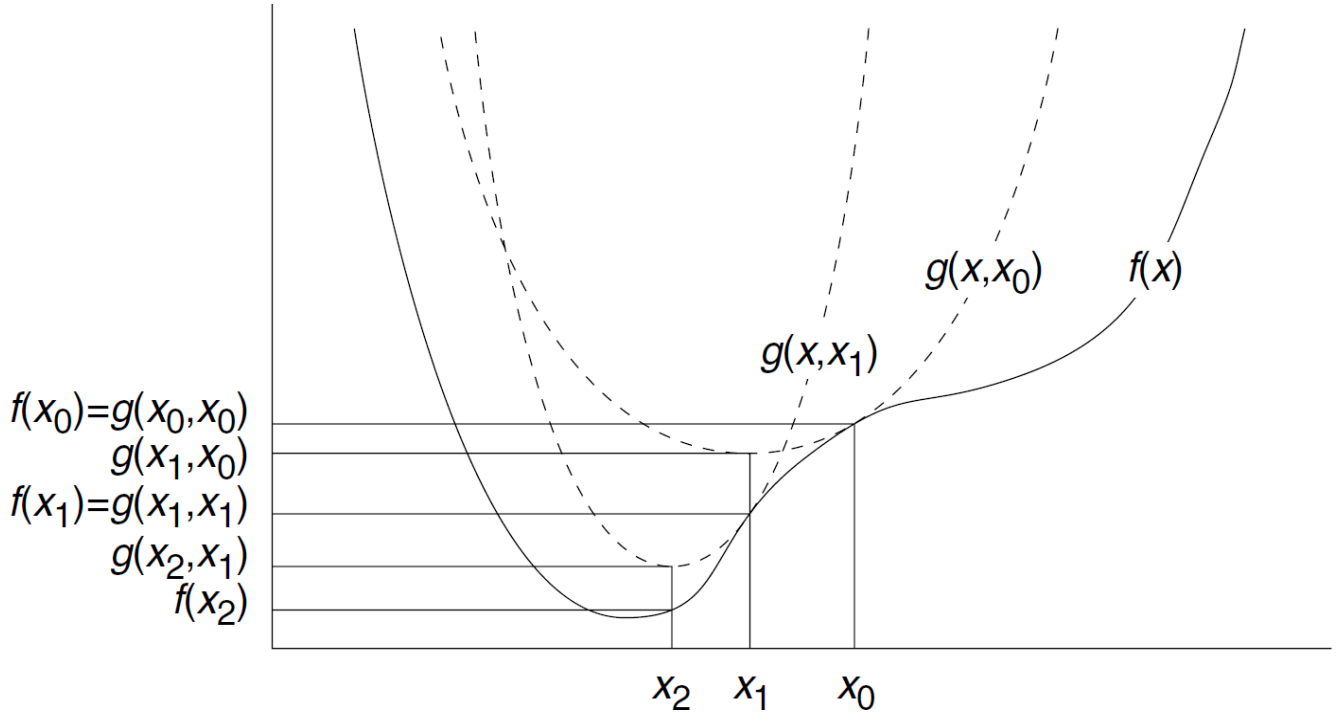
$$f(x^*) \leq g(x^*, z) \leq g(z, z) = f(z)$$



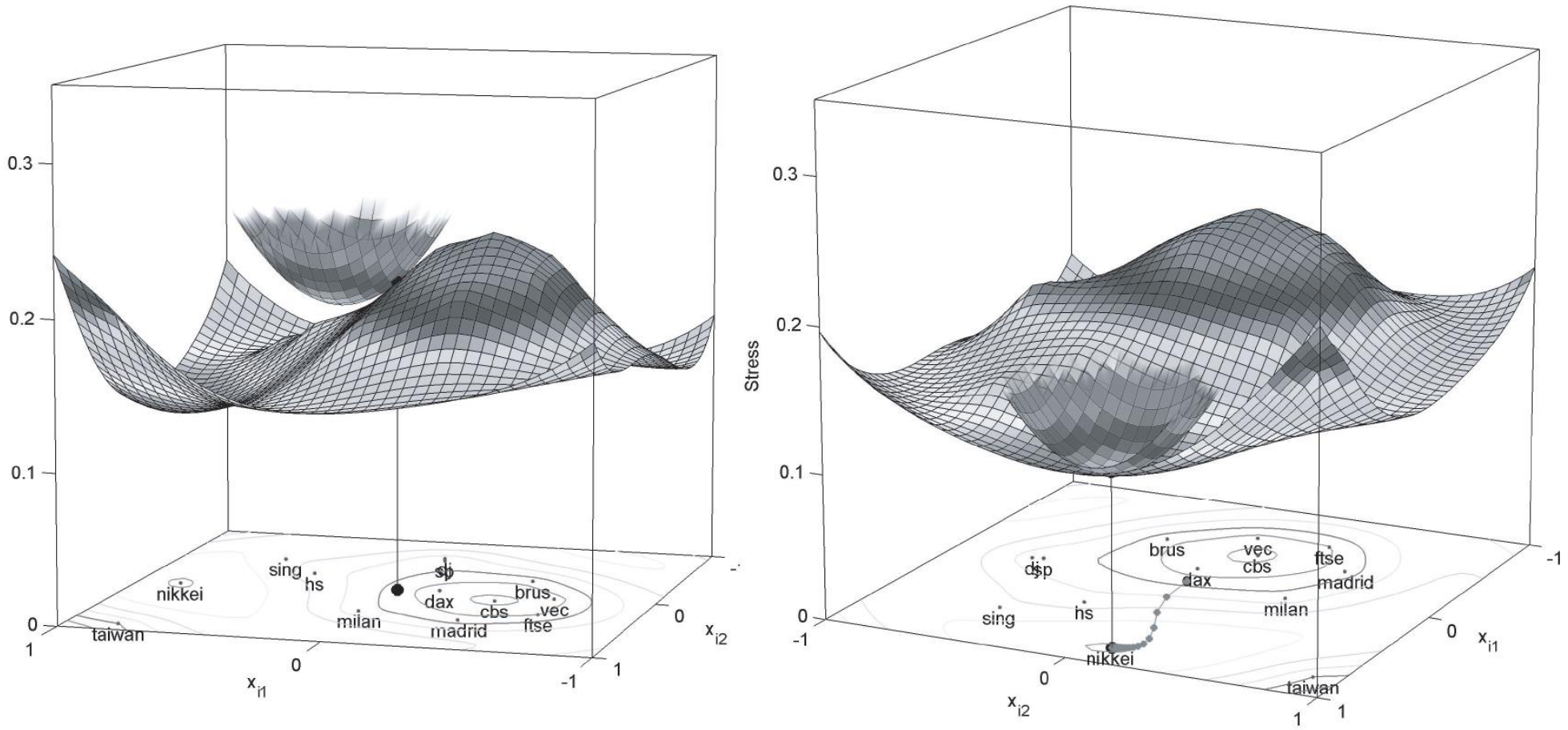
Iterative majorization (3)

- Iterative majorization algorithm

1. Set $z = z_0$
2. Find x so that $g(x, z) \leq g(z, z)$
3. If $f(z) - f(x) < \epsilon$, then stop
4. Set $z = x$ and go to 2.



Iterative majorization visualization in MDS



source: Borg, I., Groenen, P. J. F. (2005) Modern Multidimensional Scaling, Second Edition

Majorizing the stress function

$$\boldsymbol{\sigma}(\boldsymbol{X}) = \sum_{i<j} w_{ij}\left(d_{ij}(X) - \delta_{ij}\right)^2$$

$$= \sum_{i<j} w_{ij}\delta_{ij}^2 + \sum_{i<j} w_{ij}d_{ij}^2(X) - 2\sum_{i<j} w_{ij}\delta_{ij}d_{ij}(X)$$

$$= \boldsymbol{\eta_\delta^2} + \boldsymbol{\eta^2}(\boldsymbol{X}) - \mathbf{2}\boldsymbol{\rho}(\boldsymbol{X})$$

Rewriting $\eta^2(X)$

$x_{ia} - x_{ja} = (e_i - e_j)^T x_a$

Column a of the coordinate matrix

$$d_{ij}^2(X) = \sum_{a=1}^{m} x_a^T (e_i - e_j)(e_i - e_j)^T x_a = \sum_{a=1}^{m} x_a^T A_{ij} x_a = \operatorname{tr} X^T A_{ij} X$$

$$w_{ij} d_{ij}^2(X) = \operatorname{tr} X^T (w_{ij} A_{ij}) X$$

$$\eta^2(X) = \sum_{i<j} w_{ij} d_{ij}^2(X) = \operatorname{tr} X^T \left(\sum_{i<j} w_{ij} A_{ij} \right) X = \operatorname{tr} X^T V X$$

- Thus, we have a compact expression of $\eta^2(X)$ which is a quadratic function of X

Majorizing $\rho(X)$ (1)

Cauchy-Schwarz inequality (equality for $x_i = z_i$ and $x_j = z_j$

$$-\rho(X) = -\sum_{i<j}(w_{ij}\delta_{ij})d_{ij}(X)$$

$$\sum_{a=1}^{m}(x_{ia}-x_{ja})(z_{ia}-z_{ja}) \leq \left(\sum_{a=1}^{m}(x_{ia}-x_{ja})^2\right)^{\frac{1}{2}}\left(\sum_{a=1}^{m}(z_{ia}-z_{ja})^2\right)^{\frac{1}{2}} = d_{ij}(X)d_{ij}(Z)$$

$$-d_{ij}(X) \leq -\frac{\sum_{a=1}^{m}(x_{ia}-x_{ja})(z_{ia}-z_{ja})}{d_{ij}(Z)} = -\frac{\operatorname{tr} X^T A_{ij} Z}{d_{ij}(Z)}$$

Majorizing $\rho(X)$ (2)

$$-\rho(X) = -\sum_{i<j}(w_{ij}\delta_{ij})d_{ij}(X) \leq -\sum_{i<j}(w_{ij}\delta_{ij})\frac{\operatorname{tr} X^T A_{ij} Z}{d_{ij}(Z)}$$

$$= -\operatorname{tr} X^T \sum_{i<j}(w_{ij}\delta_{ij})\frac{A_{ij}}{d_{ij}(Z)} Z = -\operatorname{tr} X^T \left(\sum_{i<j}\frac{w_{ij}\delta_{ij}}{d_{ij}(Z)} A_{ij}\right) Z$$

$$= -\operatorname{tr} X^T \left(\sum_{i<j} b_{ij}(Z) A_{ij}\right) Z = -\operatorname{tr} X^T B(Z) Z$$

- The equality occurs when $Z = X \rightarrow$ majorizing inequality

$$\boldsymbol{-\rho(X) = -\operatorname{tr} X^T B(X) X \leq -\operatorname{tr} X^T B(Z) Z}$$

SMACOF

- Scaling by majorizing a convex function

$$\sigma(X) = \eta_\delta^2 + \eta^2(X) - 2\rho(X) \leq \eta_\delta^2 + \text{tr}\, X^T V X - 2\text{tr}\, X^T B(Z) Z = \tau(X, Z)$$

- $\tau(X, Z)$ is a quadratic majorizing function to be used in the iterative minimization procedure

Includes derivation of τ, setting it to 0 and computation of the matrix inverse.

1. Set initial configuration X^0 and $k = 0$
2. $k = k + 1;\ Z = X^{k-1}$
3. Update $X^k = \min_X \tau(X, Z)$
4. If $\sigma(X^{k-1}) - \sigma(X^k) < \epsilon$ stop, else go to 2.

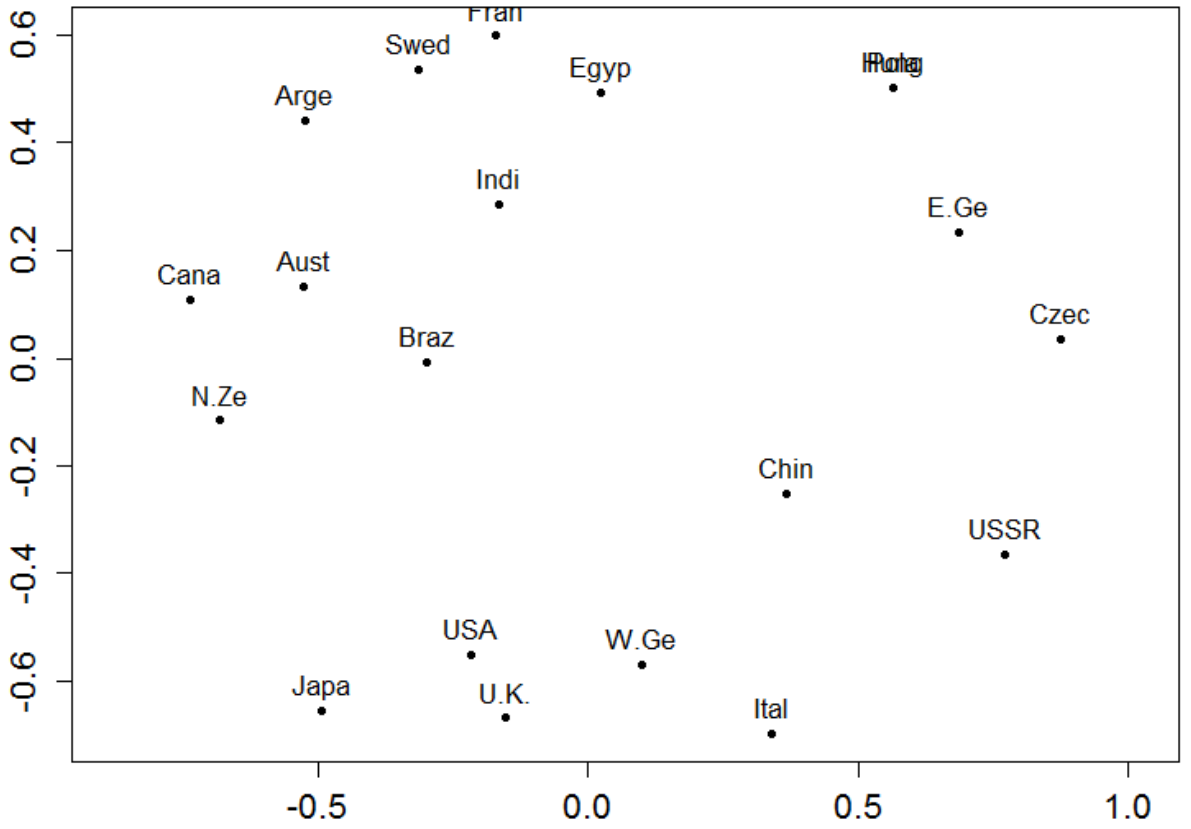
Adequacy of a MDS solution

- Configuration/MDS plot
- Scree plot
- Sheppard diagram
- Stress plot
- Bubble plot

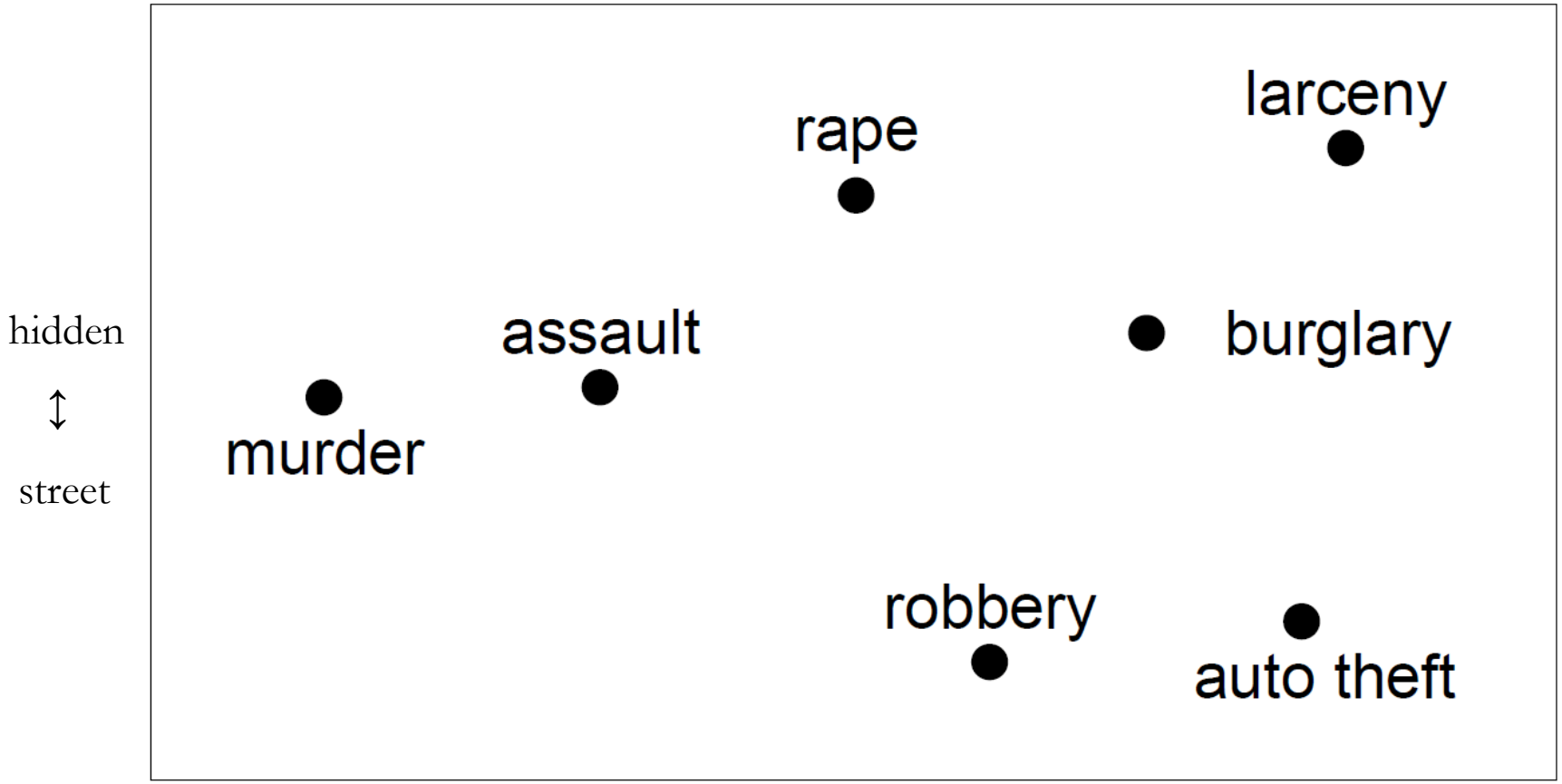
Configuration plot

Data from the New Geographical Digest (1986), analysed in Cox and Cox (2001), on which countries traded with other countries. For 20 countries the main trading partners are dichotomously scored (1 = trade performed, 0 = trade not performed). Based on this dichotomous matrix the dissimilarities are computed using the Jaccard coefficient.

- The plot is invariant with respect to rotation, reflection and shift
- The axes do not bear any meaning by definition, but they can be interpreted

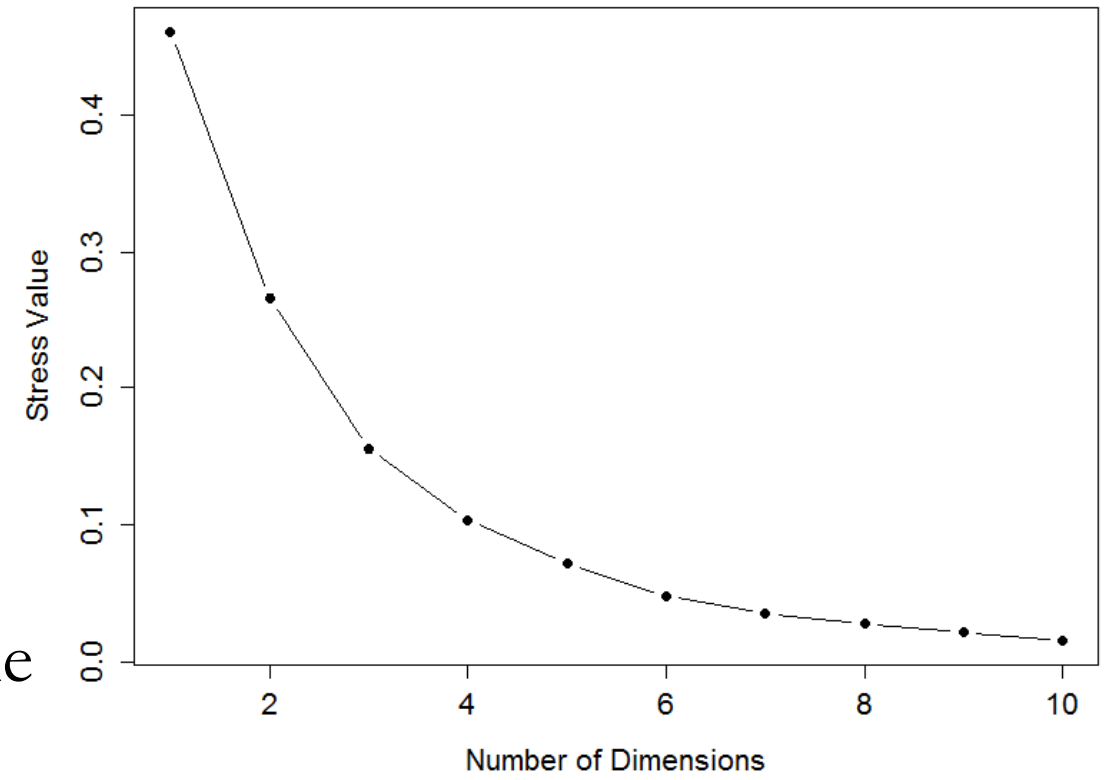


Axis interpretation



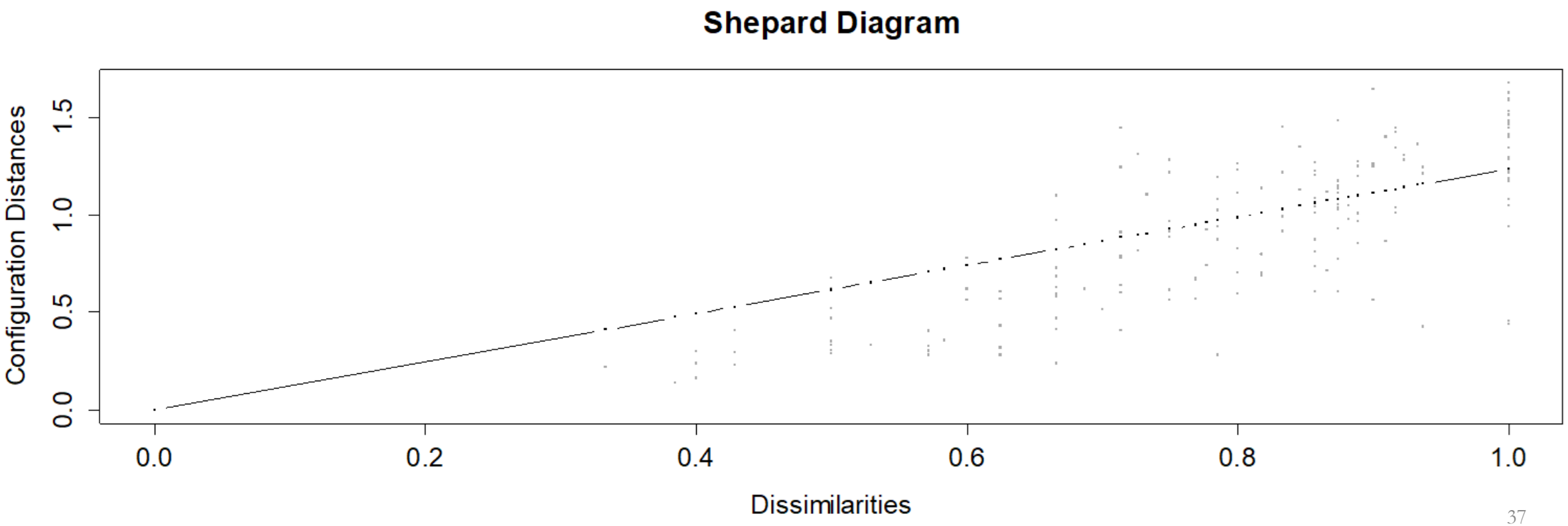
Scree plot

- The amount of stress is plotted against the number of dimensions
- How decrease in the number of dimensions decreases the stress
- It can be used to see the real dimensionality of data irrespective of the inherent noise



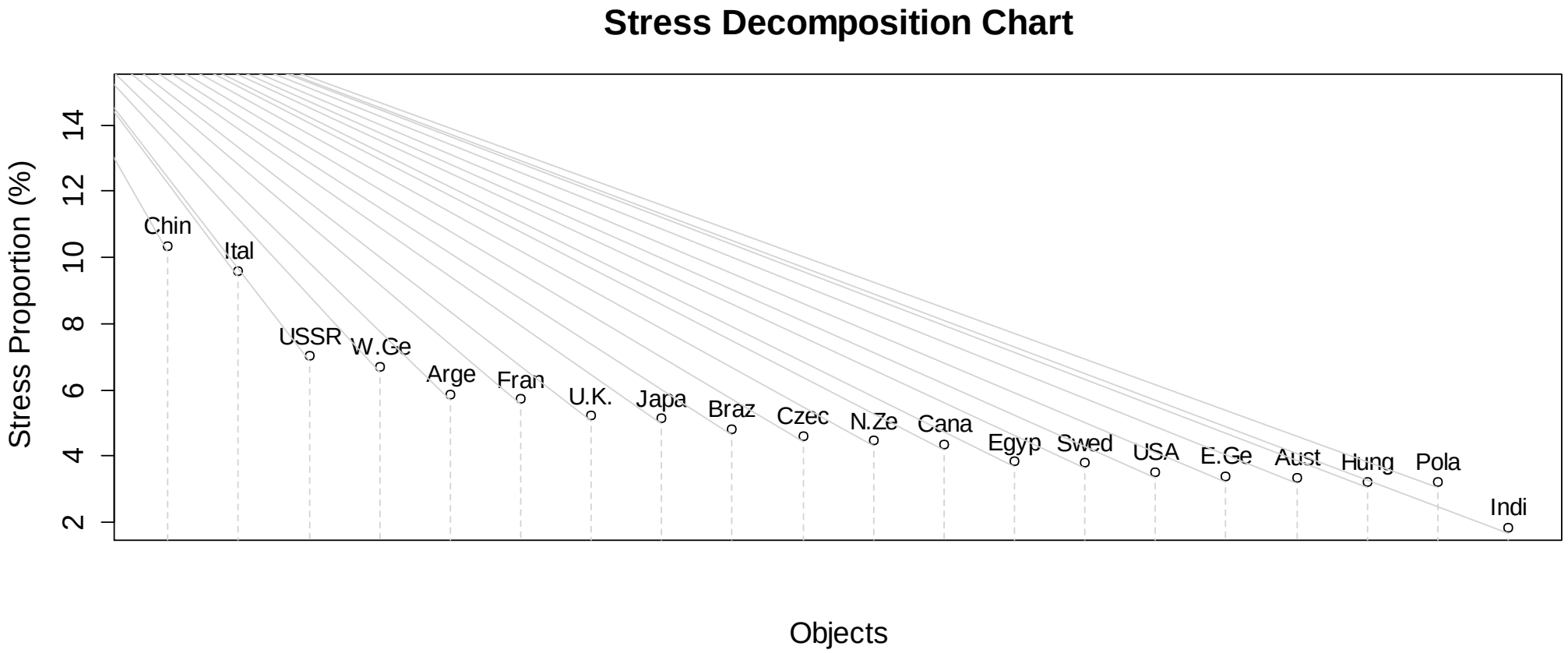
Sheppard plot

- The relationship between the proximities and the distances of the point configuration



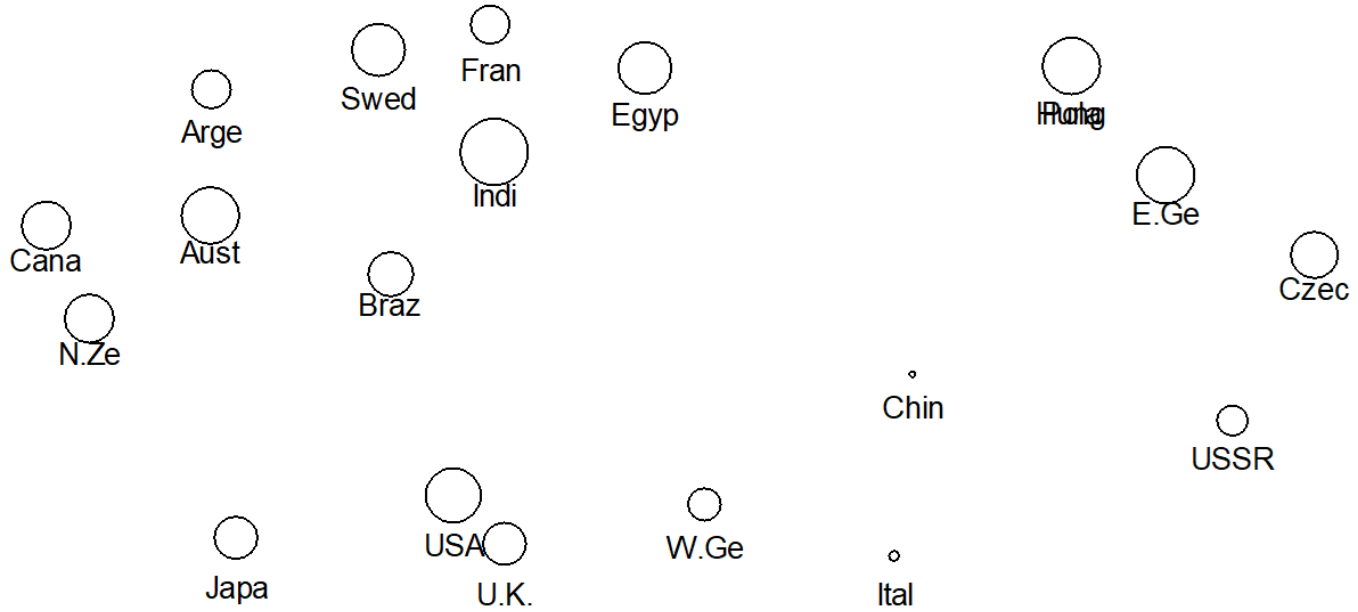
Stress plot

- Stress contribution for each point



Bubble plot

- Stressplot/confplot combined (the larger bubbles, the better the fit)



Degenerate solutions

- **Missing dissimilarities**
  - If it is possible to split the objects into two or more sets with zero between-set weights, then we are dealing with separate MDS problems → the missing similarities could be inadvertently interpreted

- **No information in the data**
  - Between-object distances fall into a small interval
    - The objects will lie on a line in 1D and on concentric circles in 2D
  - The solution is to redo MDS with transformed values (e.g., interval transformation)

Local minima

- MDS algorithms which minimize the stress function cannot guarantee obtaining of a global minimum

- One can limit the risk of ending in a local optimum **by running MDS multiple times** with different initial configurations

  - Ending in the same final configuration indicates that the minimum might be global minimum as well

MDS properties

- MDS does very **few assumptions** about the nature of the data and distance measure → well suited for a **wide variety of data**
  - PCA expects linear relationship between the coordinates
- For **large datasets**, MDS can be **slow** (holds for non-metric MDS)
- Can **stuck in local optima** since it is a numerical optimization technique (holds for non-metric MDS)
- Can be **easily explained** to non-experts

MDS in R

Classical MDS

- cmdscale (stats)
- Wcmdscale (vegan)
- smacofSym (smacof)
- dudi.pco (ade4)
- pco (ecodist)

Nonmetric MDS

- isoMDS (MASS)
- smacofSym (smacof)
- metaMDS(vegan)

```
if (!require("smacof")) {install.packages("smacof");
library("smacof")}


data(ekman)
ekman.d <- sim2diss(ekman, method = 1)
res <- smacofSym(ekman.d)


res.basic
plot(res)
plot(res, plot.type = "stressplot", ylim = c(2,15))
plot(res, plot.type = "bubbleplot")
plot(res, plot.type = "Shepard")
```

Sources

- Borg, I., Groenen, P. J. F. (2005) Modern Multidimensional Scaling, Second Edition. Springer-Verlag New York
- Cox, T. F., Cox, M. A. A. (2000) Multidimensional Scaling, Second Edition. CRC Press
- Groenen, P. J. F., Velden, M. (2004) Multidimensional Scaling. Econometric Institute Report EI 2004-15
- Wickelmaier, F. (2003) An Introduction to MDS . Sound Quality Research Unit at Aalborg University