Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Multidimensional Scaling

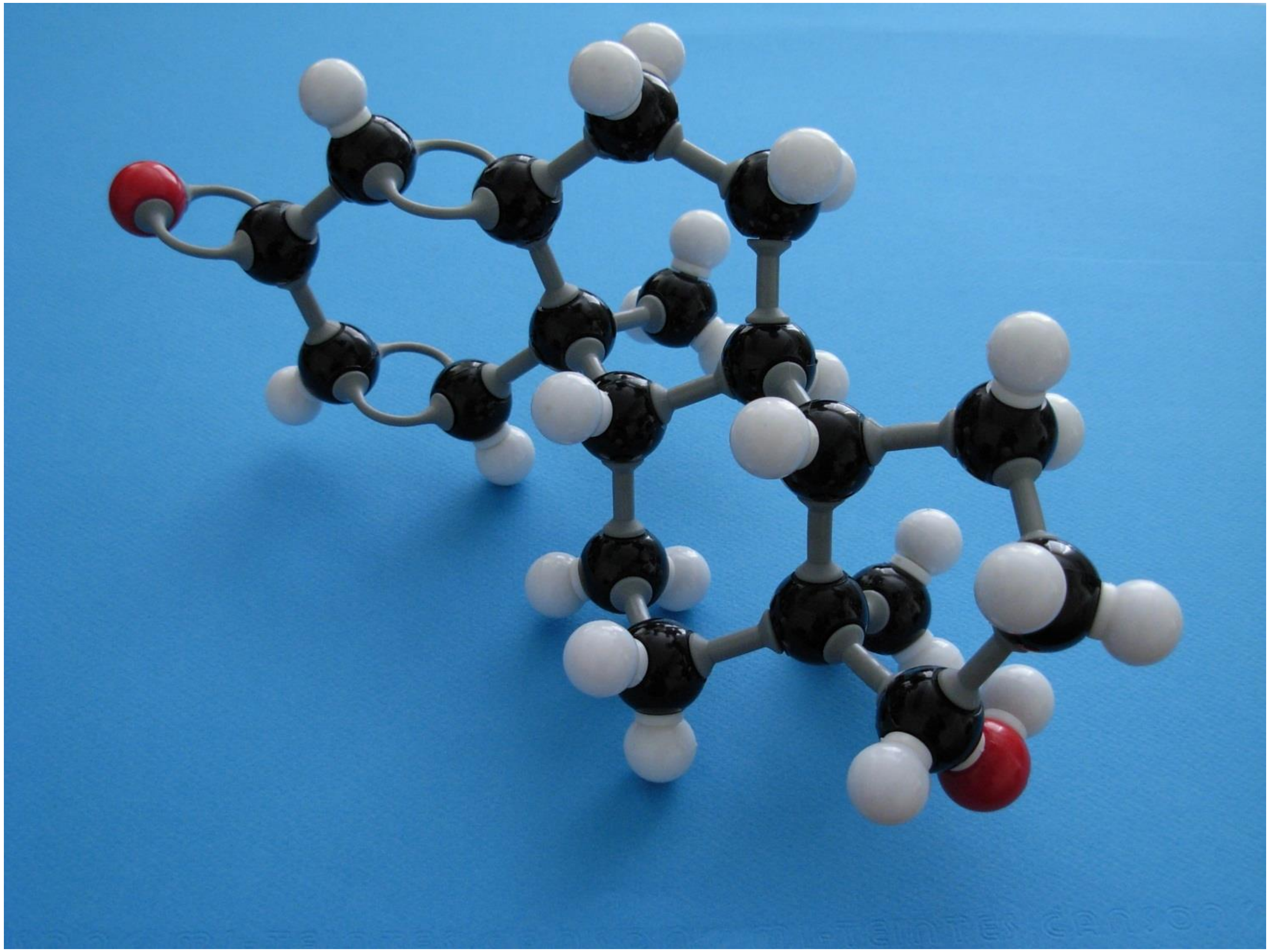
Applied Multivariate Statistics – Spring 2013

Outline

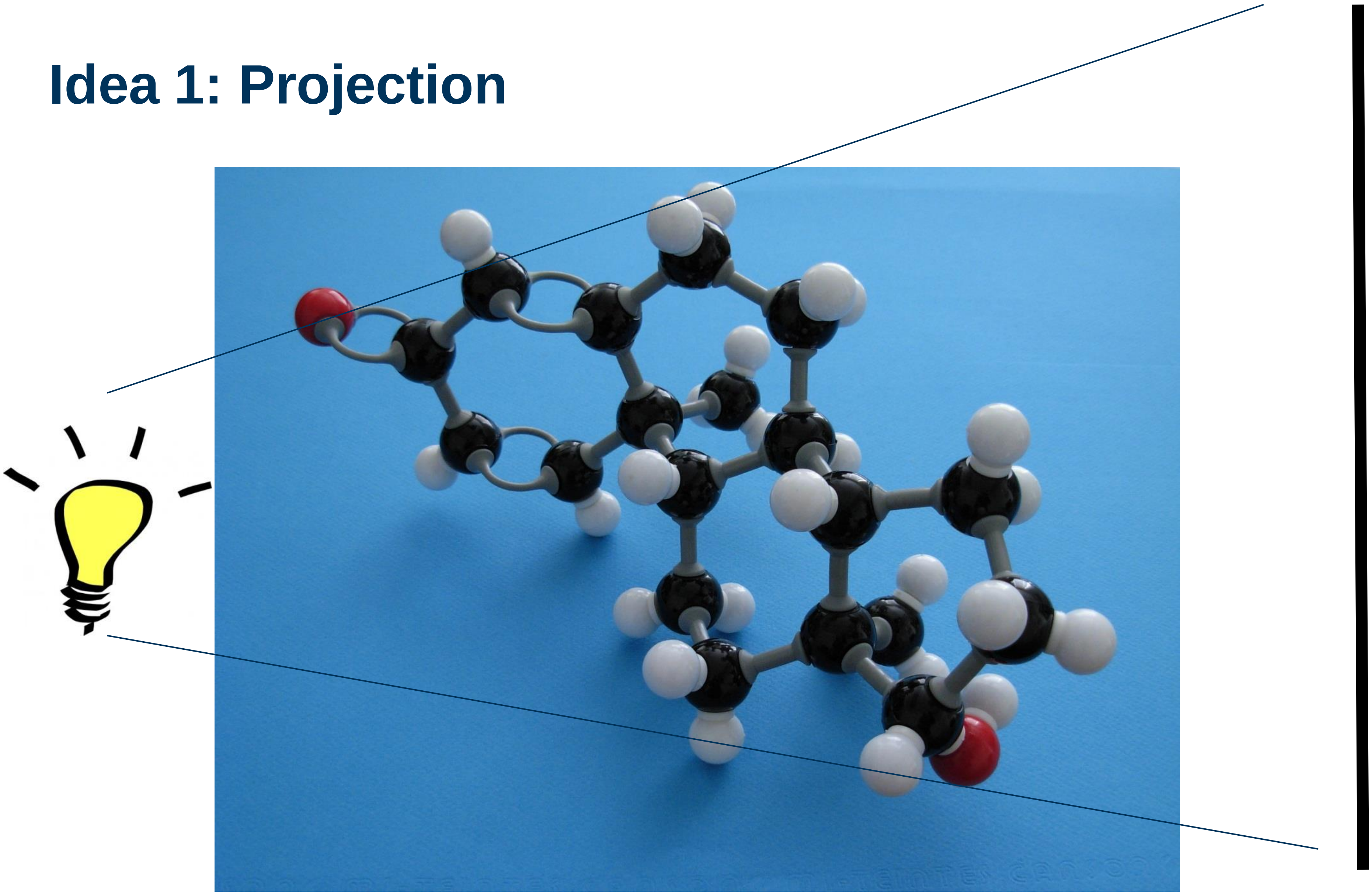
- Fundamental Idea
- Classical Multidimensional Scaling
- Non-metric Multidimensional Scaling

Basic Idea

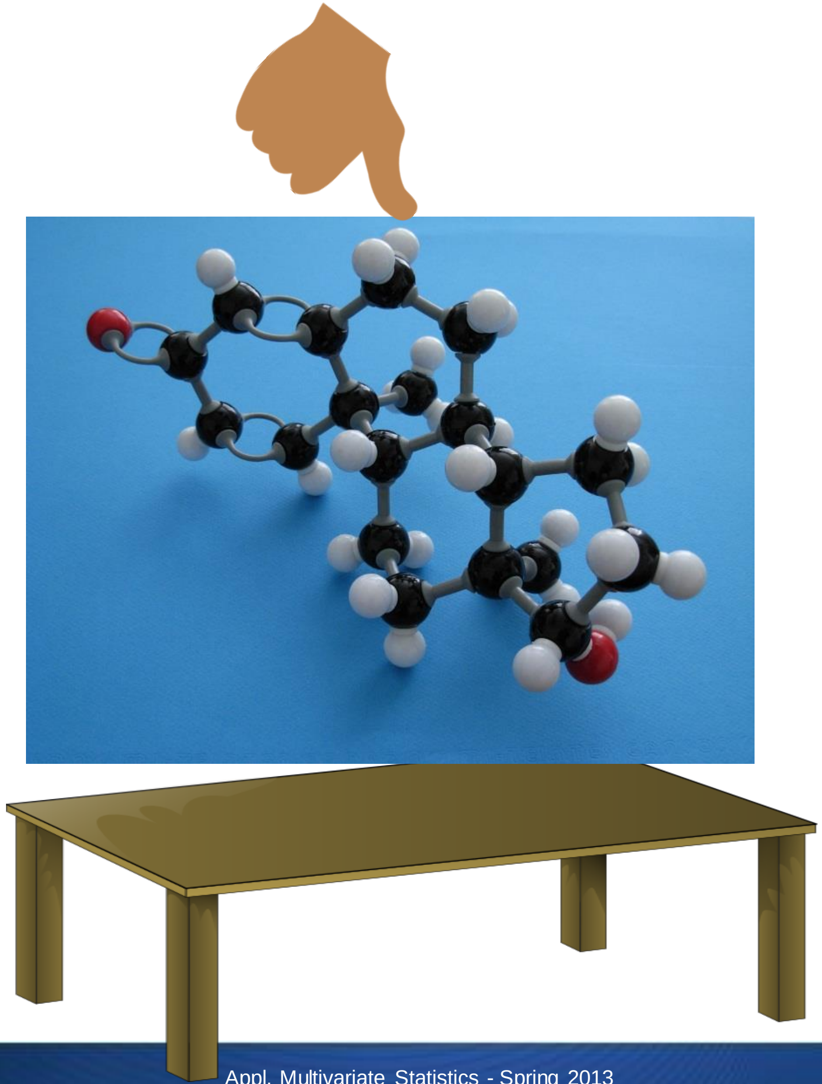
How to represent in two dimensions?



Idea 1: Projection



Idea 2: Squeeze on table

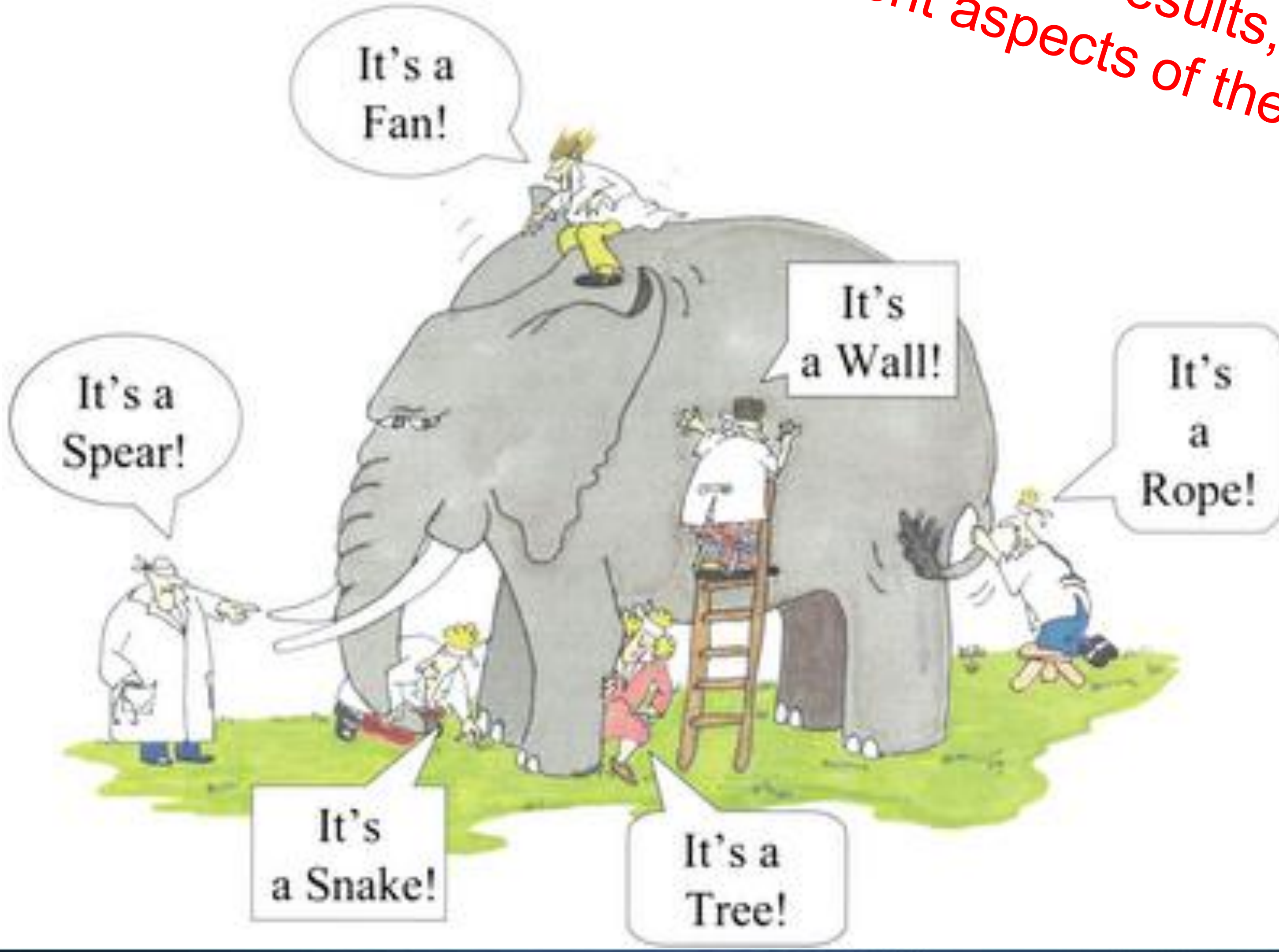


Close points stay close

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Which idea is better?

Both ideas yield correct results,
but show different aspects of the data.



Idea of MDS

- Represent high-dimensional point cloud in few (usually 2) dimensions keeping distances between points similar
- Classical/Metric MDS: Use a clever projection
  R: cmdscale
- Non-metric MDS: Squeeze data on table, only conserve ranks
  R: isoMDS

Classical MDS

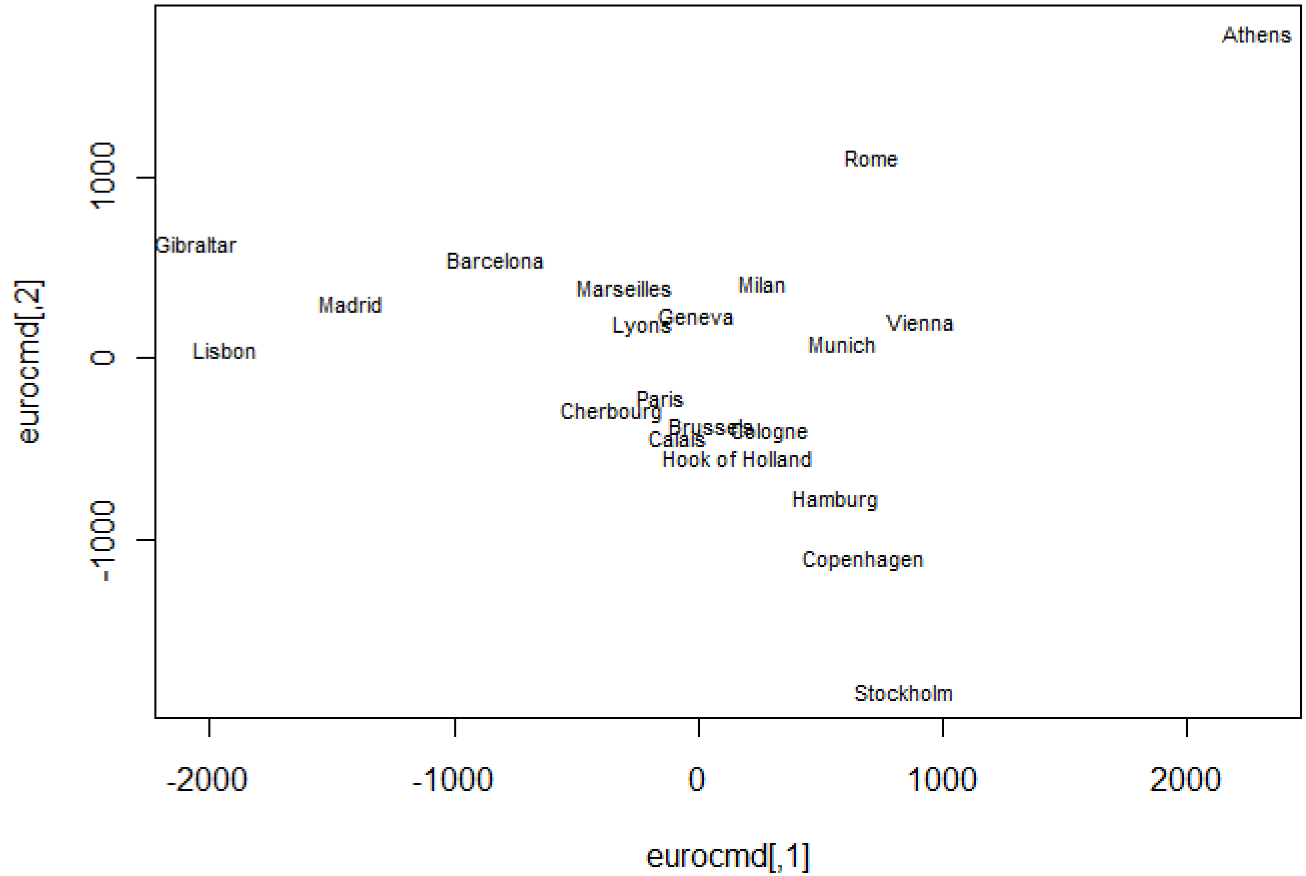
- Problem: Given euclidean distances among points, recover the position of the points!
- Example: Road distance between 21 European cities (almost euclidean, but not quite)

	Athens	Barcelona	Brussels	Calais	Cherbourg
Barcelona	3313				
Brussels	2963	1318			
Calais	3175	1326	204		
Cherbourg	3339	1294	583	460	
Cologne	2762	1498	206	409	785

⋮

Classical MDS

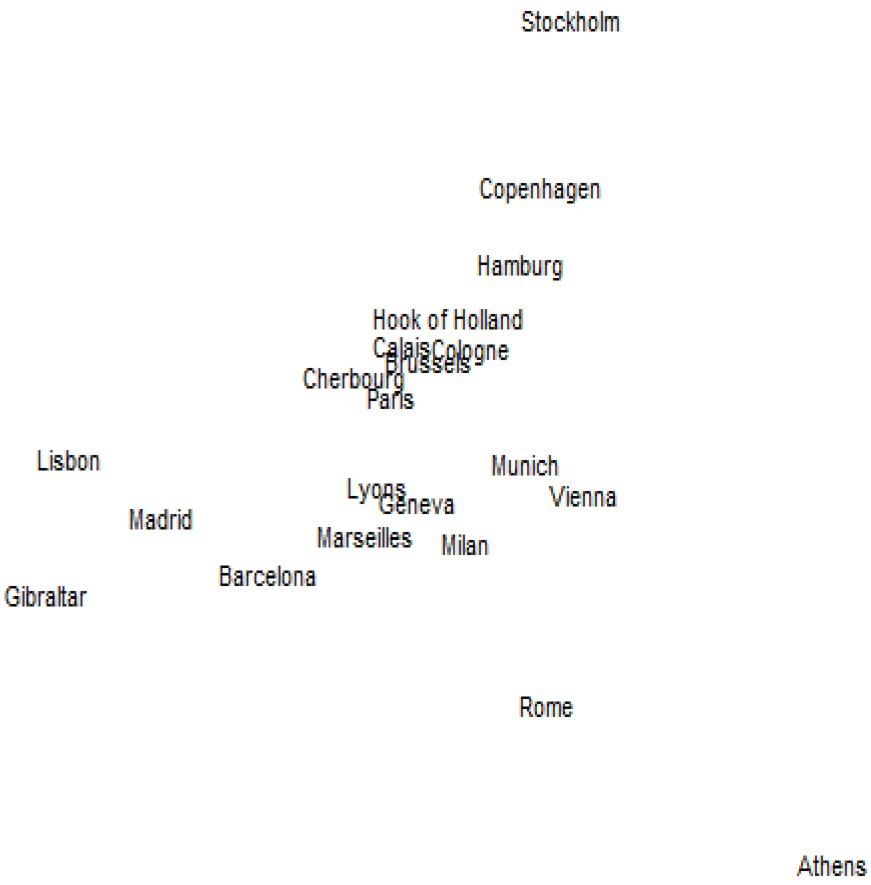
- First try:



Classical MDS

- Flip axes:

Can identify points up to
- shift
- rotation
- reflection



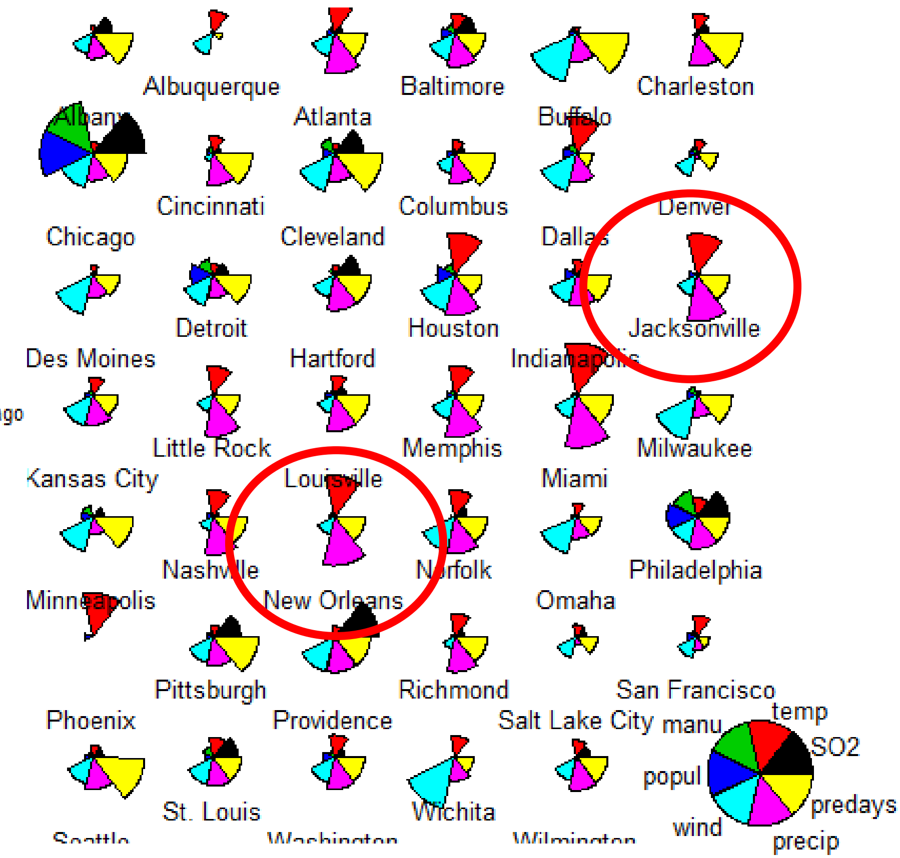
Classical MDS

- Another example: Airpollution in US cities

```
> summary(dat)
      so2              temp            manu             popul
 Min.   :  8.00   Min.   :43.50   Min.   :  35.0   Min.   :  71.0
 1st Qu.: 13.00   1st Qu.:50.60   1st Qu.: 181.0   1st Qu.: 299.0
 Median : 26.00   Median :54.60   Median : 347.0   Median : 515.0
 Mean   : 30.05   Mean   :55.76   Mean   : 463.1   Mean   : 608.6
 3rd Qu.: 35.00   3rd Qu.:59.30   3rd Qu.: 462.0   3rd Qu.: 717.0
 Max.   :110.00   Max.   :75.50   Max.   :3344.0   Max.   :3369.0
      wind            precip          predays
 Min.   : 6.000   Min.   : 7.05   Min.   : 36.0
 1st Qu.: 8.700   1st Qu.:30.96   1st Qu.:103.0
 Median : 9.300   Median :38.74   Median :115.0
 Mean   : 9.444   Mean   :36.77   Mean   :113.9
 3rd Qu.:10.600   3rd Qu.:43.11   3rd Qu.:128.0
 Max.   :12.700   Max.   :59.80   Max.   :166.0
```

- Range of manu and popul is much bigger than range of wind
- Need to standardize to give every variable equal weight

Classical MDS



Classical MDS: Theory

- Input: Euclidean distances between n objects in p dimensions
- Output: Position of points up to rotation, reflection, shift
- Two steps:
  - Compute inner products matrix B from distance
  - Compute positions from B

Classical MDS: Theory – Step 1

n * q data matrix

- Inner products matrix B = XX^T
  $b_{ij} = \sum_{k=1}^{q} x_{ik} x_{jk}$
- Connect to distance: $d_{ij}^2 = \sum_{k=1}^{q} (x_{ik} - x_{jk})^2 = ... = b_{ii} + b_{jj} - 2b_{ij}$
- Center points to avoid shift invariance
  $\left(\overline{x} = 0 \rightarrow \sum_{i=1}^{n} x_{ik} = 0 \rightarrow \sum_{i\ or\ j} b_{ij} = 0 \right)$
- Invert relationship: $b_{ij} = -\frac{1}{2}(d_{ij}^2 - d_{i.}^2 - d_{.j}^2 + d_{..}^2)$
  "doubly centered"
  (Hint for middle of page 108: Plug in (4.3) and equations on top of page 108 to show that the expression involving d's is equal to b_{ij})
- Thus, we obtained B from the distance matrix

Classical MDS: Theory – Step 2

- Since B = XX^T, we need the “square root” of B
- B is a symmetric and positive definite n*n matrix
- Thus, B can be diagonalized: $B = V \Lambda V^T$
  D is a diagonal matrix with $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$ on diagonal (“eigenvalues”)
  V contains as columns normalized eigenvectors
- Some eigenvalues will be zero; drop them: $B = V_1 \Lambda_1 V_1^T$
- Take “square root”: $X = V_1 \Lambda_1^{\frac{1}{2}}$

- Thus we obtained the position of points from the distances between all points

Classical MDS: Low-dim representation

- Keep only few (e.g. 2) largest eigenvalues and corresponding eigenvectors
- The resulting X will be the low-dimensional representation we were looking for

- Goodness of fit (GOF) if we reduce to m dimensions:

$$GOF = \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{n} \lambda_i}$$

(should be at least 0.8)

- Finds “optimal” low-dim representation: Minimizes

$$S = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(d_{ij}^2 - (d_{ij}^{(m)})^2 \right)$$

Classical MDS: Pros and Cons

+ Optimal for euclidean input data
+ Still optimal, if B has non-negative eigenvalues (pos. semidefinite)
+ Very fast
- No guarantees if B has negative eigenvalues

However, in practice, it is still used then. New measures for Goodness of fit:

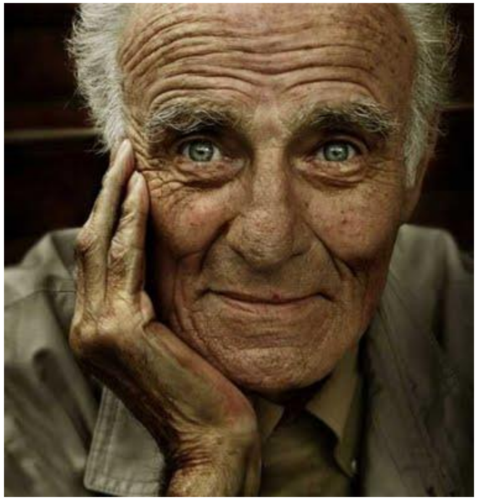
$$GOF = \frac{\sum_{i=1}^{m} \lambda_i^2}{\sum_{i=1}^{n} \lambda_i^2}$$

$$GOF = \frac{\sum_{i=1}^{m} |\lambda_i|}{\sum_{i=1}^{n} |\lambda_i|} \qquad GOF = \frac{\sum_{i=1}^{m} max(0,\lambda_i)}{\sum_{i=1}^{n} max(0,\lambda_i)}$$

Used in R function “cmdscale”

Non-metric MDS: Idea

- Sometimes, there is no strict metric on original points
- Example: How beautiful are these persons?
  (1: Not at all, 10: Very much)



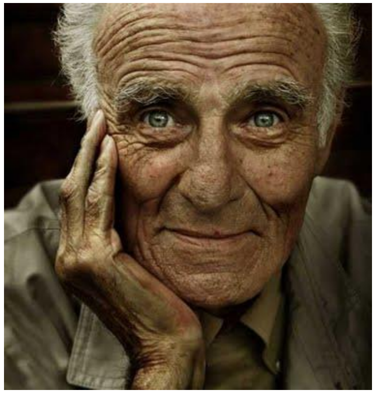
2 6 9

OR 1 5 10 ??

Non-metric MDS: Idea

- Absolute values are not that meaningful
- Ranking is important
- Non-metric MDS finds a low-dimensional representation, which respects the ranking of distances

>



>

Non-metric MDS: Theory

- δ_{ij} is the true dissimilarity, d_{ij} is the distance of representation
- Minimize STRESS (θ is an increasing function):

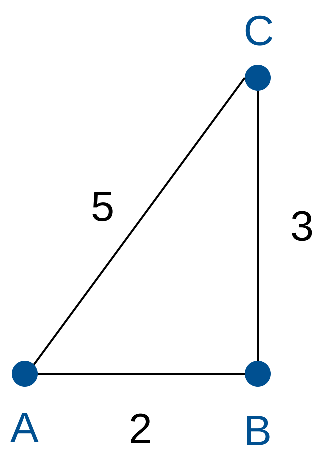
$$S = \frac{\sum_{i<j} (\theta(\delta_{ij}) - d_{ij})^2}{\sum_{i<j} d_{ij}^2}$$

- Optimize over both position of points and θ
- $\hat{d}_{ij} = \theta(\delta_{ij})$ is called "disparity"
- Solved numerically (isotonic regression);
  Classical MDS as starting value;
  very time consuming

Non-metric MDS: Example for intuition (only)

True points in

high dimensional space



Compute best

representation

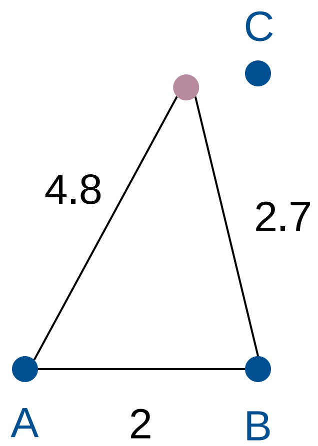
STRESS = 19.7

$$\delta_{AB} < \delta_{BC} < \delta_{AC}$$

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

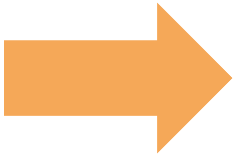
Non-metric MDS: Example for intuition (only)

True points in high dimensional space



$$\delta_{AB} < \delta_{BC} < \delta_{AC}$$

Compute best representation

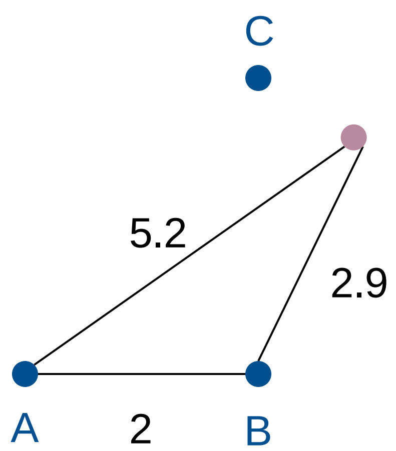


STRESS = 20.1

Non-metric MDS: Example for intuition (only)

True points in

high dimensional space



$$\delta_{AB} < \delta_{BC} < \delta_{AC}$$

Compute best

representation

Stop if minimal STRESS is found.

STRESS = 18.9

We will finally represent the "transformed true distances" (called disparities):

$$\hat{d}_{AB} = 2,\ \hat{d}_{BC} = 2.9,\ \hat{d}_{AC} = 5.2$$

instead of the true distances:

$$\delta_{AB} = 2,\ \delta_{BC} = 3,\ \delta_{AC} = 5$$

Non-metric MDS: Pros and Cons

+ Fulfills a clear objective without many assumptions (minimize STRESS)
+ Results don't change with rescaling or monotonic variable transformation
+ Works even if you only have rank information

- Slow in large problems
- Usually only local (not global) optimum found
- Only gets ranks of distances right

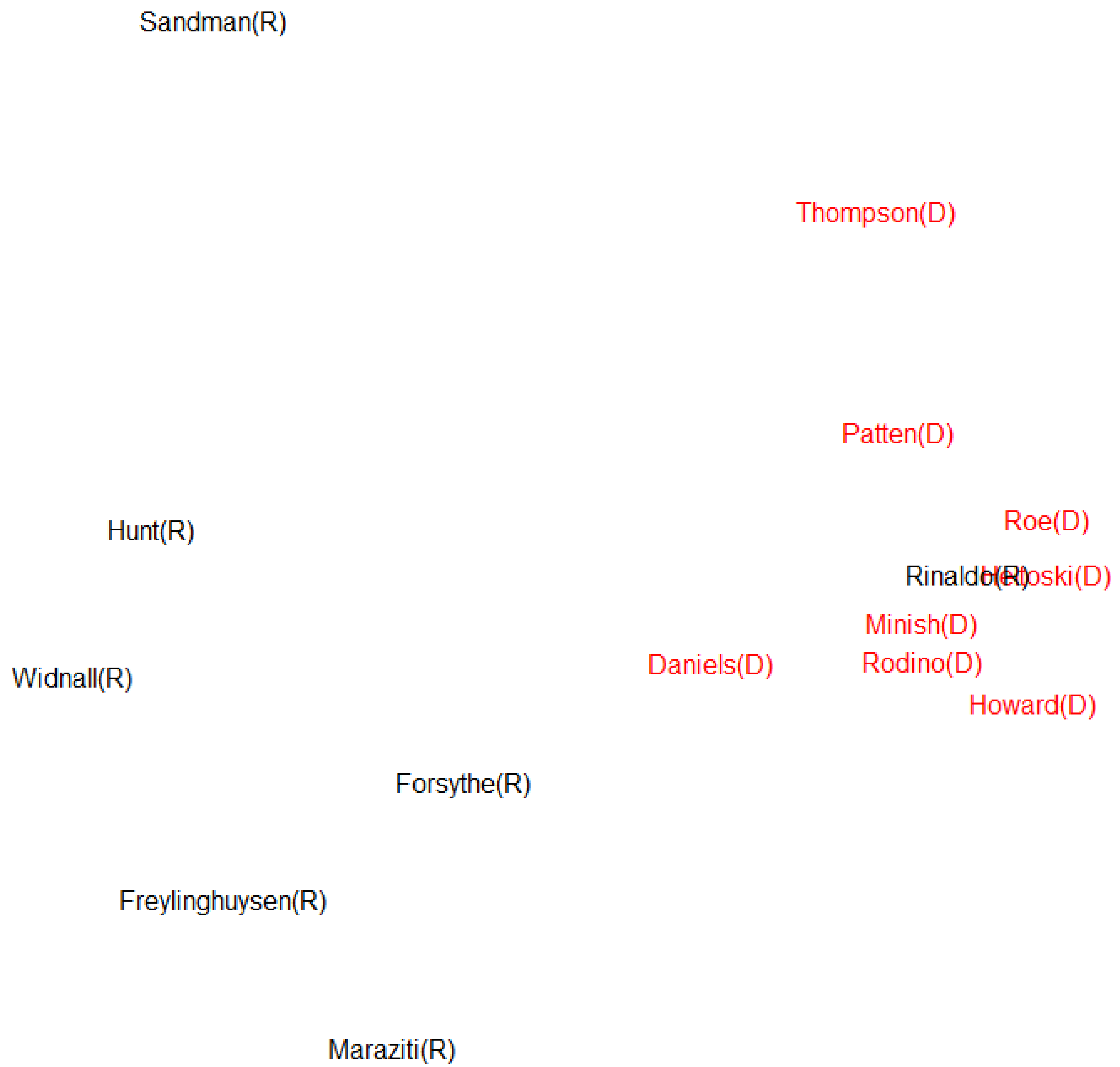
Non-metric MDS: Example

- Do people in the same party vote alike?
- Number of votes where 15 congressmen disagreed in 19 votes

	Hunt(R)	Sandman(R)	Howard(D)	Thompson(D)
Hunt(R)	0	8	15	15
Sandman(R)	8	0	17	12
Howard(D)	15	17	0	9
Thompson(D)	15	12	9	0

⋮

Non-metric MDS: Example



Concepts to know

- Classical MDS:
  - Finds low-dim projection that respects distances
  - Optimal for euclidean distances
  - No clear guarantees for other distances
  - fast
- Non-metric MDS:
  - Squeezes data points on table
  - respects only rankings of distances
  - (locally) solves clear objective
  - slow

R commands to know

- cmdscale included in standard R distribution
- isoMDS from package “MASS”