Exploration visuelle des données

Nicoleta ROGOVSCHI

nicoleta.rogovschi@parisdescartes.fr

M2-INFO

Multi-Dimensional Scaling (MDS)

Plan du cours

- Introduction et définitions
- Formulation du problème
- Algorithme
- Exemple
- Conclusions

Réduction des dimensions par extraction de caractéristiques

Deux grandes familles de méthodes :

- **Méthodes linéaires**
  - Analyse en Composantes Principales (ACP)
  - Analyse Discriminante Linéaire (ADL)
  - → Multi-Dimensional Scaling (MDS)
  - …
- **Méthodes non-linéaires**
  - Isometric feature mapping (Isomap)
  - Locally Linear Embedding (LLE)
  - Kernel PCA
  - Segmentation spectrale (spectral clustering)
  - Methodes supervisées (S-Isomap)
  - …

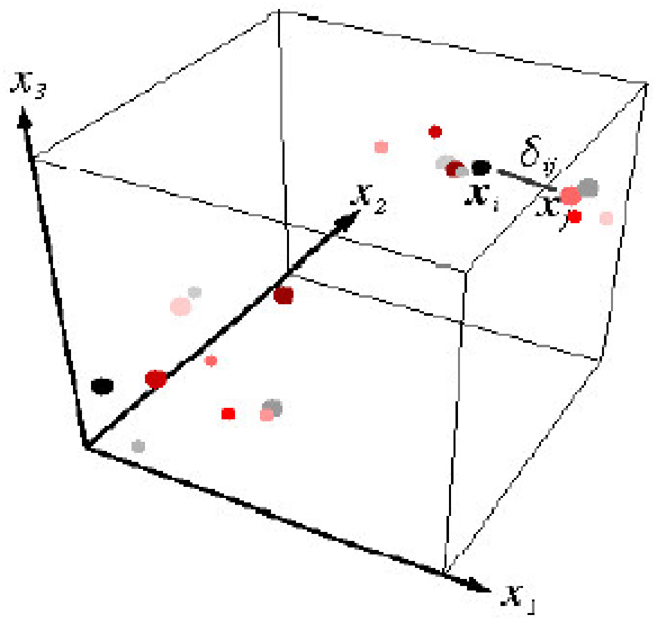
Introduction

- Multi-dimensional scaling (MDS) *(proposée par Borg et Groenen en 1997)*
  - Un ensemble de techniques de réduction de la dimension qui projettent les distances entre les observations d'un espace à grandes dimensions dans un espace de petites dimensions
  - Des techniques qui trouvent une configuration des points dans un espace de faible dimensions dont les distances inter-points correspondent aux dissimilarités dans les grandes dimensions

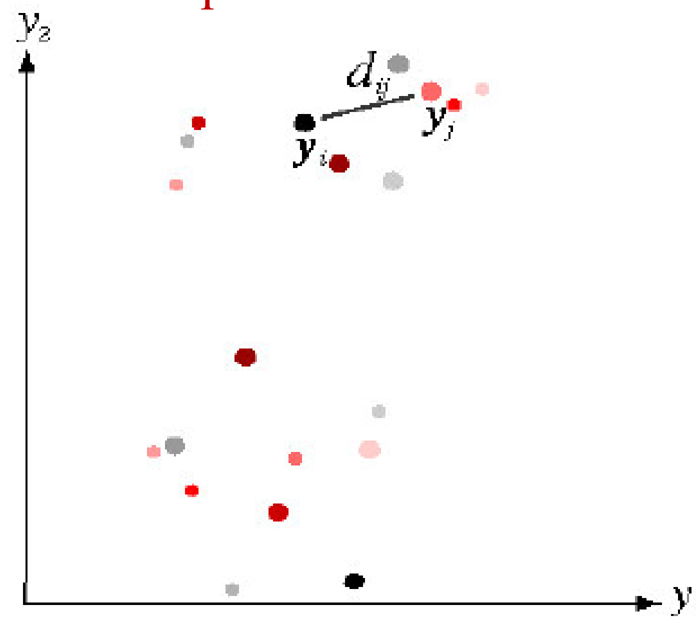
Introduction

- Capables de modéliser des structures intrinsèques complexes et de les visualiser

Espace de départ



Espace cible



Multi-Dimensional Scaling (MDS)

Dans de nombreuses applications :

- On connaît les distances entre les points d'un ensemble de données
- On cherche à obtenir une représentation en faible dimension de ces points

La méthode de positionnement multidimensionnel (MDS) nous permet de construire cette représentation

Multi-Dimensional Scaling (MDS)

- Exemple :

Obtenir la carte d'un pays en partant de la connaissance des distances entre chaque paire de villes.

- Comme l'ACP, l'algorithme MDS est basé sur la recherche des valeurs propres.

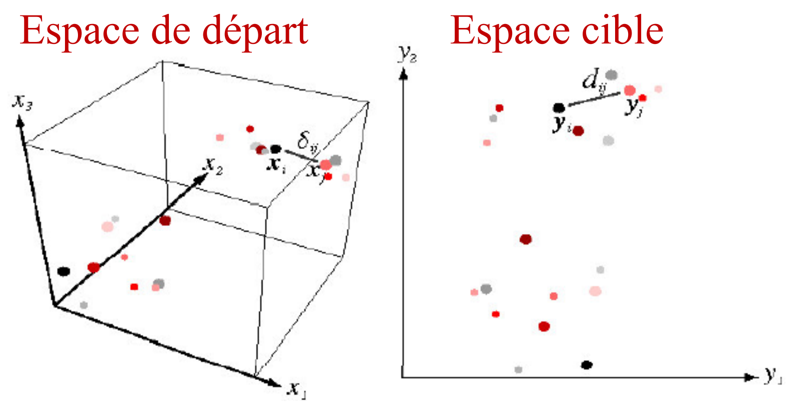
- MDS permet de construire une configuration de n points dans R^d à partir des distances entre N objets.

Multi-Dimensional Scaling (MDS)

- On a donc N(N-1)/2 distances. Il est toujours possible de générer un positionnement de N points en N dimensions qui respecte exactement les distances fournies.

- MDS calcule une approximation en dimensions $d<N$

Formulation du problème



- Soit
  - Les points $x_1,..,x_n$ dans **k** dimensions
  - On note par δ_{ij} la distance entre les points x_i et x_j
- On doit trouver
  - Les points $y_1,\ldots,y_n$ dans un espace de 2 ou 3 dimensions, tel que la distance d_{ij} entre y_i et y_j, soit proche de δ_{ij}

Fonction de coût

- On doit chercher δ_{ij} qui minimise une fonction objective
- On peut définir la fonction de coût de manière générale:

$$Fonction\_de\_coût = \sum_{i<j} (d_{ij} - \delta_{ij})^2$$

$$\delta_{ij} = \| x_i - x_j \|^2$$

$$d_{ij} = \| y_i - y_j \|^2$$

Exemples de fonctions de coût

- Les fonctions de coût possibles (Stress)
  - d_{ij} est une fonction de y_i et y_j, et pour un ensemble de données les δ_{ij} sont constant.

Disparité

$$J_{aa} = \frac{\sum_{i<j} (d_{ij} - \delta_{ij})^2}{\sum_{i<j} \delta_{ij}^2}$$ pénalise les erreurs absolues

$$J_{rr} = \sum_{i<j} \left(\frac{d_{ij} - \delta_{ij}}{\delta_{ij}} \right)^2$$ pénalise les erreurs relatives

$$J_{ar} = \frac{1}{\sum_{i<j} \delta_{ij}} \sum_{i<j} \frac{(d_{ij} - \delta_{ij})^2}{\delta_{ij}}$$ un compromis entre les deux

Critère de Sammon

Règles de mise à jour

- Règles de mise à jour

$$\nabla J_{aa}(y_k) = \frac{2}{\sum_{i<j} \delta_{ij}^2} \sum_{j \neq k} (d_{kj} - \delta_{kj}) \frac{y_k - y_j}{d_{kj}}$$

$$\nabla J_{rr}(y_k) = 2 \sum_{j \neq k} \frac{d_{kj} - \delta_{kj}}{\delta_{kj}^2} \frac{y_k - y_j}{d_{kj}}$$

$$\nabla J_{ar}(y_k) = \frac{2}{\sum_{i<j} \delta_{ij}} \sum_{j \neq k} \frac{d_{kj} - \delta_{kj}}{\delta_{kj}} \frac{y_k - y_j}{d_{kj}}$$

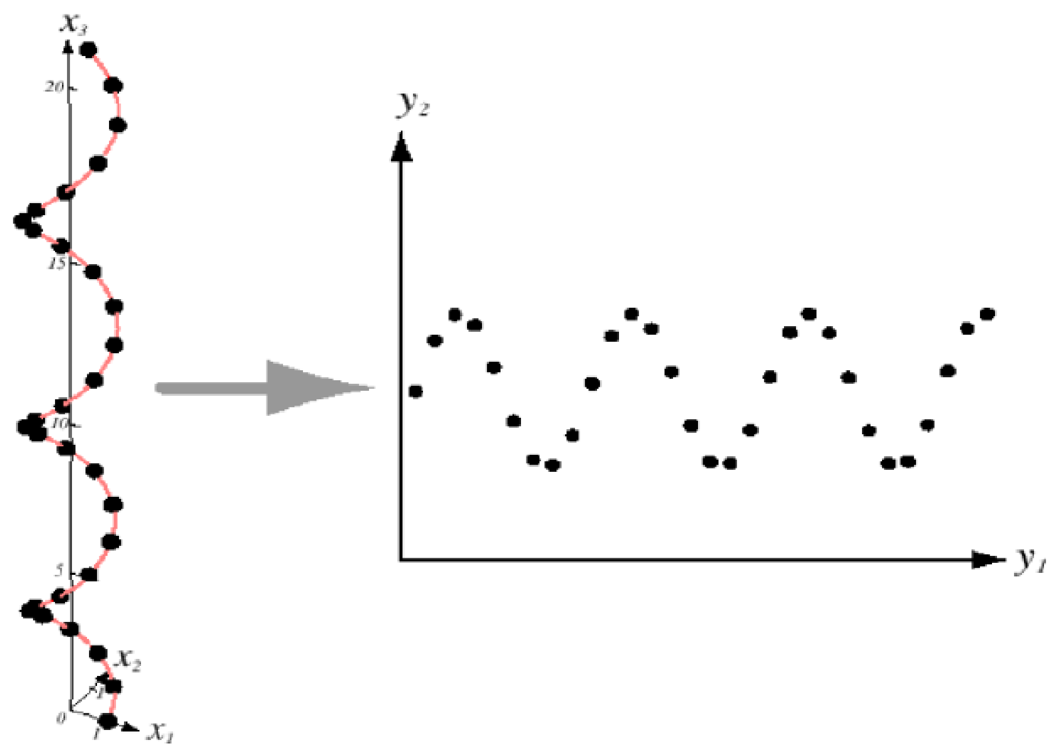
Algorithme

- On calcule ou on dispose dès le départ des distances δ_{ij}
- On initialise les points $y_1, \ldots, y_n$ (d'une manière aléatoire, par exemple)
- On tourne l'algorithme jusqu'à convergence,

$$\forall i \quad y_i \leftarrow y_i - \eta \nabla J(y_i) \quad (0 < \eta < 1)$$

Exemple

- Jeu de données artificiel : on passe d'un espace à 3 dimensions à un espace de 2 dimensions

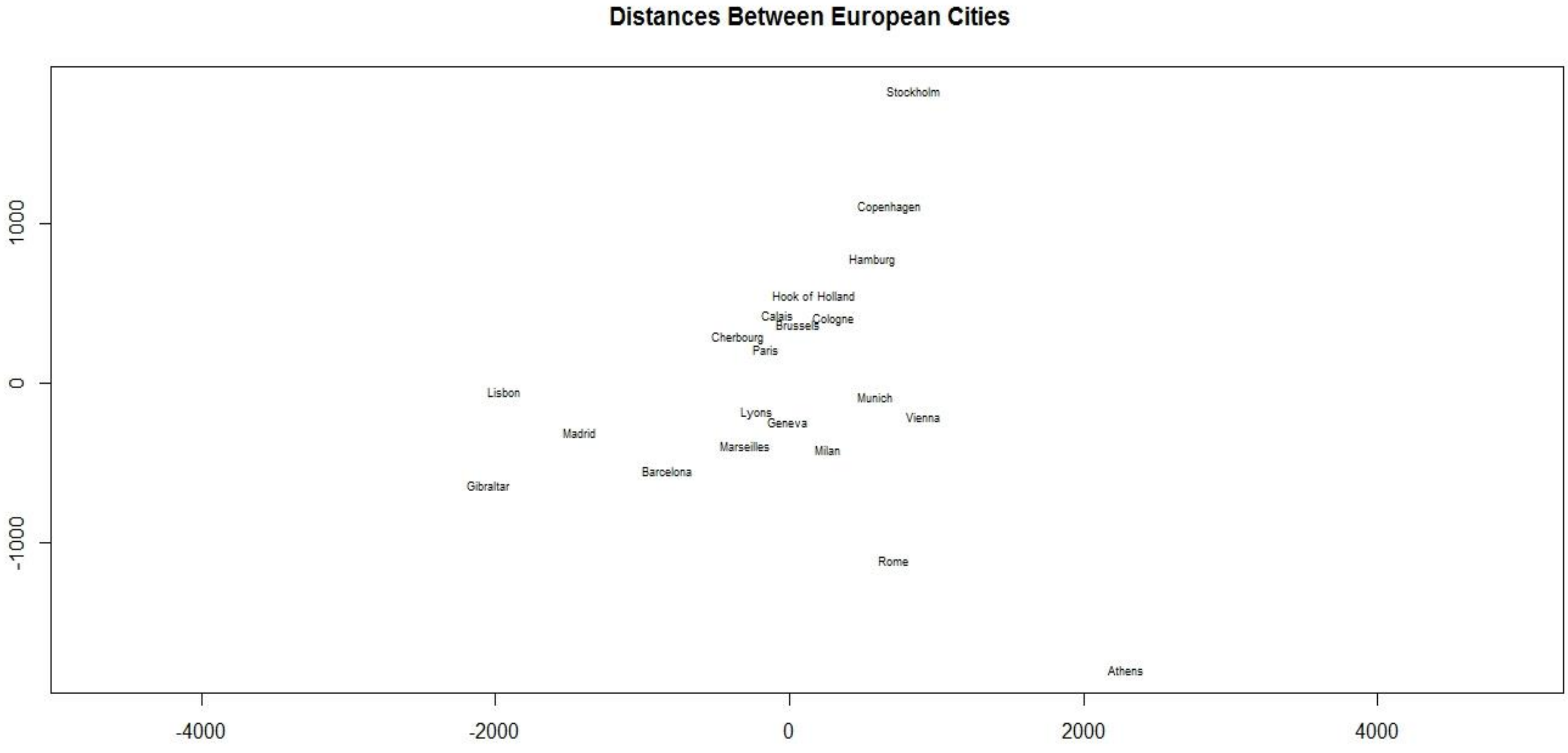


Exemple

- Le jeu de données «Eurodist» représente la distance (en km) entre 21 villes de l'Europe.
- Le jeu de données de départ doit être représente sous forme d'une matrice carrée des dissimilarités entre les variables.

	Athens	Barcelona	Brussels	Calais	Cherbourg	Cologne	Copenhagen	Geneva	Gibraltar	Hamburg
Barcelona	3313									
Brussels	2963	1318								
Calais	3175	1326	204							
Cherbourg	3339	1294	583	460						
Cologne	2762	1498	206	409	785					
Copenhagen	3276	2218	966	1136	1545	760				
Geneva	2610	803	677	747	853	1662	1418			
Gibraltar	4485	1172	2256	2224	2047	2436	3196	1975		
Hamburg	2977	2018	597	714	1115	460	460	1118	2897	

Exemple



Distances Between European Cities
Stockholm
Copenhagen
Hamburg
Hook of Holland
Calais
Brussels
Cologne
Cherbourg
Paris
Lisbon
Munich
Lyons
Geneva
Vienna
Madrid
Marseilles
Milan
Barcelona
Gibraltar
Rome
Athens
1000
0
-1000
-4000
-2000
0
2000
4000

Conclusions

- Les algorithmes MDS diffèrent par :
  – La distance utilisé dans l'espace de départ
  – Les fonctions objectives (Stress), l'utilisation de différentes fonctions de stress mènes à des résultats variés
  – La procédure d'optimisation; les MDS linéaires ne peuvent pas bien modéliser des variétés non-linéaires, tandis que les MDS non-linéaires ont besoin souvent d'utiliser un algorithme itératif