Multidimensional Scaling for large datasets

Cristian Pachón-García

(joint work with Pedro Delicado)

Universitat Politècnica de Catalunya

GRBIO, 12 July 2024

Introduction to Multidimensional Scaling

⇒ From coordinates to distances ⇒

⇒

⇐

	Amsterdam														
Atenas	3070	**Atenas**													
Barcelona	1588	3409	**Barcelona**												
Belgrado	1831	1239	2170	**Belgrado**											
Berlin	685	2556	1829	1317	**Berlin**										
Bruselas	198	3025	1409	1786	789	**Bruselas**									
Bucarest	2328	1299	2883	713	1771	2283	**Bucarest**								
Budapest	1442	1634	2102	395	922	1397	886	**Budapest**							
Burdeos	1036	3329	595	2090	1661	857	2803	2010	**Burdeos**						
Copenhague	803	3017	2087	1778	461	968	2134	1383	1840	**Copenhague**					
Dublin	878	3933	2112	2694	1530	908	3191	2305	1560	1681	**Dublin**				
Estambul	2771	1205	3110	940	2257	2726	729	1335	3030	2718	3634	**Estambul**			
Estocolmo	1427	3626	2711	2837	1070	1592	2774	1992	2464	624	2305	3327	**Estocolmo**		
Estrasburgo	674	2603	1502	1364	775	434	2077	1087	962	1033	1342	2304	1657	**Estrasburgo**	
Ginebra	1038	2698	808	1459	1139	817	2172	1294	716	1397	1570	2399	2021	383	**Ginebra**

⇐ From distances to coordinates ⇐

Multidimensional Scaling:

Dimensionality reduction based on inter-individual distances

Multidimensional Scaling (MDS)

- MDS is a family of dimensionality reduction techniques.
- Input: **D**, a $n \times n$ distance matrix between n observed objects, $\mathcal{O}_1, \ldots, \mathcal{O}_n$, elements of a metric space Ω equipped with a distance function d:

$$d_{ij} = d(\mathcal{O}_i, \mathcal{O}_j).$$

- Output: **X**, a $n \times q$ matrix, q *small*, a **low-dimensional configuration** for **D**, with rows $\mathbf{x}_i^{\top}$, $i = 1, \ldots, n$, such that

$$\delta_{ij} \approxeq d_{ij}$$

  where $\delta_{ij} = \|\mathbf{x}_i - \mathbf{x}_i\|$.
- MDS: *From distances to coordinates*.

Classical metric scaling

- Define the $n \times n$ matrix $\mathcal{D}$ with element (i, j) equal to d_{ij}^2.
- Let $\mathbf{H} = \mathbf{I}_n - (1/n)\mathbf{1}_n\mathbf{1}_n^\top$ be the $n \times n$ centering matrix.
- Then $\mathbf{Q} = -\frac{1}{2}\mathbf{H}\mathcal{D}\mathbf{H}$ is the *inner products* matrix.
- Take the spectral decomposition $\mathbf{Q} = \mathbf{V}\Lambda\mathbf{V}^\top$.
  - Attention: In general cases, some eigenvalues can be negative.
  - Assume that $\lambda_1 \geq \ldots \geq \lambda_q > 0$.
- Define $\tilde{\mathbf{X}}_q = \mathbf{V}_q\Lambda_q^{1/2}$, where $\mathbf{V}_q$ is formed by the first q columns of $\mathbf{V}$ and $\Lambda_q = diag(\lambda_1, \ldots, \lambda_q)$.
- Then $\mathbf{Q} \approx \tilde{\mathbf{X}}_q\tilde{\mathbf{X}}_q^\top$ and $\tilde{\mathbf{X}}_q$ is the q-dimensional configuration obtained from $\mathbf{D}$.

Example

Consider the distance between some cities of Europe, as shown in the following matrix:

	Athens	Barcelona	Brussels	Calais	Cherbourg	⋯
Athens	0	3313	2963	3175	3339	⋯
Barcelona	3313	0	1318	1326	1294	⋯
Brussels	2963	1318	0	204	583	⋯
Calais	3175	1326	204	0	460	⋯
Cherbourg	3339	1294	583	460	0	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋱

Table: Distances between European cities (just 5 of them are shown).

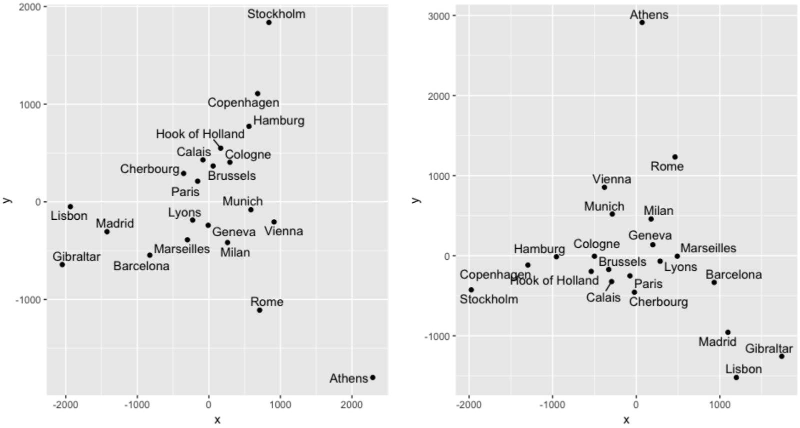


Figure: Two MDS configurations for European cities.

Non-classical metric scaling

- Let $\mathbf{D} = (d_{ij})_{i,j=1}^{n}$ be the inter-individual distance matrix.
- Fix a tentative dimension q and a $n \times q$ matrix $\mathbf{X}$. Let $\delta_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$, the Euclidean norm between rows i and j of $\mathbf{X}$.
- Metric *STRESS* (STandardized REsidual Sum of Squares):

$$\text{STRESS}_M(\mathbf{D}, \mathbf{X}) = \frac{\sum_{i<j}(\delta_{ij} - d_{ij})^2}{\sum_{i<j} d_{ij}^2}.$$

  It is a measure of the relative error made when matrix $\mathbf{X}$ is considered as a configuration for the distance matrix $\mathbf{D}$.
- Non-classical metric scaling problem:

$$\min_{\mathbf{X} \in \mathbb{R}^{n \times q}} \text{STRESS}_M(\mathbf{D}, \mathbf{X}).$$

MDS for Big Data

- When n is large, standard MDS algorithms are prohibitively memory and time consuming.
- Classical metric scaling:
  - MDS depends on eigendecomposition. The cost of it is $O(n^3)$ (Trefethen and Bau 1997).
  - It needs to store a n^2 distance matrix.
- Non-classical metric scaling: Optimization problem.
  - Number of decision variables: $O(n)$.
  - Evaluation of the objective variable, cost $O(n^2)$.
  - It needs to store a n^2 distance matrix.
  - Two algorithms with cost $O(n^2)$:
    - Majorization algorithm (SMACOF; Borg and Groenen 2005). Better using the classic MDS configuration as starting point.
    - Stochastic gradient descent, Zheng, Pawar, and Goodman (2019). See also Börsig, Brandes, and Pasztor (2020).

Algorithms for MDS with Big Data

- Several algorithms have appeared in the field of MDS as well as in that of graph viewing.
- Delicado and Pachón-García (2024)[1] review some of them and introduce two new proposals.
- Detailed analysis is given for 6 algorithms:
  - Existing algorithms:
    - *Landmark MDS* (De Silva and Tenenbaum 2004, LMDS).
    - *Fast MDS* (Yang, Liu, McMillan, and Wang 2006).
    - *Pivot MDS* (Brandes and Pich 2007).
    - *Reduced MDS* (Paradis 2021, RMDS).
  - Our proposals:
    - *Divide-and-conquer MDS*
    - *Interpolation MDS*
- All of them have computing time $O(n)$, except *Fast MDS* which is $O(n \log n)$.

[1] Delicado, P., Pachón-García, C. (2024) Multidimensional scaling for big data. *Adv. Data Anal. Classif.* 10.1007/s11634-024-00591-9

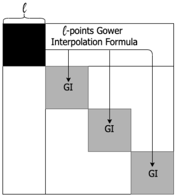
These algorithms use one of two different approaches:

1. Select a moderated large subset of subjects, run MDS on that subset to obtain a low-dimensional configuration for it, and then project all other subjects into that configuration:
   - *Landmark MDS*
   - *Interpolation MDS*
   - *Reduced MDS*
   - *Pivot MDS*

   Note: They were designed to be used with Classical MDS.
2. Divide the data set into many moderated large subsets of subjects, run MDS on each subset to obtain the corresponding many low-dimensional configurations, and then combine them to create a unique global configuration:
   - *Divide-and-conquer MDS*
   - *Fast MDS (recursive)*

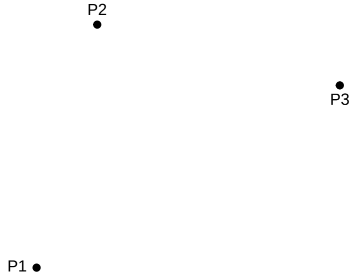
   Note: They can be easily adapted to any MDS technique.

Interpolation MDS



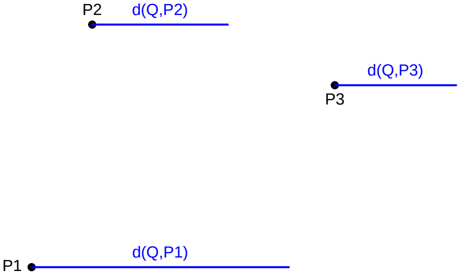
- Select ℓ random elements of the data set ($\ell << n$).
- Perform classical MDS over this subset.
- Extend the obtained results to the rest of data set, in blocks of ℓ data, by using **Gower's interpolation formula** (Gower 1968).

Gower's interpolation: Where to place a new point **Q**?



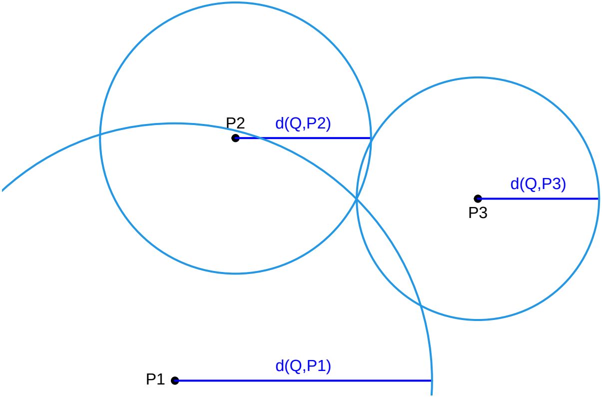
Three points, exact distances $d(Q, P_i)$.

Gower's interpolation: Where to place a new point **Q**?



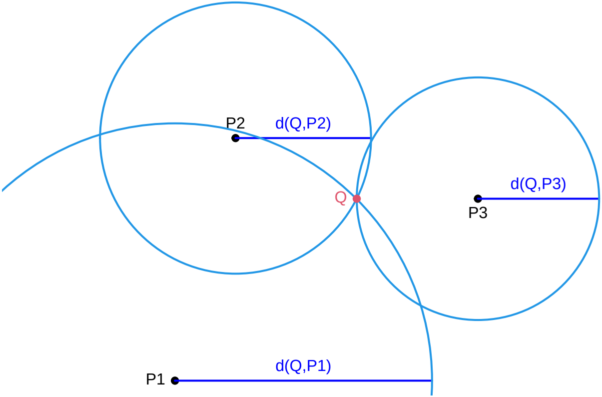
Three points, exact distances $d(Q, P_i)$.

Gower's interpolation: Where to place a new point **Q**?



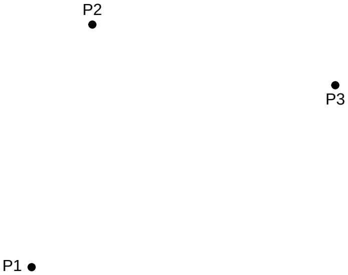
Three points, exact distances $d(Q, P_i)$.

Gower's interpolation: Where to place a new point **Q**?



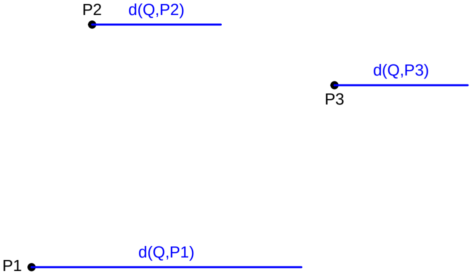
Three points, exact distances $d(Q, P_i)$.

Gower's interpolation: Where to place a new point **Q**?



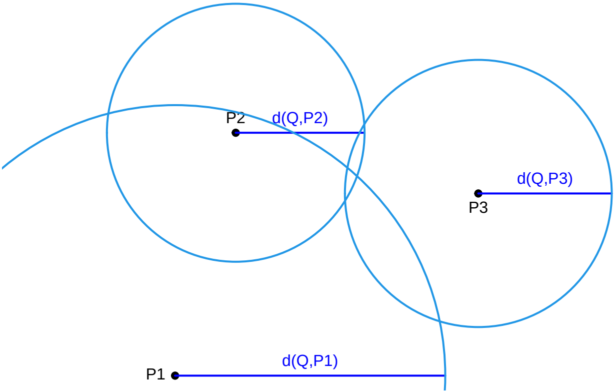
Three points, approximate distances $d(Q, P_i)$.

Gower's interpolation: Where to place a new point **Q**?



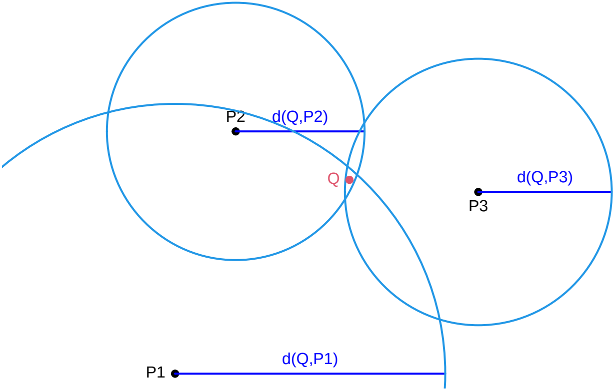
Three points, approximate distances $d(Q, P_i)$.

Gower's interpolation: Where to place a new point **Q**?



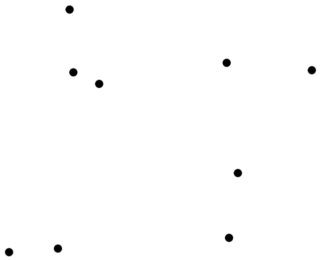
Three points, approximate distances $d(Q, P_i)$.

Gower's interpolation: Where to place a new point **Q**?



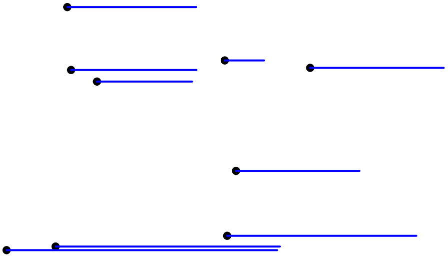
Three points, approximate distances $d(Q, P_i)$.

Gower's interpolation: Where to place a new point **Q**?



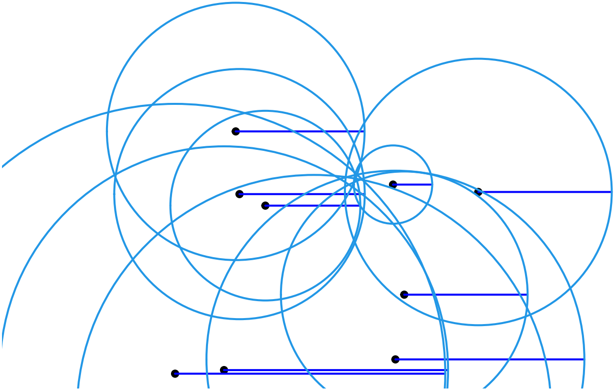
n points, approximate distances $d(Q, P_i)$.

Gower's interpolation: Where to place a new point **Q**?



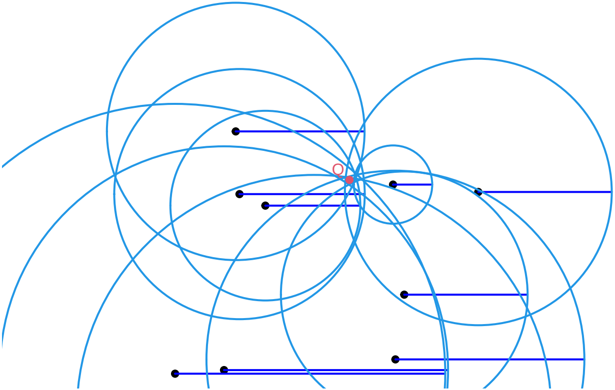
n points, approximate distances $d(Q, P_i)$.

Gower's interpolation: Where to place a new point **Q**?



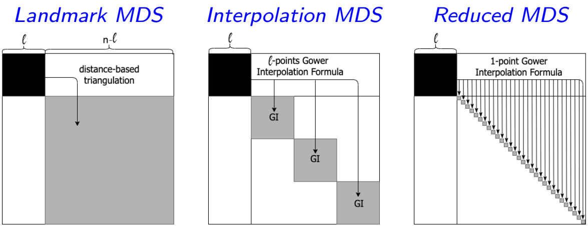
n points, approximate distances $d(Q, P_i)$.

Gower's interpolation: Where to place a new point **Q**?



n points, approximate distances $d(Q, P_i)$.

Three related algorithms



Proposition

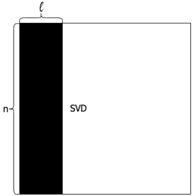
Distance-based triangulation procedure used in LMDS coincides with Gower's interpolation formula.

Different selection of the initial data subset:

- LMDS uses a MaxMin greedy optimization procedure.
- Interpolation MDS, random selection.
- RMDS, heuristic rules to ensure both central and peripheral data.

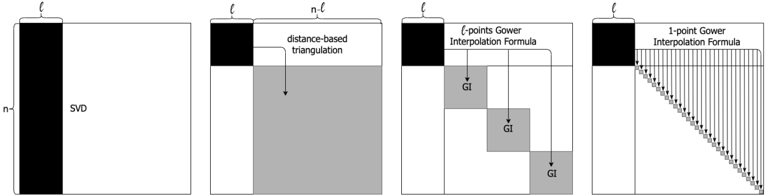
Pivot MDS (Brandes and Pich 2007).

- It is an approximation of classical MDS, based on the selection of a subset of ℓ *pivot points*.
- Let $\mathbf{C}$ be the $n \times \ell$ submatrix of $\mathbf{Q}$ containing the inner products between the pivot points and all the other points.
- The SVD of $\mathbf{C}$ is used to approximate that of $\mathbf{Q}$, whose q first eigenvectors are the pivot MDS low dimensional configuration.
- Recall that LMDS, interpolation MDS, and reduced MDS are based on the eigendecomposition of the $\ell \times \ell$ submatrix of $\mathbf{Q}$ containing only inner products of landmark points.

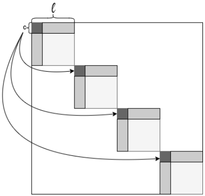


Pivot MDS (Brandes and Pich 2007).

- It is an approximation of classical MDS, based on the selection of a subset of ℓ *pivot points*.
- Let $\mathbf{C}$ be the $n \times \ell$ submatrix of $\mathbf{Q}$ containing the inner products between the pivot points and all the other points.
- The SVD of $\mathbf{C}$ is used to approximate that of $\mathbf{Q}$, whose q first eigenvectors are the pivot MDS low dimensional configuration.
- Recall that LMDS, interpolation MDS, and reduced MDS are based on the eigendecomposition of the $\ell \times \ell$ submatrix of $\mathbf{Q}$ containing only inner products of landmark points.

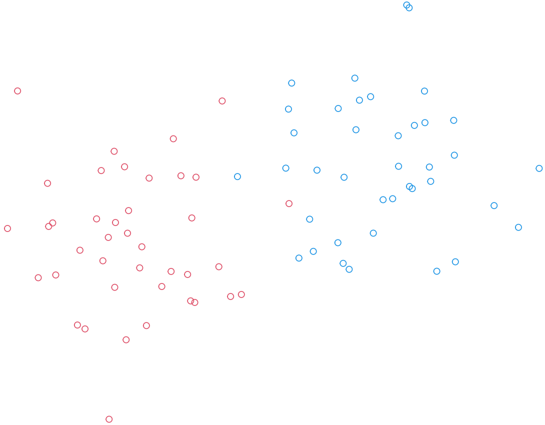


Divide-and-conquer MDS



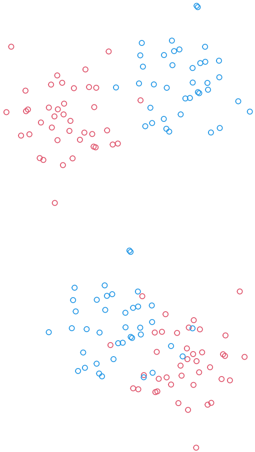
- The large data set is divided into small parts with ℓ individuals ($\ell << n$).
- All parts have c individuals in common (*connecting points*).
- MDS is performed over every part.
- The partial configurations are combined so that all the points lie on the same coordinate system.
- Connections are done one at a time by a Procrustes transformation (Borg and Groenen 2005, Chapter 20) of the c connecting points.

Procrustes transformation



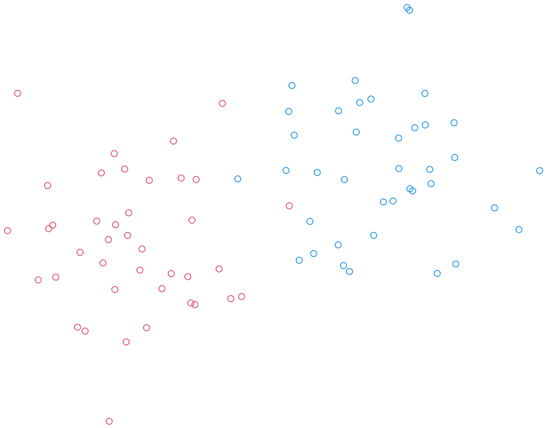
Two partial configurations from two non-linked data subsets

Procrustes transformation



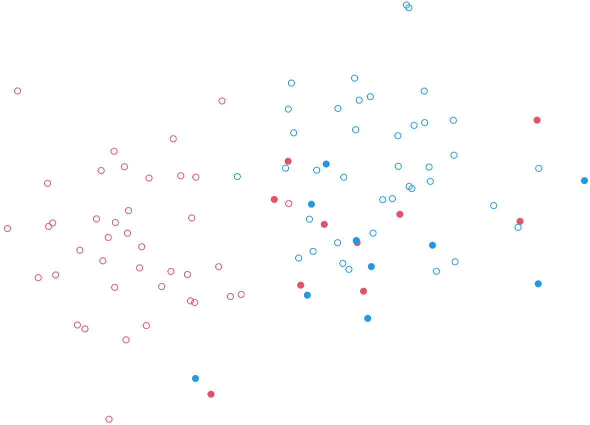
Two partial configurations: Multiple relative positions are possible

Procrustes transformation



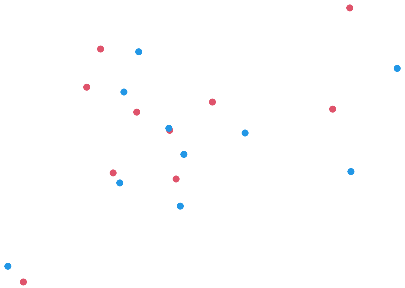
Two partial configurations from two non-linked data subsets

Procrustes transformation



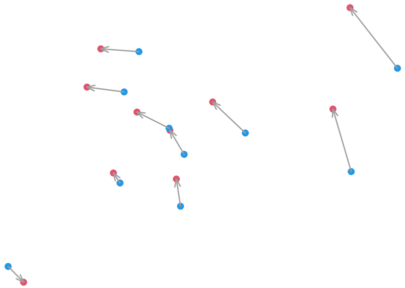
Two partial configurations: *connecting points* included

Procrustes transformation



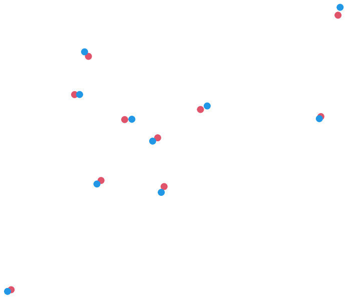
Compute the rigid transformation linking the *connecting points*

Procrustes transformation



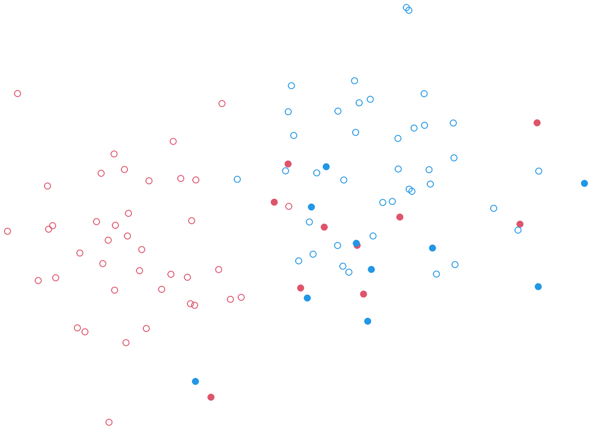
Compute the rigid transformation linking the *connecting points*

Procrustes transformation



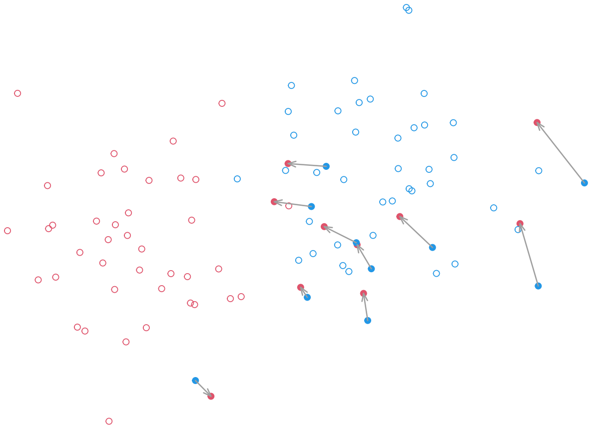
Compute the rigid transformation linking the *connecting points*

Procrustes transformation



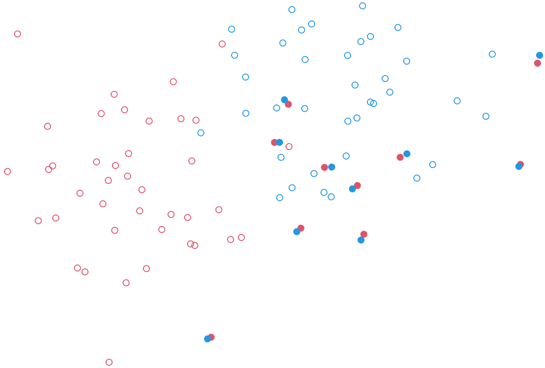
Apply the Procrustes transformation to the entire configuration 2

Procrustes transformation



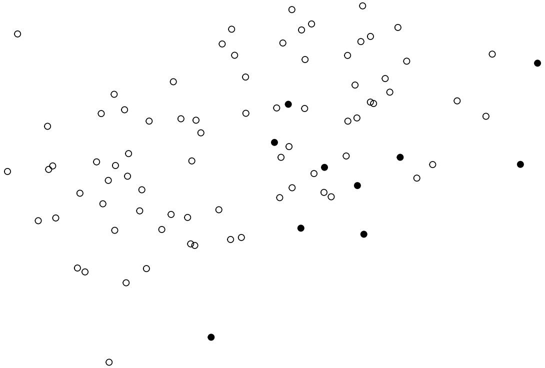
Apply the Procrustes transformation to the entire configuration 2

Procrustes transformation



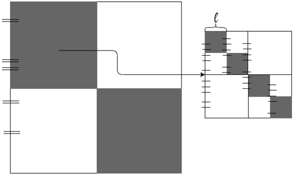
Apply the Procrustes transformation to the entire configuration 2

Procrustes transformation



A common configuration for two subsets, with connecting points

Fast MDS (Yang, Liu, McMillan, and Wang 2006). It overcomes the problem of MDS scalability using recursive programming in combination with a data set splitting strategy.



Procrustes transformations are used at each recursive step to connect the low dimensional configurations obtained for different subsets.

bigmds: An R package to do MDS with big data

- We published an R package in CRAN in 2021: `https://cran.r-project.org/web/packages/bigmds`.
- 14000 downloads since then.
- The core of the package consists of six methods:
  - landmark_mds
  - interpolation_mds
  - reduced_mds
  - pivot_mds
  - divide_and_conquer_mds
  - fast_mds
- We also implemented a Procrustes function.
- Instead of using cmdscale function for classical MDS, we use trlan.eigen function (from svd package) to perform the spectral decomposition of matrices containing inner products: *8 seconds against 15 minutes for sample size $n = 10000$.*

1. Introduction to Multidimensional Scaling
2. Algorithms for MDS with Big Data
3. A simulation study
4. A real case: The EMNIST data set
5. Conclusions

A simulation study

Simulation design:

- Sample size: 5000, 10000, 20000, 100000, 250000, 500000, 750000, and 1000000.
- Data dimension: 10 or 100.
- *Dominant dimensions:* 2 or 10, the number of columns with a variance much higher than the variance of the remaining *noisy dimensions*.

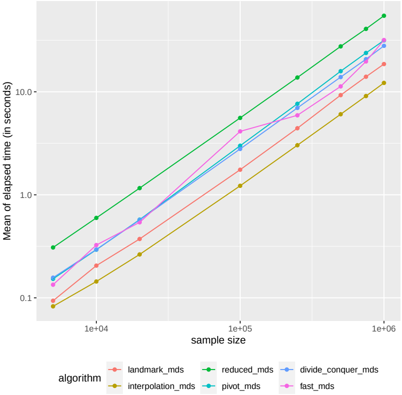
A total of 32 scenarios, each replicated 100 times.

Correlation with the true main dimensions

Quantiles of order 2.5% ($q_{0.025}$) and 97.5% ($q_{0.975}$), and mean values for the correlation coefficients between the original variables and the ones recovered by the six MDS methods.

Algorithm	$q_{0.025}$	mean	$q_{0.975}$
LMDS	0.99869	0.99950	1
Interpolation MDS	0.99868	0.99949	1
RMDS	0.99868	0.99949	1
Pivot MDS	0.99621	0.99824	0.99988
Divide-and-conquer MDS	0.99774	0.99845	0.99915
Fast MDS	0.98278	0.99417	0.99886

Results on computing time



Elapsed time (in seconds)

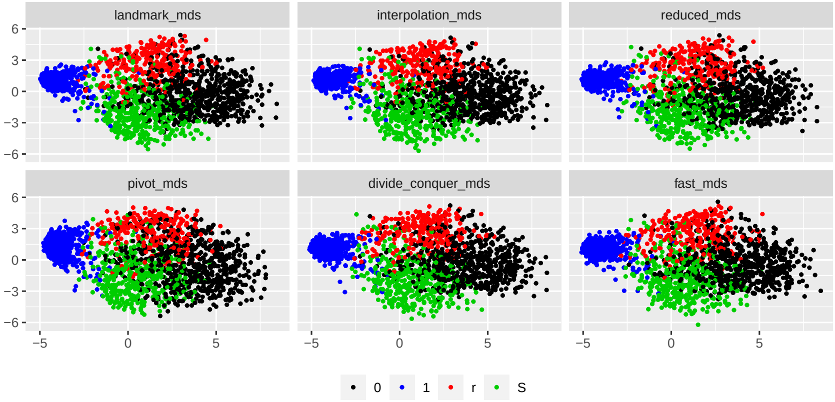
Algorithm	$q_{0.025}$	*mean*	$q_{0.975}$
LMDS	23.46	24.27	24.82
Interp MDS	18.21	18.34	18.48
RMDS	91.74	92.20	93.01
Pivot MDS	36.19	37.38	38.04
D-&-C MDS	44.57	45.06	45.74
Fast MDS	61.51	61.74	61.97

$n = 10^6$, $p = 100$, $q = 10$

1. Introduction to Multidimensional Scaling
2. Algorithms for MDS with Big Data
3. A simulation study
4. A real case: The EMNIST data set
5. Conclusions

A real case: The EMNIST data set

- The EMNIST data set (Cohen, Afshar, Tapson, and van Schaik 2017) is composed by handwritten character digits, lowercase letters and capital letters.
- In total, there are 814,255 images divided into 62 classes:
  - 10 digits (from '0' to '9'; the 49.5% of the total).
  - 26 lowercase letters (from 'a' to 'z'; 23.5%).
  - 26 capital letters (from 'A' to 'Z'; 27%).
- Each image is of size 28×28.


landmark_mds
interpolation_mds
reduced_mds
pivot_mds
divide_conquer_mds
fast_mds
6
3
0
−3
−6
−5
0
5
0
1
r
S

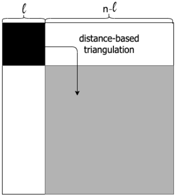
Conclusions and additional comments

- The standard MDS algorithms are not able to deal with large datasets: problems in memory and/or computing time.
- There are algorithms to overcome these difficulties.
- Two approaches: Gower's interp., or Procrustes transf.
- We have presented six of these algorithms, as well as a package in R: **bigmds**.
- In our simulation study:
  - The six MDS algorithms provide low-dimensional configurations similar to those eventually given by the classical MDS algorithm.
  - Interpolation MDS is the fastest method.
  - LMDS and pivot MDS could present memory problems.

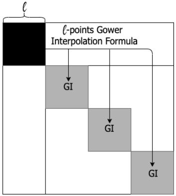
Additional comments

- Further research:
  - To combine these algorithms with other dimensionality reduction methods.
  - Non-classical metric scaling, Local MDS, ISOMAP, t-SNE, UMAP, among other.
- Alternative dimensionality reduction tools in R:
  - dimRed (Kraemer, Reichstein, and Mahecha 2018)
    18 methods.
  - Rdimtools (You and Shung 2022)
    143 methods. Very fast (it uses a C++ linear algebra library).

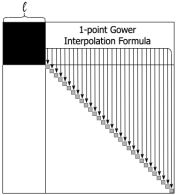
Thank You!



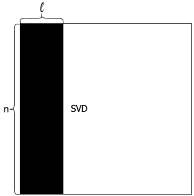
Landmark MDS



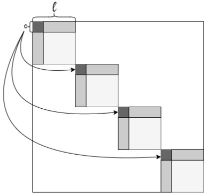
Interpolation MDS



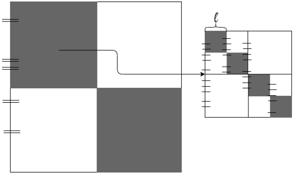
Reduced MDS



Pivot MDS



Divide-and-conquer MDS



Fast MDS

Borg, I. and P. Groenen (2005).
Modern Multidimensional Scaling: Theory and Applications.
Springer.

Börsig, K., U. Brandes, and B. Pasztor (2020).
Stochastic gradient descent works really well for stress minimization.
In D. Auber and P. Valtr (Eds.), *Graph Drawing and Network Visualization*, Cham, pp. 18–25. Springer International Publishing.

Brandes, U. and C. Pich (2007).
Eigensolver methods for progressive multidimensional scaling of large data.
In M. Kaufmann and D. Wagner (Eds.), *Graph Drawing*, Berlin, Heidelberg, pp. 42–53. Springer Berlin Heidelberg.

Cohen, G., S. Afshar, J. Tapson, and A. van Schaik (2017).
Emnist: Extending mnist to handwritten letters.
In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926.

De Silva, V. and J. B. Tenenbaum (2004).
Sparse multidimensional scaling using landmark points.
Technical report, Technical Report, Stanford University.

Delicado, P. and C. Pachón-García (2024).
Multidimensional scaling for big data.
Advances in Data Analysis and Classification Available online(Open access), 1–22.

Gower, J. C. (1968).
Adding a point to vector diagrams in multivariate analysis.
Biometrika 55(3), 582—585.

Kraemer, G., M. Reichstein, and M. D. Mahecha (2018).
dimRed and coRanking—unifying dimensionality reduction in r.
The R Journal 10(1), 342–358.
coRanking version 0.2.6.

Paradis, E. (2021).
Reduced multidimensional scaling.
Computational Statistics 37, 91–105.

Trefethen, L. N. and D. Bau (1997).
Numerical Linear Algebra.
Society for Industrial and Applied Mathematics.

Yang, T., J. Liu, L. McMillan, and W. Wang (2006).
A fast approximation to multidimensional scaling.
In *Proceedings of the ECCV Workshop on Computation Intensive Methods for Computer Vision (CIMCV)*.

You, K. and D. Shung (2022).
Rdimtools: An R package for dimension reduction and intrinsic dimension estimation.
Software Impacts 14, 100414.

Zheng, J. X., S. Pawar, and D. F. M. Goodman (2019).
Graph drawing by stochastic gradient descent.
IEEE Transactions on Visualization and Computer Graphics 25(9), 2738–2748.