UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

DISTANCE-BASED DIMENSIONALITY REDUCTION FOR BIG DATA

ADRIÀ CASANOVA LLOVERAS

Thesis supervisor
PEDRO DELICADO USEROS (Department of Statistics and Operations Research)

Thesis co-supervisor
CRISTIAN PACHÓN GARCIA (Department of Statistics and Operations Research)

Degree
Master's Degree in Data Science

Master's thesis

Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

Abstract

Dimensionality reduction aims to project a data set into a low-dimensional space. Many techniques have been proposed, most of them based on the inter-individual distance matrix. When the number of individuals is really large, the use of distance matrices is prohibitive. There are algorithms that extend MDS (a classical dimensionality reduction method based on distances) to the big data setting. In this TFM, we adapt these algorithms to any generic distance-based dimensionality reduction method.

Contents

1 Introduction, Motivation, and Objectives

1.1 Introduction

- Dimensionality Reduction definition, goal and applications.
- Examples of DR methods.
- Key points and limitations of DR methods.

1.2 Motivation

- When the number of individuals is really large, the use of distance matrices is prohibitive.
- There are algorithms that extend MDS to the big data setting.

1.3 Objectives

- Adapt these algorithms to any generic distance-based dimensionality reduction method.

2 State of the Art

2.1 Introduction

- There are many DR algorithms, linear and non-linear, but they use the distance matrix of datapoints. In big datasets, this matrix cannot fit in the system's RAM, so DR methods are not feasible. Moreover, time complexity can be prohibitive in some cases as well.

- Delicado and Pachón-García proposed new versions of MDS that handled this problem and compared them with prior algorithms [3].

- Regarding non-linear methods, Landmark Isomap [5] was proposed to adapt Isomap to large data settings. Later, in 2024, Reichmann, Hägele and Weiskopf generalized that algorithm to any DR method that would return a map between high- and low-dimensional spaces, such as t-SNE, UMAP, Autoencoders or linear methods (PCA, MDS).

- Finally, t-SNE has been specifically adapted to big data environments in Python through iterative implementations. Their details, however, are out of the scope of our project.

2.2 A few Dimensionality Reduction Techniques

2.2.1 Non-classical MDS. The SMACOF algorithm

SMACOF (Scaling by MAjorizing a COmplicated Function) algorithm is a multidimensional scaling algorithm which minimizes metric stress using a majorization technique [1]. Also known as the Guttman Transform, this technique is more powerful than general optimization methods such as gradient descent.

2.2.2 Local MDS

Local MDS [2] is a variant of MDS that handles small and large distances differently. Specifically, for large distances, a repulsive term is added to the stress function to separate points in the low-dimensional configuration.

Parameters k and τ can be tuned with k'-cross validation thanks to the LCMC (Local Continuity Meta-Criteria) (*POSSIBLE ANNEX*)

2.2.3 Isomap

Isomap [6] is a nonlinear technique that preserves geodesic distances between points in a manifold. The key insight of Isomap is that large distances between objects are estimated from the shorter ones, by the shortest path length. Then, shorter and estimated-larger distances have the same importance in a final MDS step.

The only tuning parameter of Isomap is the bandwith (ϵ or k), but there is no consensus on what is the best method to choose it.

Algorithm 1 SMACOF

Require: $D_{\mathcal{X}} = (\delta_{ij})$, the matrix of observed distances; q, the embedding's dimensionality; maximum number of iterations n_iter; convergence threshold ϵ.

Ensure: A configuration in a q-dimensional space $\tilde{\mathcal{Y}}$.

1: Initialize $\tilde{\mathcal{Y}}^{(0)} \in \mathbb{R}^{n \times q}$

2: $k \leftarrow 0$

3: **repeat**

4: Compute distance matrix of $\tilde{\mathcal{Y}}^{(k)}$: $d_{ij}(\tilde{\mathcal{Y}}^{(k)}) = \|x_i - x_j\|$

5: Compute the Metric STRESS: $STRESS_M(D_{\mathcal{X}}, \tilde{\mathcal{Y}}^{(k)}) = \sqrt{\frac{\sum_{i<j}(\delta_{ij} - d_{ij})^2}{\sum_{i<j}\delta_{ij}^2}}$

6: Compute the Guttman Transform: $\tilde{\mathcal{Y}}^{(k+1)} = n^{-1}B(\tilde{\mathcal{Y}}^{(k)})\tilde{\mathcal{Y}}^{(k)}$ where $B(\tilde{\mathcal{Y}}^{(k)}) = (b_{ij})$:

$$b_{ij} = \begin{cases} -\delta_{ij}/d_{ij}(\tilde{\mathcal{Y}}^{(k)}) & \text{if } i \neq j \text{ and } d_{ij}(\tilde{\mathcal{Y}}^{(k)}) > 0 \\ 0 & \text{if } i \neq j \text{ and } d_{ij}(\tilde{\mathcal{Y}}^{(k)}) = 0 \\ -\sum_{j \neq i} b_{ij} & \text{if } i = j \end{cases}$$

7: $k \leftarrow k + 1$

8: **until** $k \geq n\_iter$ or $|STRESS_M(D_{\mathcal{X}}, \tilde{\mathcal{Y}}^{(k-1)}) - STRESS_M(D_{\mathcal{X}}, \tilde{\mathcal{Y}}^{(k)}| < \epsilon$

9: **return** $\tilde{\mathcal{Y}}^{(k)}$

Algorithm 2 Local MDS

Require: $D_{\mathcal{X}} = (d_{ij})$, the matrix of original distances; q, the embedding's dimensionality; k, the size of neighborhoods; and τ, the weight of the repulsive term in the stress function.

Ensure: A configuration in a q-dimensional space $\tilde{\mathcal{Y}}$.

1: Compute the symmetrized k-NN graph of $D_{\mathcal{X}}$, $\mathcal{N}$

2: Let $t = \frac{|\mathcal{N}|}{|\mathcal{N}^C|} \cdot \text{median}_{\mathcal{N}}(D_{i,j}) \cdot \tau$

3: Minimize

$$\sum_{(i,j) \in N} (D_{i,j} - \|\mathbf{y}_i - \mathbf{y}_j\|)^2 - t \sum_{(i,j) \notin N} \|\mathbf{y}_i - \mathbf{y}_j\|$$

4: **return** The solution to the optimization problem, $\tilde{\mathcal{Y}}$.

Algorithm 3 Isomap

Require: $D_{\mathcal{X}}$, the matrix of observed Euclidean distances; q, the embedding's dimensionality; and ϵ or k, the bandwith.

Ensure: A configuration in a q-dimensional space $\tilde{\mathcal{Y}}$.

1: Find the k-NN or ϵ-NN graph of $\mathcal{X}$, G.

2: Compute the distance matrix of G: D_G.

3: Embed D_G to a q-dimensional space with MDS.

4: **return** The output configuration of MDS.

2.2.4 t-SNE

t-SNE (t-Distributed Stochastic Neighbor Embedding) [4] is a nonlinear dimensionality reduction technique particularly well-suited for visualizing high-dimensional data. It preserves local neighborhoods by modeling similarities between points as conditional probabilities and minimizing the difference between these probability distributions in high and low-dimensional spaces.

Algorithm 4 t-SNE

Require: $\mathcal{X} \in \mathbb{R}^{n \times p}$, the high-dimensional configuration; q, the embedding's dimensionality; perplexity $Perp$.

Ensure: A configuration in a q-dimensional space $\tilde{\mathcal{Y}}$.

1: For every datapoint i, find σ_i so that the conditional probability ditribution

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2/2\sigma_i^2)}$$

has perplexity $2^{-\sum_j p_j \log_2 p_j} = Perp$

2: Symmetrize conditional distributions: $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$ if $i \neq j$, $p_{ii} = 0$

3: Consider Student t-distributed joint probabilities for the low-dimensional data y_i:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{h \neq k}(1 + \|y_h - y_k\|^2)^{-1}}$$

4: Minimize the sum of Kullback-Leibler divergences between the joint distributions over all datapoints:

$$C(\tilde{\mathcal{Y}}) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

5: **return** The solution to the optimization problem, $\tilde{\mathcal{Y}}$.

t-SNE focuses on retaining the local structure of the data while ensuring that every point y_i in the low-dimensional space will have the same number of neighbors, making it particularly effective for visualizing clusters in high-dimensional data. The use of the Student t-distribution in the low-dimensional space addresses the *crowding problem* by allowing dissimilar points to be modeled far apart [4].

The tuning parameter $Perp$ is interpreted as the average effective number of neighbors of the high-dimensional datapoints x_i and typical values are between 5 and 50.

2.3 Multidimensional Scaling for Big Data

2.4 Landmark Isomap and the OOS Projection Framework

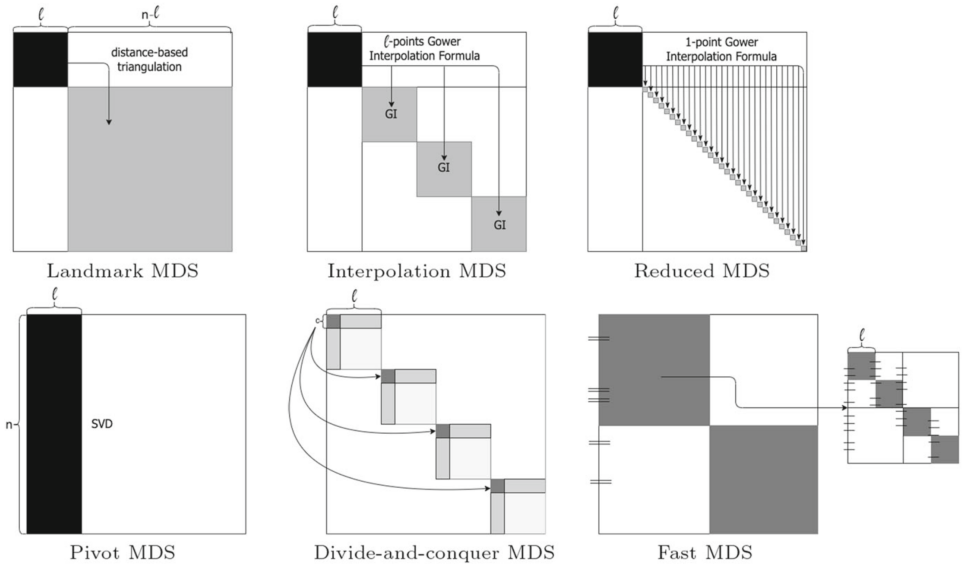

Figure 1: Schematic representation of the six MDS algorithms described by Delicado and Pachón-García [3].

3 Specification and Design of the Solution

- Jusitfy D&C in non-linear DR methods. Most techniques shown in `bigmds` are not generalizable to non-linear DR methods different to Isomap (since it uses MDS). However, D&C and recurrence can work with more general algorithms. Finally, we chose D&C because recurrence can create very small partitions that may be problematic.

- D&C algorithm

4 Development of the Proposal

- DR methods implementations: packages used, problems found during development, tuning of parameters, experiments' methodology

5 Experimentation and Evaluation of the Proposal

- Swiss Roll.
- MNIST.
- Time complexity measurements (including MDS).
- Evaluation and comparison with state of the art.

6 Analysis of Sustainability and Ethical Implications

DESCRIPTION OF THIS SECTION FROM THE REGULATION:

It must include an analysis of the impact of the following gender-related technical aspects:

- issues related to data management and analysis
- issues related to equity, where possible biases are identified and assessed both in the data and in the processes carried out in relation to data management and analysis
- actions carried out to eliminate or mitigate such biases

ACTUAL CONTENT:

- D&C can be used in normal computers, while traditional DR methods require supercomputers to work on big datasets. Hence, we will reduce computing emissions.
- Maybe (it needs testing) DR methods could emphasize biases in the data. This happens because when projecting only a few coordinates, small clusters could be left behind in the remaining not projected coordinates and do not show in the final embedding. Hence, in theory, small communities could become invisible.

7 Conclusions

References

[1] Ingwer Borg and Patrick J. F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer Series in Statistics. New York: Springer, 1997.

[2] Lisha Chen and Andreas Buja and. "Local Multidimensional Scaling for Nonlinear Dimension Reduction, Graph Drawing, and Proximity Analysis". In: *Journal of the American Statistical Association* 104.485 (2009), pp. 209–219. DOI: 10.1198/jasa.2009.0111. eprint: https://doi.org/10.1198/jasa.2009.0111. URL: https://doi.org/10.1198/jasa.2009.0111.

[3] Pedro Delicado and Carlos Pachón-García. "Multidimensional scaling for big data". In: *Advances in Data Analysis and Classification* (2024). ISSN: 1862-5355. DOI: 10.1007/s11634-024-00591-9.

[4] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605. URL: http://jmlr.org/papers/v9/vandermaaten08a.html.

[5] Vin Silva and Joshua Tenenbaum. "Global Versus Local Methods in Nonlinear Dimensionality Reduction". In: *Advances in Neural Information Processing Systems*. Ed. by S. Becker, S. Thrun, and K. Obermayer. Vol. 15. MIT Press, 2002. URL: https://proceedings.neurips.cc/paper_files/paper/2002/file/5d6646aad9bcc0be55b2c82f69750387-Paper.pdf.

[6] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. "A Global Geometric Framework for Nonlinear Dimensionality Reduction". In: *Science* 290.5500 (2000), pp. 2319–2323. DOI: 10.1126/science.290.5500.2319. URL: https://www.science.org/doi/abs/10.1126/science.290.5500.2319.

[7] Kisung You and Dennis Shung. "Rdimtools: An R Package for Dimension Reduction and Intrinsic Dimension Estimation". In: *Software Impacts* 14 (2022), p. 100414. ISSN: 26659638. DOI: 10.1016/j.simpa.2022.100414.

8 Annexes

Example citation: [7].