



**UNIVERSITAT POLITÈCNICA DE CATALUNYA**  
**BARCELONATECH**

**Facultat d'Informàtica de Barcelona**



# **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

<b>1</b>	<b>Introduction, Motivation, and Objectives</b>	<b>4</b>
1.1	Introduction . . . . .	4
1.2	Motivation . . . . .	4
1.3	Objectives . . . . .	4
<b>2</b>	<b>State of the Art</b>	<b>6</b>
2.1	A few Dimensionality Reduction Techniques . . . . .	6
2.1.1	Non-classical MDS. The SMACOF algorithm . . . . .	6
2.1.2	Local MDS . . . . .	6
2.1.3	Isomap . . . . .	7
2.1.4	t-SNE . . . . .	7
2.2	Multidimensional Scaling for Big Data . . . . .	8
2.3	Landmark Isomap and the Out-of-Core Dimensionality Reduction Framework	9
2.4	openTSNE, an outstanding implementation of t-SNE . . . . .	9
<b>3</b>	<b>Specification and Design of the Solution</b>	<b>11</b>
<b>4</b>	<b>Development of the Proposal</b>	<b>12</b>
<b>5</b>	<b>Experimentation and Evaluation of the Proposal</b>	<b>13</b>
<b>6</b>	<b>Analysis of Sustainability and Ethical Implications</b>	<b>14</b>
<b>7</b>	<b>Conclusions</b>	<b>15</b>
<b>8</b>	<b>Annexes</b>	<b>17</b>

# 1 Introduction, Motivation, and Objectives

## 1.1 Introduction

Dimensionality Reduction (DR) techniques embed high-dimensional datasets into significantly lower-dimensional spaces while preserving the structure of the original data. Their primary goal is to tackle the common challenges of working with high-dimensional data, such as sparsity (caused by the curse of dimensionality) or large computational and storage costs.

That being said, DR is mainly used for visualization purposes. In this setting, complex high-dimensional data is projected into  $\mathbb{R}^2$ , making patterns and clusters more apparent.

Many DR methods have been proposed since Principal Components Analysis (PCA) (the most famous linear method) was first introduced, each with distinct approaches and objectives. In particular, non-linear techniques have turned out to be very useful thanks to their ability to preserve complex relationships. Some examples are Multidimensional Scaling (MDS), Local MDS, Isomap, t-Distributed Stochastic Neighbor Embedding (t-SNE), Uniform Manifold Approximation and Projection (UMAP) and Autoencoder. All these methods differ in how they define and maintain relationships between points, although all try to preserve global structure and local neighborhoods.

Despite their utility, DR methods face a few limitations. Many algorithms require computing and/or keeping in memory pairwise distances between all data points, resulting in quadratic time complexity and memory requirements. This becomes prohibitive for large datasets with millions of points. Parameter selection presents challenges too, since there is no general consensus on the best way to tune them and  $k$ -fold cross validation is very costly due to the substantial time complexities of these algorithms. Furthermore, information loss is inevitable during dimensionality reduction, and understanding which aspects of the data are preserved and which are distorted is crucial to correctly interpret the results.

## 1.2 Motivation

The recent advent of large data has led to new challenges and technologies to face them. When the number of observations of a dataset is very big, distance-based DR methods become computationally prohibitive because of their quadratic time and memory complexities. For example, working with a dataset of 100,000 individuals and 10 numeric variables would require a system with about 400GB of RAM.

Recently, Delicado and Pachón-García **Delicado2024MDSBigData** studied existing and new modifications of MDS to tackle big datasets, demonstrating significant improvements in computational efficiency while maintaining embedding quality. One of the variations they came up with followed a divide-and-conquer approach, that in principle could be generalized to any DR technique based on between-points distances.

## 1.3 Objectives

The primary goal of this thesis is to propose a divide-and-conquer framework for any generic distance-based dimensionality reduction method that effectively decreases the method's time and memory complexities. Specifically, we aim to:

1. Review the literature to analyze the properties and applications of the most used DR techniques.
2. Develop a generalized framework for distance-based DR methods that leverages the divide-and-conquer strategy and the Procrustes transformation to reduce time and memory complexities.
3. Implement and parallelize the proposed framework for specific DR algorithms such as non-classical MDS, Local MDS, Isomap and t-SNE.
4. Empirically evaluate the performance of the adapted algorithms in terms of computational efficiency, size limitations and quality of embeddings on benchmark datasets of varying sizes.
5. Compare the results with the traditional counterpart of each tested DR method.
6. Provide guidelines and best practices for selecting and tuning the proposed DR methods based on dataset characteristics and desired properties of the embedding.

The successful completion of these objectives will contribute to making dimensionality reduction techniques accessible for very large datasets, democratizing their use across scientific and industrial applications where data scale is a present challenge.

## 2 State of the Art

In this section, we aim to review four dimensionality reduction techniques that we will later apply our divide-and-conquer framework to. Next, we will showcase in more detail the work of Delicado and Pachón-García with respect to MDS in big data Delicado and Pachón-García 2024 to motivate our choosing of the divide-and-conquer approach. Finally, we will discuss three more appreciable solutions to the big data problem in DR and how they relate to our proposal: Landmark Isomap, the Out-of-Core Dimensionality Reduction Framework ??? and formidable Python t-SNE implementations.

### 2.1 A few Dimensionality Reduction Techniques

#### 2.1.1 Non-classical MDS. The SMACOF algorithm

SMACOF (Scaling by MAjorizing a COmplicated Function) is a multidimensional scaling algorithm that minimizes metric stress using a majorization technique borg1997modern. Also known as the Guttman Transform, this technique is more powerful for this problem than general optimization methods, such as gradient descent.

---

##### Algorithm 1 SMACOF

---

**Require:**  $D_{\mathcal{X}} = (\delta_{ij})$ , the matrix of observed distances;  $q$ , the embedding’s dimensionality;  $n\_iter$ , the maximum number of iterations; and  $\epsilon$ , the convergence threshold.

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

- 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**  $\tilde{\mathcal{Y}}^{(k)}$
- 

#### 2.1.2 Local MDS

Local MDS **LocalMDS** is a variant of non-classical multidimensional scaling that differs in how large distances are treated. Specifically, a repulsive term between distant points is added to the stress function to further separate points in the low-dimensional configuration.

Parameters  $\tau$  (which must be in the unit interval) and  $k$  may be tuned with  $k'$ -cross validation thanks to the LCMC (Local Continuity Meta-Criteria). (*POSSIBLE ANNEX*)

---

**Algorithm 2** Local MDS

---

**Require:**  $D_{\mathcal{X}}$ , the matrix of observed distances;  $q$ , the embedding’s dimensionality;  $k$ , the size of neighborhoods; and  $\tau$ , the weight of the repulsive term.

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

- 1: Compute the symmetrized  $k$ -NN graph of  $D_{\mathcal{X}}$ ,  $\mathcal{N}$
- 2: Calculate  $t = \frac{|\mathcal{N}|}{|\mathcal{N}^c|} \cdot \text{median}_{\mathcal{N}}(D_{i,j}) \cdot \tau$
- 3: Minimize

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

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

### 2.1.3 Isomap

Isomap Tenenbaum, Silva, and Langford 2000 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.

---

**Algorithm 3** Isomap

---

**Require:**  $D_{\mathcal{X}}$ , the matrix of observed distances;  $q$ , the embedding’s dimensionality; and  $\epsilon$  or  $k$ , the bandwidth.

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

- 1: Find the  $\epsilon$ -NN or  $k$ -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.
- 

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

### 2.1.4 t-SNE

t-SNE (t-Distributed Stochastic Neighbor Embedding) **tsne** is a nonlinear dimensionality reduction technique that preserves local neighborhoods by modeling similarities between points as conditional probabilities. The difference between these probability distributions in high and low-dimensional spaces is then minimized.

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. 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 **tsne**.

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

---

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

- 1: For every datapoint  $i$ , find  $\sigma_i$  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,  $\tilde{\mathcal{Y}}$ .
- 

## 2.2 Multidimensional Scaling for Big Data

Delicado and Pachón-García Delicado and Pachón-García 2024 compared four existing versions of MDS with two newly proposed (Divide-and-conquer MDS and Interpolation MDS) to handle large data. As can be seen in figure 1, these can be grouped into four categories:

- **Interpolation-based:** Landmark MDS, Interpolation MDS and Reduced MDS apply classical multidimensional scaling to a subset of  $l \ll n$  points and then interpolate the projection of the remaining data. They differ in how the interpolation is computed: Landmark MDS uses distance-based triangulation; Interpolation MDS, the  $l$ -points Gower Interpolation Formula; and Reduced MDS, the 1-point Gower Interpolation Formula.
- **Approximation-based:** Pivot MDS approximates the SVD of the full inner product matrix with the SVD of the inner product matrix between a subset of  $l \ll n$  points and all the points in the dataset.
- **Divide-and-conquer:** In Divide-and-conquer MDS, the dataset is randomly partitioned into subsets of up to  $l \ll n$  points into which MDS is independently applied. Then, the resulting embeddings are aligned with Procrustes transformations.
- **Recursive:** Fast MDS is similar in spirit to Divide-and-conquer MDS, but it partitions the data recursively.



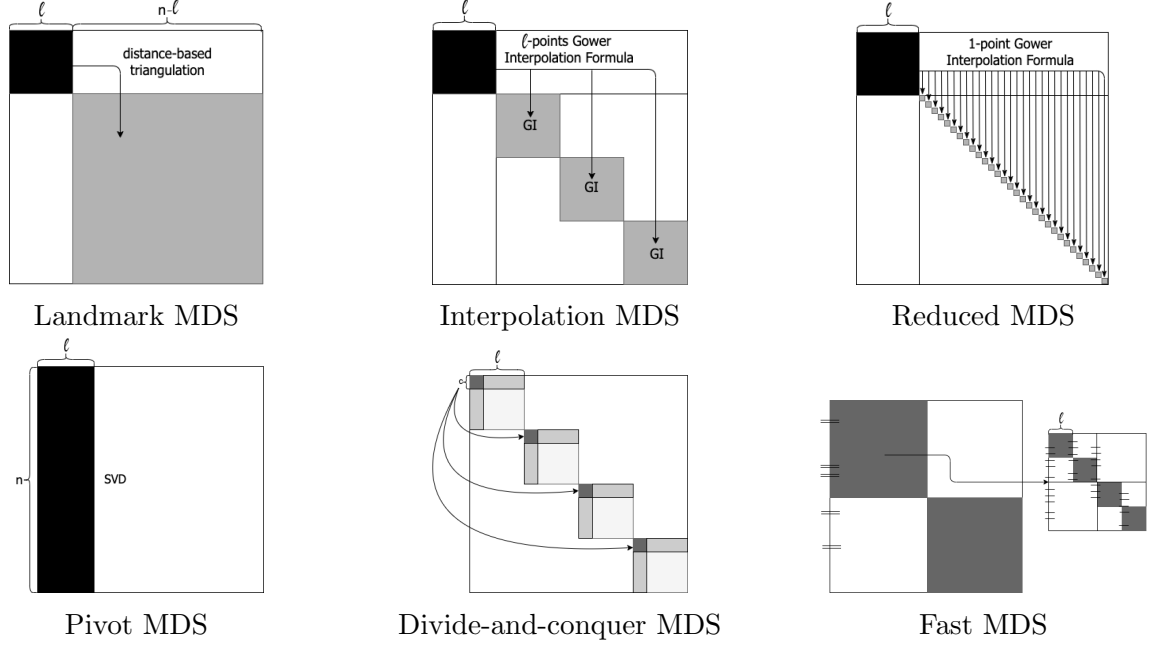


Figure 1: Schematic representation of the six MDS algorithms for big data described by Delicado and Pachón-García Delicado and Pachón-García 2024. (Source: original publication.)

## 2.3 Landmark Isomap and the Out-of-Core Dimensionality Reduction Framework

Silva and Tenenbaum **deSilvaTenenbaum2002** first introduced Landmark MDS in 2002 by applying it to Isomap (L-Isomap). This way, they reduced the time complexity of both classical multidimensional scaling and Isomap from  $\mathcal{O}(n^3)$  to  $\mathcal{O}(n^2l)$ , where  $l \ll n$  is the amount of landmark points.

Note that, similarly to Landmark MDS, other big data versions of MDS can be used with Isomap. Nonetheless, interpolation-based and approximation-based algorithms cannot be trivially generalized to nonlinear dimensionality reduction methods.

Later, in 2024, Reichmann, Hägele and Weiskopf **reichmann2024outofcoredimensionalityreduc** proposed the Out-of-Core Dimensionality Reduction Framework. Similar to Interpolation MDS, this algorithm applies a DR method that produces a mapping between high- and low-dimensional spaces (i.e. PCA, MDS, t-SNE, UMAP, Autoencoder) to a small subset of the data and then projects the remaining datapoints in blocks. In order to obtain the aforementioned mappings, Reichmann et. al. gathered different projection mechanisms for every method. PCA and autoencoders learn a parametric mapping between the original data and the embedding, so projecting new points is straightforward. For non-classical MDS, stress is minimized for a single point while keeping others fixed, a process known as *single scaling* in the literature **single-scaling**. A similar strategy is used for t-SNE **tsne-knn** and UMAP McInnes et al. 2018, which leverage the k-NN of the projecting point to initialize the optimizer.

## 2.4 openTSNE, an outstanding implementation of t-SNE

Maybe because of its popularity, t-SNE has been optimized for large data environments in Python through iterative implementations. In 2024, Poličar, Stražar and Zupan Poličar,

Stražar, and Zupan 2024 considered many of them and published **openTSNE**, a package that includes several t-SNE extensions “*to address scalability issues and the quality of the resulting visualizations*”. Furthermore, they compared their proposal with those of other popular packages in serial and parallel configurations. In figure 2 it can be seen that, thanks to these advances, the challenges of big data have already been solved in the specific case of t-SNE.

(*POSSIBLE ANNEX*)

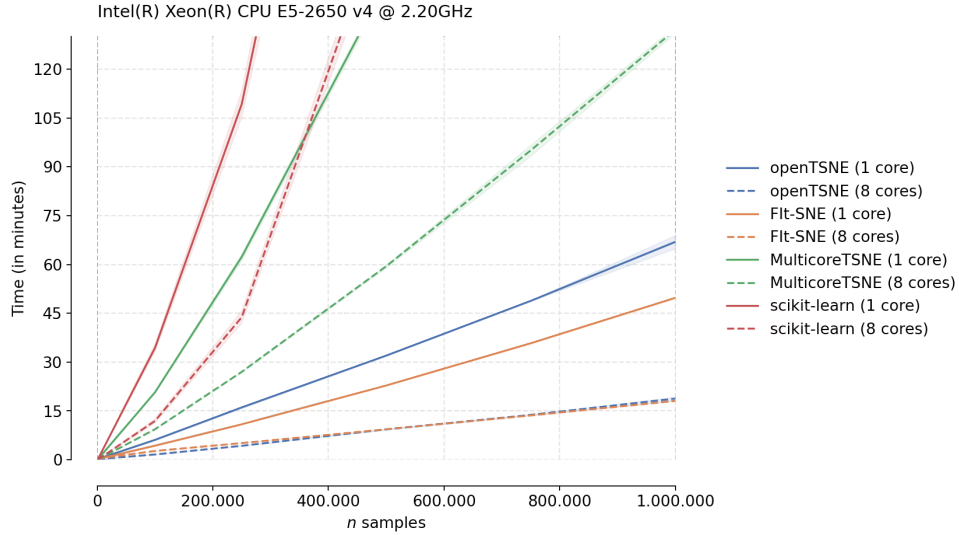


Figure 2: Benchmark of the best open-source t-SNE Python implementations Poličar, Stražar, and Zupan 2024. (Source: original ??? webpage (cite webpage too).)

### 3 Specification and Design of the Solution

- Justify 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

- Delicado, Pedro and Cristian Pachón-García (2024). “Multidimensional scaling for big data”. In: *Advances in Data Analysis and Classification*. DOI: 10.1007/s11634-024-00591-9.
- McInnes, Leland, John Healy, Nathaniel Saul, and Lukas Großberger (2018). “UMAP: Uniform Manifold Approximation and Projection”. In: *Journal of Open Source Software* 3.29, p. 861. DOI: 10.21105/joss.00861. URL: <https://doi.org/10.21105/joss.00861>.
- Poličar, Pavlin G., Martin Stražar, and Blaž Zupan (2024). “openTSNE: A Modular Python Library for t-SNE Dimensionality Reduction and Embedding”. In: *Journal of Statistical Software* 109.3, pp. 1–30. DOI: 10.18637/jss.v109.i03.
- Tenenbaum, Joshua B., Vin de Silva, and John C. Langford (2000). “A Global Geometric Framework for Nonlinear Dimensionality Reduction”. In: *Science* 290.5500, pp. 2319–2323. DOI: 10.1126/science.290.5500.2319. URL: <https://www.science.org/doi/abs/10.1126/science.290.5500.2319>.



## 8 Annexes