Distance-based dimensionality reduction for big data

Master's thesis defense

Adrià Casanova Lloveras

Thesis supervisor

Pedro F. Delicado Useros

Thesis co-supervisor

Cristian Pachón García

July 3rd, 2025

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

Table of contents

1. Introduction, motivation and objectives

Introduction, motivation and objectives

- Dimensionality reduction (DR) aims to project a dataset into a low-dimensional space.
- Most DR techniques are based on the inter-individual distance matrix $\Rightarrow$ they have quadratic memory complexity.
- There are algorithms that extend classical MDS to the big data settings.
- In this master's thesis, we adapt one of these algorithms to any generic distance-based DR method.

2. State of the art

A few dimensionality reduction techniques

Non-classical MDS

The SMACOF algorithm minimizes metric stress using a majorization technique (Kruskal, 1964a; Kruskal, 1964b).

LMDS

A repulsive term between distant points is added to the stress function (Chen and Buja, 2009).

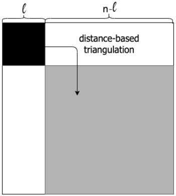
Isomap

Preserves geodesic distances between points in a manifold (Tenenbaum, de Silva, and Langford, 2000).

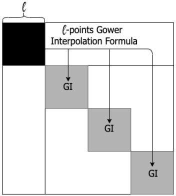
t-SNE

Models similarities between points as conditional probabilities (Maaten and Hinton, 2008).

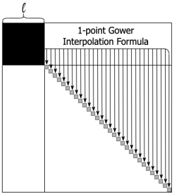
Multidimensional scaling for big data



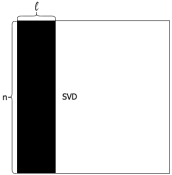
landmark MDS



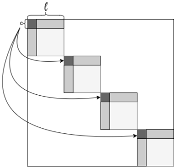
interpolation MDS



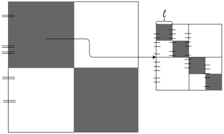
reduced MDS



pivot MDS



divide-and-conquer MDS



fast MDS

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

3. Specification and design of the solution

Divide-and-conquer dimensionality reduction (1/3)

Algorithm 1 Divide-and-conquer dimensionality reduction

Require: $\mathsf{D} = (\delta_{ij})$, the $n \times n$ matrix of observed distances; $\mathcal{M}$, the DR method; l, the partition size; c, the amount of connecting points; q, the embedding's dimensionality; and arg, $\mathcal{M}$'s specific parameters.

Ensure: $\widetilde{\mathsf{Y}}$, a configuration in a q-dimensional space.

1: **if** $n \leq l$ **then return** $\mathcal{M}(\mathsf{D}, q, arg)$

2: **end if**

3: Let $k = \lceil \frac{n-l}{l-c} \rceil$

4: Randomly partition the data: $\mathcal{P} = \{\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_k\}$ where

$$|\mathcal{P}_i| = \begin{cases} l & \text{if } i = 0 \\ l - c & \text{if } 0 < i \leq (n-l) \bmod k \\ l - c - 1 & \text{if } (n-l) \bmod k < i \leq k \end{cases}$$

Divide-and-conquer dimensionality reduction (2/3)

Sample c connecting points from $\mathcal{P}_0$: $\mathcal{C} \subset \mathcal{P}_0$
Extract distance matrix of $\mathcal{P}_0$: $\mathsf{D}_{\mathcal{P}_0} = \mathsf{D}[\mathcal{P}_0, \mathcal{P}_0]$
Apply DR method to $\mathcal{P}_0$: $\widetilde{\mathsf{Y}}_0 = \mathcal{M}(\mathsf{D}_{\mathcal{P}_0}, q, arg)$
Extract embedding of $\mathcal{C}$: $\widetilde{\mathsf{Y}}_{\mathcal{C}} = \widetilde{\mathsf{Y}}_0[\mathcal{C}, :]$
Extract distance matrix of $\mathcal{C}$: $\mathsf{D}_{\mathcal{C}} = \mathsf{D}[\mathcal{C}, \mathcal{C}]$

Divide-and-conquer dimensionality reduction (3/3)

for $i = 1$ to k **do**

  Extract distance matrix of $\mathcal{P}_i$: $\mathsf{D}_{\mathcal{P}_i} = \mathsf{D}[\mathcal{P}_i, \mathcal{P}_i]$

  Stack connecting points to $\mathcal{P}_i$: $\mathsf{D}_{\text{stack}} = [\mathsf{D}_{\mathcal{C}}; \mathsf{D}_{\mathcal{P}_i}]$

  Project the stacked data: $\widetilde{\mathsf{Y}}_{\text{stack}} = \mathcal{M}(\mathsf{D}_{\text{stack}}, q, arg)$

  Split embeddings: $\widetilde{\mathsf{Y}}_{\mathcal{C}}^{(i)} = \widetilde{\mathsf{Y}}_{\text{stack}}[:c, :]$ and $\widetilde{\mathsf{Y}}_i = \widetilde{\mathsf{Y}}_{\text{stack}}[(c+1):, :]$

  Align first and current embeddings: $\widetilde{\mathsf{Y}}_i = \text{Procrustes}(\widetilde{\mathsf{Y}}_{\mathcal{C}}, \widetilde{\mathsf{Y}}_{\mathcal{C}}^{(i)}, \widetilde{\mathsf{Y}}_i)$

end for

Combine all embeddings: $\widetilde{\mathsf{Y}}' = [\widetilde{\mathsf{Y}}_0; \widetilde{\mathsf{Y}}_1; \ldots; \widetilde{\mathsf{Y}}_k]$

Retrieve original row ordering: order = argsort($[\mathcal{P}_0; \mathcal{P}_1; \ldots; \mathcal{P}_k]$)

Set original ordering: $\widetilde{\mathsf{Y}}' = \widetilde{\mathsf{Y}}'[\text{order}, :]$

Apply PCA to center and rotate data: $\widetilde{\mathsf{Y}}$ = PCA($\widetilde{\mathsf{Y}}', q$)

  return $\widetilde{\mathsf{Y}}$

Orthogonal Procrustes transformation's derivation

Let $\mathbf{A} \in \mathbb{R}^{c \times q}$ be the target configuration and $\mathbf{B} \in \mathbb{R}^{c \times q}$ the corresponding testee. We wish to fit $\mathbf{B}$ to $\mathbf{A}$ by rigid motions. That is, we want to find the best orthogonal matrix $\mathbf{T}$ such that $\mathbf{A} \simeq \mathbf{BT}$. We will measure the $\simeq$ relation with the sum-of-squares criterion L and try to minimize it:

$$\min_{\mathbf{T} \in O(q)} L(\mathbf{T}) = \min_{\mathbf{T} \in O(q)} \operatorname{tr}(\mathbf{A} - \mathbf{BT})(\mathbf{A} - \mathbf{BT})',$$

Berge, Kiers, and Commandeur (1993) found the following solution. Let $\mathbf{U\Sigma V}'$ be the singular value decomposition of $\mathbf{A}'\mathbf{B}$, where $\mathbf{U}'\mathbf{U} = \mathbf{I}$, $\mathbf{V}'\mathbf{V} = \mathbf{I}$, and $\mathbf{\Sigma}$ is the diagonal matrix with the singular values. Then, $L(\mathbf{T})$ is minimal if $\mathbf{T} = \mathbf{VU}'$.

4. Development of the proposal

Python implementation

- `divide_conquer` implements Algorithm 1 in parallel through the `concurrent.futures` module.
- Implementations of DR algorithms used:
  - `sklearn.manifold` module (Pedregosa et al., 2011) for Isomap and SMACOF.
  - `openTSNE` (Poličar, 2023) for t-SNE.
  - A translation of the R library `smacofx` (Leeuw and Mair, 2009) for LMDS.
- Time complexity is reduced from quadratic (or cubic for Isomap) to linear.
- Space complexity is lowered from $\mathcal{O}(n^2)$ to $\mathcal{O}(l^2)$.

Test datasets

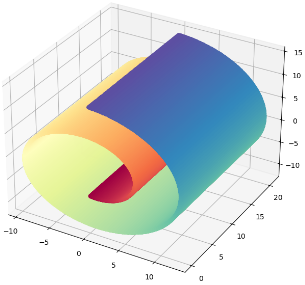


Figure 2: Swiss roll

↓

Unfolded rectangle

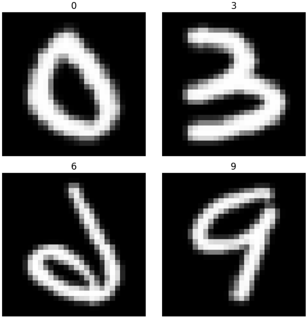


Figure 3: MNIST

↓

10 separate clusters

Experimental setup

1. Tune the bare DR method on a l points subset.
2. Apply the bare DR method on a larger subset.
3. Apply divide-and-conquer DR on a larger subset.
4. Apply the bare DR method on the whole dataset (when possible).
5. Apply divide-and-conquer DR on the whole dataset.

The testing system was an Asus ROG G513QM-HF026 laptop with

- Windows
- AMD Ryzen 7 5800H CPU
- NVIDIA RTX 3060 GPU
- 16 GB of DDR4-3200MHz RAM
- 1 TB M.2 NVMe PCIe 3.0

5. Experimentation and evalutation of the proposal

Runtime benchmarks of divide-and-conquer Isomap and divide-and-conquer t-SNE

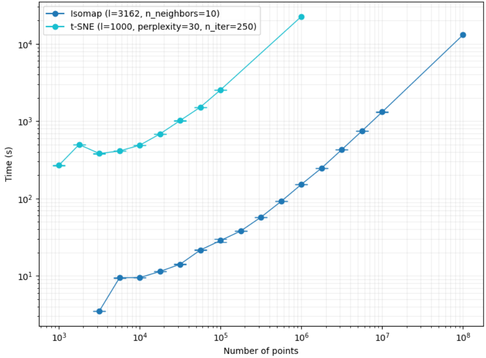


Figure 4: Runtime (s) of divide-and-conquer Isomap and divide-and-conquer t-SNE averaged over 20 experiments. Tests were performed on datasets generated on the Swiss roll manifold with sizes ranging from 10^3 to 10^8. Data was embedded into $\mathbb{R}^2$ with different parameter combinations and $c = 100$.

SMACOF's embedding of Swiss roll

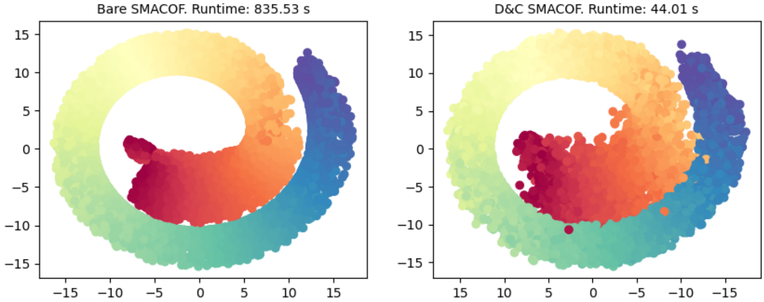


Figure 5: Comparison of the bidimensional embeddings of a 7,500 points Swiss roll dataset by bare (left) and divide-and-conquer (right) SMACOF. The arguments used were $n\_iter = 300$, $\varepsilon = 0.001$ and in divide-and-conquer there also were $l = 1000$ and $c = 100$. Color represents the angle of rotation along the Swiss roll spiral.

LMDS's embedding of Swiss roll

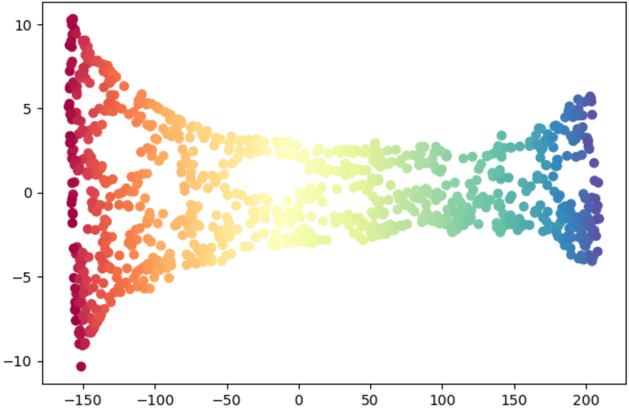


Figure 6: Bidimensional embedding of a 1,000 points Swiss roll dataset computed by LMDS with $k = 10$ and $\tau = 0.1$. Color represents the angle of rotation along the Swiss roll spiral.

Divide-and-conquer Isomap's embedding of Swiss roll

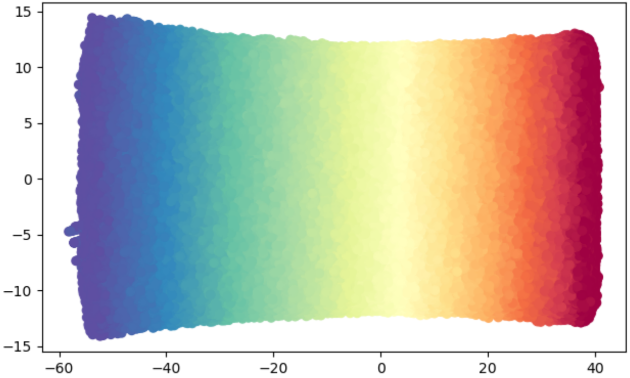


Figure 7: Bidimensional embedding of a 10^8 points Swiss roll dataset computed by divide-and-conquer Isomap with $k = 10$, $l = 3,162$ and $c = 100$. Color represents the angle of rotation along the Swiss roll spiral.

Divide-and-conquer t-SNE's embedding of Swiss roll

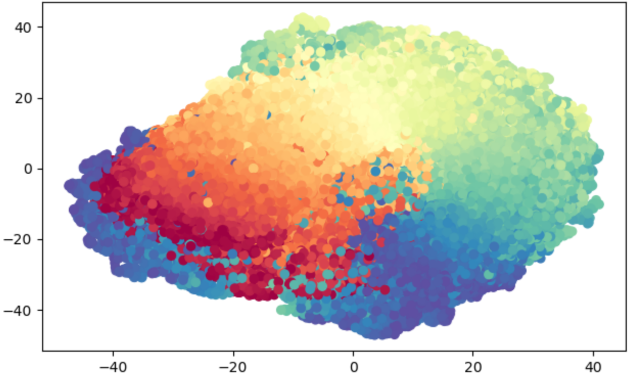


Figure 8: Bidimensional embedding of a 10^6 points Swiss roll dataset computed by divide-and-conquer t-SNE with $l = 1,000$, $c = 100$, $Perp = 30$ and $n\_iter = 250$. Color represents the angle of rotation along the Swiss roll spiral.

SMACOF’s embedding of a 5,000 points subset of MNIST (1/2)

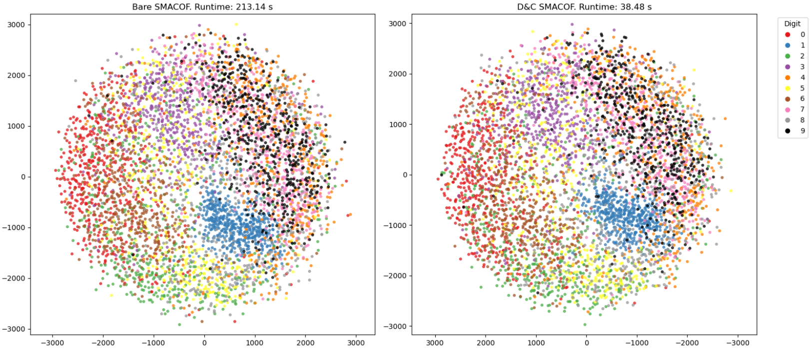


Figure 9: Bidimensional embeddings of a 5,000 points subset of MNIST by bare (left) and divide-and-conquer (right) SMACOF. The arguments we used were $n\_iter = 300$, $\varepsilon = 0.001$ and in divide-and-conquer there also were $l = 1000$ and $c = 100$.

SMACOF's embedding of a 5,000 points subset of MNIST (2/2)

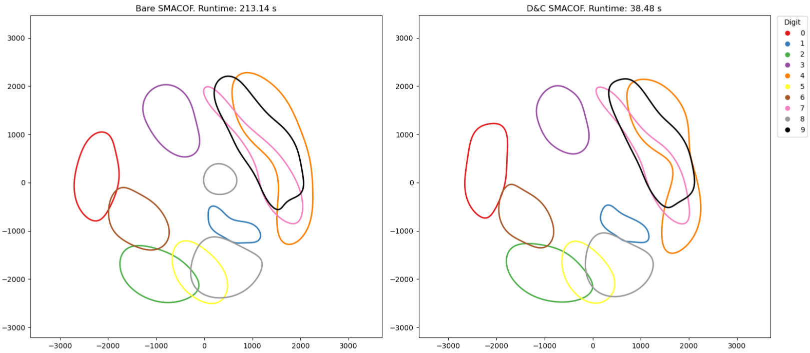


Figure 10: Kernel density estimation of the bidimensional embeddings of a 5,000 points subset of MNIST by bare (left) and divide-and-conquer (right) SMACOF. The arguments we used were $n\_iter = 300$, $\varepsilon = 0.001$ and in divide-and-conquer there also were $l = 1000$ and $c = 100$. Contour lines are at 70% of the maximum estimated density for each digit and embedding.

Divide-and-conquer SMACOF's embedding of the whole MNIST

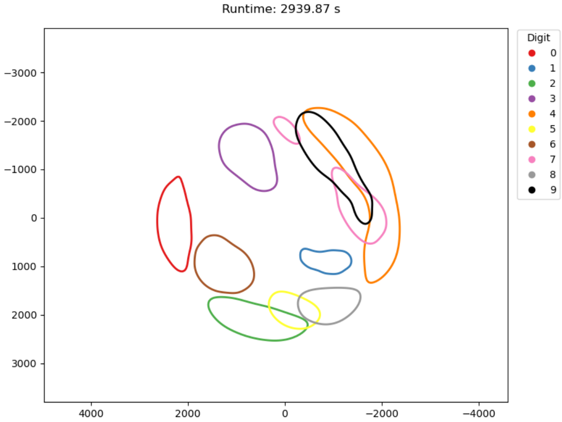


Figure 11: Kernel density estimation of the bidimensional embeddings of the whole MNIST dataset by divide-and-conquer SMACOF. The arguments used were $n\_iter = 300$, $\varepsilon = 0.001$, $l = 1000$ and $c = 100$. Contour lines are at 70% of the maximum estimated density for each digit and embedding.

LMDS's embedding of a 5,000 points subset of MNIST

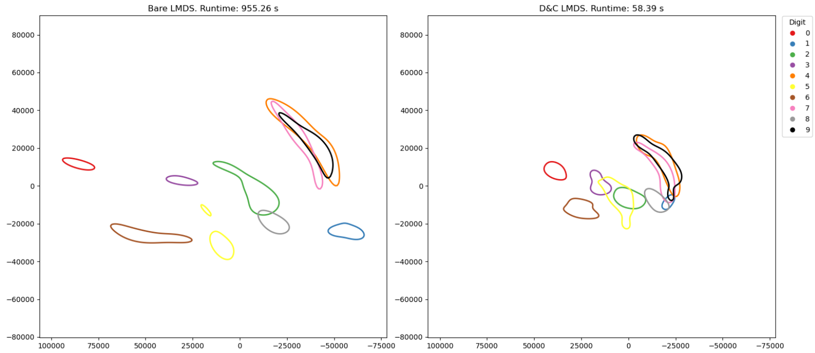


Figure 12: Kernel density estimation of the bidimensional embeddings of a 5,000 points subset of MNIST by bare (left) and divide-and-conquer (right) LMDS. The arguments used were $k = 10$, $\tau = 1$ and in divide-and-conquer there also were $l = 1000$ and $c = 100$. Contour lines are at 70% of the maximum estimated density for each digit and embedding.

Divide-and-conquer LMDS's embedding of the whole MNIST

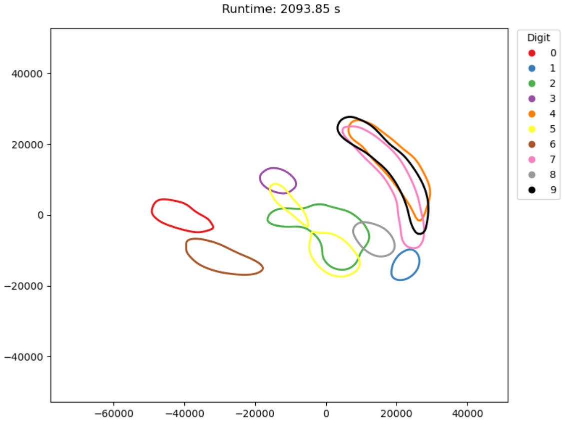


Figure 13: Kernel density estimation of the bidimensional embeddings of the whole MNIST dataset by divide-and-conquer LMDS. The arguments used were $k = 10$, $\tau = 1$, $l = 1000$ and $c = 100$. Contour lines are at 70% of the maximum estimated density for each digit and embedding.

Isomap's embedding of a 5,000 points subset of MNIST

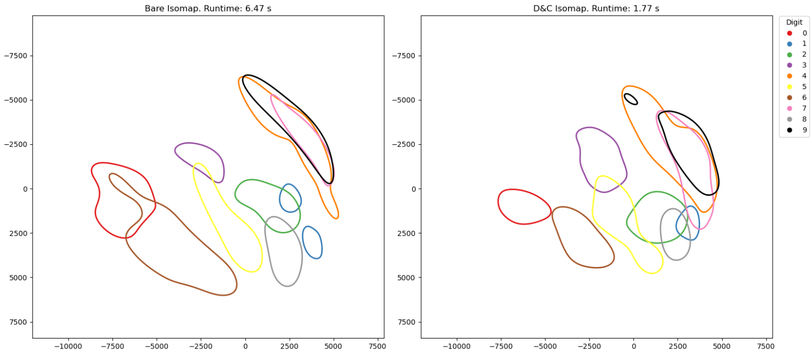


Figure 14: Kernel density estimation of the bidimensional embeddings of a 5,000 points subset of MNIST by bare (left) and divide-and-conquer (right) Isomap. The arguments used were $k = 5$ and in divide-and-conquer there also were $l = 1000$ and $c = 100$. Contour lines are at 70% of the maximum estimated density for each digit and embedding.

Divide-and-conquer Isomap's embedding of the whole MNIST

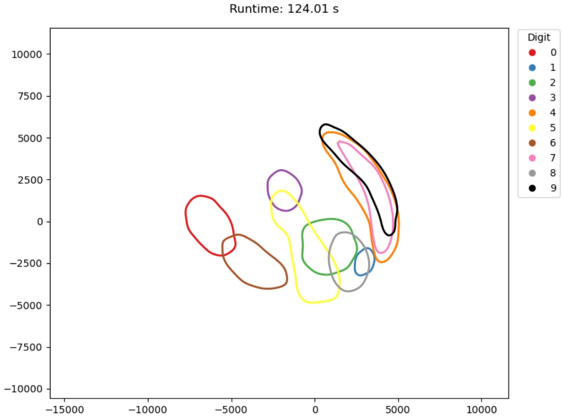


Figure 15: Kernel density estimation of the bidimensional embeddings of the whole MNIST dataset by divide-and-conquer Isomap. The arguments used were $k = 5$, $l = 1000$ and $c = 100$. Contour lines are at 70% of the maximum estimated density for each digit and embedding.

t-SNE's embedding of the whole MNIST

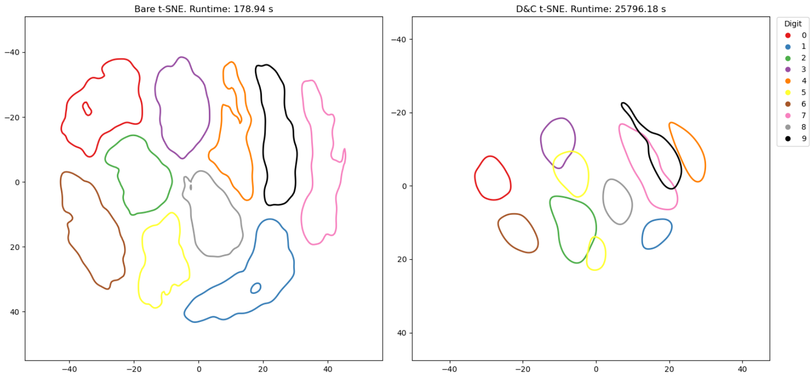


Figure 16: Kernel density estimation of the bidimensional embeddings of the whole MNIST dataset by bare (left) and divide-and-conquer (right) t-SNE. The arguments used were $Perp = 20$, $n\_iter = 100$ and in divide-and-conquer there also were $l = 1000$ and $c = 100$. Contour lines are at 70% of the maximum estimated density for each digit and embedding.

t-SNE's insconsistency

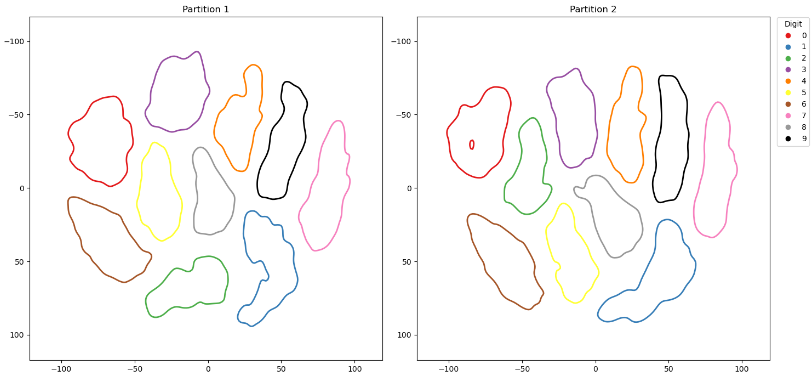


Figure 17: Kernel density estimation of the bidimensional embeddings of two halves of the MNIST dataset. Data was randomly ordered before being splitted. The DR method used was divide-and-conquer t-SNE with $Perp = 30$. Contour lines are at 70% of the maximum estimated density for each digit and embedding.

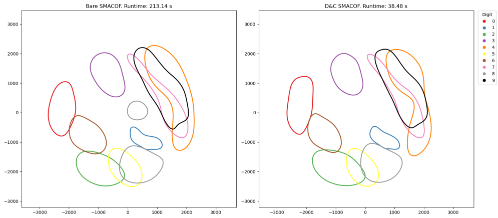
6. Analysis of sustainability and ethical implications

GHG emissions

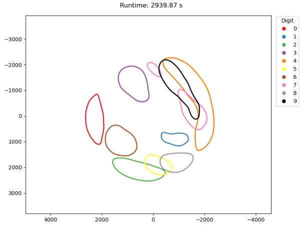
- Data centers generate significant GHG emissions due to high electricity usage.
- Our divide-and-conquer DR framework reduces runtime and hardware demands, lowering emissions.
- It enables sustainable DR by decreasing dependence on supercomputers and improving efficiency.

Visibility of small communities

- DR methods can emphasize biases.
- However, more data ⇒ more likely to represent small communities.



(a) 5,000-point subset of MNIST



(b) whole MNIST

7. Conclusions

Conclusions

- Developed a general divide-and-conquer framework for distance-based DR methods, reducing time and memory complexities.
- Achieved strong embedding quality on large datasets, notably projecting a 10^8 points Swiss roll in 3 h on a standard computer.
- Contributed to making advanced DR techniques more accessible and sustainable for big datasets.

Future work:

- Formalize the framework into a Python package.
- Analyze the effect of c on performance and embedding quality.
- Investigate why LMDS cannot unroll the Swiss roll manifold.

Conclusions

- Developed a general divide-and-conquer framework for distance-based DR methods, reducing time and memory complexities.
- Achieved strong embedding quality on large datasets, notably projecting a 10^8 points Swiss roll in 3 h on a standard computer.
- Contributed to making advanced DR techniques more accessible and sustainable for big datasets.

Future work:

- Formalize the framework into a Python package.
- Analyze the effect of c on performance and embedding quality.
- Investigate why LMDS cannot unroll the Swiss roll manifold.

Thank you!

References i

- Berge, Jos M.F. ten, Henk A.L. Kiers, and Jacques J.F. Commandeur (1993). **"Orthogonal Procrustes rotation for matrices with missing values".** In: *British Journal of Mathematical & Statistical Psychology* 46, pp. 119–134. DOI: `10.1111/j.2044-8317.1993.tb01005.x`.
- Chen, Lisha and Andreas Buja (2009). **"Local Multidimensional Scaling for Nonlinear Dimension Reduction, Graph Drawing, and Proximity Analysis".** In: *Journal of the American Statistical Association* 104.485, pp. 209–219. DOI: `10.1198/jasa.2009.0111`.
- 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`.
- Kruskal, Joseph B. (1964a). **"Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis".** In: *Psychometrika* 29.1, pp. 1–27. DOI: `10.1007/BF02289565`.

Kruskal, Joseph B. (1964b). **“Nonmetric multidimensional scaling. A numerical method”.** In: *Psychometrika* 29.2, pp. 115–129. DOI: 10.1007/BF02289694.

Leeuw, Jan de and Patrick Mair (2009). **“Multidimensional Scaling Using Majorization. SMACOF in R”.** In: *Journal of Statistical Software* 31.3, pp. 1–30. DOI: 10.18637/jss.v031.i03.

Maaten, Laurens van der and Geoffrey Hinton (2008). **“Visualizing Data using t-SNE”.** In: *Journal of Machine Learning Research* 9.86, pp. 2579–2605.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. (2011). **“Scikit-learn. Machine learning in Python”.** In: *Journal of machine learning research* 12.Oct, pp. 2825–2830.

References iii

- Poličar, Pavlin (2023). ***openTSNE. Extensible, parallel implementations of t-SNE.*** Version 1.0.2. URL: `https://opentsne.readthedocs.io/en/stable/benchmarks.html`.
- 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`.