KU LEUVEN

FACULTY OF SCIENCE

Visualizing Algorithms of Nonlinear Dimensionality Reduction Techniques

Lorenzo AMABILI

Supervisor: Prof. Jan Aerts
Visual Data Analysis Lab

1st Mentor: Jansi Thiyagarajan
Visual Data Analysis Lab

2nd Mentor: Thomas Moerman
Visual Data Analysis Lab

Thesis presented in
fulfillment of the requirements
for the degree of Master of Science
in Statistics

Academic year 2016-2017

Intended Audience

The intended audience of this thesis is all people who want to understand the dimensionality reduction process.
Thanks to the universality of visual language, the main aspects of some (nonlinear) dimensionality reduction techniques are illustrated in this work. We describe the algorithms of six popular dimensionality reduction methods not only by mathematical equations but also by a series of visualizations per each technique. This dual approach allows a general audience to understand the mechanism leading to graphical representations of high-dimensional data spaces. Following some visual storytelling, the readers are guided to gradually deal with usual issues involved in (un)supervised clustering. Comments and discussions are included in each section to induce the reader to think about relevant aspects of the subject.
Thus, we addressed the contents principally to students, although advanced topics are discussed and further reading are provided to satisfy more expert reader needs.
Obviously, prior knowledge of linear algebra and basic statistics can be helpful to comprehend all the examined material.

"The essence of independence is to be able to do something for one's self"

Maria Montessori

Acknowledgements

Studying in Leuven has been a fantastic, intense and formative experience.
The friendly international environment allowed me to grow professionally as well as personally. Working with people with different background from mine, required flexibility, curiosity and, especially, a common attitude based on the respect of diversity.
Moreover, these years in Leuven let me appreciate the study dedication of people of any nationality and helped me find great inspiration and motivation for achieving my goals. I am proud to have studied at KU Leuven, a high-level University oriented towards students offering several services and all the resources needed for enhancing individual learning.

Thus, I would like to thank all people who helped me, supported me and made me feel loved during these years. First of all, thank you, lovely mom, dad and sister. They have always given me love and I am happy to share the joy of this achievement especially with them. Then, I thank Prof. Jan Aerts. The common choice of the thesis topic has been an important key point of my academic path. Also, his supervision based on a balance between clear guidelines and freedom to explore, let me learn and work at ease. I would like also to thank my mentors Jansi and Thomas. Completing my thesis could never be possible without their suggestions, feedbacks and discussions. Moreover, I would like to thank all Professors of MSc in Statistics. Their professionalism and preparation motivated me to improve and to push my limits further. I want to thank my amazing friends, my fantastic classmates and my warm housemates. All of them colored my days and made every aspects of studying abroad even more pleasant. I thank also all my old friends who supported me and I am looking forward to see them again. Furthermore, I would like to thank the KUL football team, the KUL futsal team and, especially, the ESN football team. I enjoyed playing football with them every Sunday which gave me the feeling to be in a second family. Finally, I thank also all KU Leuven staff and especially Agora, Alma and Pangaea staff. They simplified my study life in Leuven.

"If you want to go fast, go alone. If you want to go far, go together"

African proverb

Abstract

High-dimensional data are common in many fields of application such as computer vision, biology, neuroimaging, meteorology and many others. Handling them is problematic due to the time and storage space they require and to the complexity in visualizing them. In addition, in high-dimensional data, it is likely that some attributes are correlated and this worsens the performance of statistical models. For these reasons, dimensionality reduction has become crucial for preprocessing these kind of data. A deep understanding of dimensionality reduction methods is important for selecting the appropriate technique in any different case and to interpret the results correctly. In this work, we show a series of visualizations per each dimensionality reduction technique studied (Kernel PCA, Isomap, LLE, Sammon Mapping, SNE, t-SNE) which illustrates the algorithms. The focus is on the nonlinear methods as they are more powerful but also more complex to use. Using a visual storytelling approach, we guide the reader in a learning process which combines graphical representations, practical explanations and deeper considerations about the dimensionality reduction issues. Visual examples are provided to compare how the different techniques work on the same data as well as to compare how the same technique works on different data. Furthermore, real and artificial data were used in order to enhance both practical and theoretical understanding of the algorithm use.

Acronyms

DR	Dimensionality Reduction
NLDR	Nonlinear Dimensionality Reduction
LDR	Linear Dimensionality Reduction
LLE	Local Linear Embedding
MDS	Multidimensional Scaling
PCA	Principal Component Analysis
PCs	Principal Components
SM	Sammon Mapping
SNE	Stochastic Neighbor Embedding
t-SNE	t-Student Stochastic Neighbor Embedding
kPCA	Kernel Principal Component Analysis
r.v.	Random Variable
KL	Kullback-Leibler
SVD	Singular Value Decomposition
k-NN	k Nearest Neighbors
2D	Two-dimensional
3D	Three-dimensional
N	Total number of observations
C	Cost function

Contents

List of Figures

Introduction

Motivations

Generally, the data science process is composed of three main phases in which the data are handled: the preprocessing, analysis and post-processing.
Although the analytical part is the core of this process, the preprocessing phase plays a fundamental role in the statistical process. It can significantly affect the results and without performing it, the analysis may not be able to be carried out. During the last decade, the big data market has grown exponentially thanks to the fact that nowadays is easier to collect, store and analyze high-dimensional and big data by using current technologies [103]. However, many statistical models suffer from the so-called *curse of dimensionality* [13]. In high-dimensional spaces, data tend to be sparse and hence certain statistical methods are no longer applicable unless there are a large number of observations. Moreover, working with large scale data sets can still require a large amount of memory and computation power, even if some effort to remedy this have been done [66]. Thus, dimensionality reduction methods play an important role in the statistical process allowing the data analysts to reduce the number of original data features. Furthermore, DR methods enable us to visualize the original data in 2D or 3D plots.
These methods are used extensively in many active research areas such as wood inspection, face recognition, sound source localization, speech recognition, analysis of fMRI data, supervised or semi-supervised learning problems, novelty detection, geospatial data, visualization of biomedical data, head pose estimation, gene data, image processing and data visualization [60].
The dimensionality reduction methods are usually categorized in *Feature selection* or in *Feature extraction* methods [75]. Feature selection consists of finding a subset of the original dimensions following some rules. For instance, in filtering data one excludes some dimensions and work with those which remain. Alternatively, by applying a feature extraction method one aims to transform the data from a high-dimensional space to a lower-dimensional space, usually to a 2D or 3D space.
In this thesis, we focus on feature extraction techniques and, more specifically, on nonlinear dimensionality reduction techniques. But, since the number of methods which follow in this subgroup cannot be discussed in only one paper, we focused on Sammon Mapping [46], Kernel PCA [83], Isomap [88], Locally Linear Embedding [79], Stochastic Neighbor Embedding [42] and t-Distributed Stochastic Neighbor Embedding [94].

The Big Picture

The vast use of high-dimensional data by many users with different backgrounds from statistics can lead sometimes to a generalization and misuse of NLDR methods. Due to the highly abstract nature of these techniques and to the huge number of existing methods, it is not easy to understand how they work, which technique to use and how to interpret the results. These barriers between NLDR methods and unfamiliar users can lead to either incorrect or an undesired outcome, or even the inability to work with large scale data.
In fact, the algorithms of these techniques look like 'black-box' algorithms. They consists of different steps in which the original high-dimensional data space is transformed many times through different iterative processes and/or decompositions. In addition, some of them do not work directly with the raw data but with abstract measures of (dis)similarity of the data points and, finally, the output space is obtained by extraction, projection or optimization. It is not obvious what is going on during this procedure as a sequence of different sub-algorithms take place in every DR algorithm. Besides, interpreting the final embedding can be complicated since it is not possible to compare input and the output spaces directly, especially if the number of dimensions in the original data space is high.
For these reasons, visualizing the transition of data from the input space to the output space during the dimensionality reduction procedures could help a general audience and students to grasp the main ideas behind NLDR techniques and let the readers to use and interpret them adequately in the future.
This is the main objective of this thesis.

The Objective

In the following pages, we aim to introduce the main concepts of dimensionality reduction, to explain the logic behind the above-mentioned NLDR algorithms and to suggest a critical approach to the interpretation of results with the support of visualizations.
The need to make clearer how NLDR techniques work by using a universal language arises from the use of them in many fields of application. As a consequence, it is important that the message is well conveyed to students of different theoretical backgrounds. Therefore, the focus of this work is to visualize the mapping of data from high-dimensional space to low-dimensional space while following some visual design rules.
First, we should make use of visual storytelling to represent these dynamic processes through a sequence of (static) visualizations. The user is involved in a guided learning process designed to make connections between new concepts avoiding an initial "blind" interaction. This facilitates visual comparisons without need to memorize every step *("eyes beat memory", McKinlay)*. However, as the fundamental aspects will be discussed the readers are encouraged to explore the topics of their interest, enhancing the creative learning process [71] and, for this reason, we made already available an interactive version of this project in NLDRviz [57]. Moreover, the final work should not be unjustified 3D visualizations which can be less intuitive [72] and the data-to-ink ratio should be maximized [91].
We hope that this thesis will be a useful tool for master students and, in general, for

a non-expert audience who aims to use NLDR techniques correctly. Through visual storytelling, not only the process of learning can be accelerated but it also can be delivered in a more complete way. Thus giving clues to how to visualize the mechanisms behind NLDR methods. This is essential to understand abstract concepts.

The Challenges

Generally, teaching abstract concepts is a complicated task. Especially, if several abstract topics are grouped together in the learning process.
In this work, the objects of interest are techniques which involve abstract topics such as linear algebra, topology, multivariate statistics and geometry. Therefore, our ultimate goal was to make these concepts accessible to a larger audience. During the visualization process, it is useful to consider the problem on three levels: *What* we are visualizing, *Why* we are visualizing it and *How* we are visualizing it [72]. The last component is the most technical one, typical of the visual designer and it requires different actions before ending up with the final visualization. For instance, preprocessing big data is a common option to obtain visual representations of them. Thus, the *What* we want to visualize (i.e. the dimensionality reduction methods) was the usual *How* to visualize objects [80].
Finally, the visualization techniques each consist of different algorithms themselves. This has been an issue since we wanted to limit the total number of frames per storytelling whereas several frames would have been necessary to completely illustrate every step. This would have increased the complexity of the final results which should be expressive, intuitive and consistent to be effective.

Related Works

There are not many related works which aim to visualize the NLDR algorithms. However, there are some projects which visualize statistical concepts or simpler algorithms in an efficient way.

Recently, Wattenberg et al. developed How to Use t-SNE Effectively, a visual tool which shows how t-SNE works with different artificial data sets [99]. The user can set the required hyperparameters of the algorithm for producing the final visualization. Moreover, the sometimes deceptive features of t-SNE's final output such as the distances between clusters and their size are discussed and explained. This is a must-see for anyone who wants to explore the functionalities of this technique before experiencing it themselves.

Mike Bostock, one of the key developers of D3.js [68], produced alternating static and dynamic visualizations from a series of algorithms in Visualizing Algorithms [70]. He compared different algorithms per task mixing various visual design approaches among which the storyboard approach. Yet, his work is not only efficient but also aesthetically pleasant and a lesson of visual design. The algorithms illustrated concern

sampling, shuffling, sorting and maze generation.

Daniel Kunin created a project to teach statistical theory through visualizations, Seeing Theory [25]. The work consists of a set of interactive visualizations in which statistical basic concepts including confidence intervals, ordinary least squares and hypothesis testing are visualized using D3.js. It is intuitive and users can interact and actively learn by changing the algorithm settings.

A visual introduction to machine learning by R2D3 is a fusion of machine learning and visual design, by showing how to make data analysis throughout an interactive visual storytelling [77]. Changing marks and channels as one scrolls the web page, this project illustrates the statistical processing of data, alternating different visual design methods at each step of the analysis.

An explanation of PCA is made by Victor Powell and Lewis Lehe in Explained Visually [97]. The 2D and 3D visual representations of PCA are particularly significant as the user can see the effects of dimensionality reduction directly and PCA linear data transformations in low-dimensional spaces. It is also possible to interact with the visualizations which makes the concept of maximum variance more easily retained.

Another visual explanation of abstract concepts such as momentum is a work made by Gabriel Goh, Why Momentum Really Works [34]. It describes the mechanism of the optimization algorithms focusing on its most relevant features and in the role played by the momentum.

The ongoing project by Steven Halim, VisuAlgo helps to understand data structures and algorithms, by allowing the users to learn interactively on their own [87]. The concepts illustrate mainly graph theory, machine learning and computer sciences whose many applications are available, for instance, hash tables, segment trees and shortest path. Overall, this tool is intuitive and simple to use thanks to a friendly user-interface.

For further material about dimensionality reduction, machine learning and algorithms, we also suggest referencing other interesting projects [67, 15, 32, 18, 90, 22, 39, 92].

Background Knowledge

Before diving into the core of this thesis, it is important to clarify some key concepts of the dimensionality reduction methods. A short description of the distance metrics related to NLDR methods of interest and a comparison between linear and nonlinear DR methods are made. In addition, we recall the main ideas regarding PCA and MDS (and SVD), the most widely used (linear) DR techniques over the last century.

Distance Metrics

The main goal of dimensionality reduction methods is to represent each data point of the original space by a point in a lower dimensional space attempting to preserve the neighborhood [55]. This is strongly related on the choice of distance metrics (or better, dissimilarity measures) measuring the similarity between objects and on the curse of dimensionality effects [10]. For high-dimensional data, the concept of proximity becomes meaningless and the dimensionality reduction algorithms lose effectiveness. This problem of poor discrimination between the nearest and furthest neighbors expands exponentially as the dimensionality increases [13]. Therefore, the distance measure choice affects the NLDR algorithm performance. There are several dissimilarities measures for quantitative data [35] but we focus on only some of them.

Euclidean Distance

The most popular and widely-used distance metric is the *Euclidean distance*. Its main property of giving greater emphasis to larger differences within a single variable is undesirable in high-dimensional space context. This is because high-dimensional spaces can be viewed as cubic spaces with much denser corners in comparison to the rest [48]. And the Euclidean distances must be seen as distances countable only in the sphere contained in this cubic space. Hence, only two objects located in the area in which the Euclidean distance is meaningful can be well represented by this metric. Unfortunately, this is only a small portion of data in high-dimensional spaces. It can be written mathematically as follows

$$d(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}|| \tag{1}$$

where x and y are two different data points in the n-dimensional data space.

Geodesic Distance

A generalization of the concept of a straight line, as intended for the Euclidean distance, to curved spaces is the *Geodesic distance* [88]. To give an idea of this metric, a

well-known application of it is the definition of horizontal distances on Earth. M is a geodesic if it is true that

$$d(\gamma(t_1), \gamma(t_2)) = v|t_1 - t_2| \tag{2}$$

where $\gamma : I \to M$ is a curve from an interval I to a metric space M, $t \in I$ and $t_1, t_2 \in J$ with J a neighborhood of t and $v \geq 0$ is a constant. If $v = 1$, M is the geodesic *shortest path* which measures the distance between two nodes in a graph through the number of edges in the shortest path connecting them.

Kernel-based Distance

A mathematical trick called *Kernel trick* allows us to measure the distance between data points of an implicit high-dimensional space without computing the coordinates of the data in that space. This is possible by computing the inner products between all pairs of data instead. In this way, the explicit representation of the mapping is avoided and operations on the implicit space are possible, as measuring the distances between data points. For instance, the Kernel function K should be known a priori to obtain accurate results [107]. The Kernel function is the relationship of the manifold where the data points lay and the original space. It has a mathematical form as follows

$$K_{i,j} = (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)) \tag{3}$$

where Φ is the feature map $\Phi : X \to V$, X is the input space and V is the implicit space.

Kullback-Leibler Divergence

Unlike the above-introduced distances, the *Kullback-Leibler* divergence is not really a metric. It is a measure of dissimilarity between the theoretical and empirical probability distributions [94]. The probabilistic metric space is based on distribution functions rather than real numbers, with distances from 0 to 1 instead from $-\infty$ to ∞. In other words, a dissimilarity measure is not based on the physical distance between two objects but it is a further generalization of it.
In contrast with the distance metrics, the Kullback-Leibler divergence is more robust to issues due to the high-dimensional space properties. Furthermore, it is asymmetric (i.e. $d(i, j) \neq d(j, i)$) which means that the divergences are not weighted equally.
A general equation of the KL for discrete r.v. is the following

$$KL(g(y), f(y, \theta)) = \sum_{0}^{\infty} g(y)\{log(g(y)) - log(y, \hat{\theta})\} \tag{4}$$

where $g()$ is the theoretical probability distribution and $f(,)$ is the empirical probability distribution.

The choice of distance metrics plays a critical role in defining the DR methods. As it is shown in Section: **??**, depending on the available information regarding the dimensionality reduction problem, one technique will perform better based on the dissimilarity measure involved. Factors that play a role are the number of dimensions in the original data space, number of data points, and type of intrinsic manifold nonlinearity.

Linear vs Nonlinear

Several DR methods have been proposed since the first DR technique invention, PCA [43]. During the last years, some attempts to categorize all of them have be done [55] and there is not a unique way to do it. However, one clear distinction which can be made is their nature, namely if their specified model is linear or nonlinear.
Generally, the nonlinear models are often preferred when the sub-manifold is not embedded linearly in the input space. In other words, when the most relevant dimensions of the data are nonlinearly hidden into the original data space. Since the LDR techniques are limited to second order statistics and to linear projections [106], they fail in these kind of problems whereas the nonlinear models perform much better. On the other side, NLDR techniques often need many hyperparameters and, because of that, they require more computational time, storage space and large amounts of data.
The nonlinear mapping can preserve the high-dimensional data structure in the low-dimensional representation of the input space in a more accurate way than how the linear transformation do. For this reason, NLDR techniques are widely used nowadays and we preferred to focus on these methods.

Gradient Descent and SVD

A relevant property of DR techniques is also how they obtain the optimal representation of each data point by a point in a lower dimensional space. In this work, we can divide the techniques in two subgroups by method used to find the final embedding: *optimization method-based* and *eigenvalue decomposition-based* techniques.

Gradient Descent

Also called steepest descent, the gradient descent algorithm is well-known in the field of optimization and largely used for its benefits such as working in spaces of any number of dimensions. It is mostly based on the only concept of gradient which is the multi-variable generalization of the derivative [30]. Based on the objective function, it aims to reach the optimal solution step by step following the opposite direction indicated by the gradient. There is a *step-size* parameter which determines the magnitude of transaction at each step. The algorithm stops when the maximum number of iterations has been reached or when the updated solution is close to the optimal based on a tolerance parameter. Although the algorithm is simple and quite accurate, in some cases it is possible that it confuses a local optimum with a global optimum; the two solutions have similar characteristics albeit they are different from each other.
Mathematically, it can be written as follows

$$x^{(n+1)} = x^{(n)} - \alpha^{(n)} \Delta f(x^{(n)}) \tag{5}$$

where x is a point, n is the number of iteration, α is the step-size and $\Delta f(x)$ is the gradient of $f(x)$.

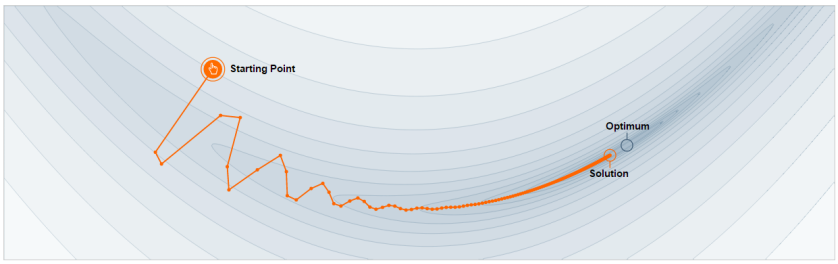


Figure 1: *Visual representation of the gradient descent algorithm* **Source:** *Why Momentum Really Works, Gabriel Goh* [34]

Another limitation of this algorithm is that it can converge slowly. For this reason, it is useful to use a *momentum* m to increase the step-size as shown below

$$m^{(n+1)} = \beta m^{(n)} + \Delta f(x^{(n)}) \tag{6}$$

$$x^{(n+1)} = x^{(n)} - \alpha^{(n)} m^{(n+1)} \tag{7}$$

It is an acceleration which speeds the optimal solution search, leading to faster convergence by dampening oscillations and creating different new ones [64].

Singular Value Decomposition

The *Singular Value Decomposition (SVD)* is a matrix decomposition of a real or complex matrix A [49]. The decomposition consists of a diagonal matrix Σ whose diagonal entries are the singular values of A and of the matrices U and V whose their columns are the left and right singular vectors, respectively. In Fig. 2, there is shown SVD of a general matrix A.

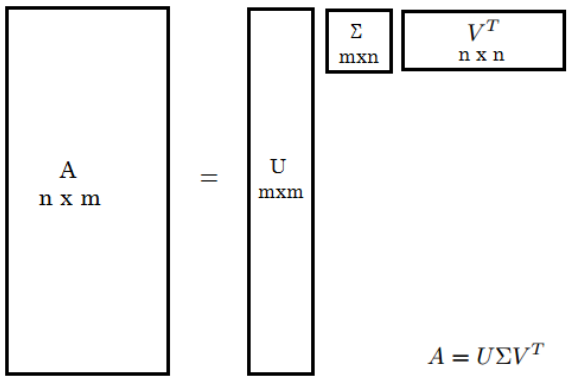


Figure 2: *SVD form of a matrix A*

The decomposition can be obtained by different algorithms. One of the most intuitive (and simple) ones is the *power method* [48]. The iterative procedure consists of a simple matrix multiplication between the matrix A and an initial vector v, divided by the norm of this product.

The algorithm always converges to an eigenvector associated to the dominant eigenvalue and it is written as follows

$$v^{(k+1)} = \frac{Av^{(k)}}{||Av^{(k)}||} \tag{8}$$

where $||Av^{(k)}||$ is the eigenvalue associated to the eigenvector $v^{(k)}$.

A special case of SVD is the *eigenvalue decomposition* [9] which applies only on symmetric matrices.
The fundamental theory of eigenvalue decomposition is based on the following equation

$$Av = \lambda v \tag{9}$$

where λ is the eigenvalue associated to the eigenvector **v**.
If a (non-zero) vector satisfies this linear equation, it is an eigenvector of the matrix **A**.
Finally, it is important to recall that the eigenvalues and the eigenvectors can be seen as length and direction of vectors and that the latter ones do not change when linear transformations are applied to the vectors.

Thus, the techniques which involve the eigenvalue decomposition have some benefits: no tuning parameters are required for executing the algorithms, there is not an iterative process which can slower the algorithm and there is no local optima issues as for the gradient descent optimization algorithm.

PCA and MDS

The NLDR methods are the general cases of the more intuitive LDR methods. In this section, we describe shortly PCA, MDS and how they are related to each other.

Principal Component Analysis

PCA consists of finding linear combinations of the original variables capturing as much variance as possible, the so-called *principal components*, and projecting the original data on them [7]. The PCs are found by eigenvalue decomposition and they are the eigenvectors associated to the largest eigenvalues of the covariance matrix. We report its mathematical expression under the form of cost function

$$U_1 = w^T X \tag{10}$$

$$C = \sum_{i=1}^{N} ||x_i - U_d U_d^T(x_i)||^2 \tag{11}$$

where U_1 is the first principal component with maximum variance and $w = [w_1, ..., w_n]$ are the weights. The solution for U can be expressed as SVD of the original data matrix X.

Multidimensional Scaling

Similarly to PCA and the other DR methods presented in this work, there are different versions of these techniques but we refer to the general definition, hence, to *classical MDS* here [63]. It aims to minimize the squared difference between the distances of two points in the higher-dimensional and lower-dimensional spaces.
One way to write it is the following

$$C = \sum_{i=1}^{N} \sum_{j=1}^{N} (x_i^T \cdot x_j - y_i^T \cdot y_j)^2 \tag{12}$$

where x_i and y_i are the data points in the higher-dimensional and in the lower-dimensional spaces, respectively.
The goal is to identify the coordinates of the N points in the low-dimensional space minimizing the cost function, also called *least squares stress function*.

PCA vs MDS

In the literature, since the invention of PCA and MDS, they have been compared often. In a nutshell, PCA aims to preserve the (co)variance of data, MDS aims to preserve the distance between data points. However, they are the same when MDS is applied to a distance matrix based on the Euclidean distance used for measuring the dissimilarity of data points in the input space. Therefore, MDS is useful when the original data is not available and only a distance matrix is defined [9]. From eq. 1 and eq. 2, it can be seen that both of them aim to minimize the squared errors between the distances measured in the two different spaces [33].

Further reading

We also recommend learning some useful concepts which have not been considered in the theoretical framework such as the choice of the optimization method [30, 74] and the different uses of PCA and MDS related to the aims of this work [52, 16, 106].
In addition, further reading is also suggested to have more details about the above-introduced basics of DR techniques [24, 55, 9, 108, 8].

Methodology

The design process of this thesis consisted of three main stages: the analysis of the NLDR algorithms, the visualization process and the final idea implementation.
First, we studied the algorithms, their characteristics and their mechanism. Running the algorithms on toy examples gave some insights on the parameter settings and that experience needed to master the techniques. As a result, we could make a first selection of the suitable visual designs limiting the design space. Afterwards, we followed some design approaches to reach a (near-)optimal solution to our research question through visual storytelling.

In the following sections, the objects of study are outlined, the algorithms of NLDR techniques (Section: *Nonlinear Data Reduction techniques*), an overview of design methods and the design strategy are provided (Section: *Design Methodology*) and a description of the data sets used for the implementations is presented (Section: *The Data Used*).

Nonlinear Data Reduction techniques

In this section, we describe NLDR techniques of interest detailing their objective functions, advantages and disadvantages. Furthermore, we illustrate the connections between each other and with MDS and PCA.

Kernel PCA

In *Kernel PCA*, principal components can be computed correctly in N-dimensional feature spaces (where N is the number of data points) that are related to the input space by some nonlinear mapping. As the name suggests, it is based on the Kernel method and this is the main difference from PCA which makes kPCA more effective when data are embedded in nonlinear manifolds [100].
The key point of kPCA is to create a NxN Kernel matrix which allows to consider a nonlinear high-dimensional mapping without working explicitly on it. In fact, we extract the projections of data on the PCs of the actual N-dimensional feature space by eigenvalue decomposition of the Kernel matrix and not the PCs themselves [38].
Its cost function is the following

$$C = \sum_{i=1}^{N} ||\Phi(x_i) - U_q U_q^T \Phi(x_i)|| \tag{13}$$

where we assume $\sum_{i=1}^{N} \Phi(x_i) = 0$ and $K = \Phi(x_i)^T \Phi(x_i)$ is the Kernel matrix.

The algorithm of Kernel PCA

- Computing the Kernel matrix by Kernel function

- Eigenvalue decomposition of the Kernel matrix (by power method)

- Projection of the Kernel matrix to the eigenvectors with the greatest eigenvalues

Thus, the advantage of using kPCA is to apply dimensionality reduction to data based on the eigenvalue decomposition of an even higher dimensional space showing the right characteristics of the manifold in which the data points lay (assuming the chosen Kernel as correct).
The main drawbacks in using kPCA are the fact that the Kernel function must be known a priori and is impractical to use it on large data sets even though some effort to improve it have been done [84].

Isomap

Isomap is a global nonlinear generalization of MDS.
In a nutshell, it consists of computing the neighbors of each data point in high-dimensional data space, usually by the k-NN method and represents it as a weighted (neighborhood) adjacency graph G. Then, the geodesic distances are estimated by computing the shortest paths (by Dijkstra's algorithm or Floyd's algorithm, for example) through the undirected edges connecting neighbors in G for all pairs of data points. Successively, MDS (or PCA) is applied to the new matrix obtained to find a lower-dimensional embedding in Euclidean space for the reconstructed data points.
In the following equation, we show what the cost function looks like

$$C = \sum_{i=1}^{N} ||\tau(D_G) - \tau(D_Y)|| \tag{14}$$

where τ is an operator defined by $\tau(D) = -HSH/2$, where $S_{ij} = D_{ij}^2$ is the matrix of squared distances and $H_{ij} = \delta_{ij} - 1/N$ is the *centering matrix*. D_Y and D_G denote the matrices of Euclidean and geodesic distances, respectively. Hence, the cost function of Isomap is, basically, an Euclidean norm of the difference between geodesic and Euclidean distance matrices.

The algorithm of Isomap

- Determining the neighbors of each point

- Constructing a neighborhood graph

- Computing shortest path between all the nodes (as GD)

- Computing lower-dimensional embedding (MDS/PCA)

Computing the geodesic distances in the original data space allows us to take into account the nonlinear manifold in which the data points lay, assuming that the G graph is connected and the neighborhoods on graph reflect those on manifold. One hyperparameter has to be set generally, the number of neighbors k but this can be an issue. In fact, if k is too large, the *short-circuits* errors can occur which can alter all the reconstruction in low-dimensional space whereas if k is too small, the neighborhood graph may become too sparse to estimates geodesic paths accurately [12].
Furthermore, Isomap can be slow due to the steps needed to accomplish a final embedding as the cost function is minimized in the high-dimensional space (eq. 14), although the optimization problem is convex [60].

Locally Linear Embedding

Locally Linear Embedding (LLE) computes a low-dimensional neighborhood preserving embedding of high-dimensional data through estimated reconstruction weights of each neighborhood.
Assuming that weights can define the patches on the manifold as long as they characterize the local geometry in the input data space, LLE defines a linear mapping of the original data consisting of translation, rotation, and rescaling [81]. However, some constraints on the reconstruction weights must to be set to maintain intrinsic geometric properties of data. More specifically, the weights have to be invariant to linear transformations and each data point has to be reconstructed only from its neighbors, which means that we set $\sum_{j=1} W_{ij} = 1$ and, if the ith point is not a neighbor of the jth point, $W_{ij} = 0$.
Thus, in LLE each point can be written as a linear combination of its neighbors.

To compute the weights we minimize the following cost function

$$C = \sum_{i=1}^{N} ||\vec{Y_i} - \sum_{j=1}^{N} W_{ij}\vec{Y_j}|| \tag{15}$$

where W is the weight matrix for the local reconstruction.

The algorithm of LLE

- Determining the neighbors of each point

- Constructing a neighborhood graph

- Finding the reconstruction weights for each neighborhood

- Computing lower-dimensional embedding (MDS/PCA)

Assuming that each data point and its neighbors lay on an approximately linear subspace, LLE gives (globally) highly nonlinear embeddings of the data with local linear properties preserved [55]. An advantage of LLE is the convex optimization, in eq.

15. The objective function is computed in the low-dimension space which saves computational time. On the other hand, this algorithm fails if there is some noise in the manifold or outliers. As a consequence, the points on the map space are collapsed as the constraints are cheated by the extreme values [60]. In addition, the hyperparameter of k-NN has to be set a priori as for Isomap.

Sammon Mapping

Sammon mapping is one of the first NLDR techniques which aims to identify geometric relationships among subsets of the data vectors in the input space. Simply using weighted Euclidean distances, it assigns more importance on the small distances preserving the relationships between nearby points [40]. In this manner, the cost function is optimized by the gradient descent algorithm which minimizes the differences between corresponding inter-point distances in the two high- and low-dimensional spaces.
The Sammon's stress can be written as follows

$$C = \frac{1}{\sum_{i<j}^{N} d_{ij}^{*}} \sum_{i<j}^{N} \frac{(d_{ij}^{*} - d_{ij})^2}{d_{ij}^{*}} \tag{16}$$

where d_{ij}^{*} are the distances between data points in the original space and d_{ij} are the distances between data points in the low-dimensional space.

The algorithm of Sammon mapping

- Computing the distance matrix *D** for the higher-dimensional space

- Initialization of the projections by PCA or normal distribution

- Computing the distance matrix *D* for the lower-dimensional space

- Minimizing the difference between *D** and *D*

Although SM is able to generalize, it shows many limitations such as its inefficiency with complex high-dimensional structures or with a large number of data vectors to handle [28]. Moreover, it is common to encounter a local optimum problem due to a non-convex optimization problem. Furthermore, its stress is mostly on preserving the local structure of data accurately caring less about its global structure. Typically, this results with a circular final embedding.

Stochastic Neighbor Embedding

Stochastic Neighbor Embedding (SNE) is a probabilistic approach to represent high-dimensional objects in a low-dimensional space while preserving the neighbor identities, the topology of data space. Unlike the previous methods, a Gaussian is centered on each object

in the high-dimensional space and the densities are used to define a probability distribution over all the potential neighbors of the object, constructing an embedding based on probable neighbors [42].
Therefore, SNE does not require that each high-dimensional point is associated with only a single location in the low-dimensional space. This is a more precise representation of the original input space since each single point belongs to several disparate locations in the low-dimensional space than by assigning exact coordinates.
Thus, the sum of KL divergences over all data points is minimized by using the gradient descent method as shown in eq. 7 instead of a difference of metrics. Despite that, other divergences can also be successfully used [17].
The similarity of data points in the higher-dimensional space is the conditional probability that a point x_i would be a neighbor of point x_j given that neighbors would be chosen in proportion to their probability density under a Gaussian centered at x_i as the following equation shows

$$p_{j|i} = \frac{exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i}^{N} exp(-||x_i - x_k||^2/2\sigma_i^2)} \tag{17}$$

where $p_{j|i}$ is the conditional probability between data points in the original space, σ_i is the variance of the Gaussian and $p_{i|i} = 0$.

Analogously, the similarity of points in the lower-dimensional space is computed

$$q_{j|i} = \frac{exp(-||y_i - y_j||^2)}{\sum_{k \neq i}^{N} exp(-||y_i - y_k||^2)} \tag{18}$$

where $q_{j|i}$ is the conditional probability between data points in the lower-dimensional space, $\sigma_i = \frac{1}{\sqrt{2}}$ and $q_{i|i} = 0$.

The cost function of SNE is as follows

$$C = \sum_{i=1}^{N} KL(P_i||Q_i) = \sum_{i=1}^{N} \sum_{j=1}^{N} p_{j|i} log \frac{p_{j|i}}{q_{j|i}} \tag{19}$$

where P_i represents the conditional probability distribution over all data points x_j with $j \neq i$ given the point x_i and Q_i represents the conditional probability distribution over all points y_j with $j \neq i$ given the point y_i.

The algorithm of SNE

- Computing the distance matrix *D** for the higher-dimensional space

- Converting *D** to P

- Computing the distance matrix D for the lower-dimensional space

- Converting D to Q

- Minimizing the KL divergence between P and Q

In the algorithm of SNE, D^* and D are the distance matrices of original and mapped space computed by using Euclidean distance as in SM. Theoretically, this algorithm should associate nearby points with a relatively high conditional probability, whereas distant points will have a low conditional probability.

However, SNE presents a similar issue to the previously-introduced techniques called the *crowding problem*. Data points tend to be attracted towards the center of the map by forces connecting each point to the extreme points. If there are more than few, natural clusters do not form due to this phenomenon [94].
Furthermore, the SNE's algorithm includes more than one free parameter to be chosen a priori. In particular, the *perplexity* plays an important role as it determines the effective number of neighbors.

t-Distributed Stochastic Neighbor Embedding

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a variation of SNE. The main difference between them is the use of Student-t distribution with one degree of freedom instead of Gaussian distribution for computing the similarity matrix Q in eq. 6.
This innovation solves the crowding issue since the Student-t distribution employed by t-SNE is a Cauchy distribution which is a special case of the Gaussian distribution with heavy tails. As a result, the previously-considered extreme values are now considered as standard points softening substantially the forces derived by them [94].
Besides, t-SNE aims to minimize a symmetric version of the SNE cost function which is based on joint probability distributions. The cost function of t-SNE is symmetric because it benefits with a nice property, $p_{ij} = p_{ji}$ as well as $q_{ij} = q_{ji}$.
However, the joint probabilities referred to the high-dimensional data points have to be defined as shown in eq. 20 instead of a more deducible form as in eq. 21. Turning conditional probabilities into pairwise probabilities, each data point contributes significantly to reduce the cost function.
This is shown in the following equation

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{20}$$

where p_{ij} is the symmetric conditional probability and n is the number of data points.

The similarity of points in the lower-dimensional space is computed through a Student-t Kernel as follows

$$q_{ij} = \frac{exp(-||y_i - y_j||^2)}{\sum_{k \neq l}^{N} exp(-||y_k - y_l||^2)} \tag{21}$$

The cost function of t-SNE is the following

$$C = \sum_{i=1}^{N} KL(P||Q) = \sum_{i=1}^{N} \sum_{j=1}^{N} p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{22}$$

where P and Q are the joint probability distributions.

The algorithm of t-SNE

- Computing the distance matrix D^* for the higher-dimensional space

- Converting D^* to P

- Computing the distance matrix D for the lower-dimensional space

- Converting D to Q

- Minimizing the KL divergence between P and Q

In terms of efficiency, it is a remarkable improvement of SNE by alleviating the crowding problem and its superiority among the unsupervised parametric DR techniques has been shown [59]. In addition, a t-SNE variation can also successfully visualize non-metric similarities (e.g. word associations or event co-occurrences) by constructing a collection of maps [95]. However, t-SNE is also characterized by a non-convex optimization problem and its convergence can be slow or erroneous.

Relationships among the techniques

NLDR methods are strongly related to each other. Starting from PCA and MDS, they are actually equivalent. It can be seen from their cost functions in eq. 12 and in eq. 16 that they aim to minimize the same criterion [55]. SM is a nonlinear (weighted) version of MDS for a certain metric distance scaling [16]. The difference between them is the greater stress on representing nearby points in the original space as nearby points in the mapped space as well by SM.

SNE and t-SNE are similar based on the same concept of using probabilities instead of coordinates of data points for mapping as discussed previously. Isomap and LLE are also similar as shown by their algorithm. They are *spectral methods* based on an initial k-NN search to construct an adjacency graph of the data in which the similarities among data are computed. However, whereas both Isomap and LLE map nearby points on the manifold to nearby points in final embedding, only Isomap aims to map distant points as distant points. On the other hand, LLE is more computationally efficient and it performs better on manifolds whose local geometry is close to Euclidean but whose global geometry may not be [26]. This is why they are considered global and local NLDR methods. Finally, they also share the same issues: the short circuits problem and the collapse problem [26, 82].

It is also possible to show that Isomap and LLE (and PCA and MDS) are special cases of kPCA [33]. Therefore, SM is also a special case of the kPCA since it is a variation of MDS [101]. This can be explained by the fact that all the dissimilarities measures presented in this section can be seen as Kernel functions. Thus, making a larger deduction, SNE and t-SNE are a probabilistic version of MDS [95]. Hence, they are also connected to kPCA which can be considered a general form of the DR techniques.

Further reading

The field of nonlinear data reduction techniques is vast and there exists many different variants of each methods illustrated above [23]. Therefore, in order to provide a wider theoretical framework, we suggest consideration of some applications of kPCA [104, 98], some effort done to exploit SM's properties [28, 14] and to improve its performance [73] and how LLE and Isomap are strongly related to methods based on probabilistic approach for constructing the distance matrix [53]. Moreover, a variant of SNE, UNI-SNE, which is much better at showing the boundaries between classes [21] and the variation of t-SNE to speed its convergence up [93] should be checked. Finally, to have more details about robust DR methods, NLDR techniques in general and about future developments of this field some further readings are suggested [29, 55, 96].

Design methodology

The visualization process is a kind of algorithm itself. There is an input, a object of interest and there is an output, the final visualization. Then, a series of actions must be executed. They can be cycles or conditioned tasks. Altogether they create the process flow and its structure is well defined by a pipeline model. Similarly to dimensionality reduction process, more than one technique can perform adequately in a certain case since they share some properties. However, one of them performs better in that particular case. Once the technique has been selected following certain criteria, some hyperparameters must be set. Several possible solutions can be found by editing the initial settings and, the process can also end finding a local optimum. In that case, the process must be initiated again. Furthermore, the optimum could never be reached, although it could be well approximated.
Based on the mechanism of the NLDR algorithms, we explored some visual design methods which led us to obtain a series of visual storytellings.

Visual Storytelling

The main reason this work has been conceived is to provide a useful tool for educational purposes. Consequently, we focused on visual storytelling as means to communicate with and to involve our audience in the learning process.
Learning through storytelling can create strong links between theory and practice. Making use of visualizations, text and various levels of interaction, this technique enhances the accessibility and comprehension of material of interest simplifying complex concepts [54]. Therefore, narrative visualization adds to the communication expressiveness and efficiency with different audiences and topics thanks to its flexibility.
Although it may seem to be a simple design method, sophisticated and powerful storytellings can also be made, conveying abstract concepts connected and expressed by multiple means. Taking advantage of the iterative optimization methods involved in some NLDR techniques, we selected the *partitioned poster*, *comic strip* and *video* as genres of interest since the time factor can be the transition key of story [85]. In traditional stories, usually the frame order corresponds with time which is closely related to causality. As a consequence, by providing the causal relationships between facts

and events all the singular parts are connected together in a cohesive and consistent structure [62].
In addition, the relationships between elements of different frames are enhanced by diversifying (un)labeled data by different colors and shades. More specifically, in case of unlabeled data, the brushing & linking technique can be used to interactively explore patterns in the data [69]. While selecting some items in one plot, they are also highlighted in all other plots. This allows to see how contiguous regions are distributed within different plots [72].

Therefore, we combined object of interest, audience experience regarding the topic and an appropriate set of design techniques for designing the visual storytellings. The possible solutions are infinite but we used some criteria to thin this long list out. For instance, a measure of effectiveness and expressiveness can compare the different visualizations based on their capacity to convey accurately all information, and only that information, to the audience [61]. Also sacrificing details in order to facilitate the understanding, the so-called *data-to-ink ratio* rule [78, 91] is a good selection criterion.
In conclusion, as first choice we opted for a *passive storytelling* in which we decided where the reader attention should be directed. This ensures that the readers are guided during this cognitive process. Besides, static visualizations help to link and easily retrieve all information provided by different frames and enhance comparison between them. On the other hand, *semi-interactive storytelling* could stimulate greater involvement by the readers through interactive exploration of the graphical representations. They can take control of each section and investigate the relationships between various elements. Furthermore, it can also increase the transparency and credibility of the visual data stories, making it more meaningful for us to share our analysis processes and considerations [50].

The Design Strategy

The visualization of high-dimensional spaces has always been an issue. As matter of fact, until the 13th century we were not able to properly represent a 3D space. Only during the Renaissance, through the masterpieces of Filippo Brunelleschi and thanks to Leon Battista Alberti who wrote "De Pictura", the concept of *perspective* appeared for the first time. In Fig. 3, one of the first studies about the perspective is shown.

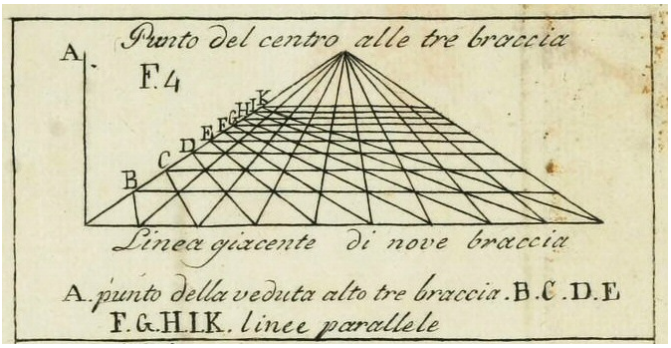


Figure 3: *Visual explanation of the perspective from "De Pictura", written by Leon Battista Alberti in 1518*
Source: Wikipedia

Without listing all the optics and scientists who contributed to the progress made in visual design, nowadays we boast of advanced and elegant techniques to visualize high-dimensional and large data [65, 19, 102].
In this project, we considered the *scatter plot* essential since the output space of interest is 2D and the objects are data points. Although *parallel coordinates* provide an efficient and flexible data visualization, they are not practical with large dimensionality [45, 86]. Alternatively, the *heat map* and *adjacency matrix* are suitable with high-dimensional data [19]. Finally, we considered the *dimensional stacking*, a recursive projection method [65]. However, its difficult interpretation makes it hard to use.
Thus, in order to accomplish the goal of finding an efficient visual design, we followed the 5-design-sheets approach which suggests an initial brainstorming as first step [89]. During brainstorming, we set some rules based on structured games which enhanced the creative process [37]. When problems are highly abstract, the imagination has to be released seeking also for counterintuitive and absurd solutions. Through some focused games, it is easier to let the imagination go.
This diverging phase was, then, followed by an exploration stage. This time, the objective was not to push further the limits of our imagination, but rather to let conceptual ideas emerge from the connection of creative sparks generated initially. The solution started to be shaped and elaborated evaluating all the ideas with a critical point of view and making the final decisions specifically based on the practical aspects. In addition, it has been more effective to sketch potential solutions actualizing prototypes before converging to the final output.
This approach could also be combined to other common approaches in visual design in order to enhance the design development. Therefore, we engaged a small audience during the first step. We showed them some sketches asking the issues they were encountering in interpreting the visualizations and their needs in using NLDR techniques. Thanks to their feedback, we could adjust the visualization design based on their needs and impressions avoiding self-design for goal-focused design [44]. Furthermore, when we started sketching design concepts we also followed the *anti-solution* approach. It consists of progressively identifying what is not working and to add boundaries in the design space. In this way, it enables a faster convergence to the final solution.

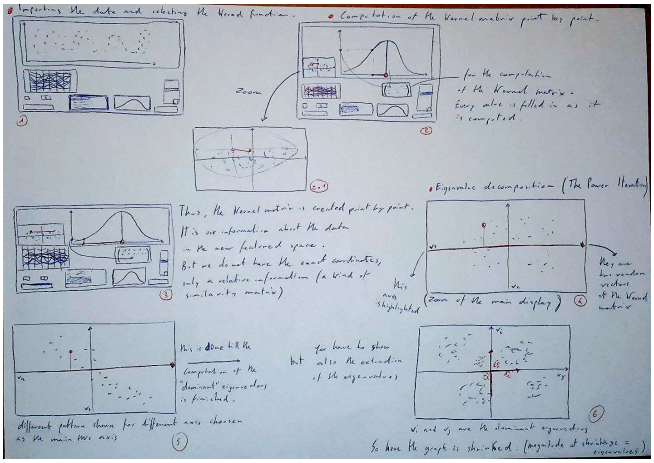


Figure 4: *Initial sketch of the visualization of the Kernel PCA algorithm*

In Fig. 4, a visualization of the kPCA's algorithm is shown. An interactive tool with multiple displays was conceived as the best way to illustrate all the steps of kPCA, including the Kernel matrix computation and the power method process for finding the eigenvectors with largest eigenvalues. Through multiple displays, the user could monitor every computation and its effects immediately. Since every view was based on a different visual technique, this tool could adapt to various data types and data transformations which occur during the DR process. An interactive visual storyboard to combine all the techniques necessary to visualize each task of the process, when possible, in only one tool.
Although it could be an exhaustive way to visualize the algorithm process, it was too complex to implement in a short time and it did not match all the eligibility criteria set initially.
Therefore, a visual storytelling approach based on multiple graphical representations (shown in Fig. 5) was preferred due to its easier interpretation and we modeled the final visualizations from this concept.

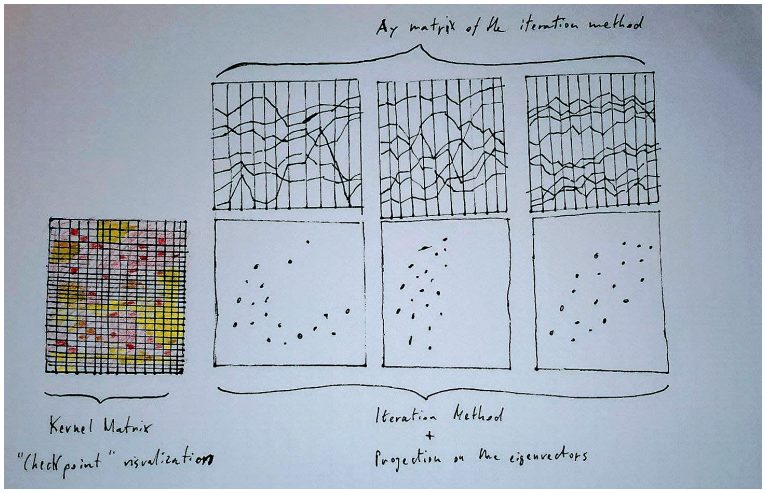


Figure 5: *Sketch of the visual storytelling of Kernel PCA*

After the definition of the visual design, we implemented the project. Working with the DRtoolbox made available by Laurens van der Maaten [51], all data processing was performed off-line using a commercial software package (*MATLAB ® R2016b*) whereas the visualizations were created by using *D3.js* [76].

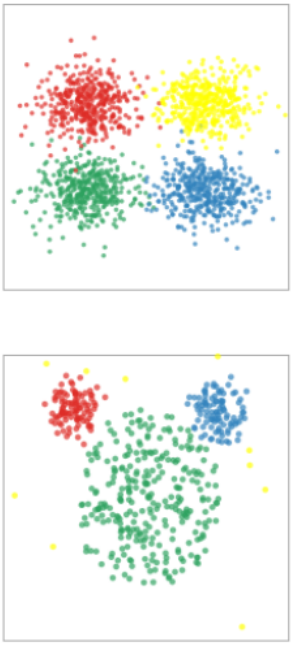
The Data Used

In this section we provide a description of the data used for getting our results. Most of them are data sets popular in the statistical word such as Swiss Roll data and data sets from Kaggle or UCI Machine Learning Repository [1, 3, 27]. They are divided in two subgroups, *Artificial Data* and *Real Data*. For all the data sets, we mention their size in the form NxM where N is the number of data points and M is the number of dimensions.

Artificial Data

First, we created two data sets and we referred to them as *Clustered data*, 1000x100 and *Non-Clustered data*, 1000x40. The first one is structured data separable in seven clusters whereas the latter one is simply a data matrix randomly generated from a Normal distribution with five classes assigned also randomly. Both of them have been generated by using the platform Dataset Generator (datgen) [31].

For the special cases, *from 3D to 2D* and *from 2D to 1D*, we employed the so-called *Mickey Mouse data*, 500x2, as the three clusters of data form shapes similar to the head and ears of the popular cartoon character. In addition, some noise was added too. Next, an artificial data matrix 1600x2 with four well-separated clusters was used in an example and we called it Clustered data - 2D [27].
Then, we created another classical DR example, the *Circle data*, 1671x2, composed of circle-shaped data with some points also in the middle of circle. The last three data sets described are shown in Fig. 6. Swiss Roll data is structured data embedded in a 3D space after a specific mapping. Finally, a data matrix was randomly generated from four different random distributions forming 4 clusters in a 3D space, *Clustered data - 3D*, 1250x3. In Appendix 1, in Fig. 26, there are the graphical representations of this data viewed from two different angles and of the Swiss Roll data.



Real Data

The real data structures differ from each other leading to interesting results to compare and discuss.
Ionosphere data, 351x34, regards classification of radar returns from the ionosphere [5]. *Churn data*, 5000x17, is a private data set and is about the churns in a telephone company. We also used Semeion Handwritten Digits data, 1593x256, which, unfortunately, it has no labels but an important characteristic [6]. Usually, these kind of data present numeric values from 0 and 1 but Semeion data were dichotomized at 0.5. Hence, it contains only 0 or 1 as value greater than 0.5 were considered as 1 and values less or equal to 0.5 as 0. Since some information is lost, we did not expect much in terms of DR results but it can be interesting to see how differently the discussed NLDR methods deal with this data. Moreover, we can explore data patterns by using brushing & linking technique although the data is unlabeled. Lastly, two examples are made by using Mushroom data, 3000x23, and Breast Cancer Wisconsin data, 569x32 [2, 4].

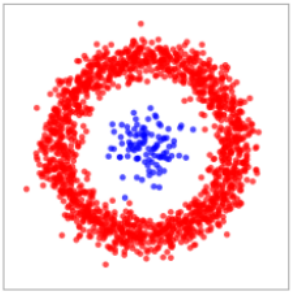


Figure 6: From the top: the *Clustered data - 2D*, *Mickey Mouse data* and *Circle data*

Results

"Vision is an intelligent form of thought"

Andreas Gursky

The results of the visualization process have been grouped per data and NLDR technique used to give a reading order and a logical thread to readers.
A visual storytelling approach was followed for comparing the SM, SNE and t-SNE actions by illustrating data in the lower-dimensional space at iteration 1, 10, 30, 100, 300 for all the data sets above-described. This was possible using scatter plots as we set that output spaces have to be in 2D. The iterations and number of frames were so selected to show how the lower-dimensional space is optimized starting from a random initialization by the gradient descent optimization algorithm. The chosen sequence does not grow linearly because the greatest efforts are made during the first iterations (from iteration 1 to 30) and the momentum (in SNE and t-SNE) is augmented at iteration 250. In this way, the evolution of output space is expressed more effectively. Besides, as most of the data were labeled, in some examples we show the issues of unsupervised clustering removing the labels and visually exploring the data by brushing & linking.

Unfortunately, humans are not able to perceive more than 3D in a space and that is also why we perform DR techniques. Hence, as the visualizations of original data spaces are not available for all the data sets used, we also worked with low-dimensional data allowing for direct comparisons between input and output spaces.
Furthermore, we made visual storytellings per technique (Appendix 2) to compare the results of performing NLDR methods to various types of data. In this manner, a deeper overview of the algorithm process of every single technique is also presented.

However, to visualize the techniques performed by eigenvalue decomposition (i.e. Isomap, kPCA and LLE), we had to slightly change our approach due to the different optimization not based on the gradient descent methods. As we introduced in Section: *Background Knowledge*, the nature of these algorithms is different. The power method does not provide the same progressive evolution of the output space. After a couple of iterations, an approximation of the optimal solution is already defined and a sequence of visualizations is meaningless here. Whereas it is much more effective to show how the low-dimensional space is optimized based on the objective function if a gradient descent method is performed.

Thus, we made a visual storytelling of the output spaces obtained by projecting the initial data on the eigenvector associated to the largest eigenvalue and on the eigenvectors associated to the second largest eigenvalue, to third largest one and so on. This is shown in Fig. 27, in Appendix 1. However, it is not that informative and the comparisons lose interpretation since the evolution of the story is not based on the algorithm process but from the different "points of view" of the data. Hence, we ended up comparing the final outputs of these methods applied on different data, including the low-dimensional ones.

Thanks to the variety of data and NLDR techniques, we could cross compare the final visualizations and draw conclusions on our findings. In the following sections, the readers will be guided through the presentation and discussion of results, starting from the visualizations per artificial data, per low-dimensional data and per real data. In this learning path, they will observe how NLDR methods work on synthetic data to understand the theoretical aspects of the algorithms and on real data to understand their behavior in more complex cases.

Visual Storytelling of SM, SNE and t-SNE

In the visualizations related to SM, SNE and t-SNE the algorithm initialization is random. Another option could have been performing PCA on the original data but we preferred to have a similar first frame for all the stories. It is also important to remark that the optimization of the SM's algorithm was performed by the *Newton method*. It is similar to the gradient descent but it converges faster as it involves also the use of Hessian matrix [30]. Furthermore, all the results have been obtained after running the algorithm several times and setting different combinations of hyperparameters.

Artificial Data

The first case analyzed is data without any pattern shown in Fig. 7. Firstly, we can observe that data at the final iteration (i.e. i = 300) are arranged similarly to data at first iteration, that is randomly, for all the techniques. This is because the data is unstructured. However, the shapes of the agglomeration of points are slightly different and this is due to their cost functions. For instance, the SM's embedding is circular with approximately uniform density which is typical from this technique as explained above. In the other frames, a "Celtic cross" shape can be seen which may be due to the critical points of the Hessian matrix involved in the optimization [30].

The NLDR techniques performance are different with data well-clustered. In Fig. 8, it can be seen that t-SNE correctly separated the seven clusters while SNE encountered the crowding problem: some extreme values force the mass of points to collapse into the center and it is not possible to identify patterns. In contrast, the SM roughly succeeded in identifying classes, although it failed to form clusters. This is due to its characteristic of focusing on local distances instead of global ones.

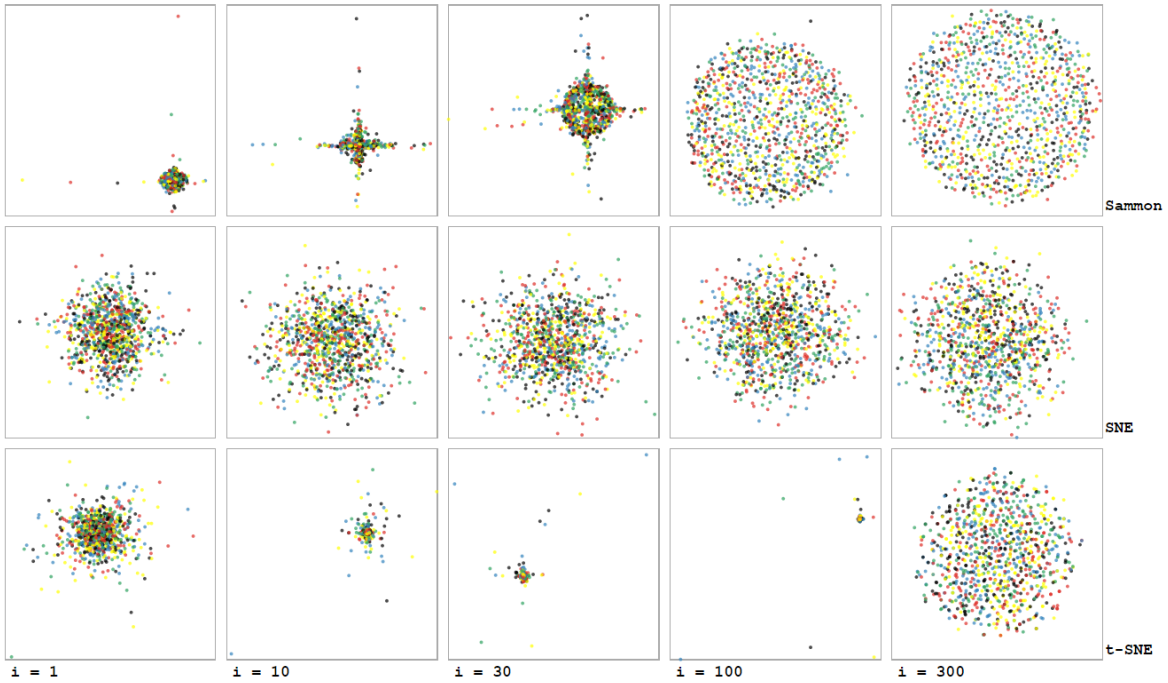


Figure 7: *Visual storytelling of the algorithm of SM, SNE and t-SNE (per row) at different iteration (per column) performed on Non-Clustered data*

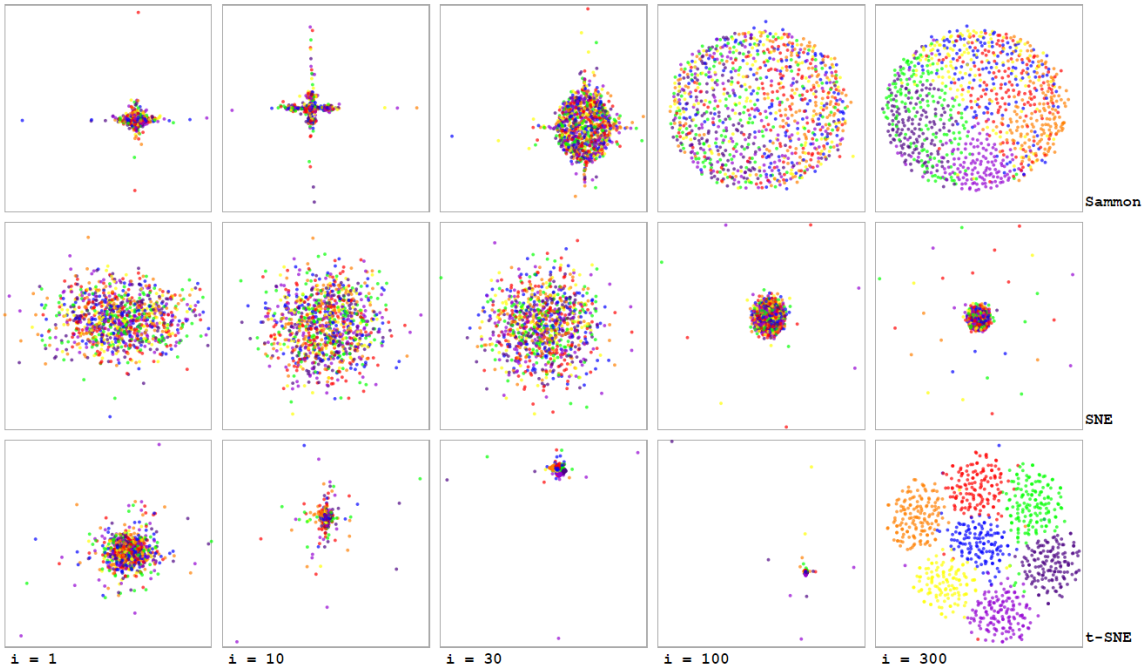


Figure 8: *Visual storytelling of the algorithm of SM, SNE and t-SNE (per row) at different iteration (per column) performed on Clustered data*

From 3D to 2D

In the previous toy examples, it was possible to get familiar with dimensionality reduction. By reducing low-dimensional spaces (3D or 2D) to even lower-dimensional spaces (2D or 1D), we can compare the solution embedding with the original data space in Fig. 26, in Appendix 1.

In Fig. 9, there is a visual storytelling related to Swiss Roll data. Even if some clusters are defined, both t-SNE and SNE did not manage to separate points belonging to different clusters (from i = 30 to i = 300). These algorithms consider the wrapped points as unique clusters since we assume them in an Euclidean space [21]. The boundaries between classes are less defined by SNE whereas they are well-delineated by t-SNE. This due to the different properties between the Gaussian and Student-t distribution. Any point more distant than 2-3 standard deviations from the mean value can be considered extreme value by using the Gaussian distribution. As a consequence, the multiple attractive forces between these points and the rest of points crush together the latter ones in the center of the map iteration after iteration [94]. Furthermore, it is noticeable that t-SNE converges faster than SNE in Fig. 9, from iteration 10 and 30. Although SM preserved the original data structure, it failed to unfold the data manifold.
A different scenario is illustrated in the visualizations in Fig. 10. Overall, all the techniques achieved in revealing patterns successfully. The SM did not reach the 300th iteration which means that not much progress was made and the algorithm converged. Unlike the last example, this data was not embedded in a complex nonlinear manifold as shown in Fig. 26 and the techniques performed better.

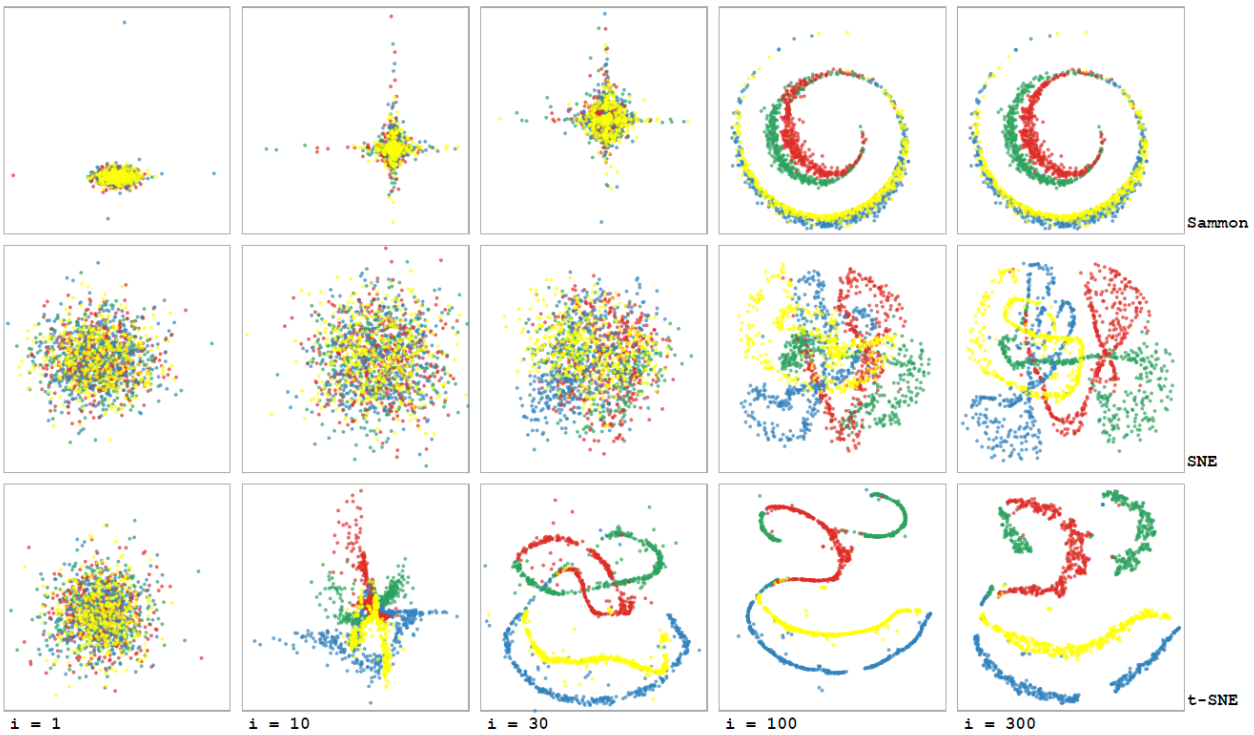


Figure 9: *Visual storytelling of the algorithm of SM, SNE and t-SNE (per row) at different iteration (per column) performed on Swiss Roll data*

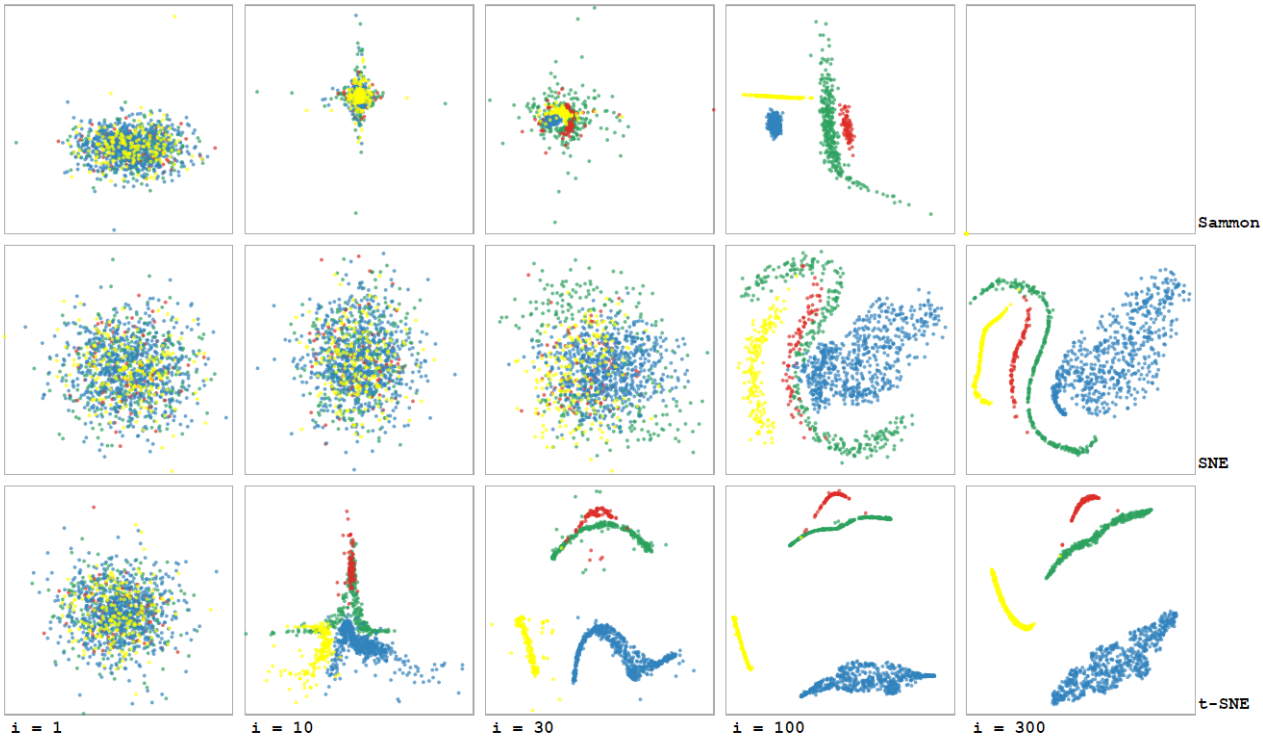


Figure 10: *Visual storytelling of the algorithm of SM, SNE and t-SNE (per row) at different iteration (per column) performed on Clustered data - 3D*

From 2D to 1D

Reducing a 2D space to a 1D space has also the advantage to be able to visualize the original input space and to compare it with the output space. We show the points on a space spanned by the 1D final embedding rotating it 45 °. Applying this jitter to the data, we make it more visible facilitating the interpretation of results. The stress is on the extremes of 1D spaces as we consider them more interesting. However, the final embedding structure was not subjected to any alteration and it is still in one dimension.

In Fig. 11, there is a visual storytelling related to Clustered data. The SM's algorithm stopped after some iterations most probably reaching a local optimum. This can be deduced as it optimized the distance between points belonging to two classes (i.e. blue and yellow) but it ignored the classification of points of the other two classes. Differently, the probabilistic DR methods progressively reached a (near-)optimal solution. In particular, this story shows how the gradient descent algorithm carries gradually the output space towards an optimal solution. As this example is simple, the path along the gradient can be imagined as approximately linear as the optimization problem is almost convex.

In Fig. 12, SM does not reach the maximum iteration but finding a solution earlier this time. Similar results for SNE and t-SNE. Thus, in conjunction with Fig. 13, we can observe that the SM is prone to find local optima in comparison to the other two techniques. This behavior is emphasized when the data structure is more complex such as Circle data's where the optimization problem is not close to be a convex.

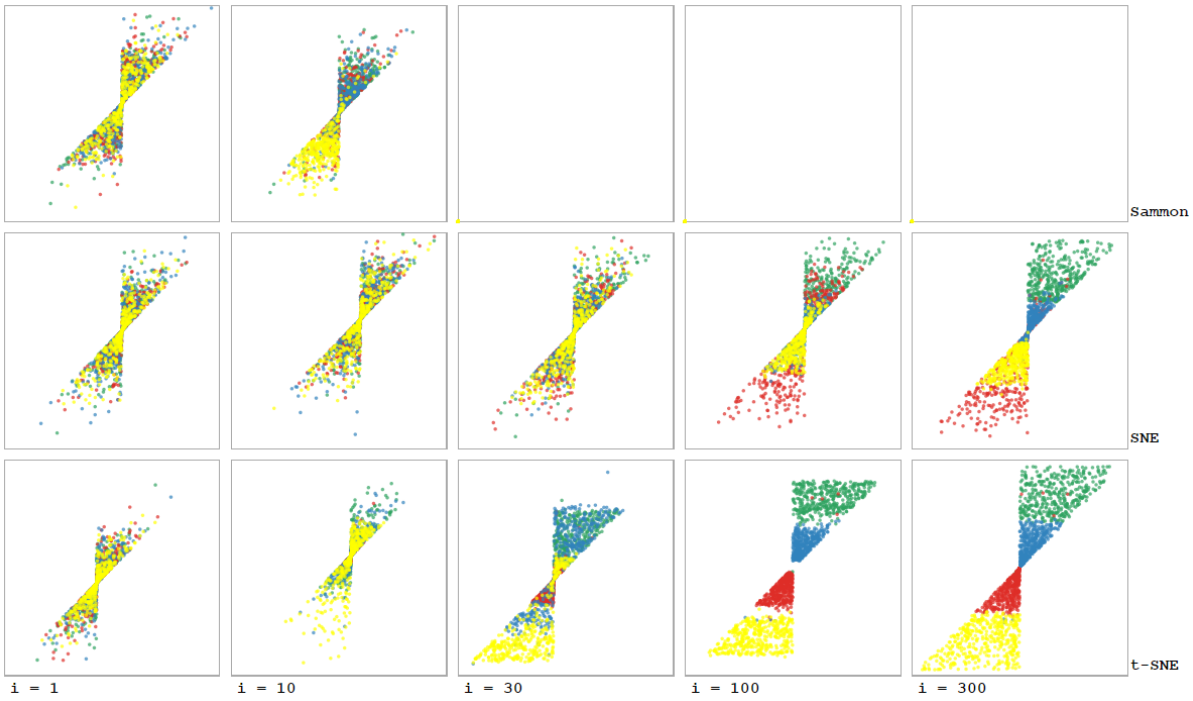


Figure 11: *Visual storytelling of the algorithm of SM, SNE and t-SNE (per row) at different iteration (per column) performed on Clustered data - 2D*

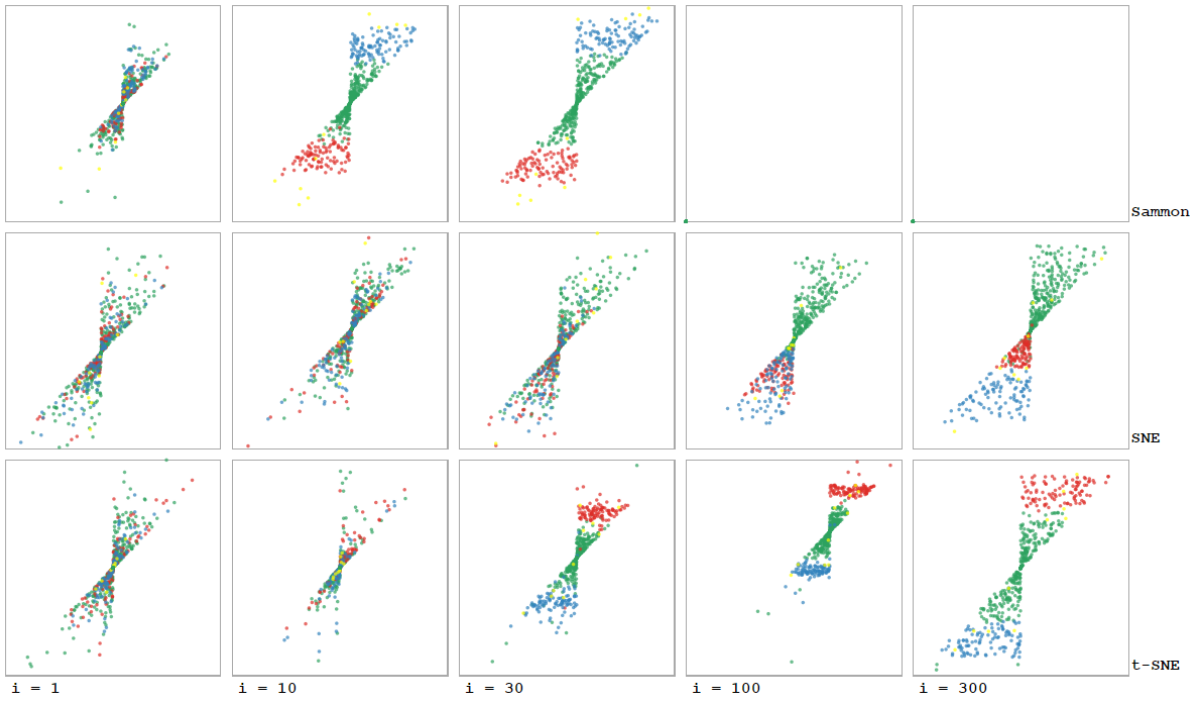


Figure 12: *Visual storytelling of the algorithm of SM, SNE and t-SNE (per row) at different iteration (per column) performed on Mickey Mouse data*

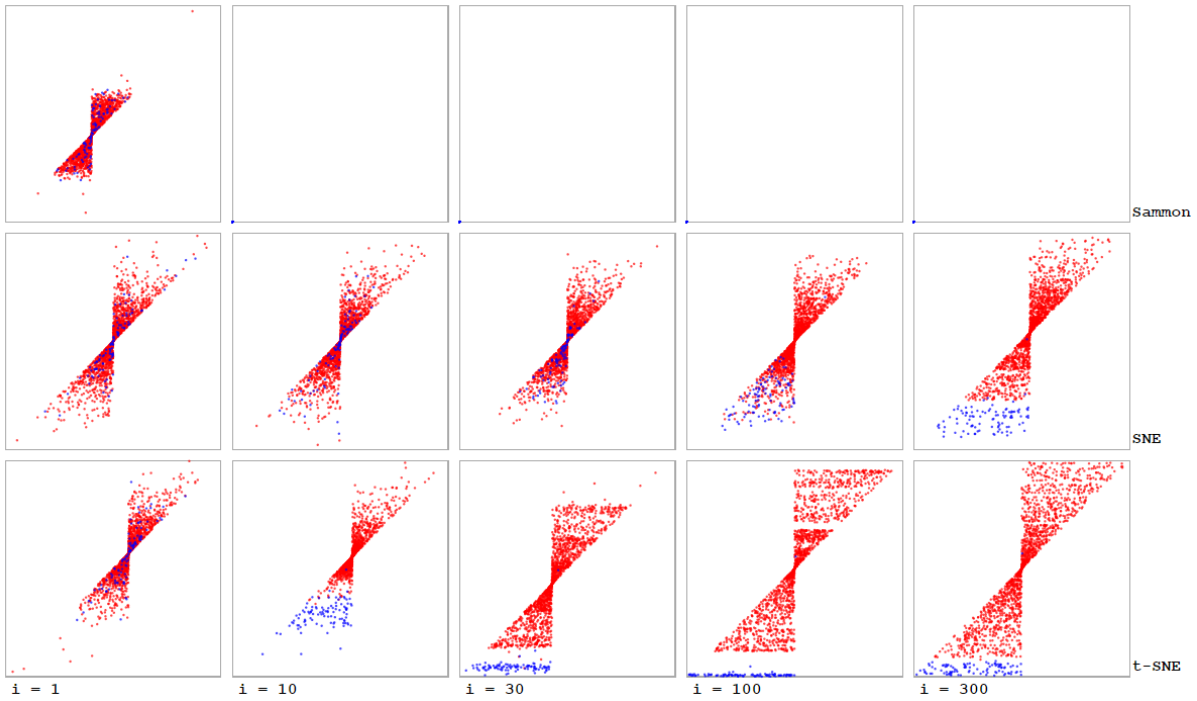


Figure 13: *Visual storytelling of the algorithm of SM, SNE and t-SNE (per row) at different iteration (per column) performed on Circle data*

Real Data

All what we have ascertained from the previous visualizations can now appear confusing and not justified as real data spaces can be much more complex than artificial ones. They are not controlled by us and we cannot prove that the final embedding is perfectly representing the original space, although labels are available. In addition, we are focusing on explaining how and why NLDR techniques work in a certain way without having any additional information by data analysis.

Starting from Fig. 14, we notice a different but related behavior of the three techniques. The most interesting pattern evolution is from t-SNE which show some clustering after only 30 iterations. Although all data lay in a small region, the data points are separated per class and a circular shape is formed hiding apparently a third dimension. This subtopic is discussed extensively in Section:*Remarks*. However, the violet points are divided in two different clusters so that one cluster is composed of violet as well as orange points. t-SNE split one class in two different clusters also in the Swiss Roll example and we will deepen this topic soon. A similar behavior is observed for SNE and SM even if the violet cluster is more compact there.
Recalling Fig. 9, similar features can be identified in both storyboards. For instance, the SM final embedding appears to be flattened in contrast to SNE and t-SNE results. The violet cluster forms a sort of spiral for SM and SNE while t-SNE splits it in two subgroups in Ionosphere data as also occurred in Swiss Roll data. Thus, we guess that Ionosphere data points initially laid on a highly nonlinear embedding and for the same reasons as before the solution is biased. However, this cannot be checked as the original data structure cannot be visualized.

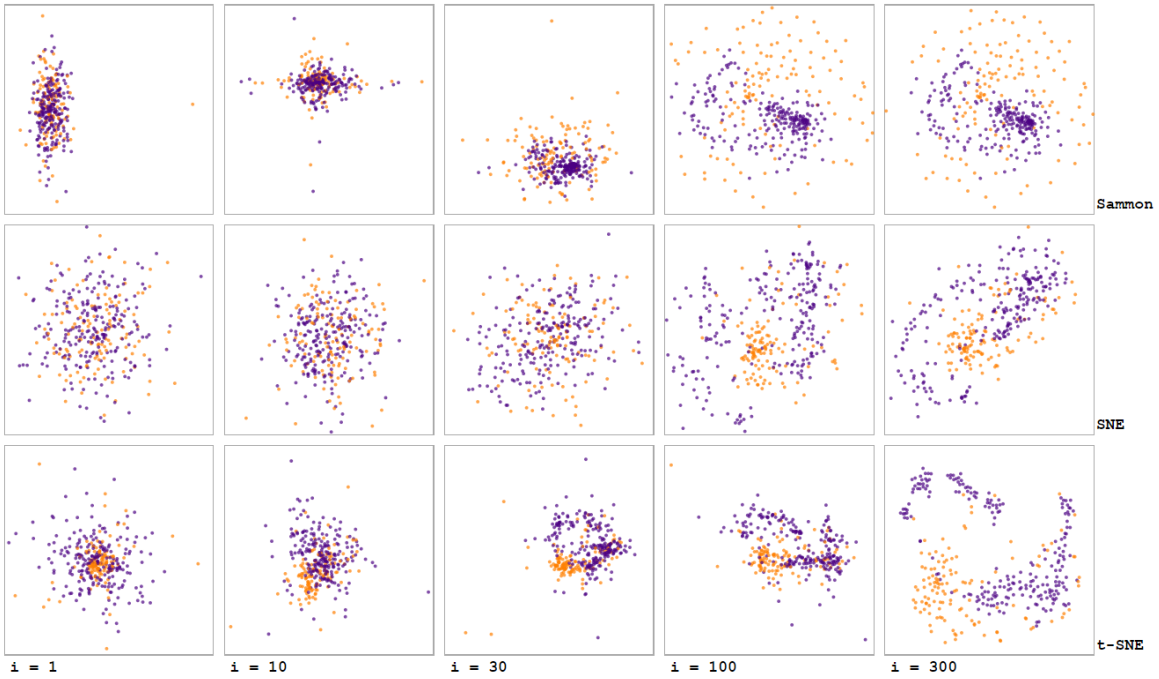


Figure 14: *Visual storytelling of the algorithm of SM, SNE and t-SNE (per row) at different iteration (per column) performed on Ionosphere data*

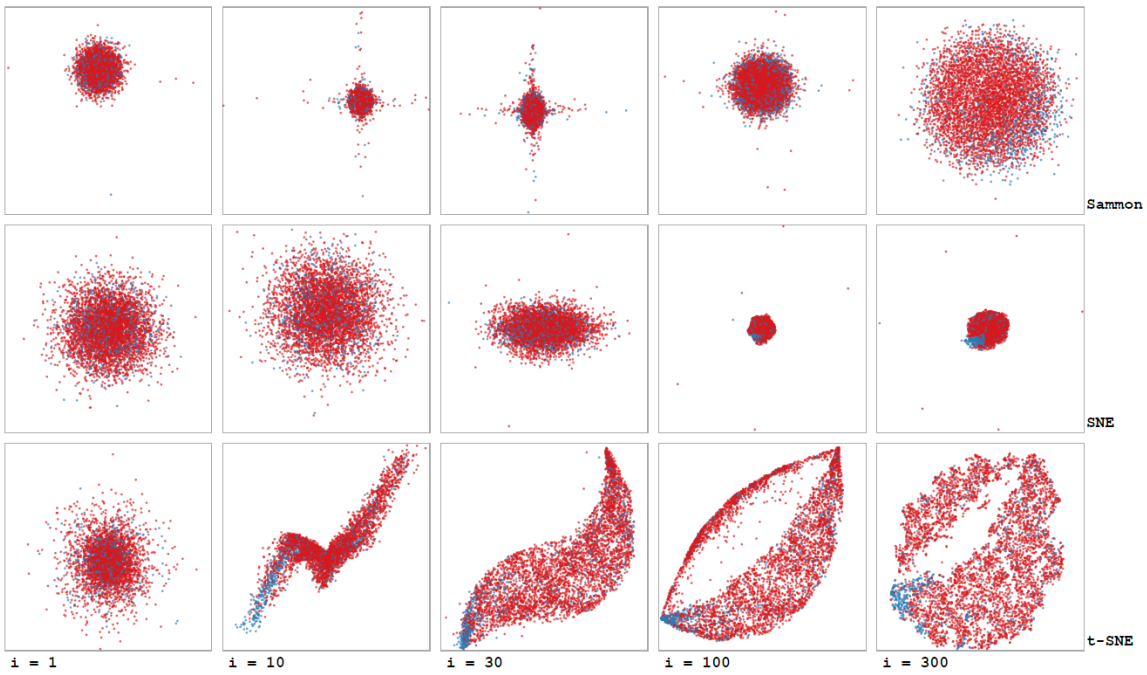


Figure 15: *Visual storytelling of the algorithm of SM, SNE and t-SNE (per row) at different iteration (per column) performed on Churn data*

An analogous scenario is shown in the Churn data example illustrated in Fig. 15. In the last two frames, the lower-dimensional space seems to be in 3D instead of 2D as if there is an intrinsic dimension on the 2D final embedding. This can mean that the correct number of dimensions needed to preserve global and local properties of the high-dimensional manifold is not 2 but 3 [107]. In addition, the imbalance of data does not help to clarify much as only a small group of point has a different label. Similarly to the previous example, the original data structure cannot be visualized and we cannot verify our findings.

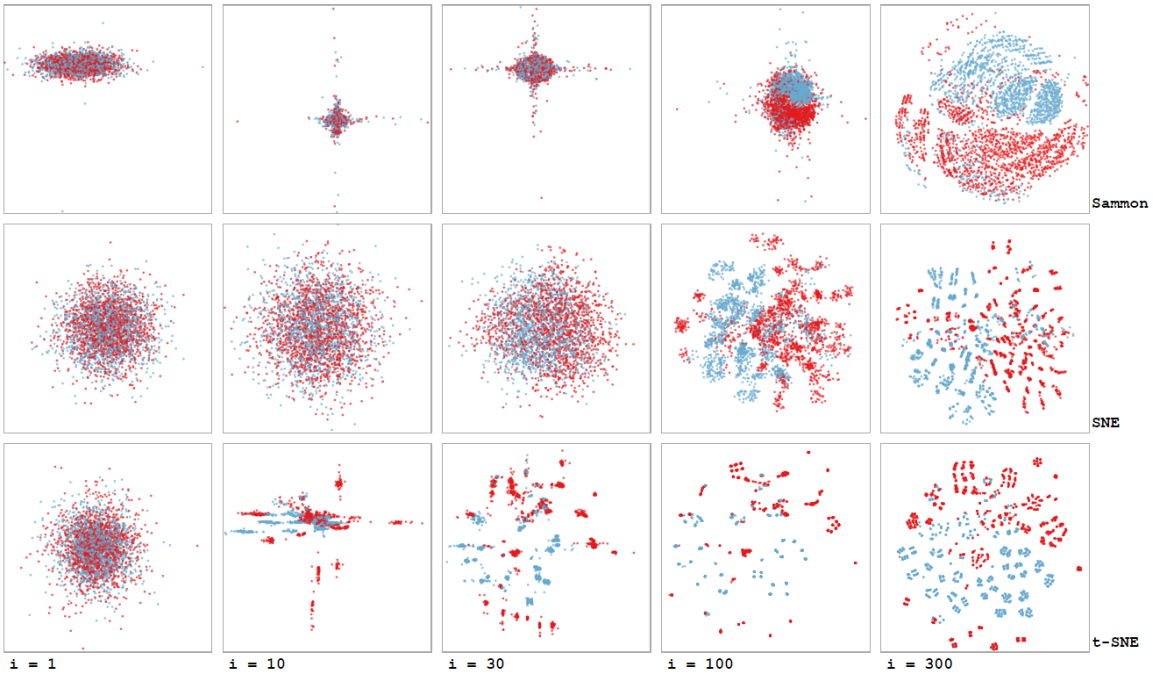


Figure 16: *Visual storytelling of the algorithm of SM, SNE and t-SNE (per row) at different iteration (per column) performed on Mushroom data*

This is not the only issue when interpreting the final embeddings of NLDR techniques. It is also common to observe an unexpected outcome for shape, structure or bad matching with labels as illustrated by storytelling related to Mushroom data in Fig. 16. A multitude of clusters is found by SNE and t-SNE. Even if they located most of the red and blue points in two different regions, there are not only two main clusters in the map. In case of unsupervised clustering (i.e. without information about classes) on this data, it would be challenging to assigns points to the classes.
Thus, in Fig. 17, we did not use labels to color points in the map differently rather we highlighted a group of clusters which are relatively close to each other. The result is a wrong classification as shown by the labels in Fig. 18. Although the selected clusters are located nearby on the map, they belong to different classes. This solution is misleading as there is no correspondence between the right solution indicated by the labels (i.e. two classes) and the clustering solution (i.e. dozens of classes). However, the latter solution may not be wrong neither.

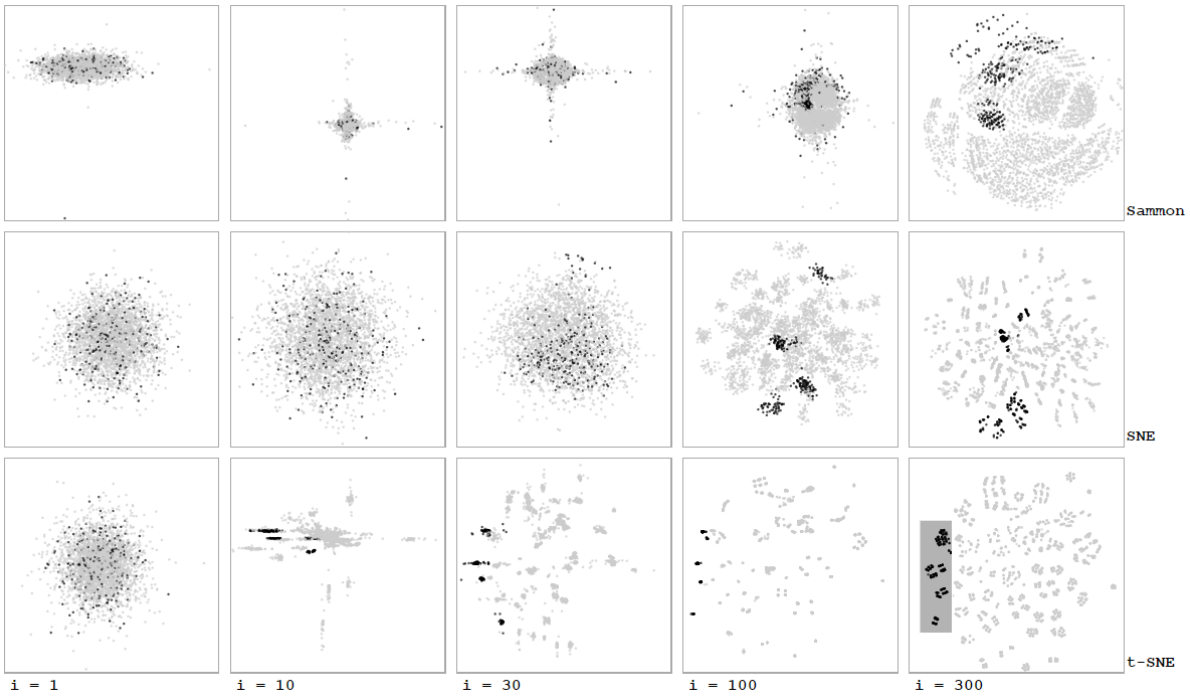


Figure 17: *Visual storytelling of the algorithm of SM, SNE and t-SNE (per row) performed on Mushroom data with highlighted regions and without labels*

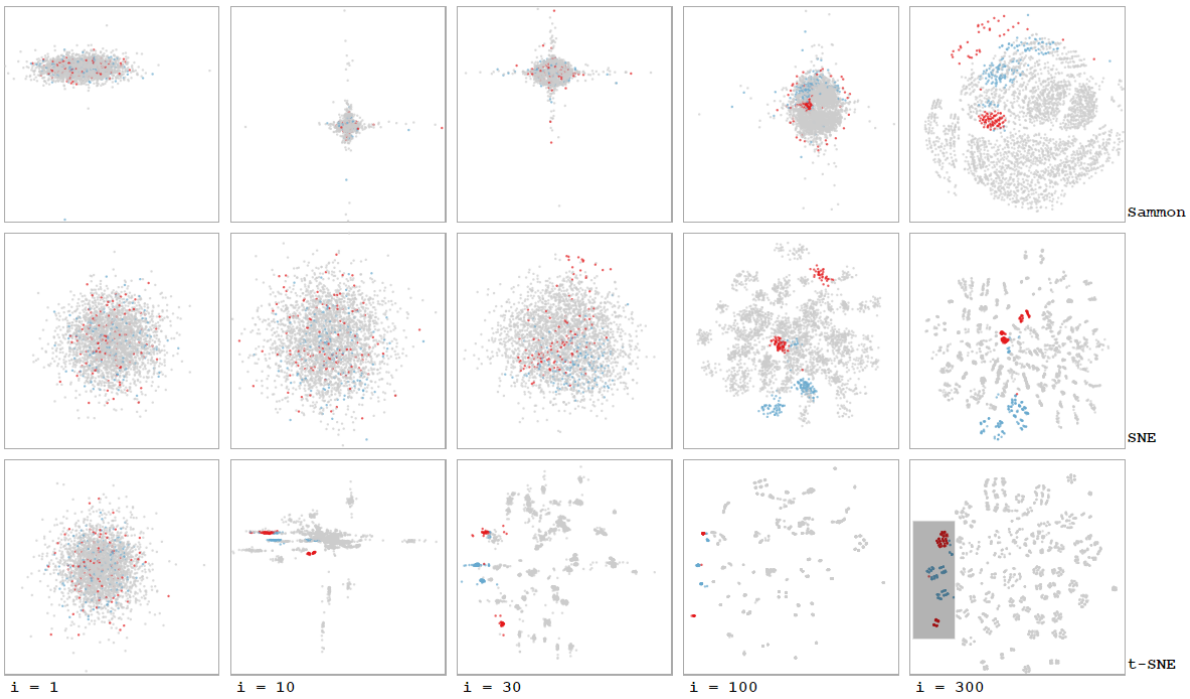


Figure 18: *Visual storytelling of the algorithm of SM, SNE and t-SNE (per row) performed on Mushroom data with highlighted regions*

Looking at the description of the original data, it can be seen that it consists of 23 different mushrooms species which should be classified by edibility.
Thus, based on data features, the probabilistic DR techniques may have identified the mushroom species which are also data classes. The algorithms represent all data information as an Euclidean distance matrix and, based on it, they define the probability that each point is a neighbor of another point for all of them.

Therefore, it is possible that they find other patterns in the data, if present, as well. The NLDR methods should be considered as sorts of metal detectors where the metal is the pattern structure. They reveal all the patterns and not only the target pattern defined by labels. Hence, it is useful to know well the structure and description of data as well as NLDR algorithm operations before performing DR to avoid misleading interpretation of the results.

A completely different scenario occurs in the Breast Cancer data example illustrated in Fig. 19. Most of the time, the cluster shapes and the location of points on the map are less meaningful in t-SNE since it focuses to define and visualize clusters as clearly as possible. Differently, SM can reveal how the local relationships among points are. As a consequence, the green cluster is compacted since the first iterations whereas the orange one is more spread. We can deduce that the two clusters are well defined in the original data space but the green one lays in a smaller region than the other one.

The last example presented is about Semeion data. It differs from the classical handwritten digits data for the dichotomization which caused a loss of information as shown in Fig. 20. Thus, the final solution of t-SNE shows only half of the expected clusters (i.e. 10 digits) clearly. The isolated group of points (e.g. on the top right side or on the left side) are supposed to be the well-written digits while in the center there should be mistakable digits.

Since we do not have information on the classes for this data set, we highlighted one potential cluster in the map center as illustrated in Fig. 21. The differences between evolution of that cluster during the optimization process of SM, SNE and t-SNE is due to the optimization algorithm characteristics outlined previously.

Furthermore, as the SM's algorithm suffers the high-dimensionality of original data, its results are similar to those shown for Clustered data case in Fig. 8, due to the Euclidean distance employment. Whereas SNE is less efficient than t-SNE in defining clusters due to the lighter tails of the Gaussian distribution.

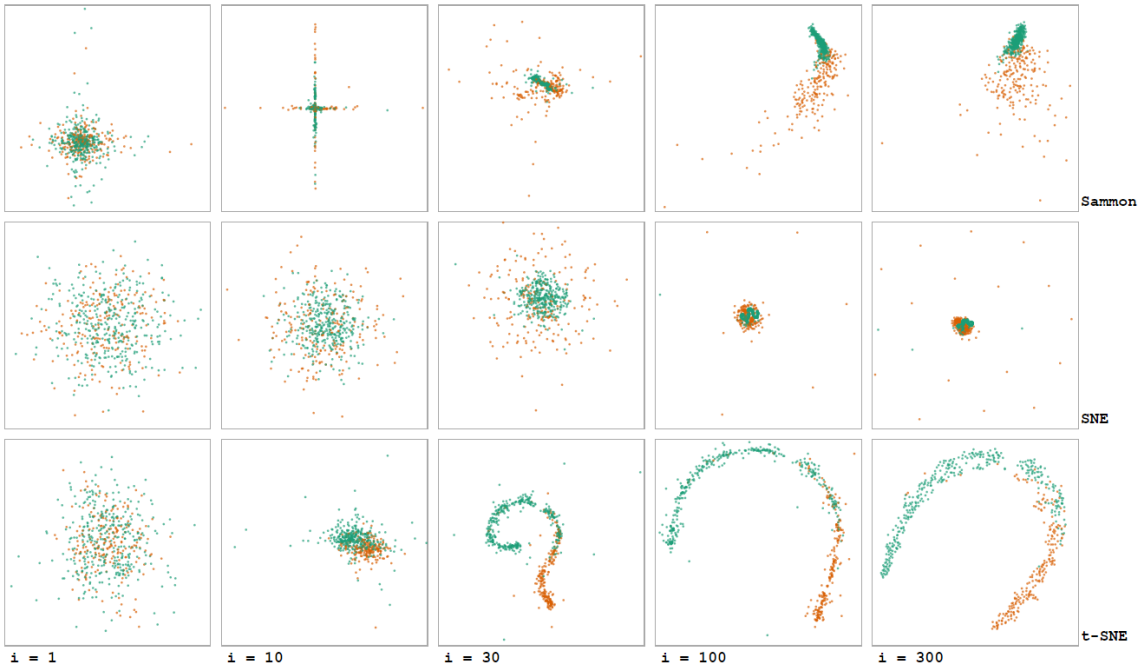


Figure 19: *Visual storytelling of the algorithm of SM, SNE and t-SNE (per row) at different iteration (per column) performed on Breast Cancer data*

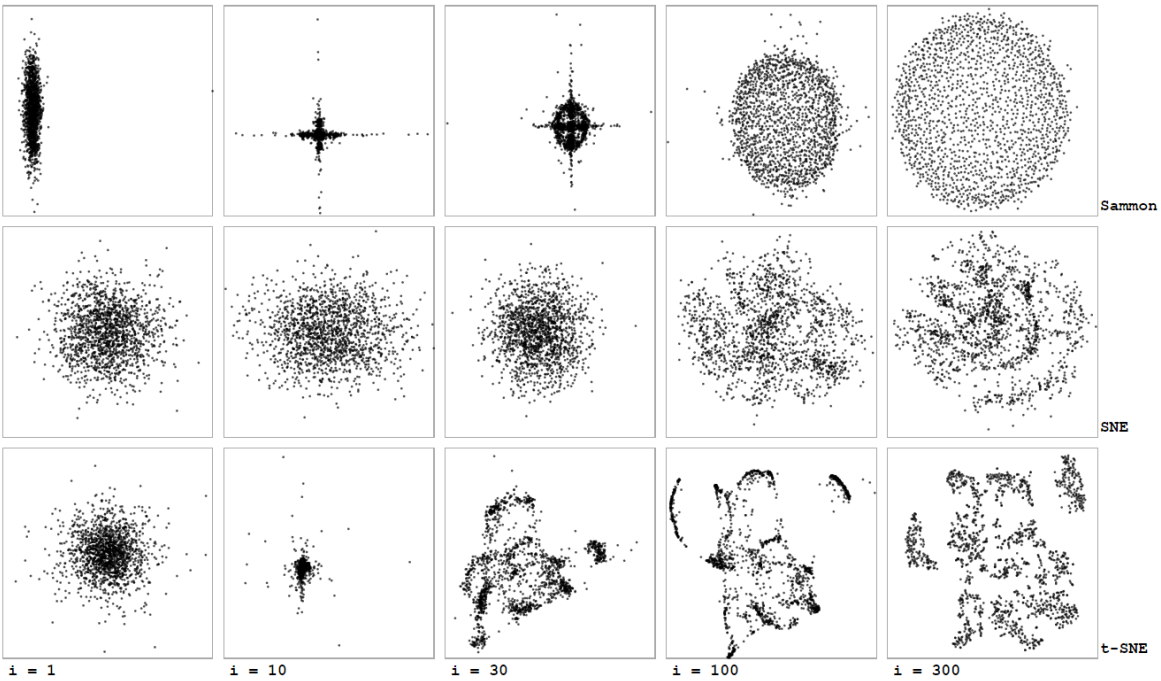


Figure 20: *Visual storytelling of the algorithm of SM, SNE and t-SNE (per row) at different iteration (per column) performed on Semeion data*

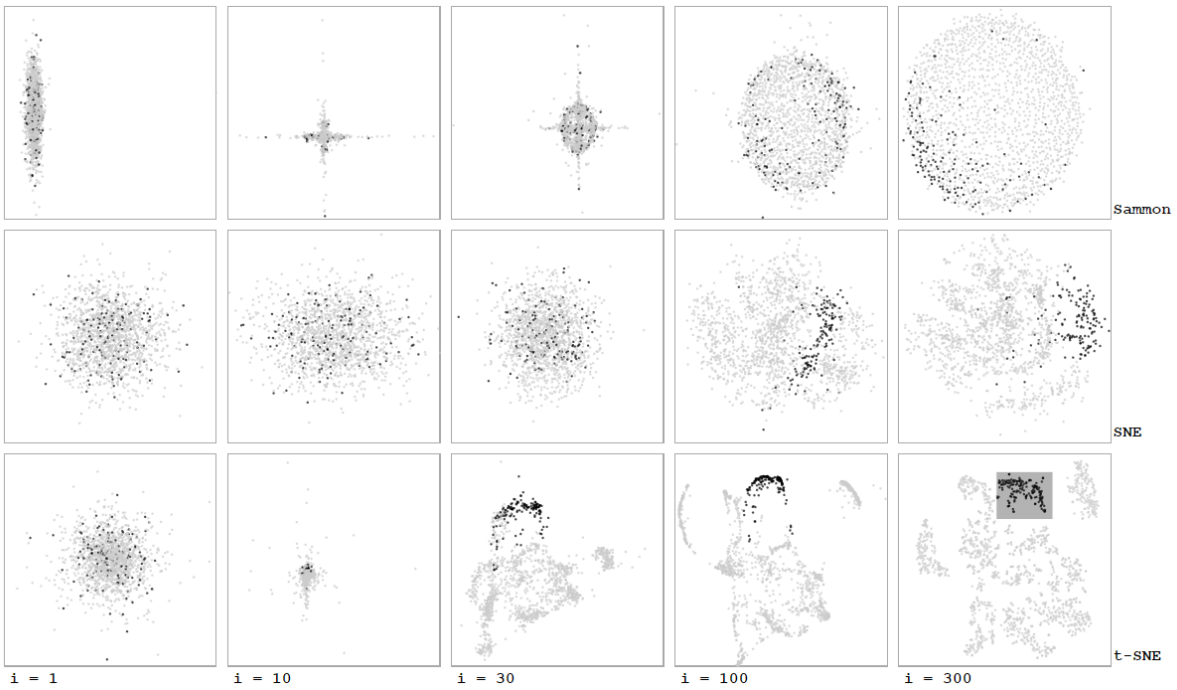


Figure 21: *Visual storytelling of the algorithm of SM, SNE and t-SNE (per row) at different iteration (per column) performed on Semeion data with highlighted regions*

Visual Storytelling of kPCA, LLE and Isomap

Although it was not possible to show the optimization algorithm to create final embeddings as clearly as we did for the previous techniques, we present three visual storytellings of the kPCA, LLE and Isomap's algorithms performed by eigenvalue decomposition on artificial data, 2D artificial data and real data, illustrated in Fig. 22, 23 and 24, respectively.

As we already mentioned, the dimensionality reduction processes based on eigenvalue decomposition are more difficult to represent graphically in comparison to those based on the gradient descent method. To explain why, we can compare these two procedures to the approach of two different schools of art to make a work. In both of them, the procedure which leads to reproduce (to perform dimensionality reduction) a desired object (the output space) is performed by observing carefully the real object (the input data space) and measuring its proportions, sizes and details (the relationships between data points). At this stage, every artist (any existing DR technique) works in a different way following his own ideas on the relevance of object physiognomy (the local and/or global geometry) and using their preferred tools (the dissimilarity measures). In this way detailed notes (the dissimilarity distances) on the object features are made.

The artists of the "Optimization" school are sculptors and, afterwards, they start assembling randomly some material (the initialization of the output space) and modeling it day by day (the iterative process) so that it would bear a good resemblance to the real object (minimizing the cost function) until they were stopped (the maximum iteration) or satisfied (the local/global optima) of their work.
In contrast, the artists belonging to the school of "Decomposition" are specialized in visual effects (the projections). They make use of their notes to choose the best spot (the matrix to factorize) and its best framing (the eigenvectors with highest eigenvalues) where they generate shadows and holograms (the projections of the output space) of the real object.
The work process in the two schools are conceptually different and not meaningfully comparable. However, their final works and the quality of them are. On this principle, we made the following visualizations comparing them successively.

Finally, it is also important to remark that the choice of hyperparameters related to these NLDR techniques affects more the final results and several attempts have been done to obtain meaningful graphical representations. Generally, we used a Gaussian Kernel function ($\mu = 0$ and $\sigma = 1$) when possible (otherwise a linear Kernel function was used) and we set the number of neighbors k equal to 12.

Artificial Data

In the artificial data examples, we can observe how NLDR methods of interest work. Looking at Fig. 22, it is clear that these techniques are more difficult to interpret and robustness in parameter setting changes.

In the column on the right, there is shown Swiss Roll data which is a classical example to prove the properties of Isomap and LLE but our results are not satisfying. LLE partially succeeded in unfolding the data while Isomap obtained a poor embedding similar to the SM's. This can be due to bad initial settings. kPCA output shows also a bad choice of the Kernel function as it indicates no variation among blue and yellow points.

In the structured and unstructured data, namely with Clustered data and Non-Clustered data, they did not perform greatly neither. However, a typical shape by LLE is observable in the Non-Clustered data example. It is evident that the constraint of unit variance is too simple to preserve the global structure of data. This is a major disadvantage as most of the data collapse in small region of the map worsening its readability. Moreover, for the well-clustered data, we used a linear kPCA (i.e. PCA) but the embedding is nonlinear. As a result, the generated clusters overlapped. In another way, Isomap identified some clusters but it arranged them in a confused way on the map. Oppositely to LLE, usually it succeeds to preserve the global geometry of the input space while it fails in maintaining its local properties [82]. Both these last NLDR methods are not based on the probability distributions of data points and, for this reason, they are not able to neglect how data points are distributed in the original space focusing more either on local geometry (e.g. LLE) or global geometry (e.g. Isomap).

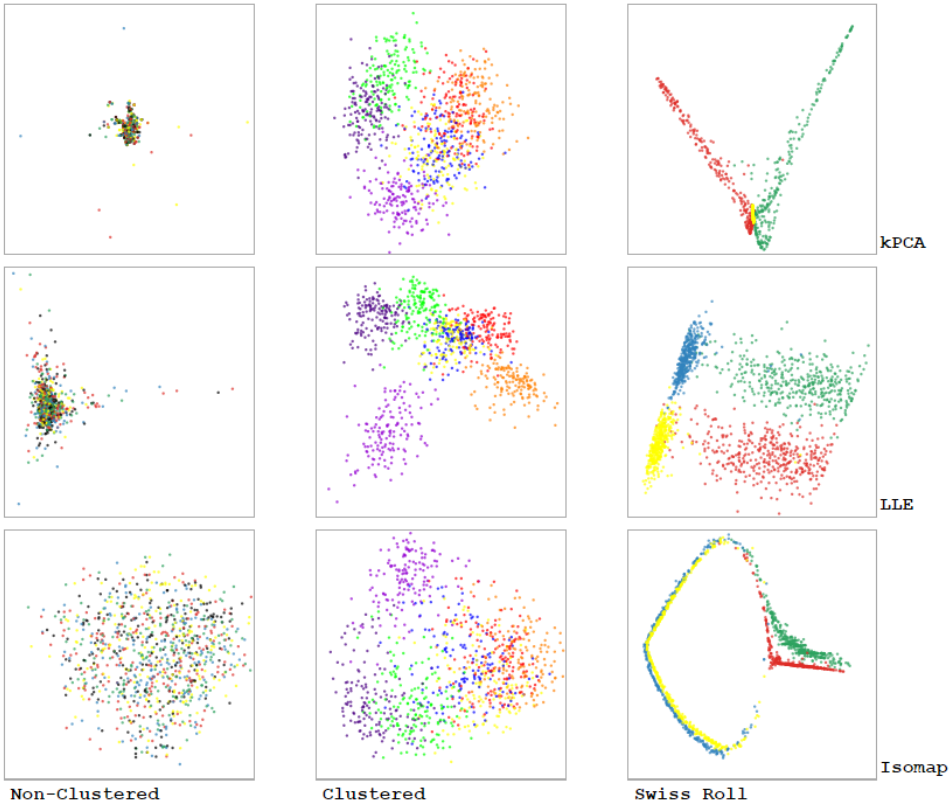


Figure 22: *Visual storytelling of the algorithm of kPCA, LLE and Isomap (per row) performed on different artificial data (per column)*

From 2D to 1D

Referring to one-dimensional final outputs, the comparison between the current NLDR techniques is simplified. From Fig. 23, there are illustrated the final embeddings of kPCA, LLE and Isomap performed to 2D artificial data.

None of them seems to be efficient and practical to use, except in reducing Mickey Mouse data. For instance, LLE did not succeed in separating the blue and the green clusters as the k-NN search was based on the default Euclidean distances and not on more sophisticated and appropriate rules (e.g. choosing all points within a ball of fixed radius) [79].

Nevertheless, Isomap succeeded in it even if k-NN search was also set with Euclidean distances. This can be explained by the fact that LLE kept the red cluster closer to the blue cluster rather than to the bottom part of the green one by the reconstruction local weights. In contrast, computing the shortest paths of all data points, Isomap evaluated the three different regions as globally far and distinct.

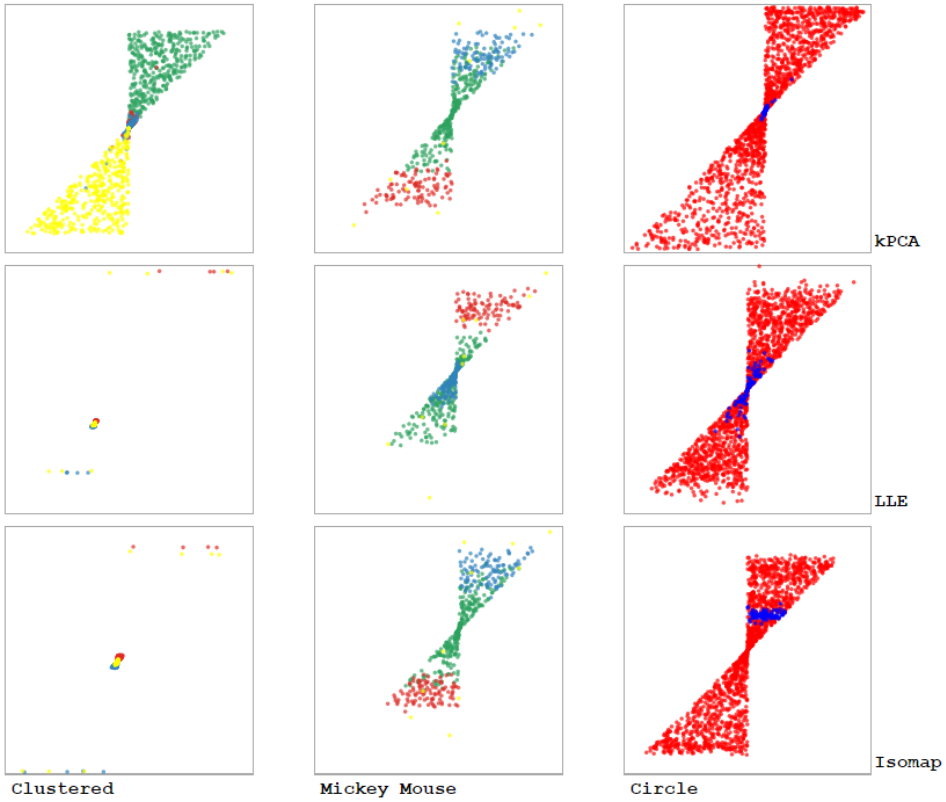


Figure 23: *Visual storytelling of the algorithm of kPCA, LLE and Isomap (per row) performed on different artificial 2D data (per column)*

Real Data

The next examples offer an overview of how kPCA, LLE and Isomap perform on real data. As before, the identification of clusters is not straightforward. In Fig. 24, there are illustrated all the final embeddings and we can make a cross comparison.

Clearly, Breast Cancer data structure is well-represented also by this set of techniques. As we used a linear Kernel PCA, we deduce that the initial embedding is almost linear. However, the methods did not perform well with Semeion data since we expected some clusters representing the 10 digits but there are no clear patterns in the final maps. This is due to the dichotomization and to the high-dimensionality of data, especially. According to previous studies, the performance and quality of the k-NN algorithm are questionable [41] and it is a crucial step for LLE and Isomap.

From the Churn data example, it can be observed that in kPCA final embedding the points are randomly distributed and that most of the points belonging to the 'churn' class (i.e. the blue points) is under one of Gaussian distribution tails. This also makes sense as Churn data points are people and they can easily be normally distributed based on call habits with churns identified as extreme values. However, it did not classify the data whereas it was done by Isomap and also by t-SNE.

In the Mushroom data example, Isomap made good efforts in grouping data correctly. Whereas the previous set of techniques found several clusters, Isomap's final embedding presents well-grouped points, although they are still laying on a nonlinear manifold. A good explanation can be the high nonlinearity of the initial embedding and the use of geodesic distance and shortest path as criteria for constructing the lower-dimensional space by Isomap [53].

In contrast, Kernel PCA and LLE's outputs are not interpretable as all the points collapsed. As explained above, this event is common and in the plot related to Churn data there is another display of it. Finally, the results for Ionosphere data express the complexity in reconstructing the original data space due to its topology.

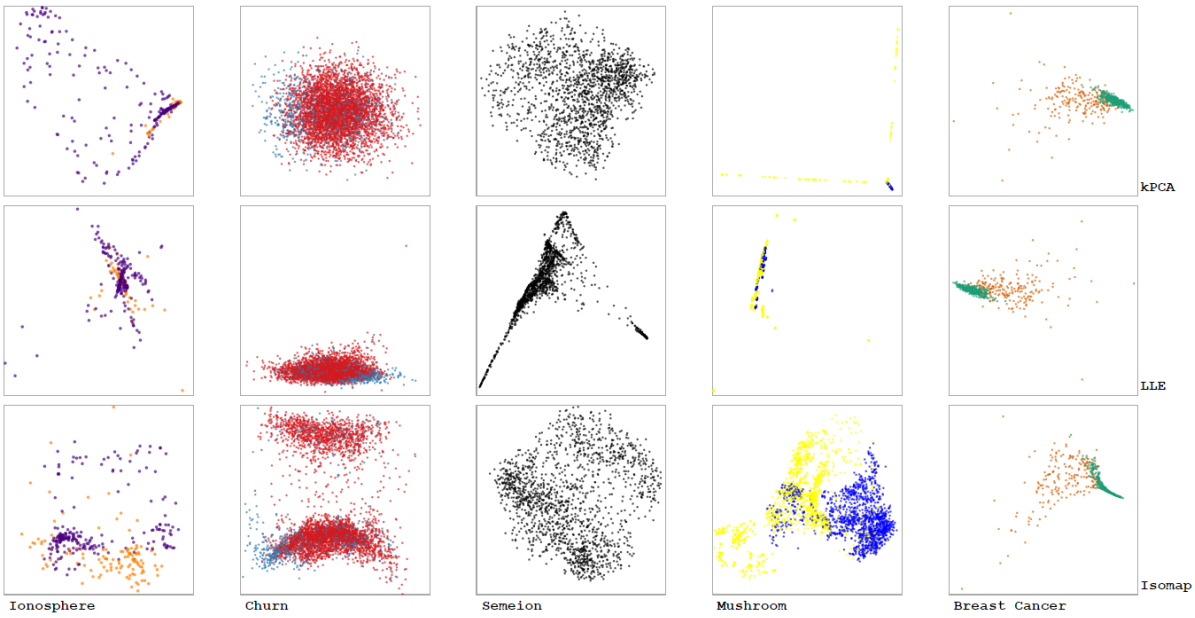


Figure 24: *Visual storytelling of the algorithm of kPCA, LLE and Isomap (per row) performed on different real data (per column)*

Remarks

The interpretation of (un)supervised clustering by using NLDR techniques is still troublesome, although the constant progresses to improve the existing methods [23]. Analyzing the t-SNE final embeddings in the Ionosphere and Churn data examples illustrated in Fig. 33, we should not consider the cluster shapes relevant according to other works [99]. Therefore, we should neglect the evident curvilinear shapes of data and consider the data separation obtained, even if a third (intrinsic) dimension seems to be embedded. In addition, looking at Isomap output space in Fig. 24, points of both data are well-separated in two clusters confirming this interpretation.

Setting the same parameters, we performed t-SNE on the same data again, except of the number of output dimensions which now is set to 3 instead of 2. The results are shown in Fig. 28 and 29 in Appendix 1. Ionosphere data shape is highly curvilinear and slightly similar to a Möbius strip while Churn data shows a sinusoidal wrapped shape. From these plots, it is evident that the 2D and 3D final embeddings are similar for both cases. This means that no much information is lost reducing the original space to 2D instead of 3D. But there is an intrinsic dimension represented by the red line which was not detected by t-SNE, as shown in Fig. 25. The points A and B seem to be close to each other but, along the manifold where they really lay, they are distant. This interpretation is supported also by the fact that most yellow points are grouped in one extreme of this surface.

It would be obvious to show the results of Isomap performed on this already reduced data, however, it would not be much useful.
According to a previous studies [12, 88], an Isomap drawback is its inability in unfolding manifolds with holes (i.e. "short-circuit errors") and this explains all the picture.
In Fig. 24, Isomap separated points in that manner because it cannot do imputation basically. This means that if some points are missing due to data availability, for instance, Isomap is not able to take into account it in the computation of final embedding attributing it to the data structure. As a result, only the manifold surface covered by data points is embedded and not the entire manifold itself.

This shows that real data world are various and none of the discussed techniques performs best in all cases examined. Although we showed that the probabilistic approach can be considered the best solution for clustering, they are less efficient in unfolding curvilinear embeddings properly. Differently, Isomap focus is on that aspect since it uses geodesic distances which read the curvatures of data structure.

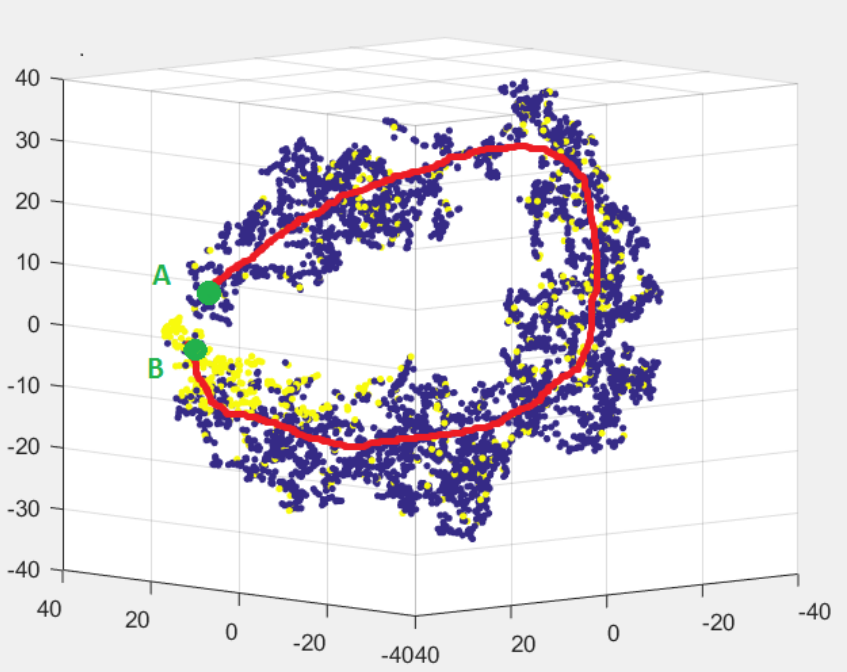


Figure 25: *The 3D final embedding by using t-SNE on Churn data. The red line indicates the intrinsic dimension*

Combining different techniques and approaches can also be useful as shown [52, 105], both sequentially as it could be possible with Churn data and in parallel as it could be possible with Mushroom data. The visualizations related to the latter data and illustrated by t-SNE revealed many interesting (unexpected) clusters failing in separating correctly labeled points. Performing Isomap in parallel made the understanding of Mushroom data structure clearer: it presents a highly nonlinear (i.e. curvilinear) nested structure, as the data points can be split in two clusters (global structure) which can also be separated in many other sub-clusters (local structure).

Finally, in the Breast Cancer data example in Fig. 19, the lower-dimensional space could have been reduced in one dimension. The curvilinear manifold of data points composed of two elongated clusters is one-dimensional and a second dimension to represent the original data structure is redundant.
Analogously to previous statements, we did not test this assumption and we cannot check it by looking at the original data space as it is high-dimensional.

Discussion

In this work, we compared six popular NLDR methods by visualizing their algorithms and final outputs. Using multiple visual storytellings embedded in a broader storytelling which is this entire paper, we attempted to give more insights on how to use these techniques.

Therefore, discussing the results obtained we aimed to involve the readers in interpreting the visualizations by themselves. The interpretation complexity can be substantially reduced by associating a basic theoretical knowledge of the algorithms to the graphical representation elements which enables to go from the visualizations to the theory and the other way around as well [36].

Unfortunately, some of the algorithms did not include an iterative method for finding the best embedding sufficiently long and we had to change some storytelling characteristics to make it efficient and expressive. However, it was not possible to compare how the optimal final embedding was reached for all the techniques.

The order in which we presented the results was decided on purpose. We gradually introduce our audience into the learning process through toy examples. Initially, just understanding the basic concepts of the dimensionality reduction process is not obvious. The low-dimensional examples aimed to facilitate a direct comparison between input and output spaces and to have an overview of the dimensionality reduction process easily.
Alternatively, the real data cases illustrated the issues in working with highly nonlinear data spaces. In this way, our willing was to encourage readers to move from concrete examples to abstract thinking developing a critical reasoning about the topic [71].

In the previous chapter, we discussed about t-SNE in particular concluding that unambiguous output interpretations are not possible in DR applications in general as it is remarked in the next section.
In fact, kPCA is theoretically powerful but is unpractical since it requires to know the Kernel function a priori. LLE has good properties which can be useful if combined with the right choice of distance metrics and parameter setting as we outlined in Fig. 23. However, the unit variance constraint is defective.

As the purpose of this thesis is to explain concepts through examples and visualizations, we did not perform dimensionality reduction (PCA, for instance) prior to the k-NN algorithm. The Isomap and LLE's algorithms involve a k-NN search which is affected by the curse of dimensionality and its application quality is not reliable [58].

Although SM achieves to represent some original data space properties efficiently, it generates circular maps without well-defined clusters. SNE is rough in practice as the frequent crowding problem annihilates the readability of its graphical outputs.

Besides, we stressed the importance of the dissimilarity measure choice since it characterizes the NLDR methods significantly affecting the results. For instance, the KL divergence based on (conditional) probabilities is more robust against high-dimensionality issues while the shortest path can identify and unfold curvilinear surfaces.

The choice of how to obtain the final embedding is also relevant but not always possible. Although the optimization methods facilitate the visualization of the NLDR algorithms, they present some issues. In fact, depending on the optimization complexity (i.e. on the cost function) it is common to find a local optimum or that the algorithm does not converge in some cases. As a consequence, every algorithm run leads to different results, especially if the initialization is randomly executed [64], and this can be confusing. On the other hand, SVD can be computationally expensive in case of large data sets and not possible when only the distance matrix is available. However, they are equivalent in certain cases as explained previously.

Furthermore, in this project we made use of labels to interpret results assuming that the data features are relevant for classifying data. Sometimes is possible that there are underlying latent variables which are ascertainable with certainty only by data analysis [56]. Finally, we showed how difficult is to perform unsupervised clustering on real data. Even if visual storytellings are provided with features such as brushing & linking technique, it is not straightforward to identify correctly the data structure, especially by using certain NLDR methods or in case of a high number of data features.

Conclusions

Preprocessing high-dimensional data is complex and requires a good understanding of the existing DR techniques. As dimensionality reduction of high-dimensional data spaces is a strongly abstract process, we made use of visual storytellings to provide an introduction of this topic. The main purpose was to visualize the algorithm of some NLDR techniques, namely kPCA, LLE, Isomap, SM, SNE and t-SNE and to guide a general audience through a learning path provided with conceptual explanations, practical examples, graphical representation and cross comparisons.

Using artificial and real data of different types, we showed various scenarios in which some features of each technique were highlighted. Some toy examples were illustrated of which some included performing dimensionality reduction from a 3D to a 2D space and from a 2D to 1D space. In this way, we attempted to reduce the degree of mathematical experience needed for the assimilation of these abstract concepts.

Moreover, we compared the unsupervised and supervised clustering. It is evident that, although a lot of effort has been done until now, there is still room for improvements in this topic, especially regarding unsupervised clustering. The final embedding interpretation of these cases is an issue in dimensionality reduction due to a complete blind search in highly complex (usually nonlinear) spaces and none of the discussed techniques can be considered universally efficient.

However, t-SNE is the most sophisticated existing technique as it identifies clusters efficiently in many types of data. Isomap is also useful because it compensates the deficiencies of t-SNE. When the nonlinear manifold are highly curvilinear and wrapped in the original data space, t-SNE fails as its algorithm is constructed on Euclidean distances whereas Isomap is constructed on geodesic distances. Therefore, it would be interesting to combine the characteristics of these two techniques (i.e. geodesic distances and probabilistic approach) in future work. Otherwise, a combined use of both techniques is suggested in certain circumstances.

Furthermore, all the singular visualizations in this work can be seen as frames of a wider and unique visual storytelling composed of several chapters. The story order matters but depending on the reader level of experience some stories can be skipped to explore freely the features of more interest. Each story is independent but it is possible to fully understand some interesting underlying properties of NLDR techniques only by connecting and comparing visualizations of different stories. With the additional comments and digressions, we aimed to release a useful and variegated guide for satisfying needs of both students and researchers of any field.

Future Works

In this work, we followed some criteria to provide an efficient and accessible introduction to NLDR methods. We avoided 3D visualizations as much as possible and dynamic graphs which can be often misleading and unpractical.

However, an interactive version of this work is already public in NLDRviz [57]. In the future, we aim to extend it to allow users to perform NLDR techniques by themselves on their own data. This interactive tool could be used for studying and for investigating data and techniques of one's own interest. The reconstruction of final embeddings will be shown not only as frames but also as atomic videos to improve the algorithm understanding. In addition, some extensions could make it even more sophisticated letting people decide in which part of the algorithm the major focus should be. Finally, the most ambitious objective is to let people edit the algorithms and create their own new ones, making this platform a DR technique laboratory.

Furthermore, implementing Isomap, LLE and kPCA with an iterative method instead of an eigenvalue decomposition may be able to improve the visual storytelling and enhance comparison with SM, SNE, and t-SNE. Some attempts have been done to change the algorithm of these techniques to facilitate the visualization of DR process. For instance, in Fig. 30, there is a sequence of visualizations which capture the output space evolution generated by kPCA. Instead of PCA, the output is optimized by the Generalized Hebbian Algorithm which is a feedforward neural network model [20]. This can be a valid alternative to SVD and a way to investigate more in the future. Other solutions can be also the Arnoldi iteration [11].

Finally, receiving feedback during the initial phase as well as during the testing phase of the project could let us improve the storytellings based on the user needs and to check their effectiveness and expressiveness. In future works, a statistical validation of the results and an evaluation of the visual storytelling method by randomly selecting a sample of the intended audience can improve this work.

Appendix 1

Other figures

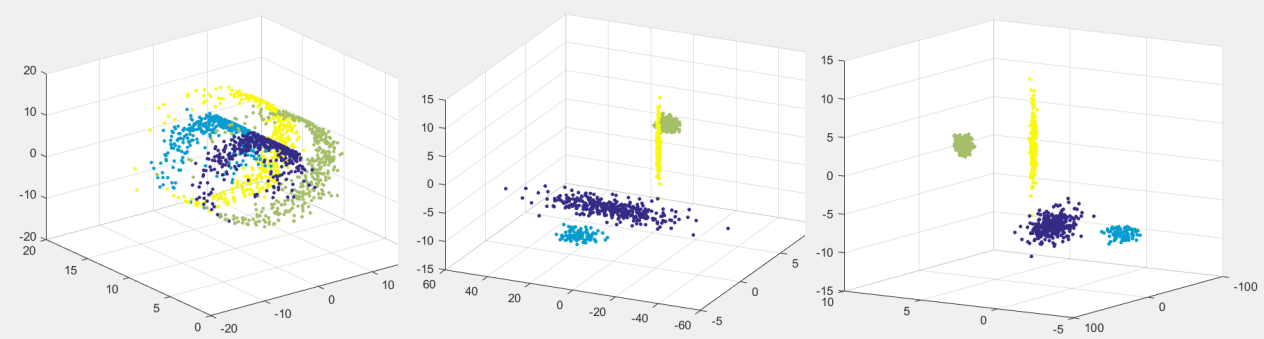


Figure 26: *Swiss Roll data (on the left) and Clustered data observed from two different angles*

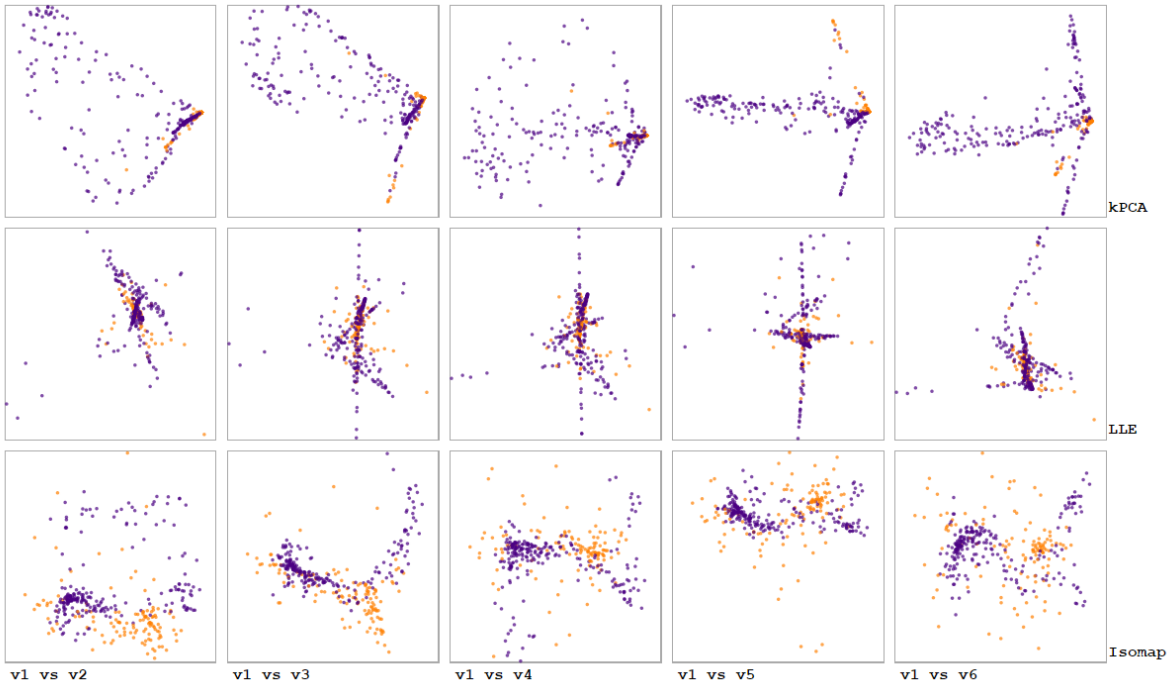


Figure 27: *Visual storytelling of the algorithm of kPCA, LLE and Isomap (per row) for different combinations of eigenvectors (per column) performed on Ionosphere data*

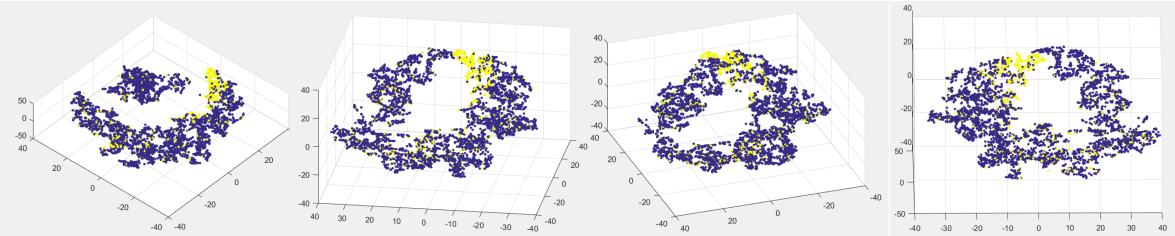


Figure 28: *The 3D final embedding by using t-SNE on Churn data observed from different angles*

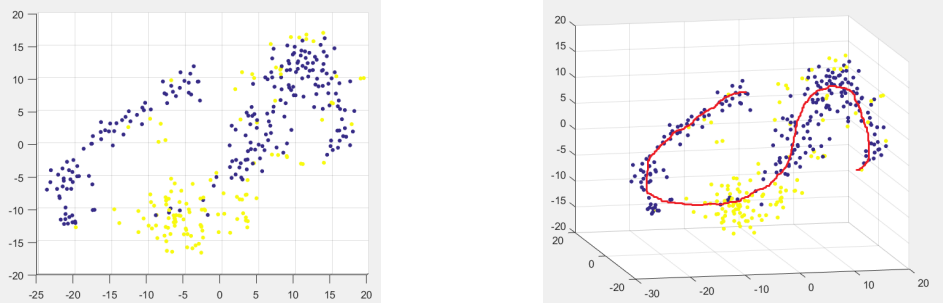


Figure 29: *The 3D final embedding by using t-SNE on Ionosphere data. On the figure on the right, the red line indicates the intrinsic dimension*

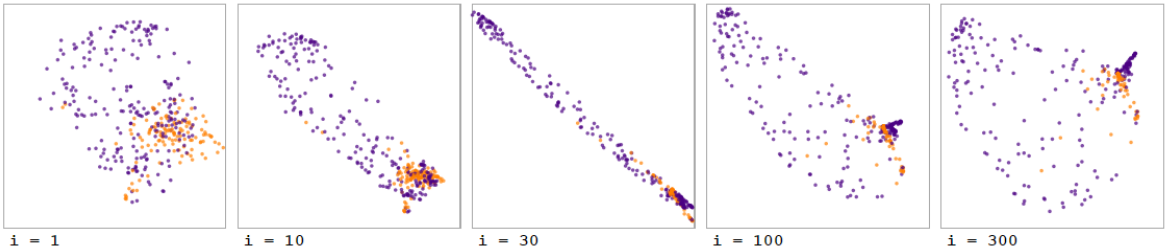


Figure 30: *Visual storytelling of kPCA at different iteration (per column) of the Generalized Hebbian Algorithm (GHA) performed on Ionosphere data*

Appendix 2

Visual Storytelling per Technique

Visualizing storytelling per technique highlights pros and cons of the algorithms characteristics where each storyboard represents the DR technique profile.
Fig. 31 shows the tendency of SM to create uniform dense circular-shaped final embeddings due to the simple cost function based on the Euclidean distance. Differently, the SNE final outputs highlight the crowding problem as shown in Fig. 32. Although the points belonging to the same class seem to be arranged correctly close to each other, they tend to collapse in the map center.
This issue is not observed in t-SNE which successfully separates clusters as illustrated in Fig. 33. Despite that, it struggles to recognize and unfold embeddings with curvilinear shapes adequately. Consequently, it can split data from the same class in more clusters. This can be attributed to the use of joint/conditional probabilities based on a Euclidean distance matrix to define dissimilarities between data points.

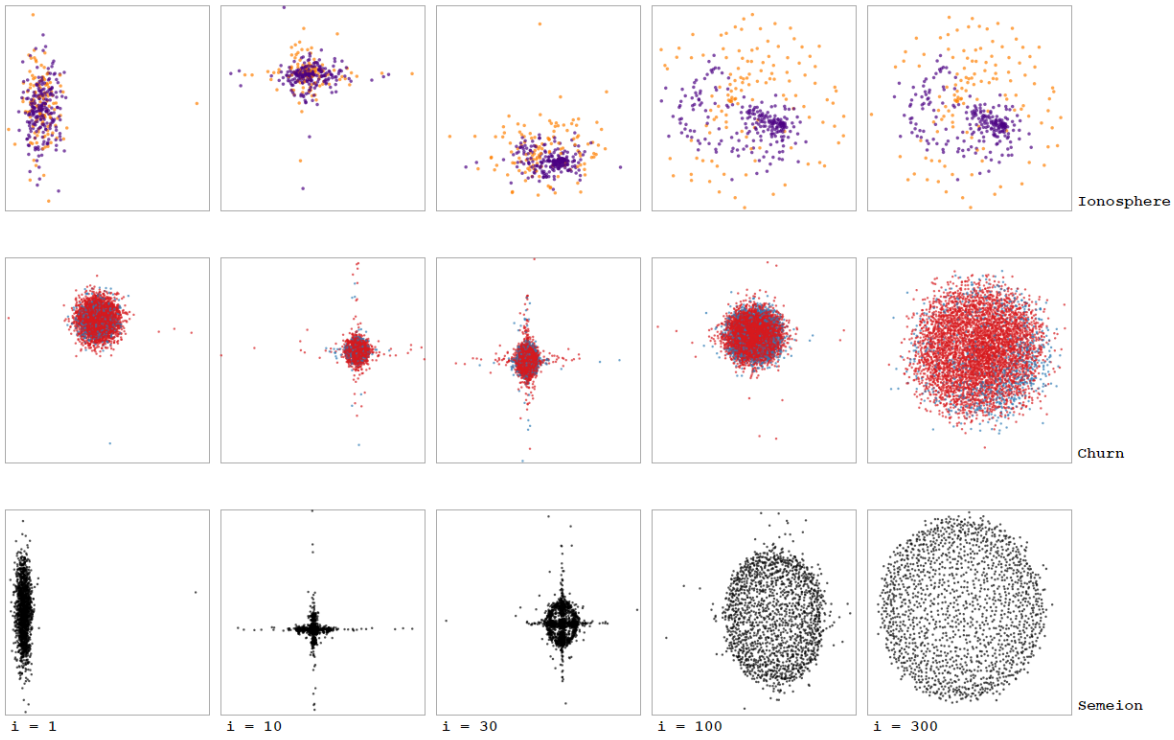


Figure 31: *Visual storytelling of the algorithm of SM at different iteration (per column) performed on Ionosphere data, Churn data and Semeion data (per row)*

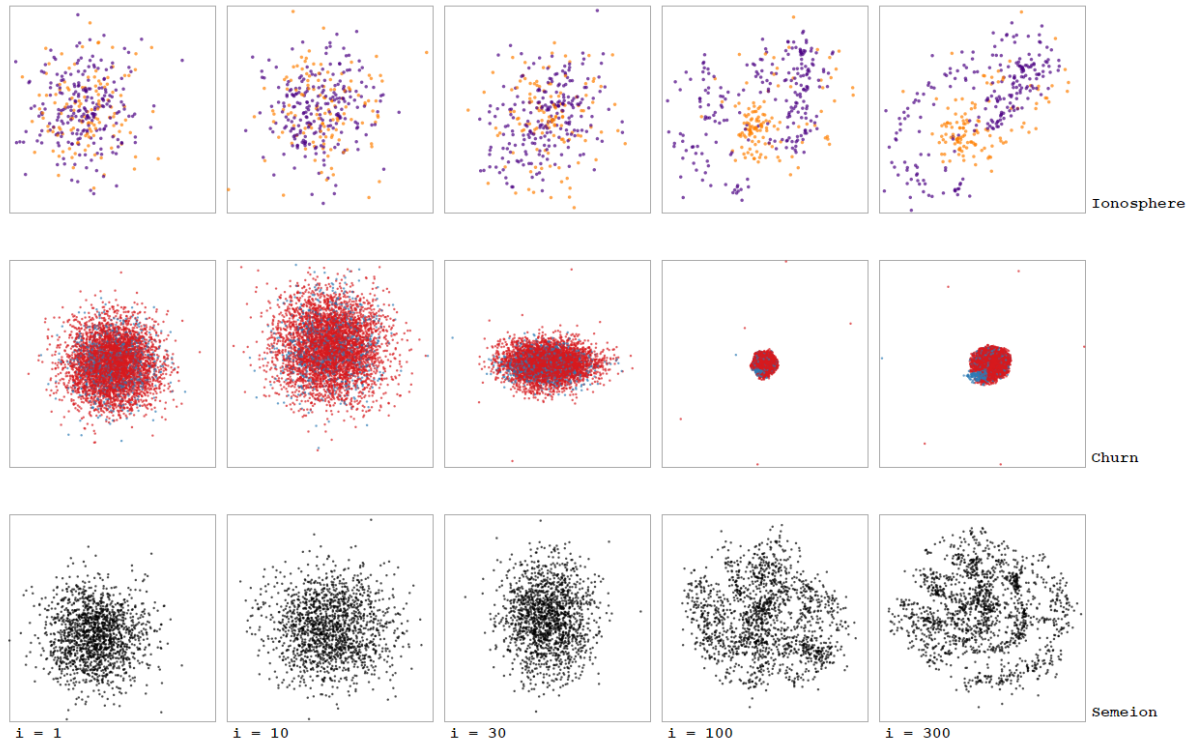


Figure 32: *Visual storytelling of the algorithm of SNE at different iteration (per column) performed on Ionosphere data, Churn data and Semeion data (per row)*

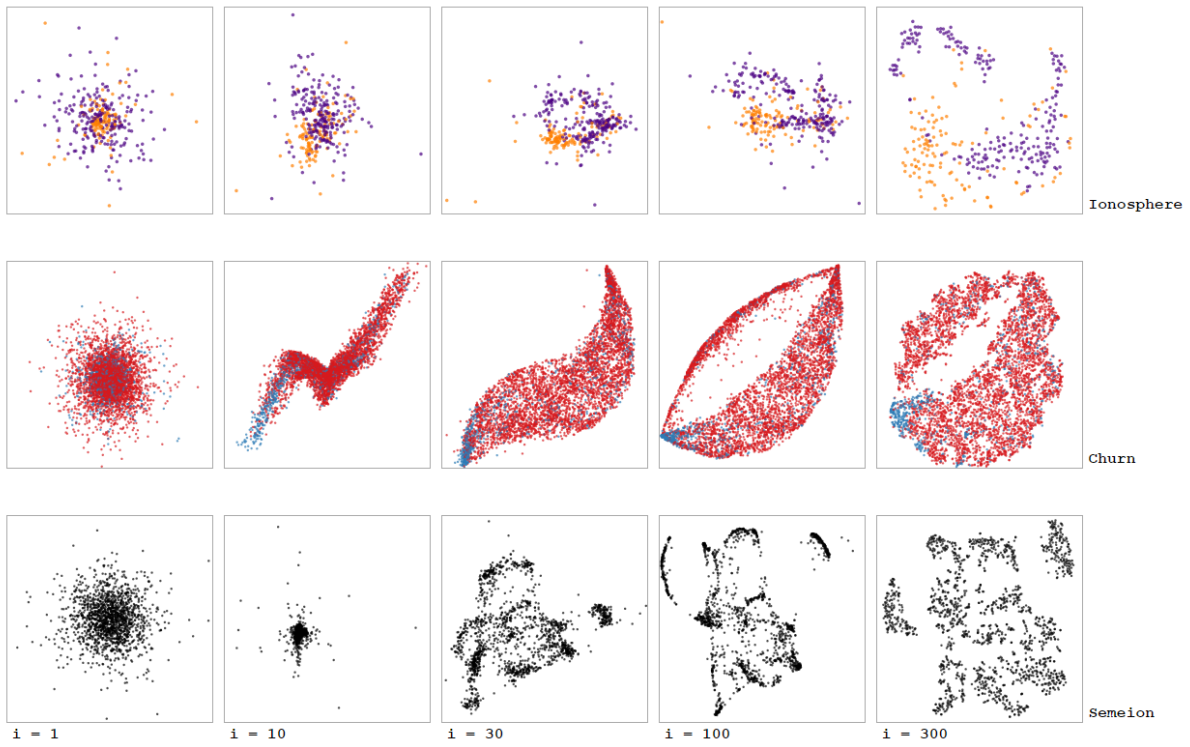


Figure 33: *Visual storytelling of the algorithm of t-SNE at different iteration (per column) performed on Ionosphere data, Churn data and Semeion data (per row)*

References

[1] Kaggle: Your Home for Data Science. https://www.kaggle.com/.

[2] Mushroom Classification | Kaggle. https://www.kaggle.com/uciml/mushroom-classification.

[3] UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/.

[4] UCI Machine Learning Repository: Breast Cancer Wisconsin (Diagnostic) Data Set. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic).

[5] UCI Machine Learning Repository: Ionosphere Data Set. https://archive.ics.uci.edu/ml/datasets/ionosphere.

[6] UCI Machine Learning Repository: Semeion Handwritten Digit Data Set. http://archive.ics.uci.edu/ml/datasets/semeion+handwritten+digit.

[7] Hervé Abdi and Lynne J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

[8] CC C Aggarwal. Re-designing distance functions and distance-based applications for high dimensional data. *ACM SIGMOD Record*, 30(1):256–266, 2001.

[9] Charu C. Aggarwal. *Data Mining: The Textbook*. 2015.

[10] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional space. *Database Theory – ICDT 2001*, pages 420–434, 2001.

[11] W E Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. (2):17–29, 1950.

[12] Mukund Balasubramanian and Eric L Schwartz. The isomap algorithm and topological stability. *Science (New York, N.Y.)*, 295(2002):7, 2002.

[13] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When Is " Nearest Neighbor " Meaningful ? pages 217–235, 1998.

[14] Its'hak Dinstein Boaz Lerner, Hugo Guterman, Mayer Aladjem and Yitzhak Romem. On pattern classification with Sammon's Nonlinear Mapping - An Experimental Study. *Mathematics, Applied Management, Production*, 31(4):371–381, 1998.

[15] Jamis Buck. Buckblog: Maze Generation: Algorithm Recap. http://weblog.jamisbuck.org/2011/2/7/maze-generation-algorithm-recap.

[16] Andreas Buja, Deborah F Swayne, Michael L Littman, Nathaniel Dean, Heike Hofmann, and Lisha Chen. Data Visualization with Multidimensional Scaling. 06511:1–30, 2007.

[17] Kerstin Bunte, Sven Haase, Michael Biehl, and Thomas Villmann. Stochastic neighbor embedding (SNE) for dimension reduction and visualization using arbitrary divergences. *Neurocomputing*, 90:23–45, 2012.

[18] Nicky Case. Explorable Explanations. http://explorableexplanations.com/.

[19] Antony Unwin Chun-houh Chen, Wolfgang Hardle. Handbook of Data Visualization.

[20] Original Contribution. Optimal Unsupervised Learning in a Single-Layer Linear Feedforward Neural Network. 2:459–473, 1989.

[21] James Cook, Ilya Sutskever, Andriy Mnih, and Geoffrey Hinton. Visualizing Similarity Data with a Mixture of Maps. *International Conference on Artificial Intelligence and Statistics*, (1):67—-74, 2007.

[22] Aldo Cortesi. Sortingalgorithmvisualisation.

[23] Robert Cowell. *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*. 2005.

[24] D R Cox, V Isham, N Keiding, T Louis, N Reid, R Tibshirani, H Tong, Monte Carlo, Methods J M Hammersley, and D C Handscomb. *Monographs on statistics and applied probability*. Number 1960. 2001.

[25] Daniel Kunin. Seeing Theory. http://students.brown.edu/seeing-theory/.

[26] V de Silva and J B Tenenbaum. Global Versus Local Methods in Nonlinear Dimensionality Reduction. pages 705–712, 2003.

[27] Dinoj Surendran. Swiss Roll Dataset. http://people.cs.uchicago.edu/{~}dinoj/manifold/swissroll.html.

[28] Witold Dzwinel. How to make sammon's mapping useful for multidimensional data structures analysis. *Pattern Recognition*, 27(7):949–959, 1994.

[29] Alessio Farcomeni and Luca Greco. *Robust methods for data reduction*. 2015.

[30] R Fletcher. Practical methods of optimization, 1986.

[31] Gabor Melli. Dataset Generator - Perfect data for an imperfect world. http://www.datasetgenerator.com/.

[32] David Galles. Data Structure Visualization. https://www.cs.usfca.edu/{~}galles/visualization/Algorithms.html.

[33] Ali Ghodsi. Dimensionality reduction a short tutorial. *Department of Statistics and Actuarial Science, Univ. of Waterloo, Ontario, Canada*, 37:38, 2006.

[34] Gabriel Goh. Why Momentum Really Works. http://distill.pub/2017/momentum/.

[35] A Ardeshir Goshtasby. *Similarity and Dissimilarity Measures*, pages 7–66. Springer London, London, 2012.

[36] S. Gratzl, A. Lex, N. Gehlenborg, N. Cosgrove, and M. Streit. From Visual Exploration to Storytelling and Back Again. *Computer Graphics Forum*, 35(3):491–500, 2016.

[37] Dave Gray and James Macanufo. Gamestorming.

[38] Detlef Groth, Stefanie Hartmann, Sebastian Klie, and Joachim Selbig. Kernel Principal components analysis. *Methods in molecular biology (Clifton, N.J.)*, 930(4):527–47, 2013.

[39] Naftali Harris. Visualizing K-Means Clustering. https://www.naftaliharris.com/blog/visualizing-k-means-clustering/.

[40] Paul Henderson. Sammon Mapping. *Pattern Recognition Letters*, 18:1307–1316, 1997.

[41] Alexander Hinneburg, Charu C Aggarwal, and Daniel a Keim. What is the Nearest Neighbor in High Dimensional Spaces? *Proceedings of the 26th VLDB Conference*, pages 506–515, 2000.

[42] Geoffrey E Hinton and Sam T Roweis. Stochastic neighbor embedding. *Advances in neural information processing systems*, pages 833–840, 2002.

[43] Harold Hotelling. Analysis of a complex statistical variables into principal components.

[44] Jared M. Spool. 5 Design Decision Styles. What's Yours? https://articles.uie.com/five{_}design{_}decision{_}styles/.

[45] Jimmy Johansson, Patric Ljung, Mikael Jern, and Matthew Cooper. Revealing structure within clustered parallel coordinates displays. *Proceedings - IEEE Symposium on Information Visualization, INFO VIS*, pages 125–132, 2005.

[46] Jr. John W. Sammon. A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computers*, C(5), 1969.

[47] James M Joyce. *Kullback-Leibler Divergence*, pages 720–722. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[48] Ravi Kannan and John Hopcroft. *Computer Science Theory for the Information Age*. 2012.

[49] Baker Kirk. Singular Value Decomposition Tutorial. 2005:14–20, 2005.

[50] Jennifer Frazier Kwan-Liu Ma, Isaac Liao, Helwig Hauser, Helen-Nicole Kostis. Scientific Storytelling Using Visualization. 2012.

[51] Laurens van der Maaten. t-SNE. https://lvdmaaten.github.io/tsne/.

[52] Neil Lawrence. Probabilistic Non-linear Principal Component Analysis with Gaussian Process Latent Variable Models. 6:1783–1816, 2005.

[53] Neil D Lawrence. Spectral Dimensionality Reduction via Maximum Entropy. 15:51–59, 2011.

[54] Bongshin Lee, Nathalie Henry Riche, Petra Isenberg, and Sheelagh Carpendale. More Than Telling a Story: Transforming Data into Visually Shared Stories. 2015.

[55] John a Lee, John Lee, and Michel Verleysen. *Nonlinear Dimensionality Reduction*. 2007.

[56] Elizaveta Levina, Ann Arbor Mi, and Peter J Bickel. Maximum Likelihood Estimation of Intrinsic Dimension.

[57] Lorenzo Amabili. NLDRviz. https://lorenzoamabili.github.io/.

[58] Gabor Lugosi Luc Devroye, Laszlo Gyorfi. *A Probabilistic Theory of Pattern Recognition*. 1996.

[59] Laurens Van Der Maaten. Learning a Parametric Embedding by Preserving Local Structure. *JMLR Proceedings vol. 5 (AISTATS)*, pages 384–391, 2009.

[60] Laurens Van Der Maaten, Eric Postma, and Jaap Herik. Dimensionality Reduction : A Comparative Review. (April), 2009.

[61] Jock Mackinlay. Automating the Design of Graphical Presentations of Relational Information. 5(April 1986):110–141, 1987.

[62] Kosara Jock Mackinlay Robert. Storytelling: The Next Step for Visualization. pages 44–50, 2013.

[63] K V Mardia and T Kent. *Multivariate Analysis*.

[64] James Martens and Geo Hinton. On the importance of initialization and momentum in deep learning. (2010), 2012.

[65] A.R. Martin and M.O. Ward. High Dimensional Brushing for Interactive Exploration of Multivariate Data. *Proceedings Visualization '95*, pages 271–278, 1995.

[66] Andrew McCallum, Kamal Nigam, and Lyle L.H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 169–178, 2000.

[67] Duncan Meech. Animated Algorithms. http://www.algomation.com/.

[68] Mike Bostock. D3.js - Data-Driven Documents. https://d3js.org/.

[69] Mike Bostock. Scatterplot Matrix Brushing - bl.ocks.org. https://bl.ocks.org/mbostock/4063663.

[70] Mike Bostock. Visualizing Algorithms. https://bost.ocks.org/mike/algorithms/.

[71] Maria Montessori. *The Montessori Method : the origins of an educational innovation: including an abridged and annotated edition of Maria Montessori's The Montessori Method*, volume 1. 2004.

[72] Tamara Munzner. *Visualization Analysis and Design*. 2014.

[73] E Pekalska, D. de Ridder, R. P. Duin, and M. A. Kraaijveld. A new method of generalizing Sammon mapping with application to algorithm speed-up. pages 221–228, 1999.

[74] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. (December), 2015.

[75] Pavel Pudil and Jana Novovičová. *Novel Methods for Feature Subset Selection with Respect to Problem Knowledge*, pages 101–116. Springer US, Boston, MA, 1998.

[76] Nick Qi Zhu. *Data Visualization with D3.js Cookbook*. 2013.

[77] R2D3. A visual introduction to machine learning. http://www.r2d3.us/visual-intro-to-machine-learning-part-1/.

[78] Margaret Rangecroft and E. R. Tufte. Envisioning Information. *Applied Statistics*, 41(1):227, 1992.

[79] Lawrence K Roweis Sam T.Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323, 2000.

[80] Dominik Sacha, Leishi Zhang, Michael Sedlmair, John A. Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C. North, and Daniel A. Keim. Visual Interaction with Dimensionality Reduction: A Structured Literature Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):1–1, 2016.

[81] Lawrence K Saul and Sam T Roweis. An introduction to Locally Linear Embedding. pages 1–13, 2000.

[82] Lawrence K Saul, Kilian Q Weinberger, and Daniel D Lee. Spectral Methods for Dimensionality Reduction.

[83] B Scholkopf, a J Smola, and K R Muller. Kernel Principal Component Analysis. *Computer Vision And Mathematical Methods In Medical And Biomedical Image Analysis*, 1327:583–588, 2012.

[84] Nicol N Schraudolph, G Simon, and S V N Vishwanathan. Fast Iterative Kernel PCA. (1).

[85] Edward Segel and Jeffrey Heer. Narrative Visualization : Telling Stories with Data. 16(6):1139–1148, 2010.

[86] H. Siirtola. Direct manipulation of parallel coordinates. *2000 IEEE Conference on Information Visualization. An International Conference on Computer Visualization and Graphics*, pages 373–378, 2000.

[87] Steven Halim. VisuAlgo - visualising data structures and algorithms through animation. https://visualgo.net/en.

[88] Joshua B Tenenbaum, Vin de Silva, and John C Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.

[89] Think Up Themes Ltd. Five Design Sheet | Design Methodology for Visualisation. http://fds.design/.

[90] Toptal. Sorting Algorithm Animations. https://www.toptal.com/developers/sorting-algorithms.

[91] Edward R Tufte. The visual display of quantitative Information, 2008.

[92] Stanford University. Visualizing K-Means Clustering. http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html.

[93] Laurens Van Der Maaten. Accelerating t-SNE using tree-based algorithms. *Journal of machine learning research*, 15(1):3221–3245, 2014.

[94] Laurens Van Der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[95] Laurens Van Der Maaten and Geoffrey Hinton. Visualizing non-metric similarities in multiple maps. *Machine Learning*, 87(1):33–55, 2012.

[96] Michel Verleysen and John A Lee. Nonlinear Dimensionality Reduction for Visualization. *Neural Information Processing*, pages 617–622, 2013.

[97] Lewis Lehe Victor Powell. Explained Visually. http://setosa.io/ev/.

[98] Quan Wang. Face Recognition and Active Shape Models. 1995.

[99] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to Use t-SNE Effectively. http://distill.pub/2016/misread-tsne/, 2016.

[100] Kilian Q Weinberger and Lawrence K Saul. Learning a Kernel Matrix for Nonlinear Dimensionality Reduction. 2004.

[101] Christopher K I Williams. On a Connection between Kernel PCA and Metric Multidimensional Scaling. pages 11–19, 2002.

[102] Graham Wills. *Statistics and Computing*. 2012.

[103] Ian H. Witten, Eibe Frank, and Mark a. Hall. *Data Mining: Practical Machine Learning Tools and Techniques, Third Edition*, volume 54. 2011.

[104] Huilin Xiong and Xue-wen Chen. Kernel-based distance metric learning for microarray data classification. 11, 2006.

[105] Li Yang. Sammon' s Nonlinear Mapping Using Geodesic Distances. pages 4–7.

[106] Hujun Yin. Nonlinear multidimensional data projection and visualisation. *Intelligent Data Engineering and Automated Learning*, pages 377–388, 2003.

[107] Mohammed J Zaki and Meira Jr. Meira. *Data Mining and Analysis: Fundamental Concepts and Algorithms*. 2013.

[108] Lingsong Zhang, J S Marron, Haipeng Shen, and Zhengyuan Zhu. Singular Value Decomposition and Its Visualization. pages 1–38, 2007.

KU LEU
Oude Markt 13 - bus 5
3000 LEUVEN, BEL
tel. +32 16 324010 fax +32 16 324
www.kuleuve