Proximities in dimensionality reduction

John Aldo Lee, Cyril Bodt De, Ludovic Journaux, Lucile Sautot

Proximities in dimensionality reduction

John Aldo Lee Cyril de Bodt Ludovic Journaux Lucile Sautot

May 17, 2022

- **John Aldo Lee** was born in 1976 in Brussels, Belgium. He received the M.Sc. degree in Applied Sciences (Computer Engineering) in 1999 and the Ph.D. degree in Applied Sciences (Machine Learning) in 2003, both from the Université catholique de Louvain (UCLouvain.be, Louvain-la-Neuve, Belgium). He is currently a UCLouvain Professor and a Senior Research Associate with the Belgian fund of scientific research (Fonds de la Recherche Scientifique, F.R.S.-FNRS). His current research interests are at the crossing of artificial intelligence, machine learning, information visualisation, image processing, medical imaging, radiation oncology, and medical physics. He is currently the head of the Centre for Molecular Imaging, Radiotherapy, and Oncology (MIRO) within the Institute of Experimental and Clinical Research (IREC) next to the Saint-Luc University Hospital (Belgium). He is also affiliated with the UCLouvain ICTEAM Institute (Information and Communication Technologies, Electronics, and Applied Mathematics). He is author or coauthor of more than 160 documents listed on Scopus. He is also an author of a book entitled "Nonlinear Dimensionality Reduction" and published by Springer in 2007.

- **Cyril de Bodt** was born in Brussels, Belgium, in 1993. He received the M.Sc. degree in Applied Sciences (Mathematical Engineering) in 2016 and the Ph.D. degree in Engineering Science and Technology in 2020, both from UCLouvain, Belgium. He is now a Postdoctoral Research Fellow in the Human Dynamics group of the Massachusetts Institute of Technology (MIT) Media Lab. His research interests include dimensionality reduction, machine learning, visualization, clustering, optimization and data mining.

- **Ludovic Journaux** was born in 1976 in Dijon, France. He received the Ph.D. degree in computer sciences from the University of Burgundy, France, in 2006. He actually associate professor with AgroSup Dijon since 2008 and at the Burgundy computer science laboratory at the University of Burgundy. He is interested in machine learning, data sciences, Signal and image processing, and high multidimensional data processing.

- **Lucile Sautot** was born in 1989 in Grenoble, France.She received her Ph.D. degree in Computer Sciences in 2015, both from the Université de Bourgogne (Dijon, France). She is associate professor at AgroParisTech (French public institute of life and environmental sciences and industries). Her research interests are the design of information systems for agriculture and environment. It's her first chapter in a book.

Abstract

Dimensionality reduction aims at representing high-dimensional data in a lower-dimensional representation, while preserving their structure (clusters, outliers, manifold). Dimensionality reduction can be used for exploratory data visualization, data compression, or as a preprocessing to some other analysis in order to alleviate the curse of dimensionality. Data structure is usually quantified with indicators, like covariance between variables, or pairwise proximity relationships, like scalar products, distances, similarities, or neighbourhoods. One objective of this chapter is to provide an overview of some classical and more recent methods of dimensionality reduction, to shed some light on them from the perspective of analyzing proximities, and to illustrate them with multivariate data that could be typically encountered in social sciences. Complementary aspects like quality assessment and alternative metrics are briefly developed.

1 Introduction

Data collection becomes easier and cheaper every day. One purpose of accumulating large amounts of data (*Big Data*) and making it publicly available (*Open Data*) is essentially the hope that analysts will be able to extract some useful knowledge or information out of it. Information means here that data is hoped not to be just random, that some structure or patterns can be spotted and possibly interpreted. Beyond retrospective analysis of data, information is expected to be valuable in prospective applicative cases, where the same patterns could be found back or even predicted in new data. Extracting information can thus be seen as summarizing large amounts of data into exploitable, more concise knowledge.

Both structure and pattern are terms that connote some visual or geometrical perception in space. Let us leave aside text data, which are mostly linear, to focus rather on multivariate data. Then, a spatial representation is more or less straightforward. Each datum or observation can be thought of as a point (or individual sample) in a multidimensional space, with one axis (or dimension) associated with each variable (or attribute). Structure in data means that observations are not occupying space in a dense, entropic, and trivial way, like uniform or normal distributions that would span all dimensions indistinctively. Instead, some regions of space are expected to be (nearly) empty. Abstract structures that live in a data space are for instances subspaces, clusters, or outliers.

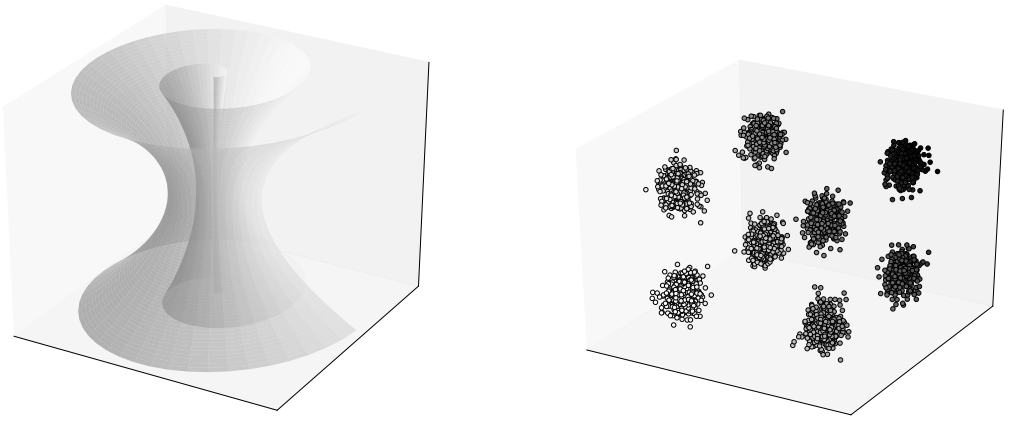
Subspaces can be linear, like hyperplanes. A straight line or a plane in our three-dimensional space are typical examples. More complex, curved hypersurfaces, are said to be nonlinear and termed manifolds. Any curve, a paraboloid, or an ellipsoid are all one- or two-dimensional manifolds embedded in our three-dimensional space (Figure 1a).

Clusters are densely populated regions of space, separated by zones where data is much more sparse (Figure 1b). To some extent, modes of a distribution can be thought of as clusters. Outliers are then rare and isolated observations lying in between or outside clusters.

Abstract structures like subspaces, clusters, and outliers are high-level objects that are loosely defined and difficult to identify in data. The task of finding them is facilitated by considering lower-level indicators that characterise structure. Such indicators aim for instance at quantifying possible linear and nonlinear dependences between variables, like covariance and mutual information. Other indicators deal with pairwise proximity relationships between data, like affinities, distances, or even scalar products.

Clustering and outlier detection are the task of identifying possible clusters and outliers in data, usually by labeling instances. Projection methods and manifold learning aim to spot linear and nonlinear subspaces, typically by providing a lower-dimensional representation of data. Such representation, often called projection or embedding, should ideally keep just as many dimensions as necessary. For instance, one dimension is enough to represent a curve, a plane or curved surface requires two dimensions, but more complex structures might need more dimensions to be correctly represented. Looped structures like a circle or spherical shell might need more than one or two dimensions to be rendered faithfully.

Dimensionality reduction (DR) is a task even more general than projecting or embedding [CPn97, LV07]. It encompasses them, but could also achieve other goals, like forcing a two-dimensional representation even though the underlying structure in data might span more than two dimensions. A typical application of DR is exploratory data visualisation. A two-dimensional representation turns out to be visually convenient, whereas the actual intrinsic dimensionality or



(a) (b)

Figure 1: Figure 1a shows a surface in shades of grey that depicts an hourglass-shaped Swiss roll, as an instance of a two-dimensional subspace embedded in our usual three-dimensional space. Figure 1b presents eight dense clusters, separated by large empty regions in the three-dimensional space.

even the true nature of the underlying structure is yet to be discovered. For instance, DR can yield a representation where not only manifolds, but also clusters and outliers can be visually detected. Beyond visualisation, DR can also be used to compress data, by keeping their principal components of variability and getting rid of the minor ones, like those only affected by statistical noise.

How does DR work intuitively? The general idea is to capture the abstract, high-level structure of data (manifolds, clusters, etc.) by evaluating appropriate low-level structure indicators (scalar products, distances, affinities, neighbourhoods, etc.) in the multivariate data space. Next, a lower-dimensional representation of data (projection, embedding, etc.) is devised such that the same indicators, now computed on the lower-dimensional representation space, yield nearly identical values. In that sense, the result of DR might be deemed to represent data faithfully. For example, clusters might be faithfully rendered if small intra-cluster and long inter-cluster distances are preserved in their low-dimensional representation. Distance preservation has been a very popular principle of DR for many decades, although it can be flawed in certain cases.

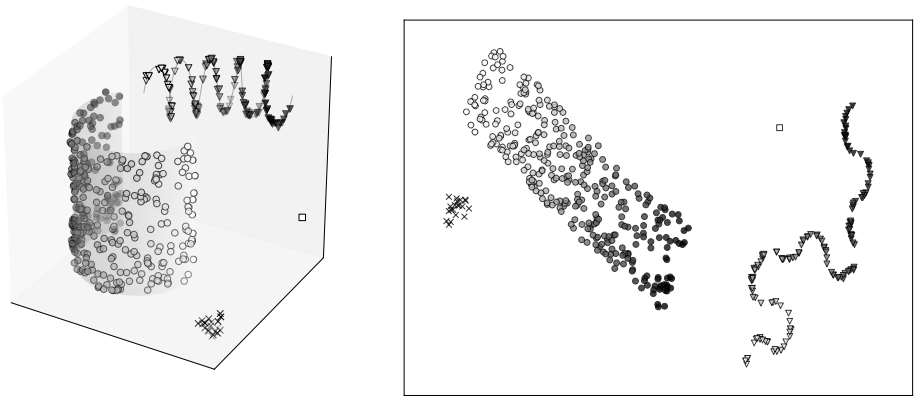
Over the course of its history, DR has moved from rather global indicators like covariance between dimensions to more and more local ones, like pairwise distances, similarities, or probabilistic neighbourhoods. Most modern DR methods focus now on preserving local proximity from the high-dimensional data space to the lower-dimensional representation. Their objective could be briefly put as: represent dissimilar data far apart and similar data close to each other. Their quality assessment follows the same principle.

Figure 2 illustrates with a toy example how modern DR methods can capture the proximities and neighbourhoods in high-dimensional data. Markers in Fig. 2a represent three-dimensional data points. Obviously, these have a complex and particular distribution in the three-dimensional space. Namely, the circles are sampled from a two-dimensional surface (half a cylinder), the triangles are distributed on a one-dimensional subspace (a spiral), the crosses define a smaller cluster, and the single square seems to be an outlier, as it is lying alone and far from other data points. In real applications, noise can affect measurements, explaining why the circles and triangles slightly deviate from their respective two- and one-dimensional subspaces.

The application of a DR method to data shown in Figure 2a should render the four types of identified structures. This goal is achieved in Figure 2b, where a two-dimensional embedding is obtained by running the *Neighborhood Retrieval Visualizer* (NeRV) [VPN+10]. In particular, circles are now distributed on a piece of plane, which means that the half cylinder were nicely unrolled. The spiral is correctly represented as a one-dimensional string. The crosses still lie in a small cluster, while the square remains an outlier. Hence, as previously underlined, DR attempts to represent dissimilar data far apart in the embedding and, conversely, similar data close to each other. Here, shades of grey are used only as a visual cue to allow seeing corresponding points on figures 2a and 2b.

In a practical cases, data dimensionality can be much higher, making direct visualization very

difficult. Therefore, a representation such as in Fig. 2a is not available. This highlights the interest of low-dimensional embeddings provided by DR methods, as they often give users some insights about the data structure through visualization, like in Figure 2b.



(a) (b)

Figure 2: Illustration of some concepts of dimensionality reduction. Figure 2a: Three-dimensional data space. Figure 2b: Two-dimensional embedding obtained after applying a modern DR method.

The rest of this chapter is organized as follows. Section 2 briefly reviews the history of DR and its main trends over time. Section 3 presents the different ways to categorize DR methods, considering properties of their respective models and optimization techniques. Section 4 defines the necessary symbols and concepts that are used further. After a few definitions, section 5 describes a few emblematic DR methods in more details, to illustrate their main categories and principles. Section 6 briefly discusses how to assess quality of DR, whereas Section 7 illustrates and compares DR methods with data. Section 8 deals with alternative metrics and distances. Section 9 draws the conclusions and sketches some open issues and perspectives.

2 Brief history of DR

The history of DR spans more than a century and it has benefitted from trends and progress in other fields like social sciences (psychometry, econometrics, and statistics in the broad sense), artificial intelligence (machine intelligence), computer science (graph theory), and applied mathematics (optimization theory).

The history of DR begins with principal component analysis, a linear projection method that has been discovered independently several times in different contexts [Pea01, Hot33, Kar46, Loe48] between 1901 and 1948. Starting from coordinates of points in some space, PCA looks for the projection that both preserves most of the observed variance and minimizes the residues between the initial points and their projection onto the linear subspace [Jol86]. PCA is described in Section 5.1.1.

Similar to PCA, classical metric multidimensional scaling (MDS) aims to project points that are known through pairwise relationships like similarities or dissimilarities [YH38, Tor52, Gow66]. In the particular cases where these relationships are scalar products or Euclidean distance, classical metric MDS turns out to yield the same solution as PCA. Section 5.2.1 deals with this early version of MDS.

Both PCA and classical metric MDS are considered to project data in a usual, linear way, if distances are Euclidean. Starting from the late 1960s, metric MDS, also known as stress-based MDS, generalizes classical MDS and allows for nonlinear mappings of data, thanks to explicit formalization of a very generic and tunable criterion of distance preservation from one space to the other [She62, Sam69, DH97]. This criterion is an objective function that is often called *stress*, which assesses how pairwise distances d_{ij} computed from the low-dimensional coordinates match the corresponding distances δ_{ij} in the high-dimensional data space. In particular, this framework allows for putting more weight on the preservation of short distances as compared to longer ones. This has been the first attempt to implement the intuition that proximities [She62] are more important than more distant, looser relationships [Sam69, DH97].

In the sixties and seventies, yet another generalization of MDS was investigated, with the early intuition that distances measured in two spaces of largely differing dimensionalities are not necessarily comparable [Don00, FWV07]. Nonmetric MDS relaxes strict distance preservation ($\delta_{ij} = d_{ij}$ for all i, j) into ordinal distance preservation ($\delta_{ij} \leq \delta_{ik} \Rightarrow d_{ij} \leq d_{ik}$ for all i, j, k) [Kru64, TYdL77]. Section 5.2.2 describes stress-based (metric and nonmetric) MDS.

In the eighties, artificial neural networks raised much interest, with a peak when backpropagation was popularized in 1986 [RHW86]. A few years earlier, a neural model known as self-organizing feature maps (SOMs) was published [Koh82]. This model performs both vector quantization (or clustering, like K-means) and a particular form of nonlinear DR, particularly suited for data visualization. The neural trend continued in the nineties with the advent of auto-associative networks, composed of an encoder and decoder in sequence [Kra91, Oja91, UNN91, DC93]. Difficulties to train such networks with many layers of neurons stacked in the encoder and decoder were addressed only recently in the early years of the deep learning trend [HS06]. Section 5.3 investigates these neural approaches of DR based on vector quantization or auto-association.

In the late nineties and early 2000s, developments in DR were influenced mainly by the interest in kernel techniques in the booming field machine learning. Briefly put, the so-called kernel trick theorizes the fact that some particular functions termed *kernels* induce a scalar product in an unknown, possibly very high-dimensional space, without dealing explicitly with coordinates (or *features*) in that space. The pioneering work in that direction was kernel PCA, which 'kernelizes' classical metric MDS into a method of nonlinear mapping [SSM98]: MDS is applied in the feature space induced by a nonlinear kernel, instead of the raw data space. Other methods followed, investigating numerous kernels and various ways to build them, like graph-approximations of geodesic distances [TdSL00a], commute-time distance [YVW+05], (heat) diffusion kernels [CLL+05], or even optmized kernels [WS06]. All these methods are closely related to graph theory [BN02, RS00, DG03] and spectral clustering [SFYD04, BVP+03]. While these methods are appealing theoretically, they entail usually some arbitrariness in the choice of the kernel or difficulties in its optimization. Section 5.4 deals with spectral methods of DR, with a focus on kernel PCA and Laplacian eigenmaps.

At the time of writing, the latest trend in DR investigates methods of stochastic neighbour embedding (SNE), which started in 2002 [HR03]. The next milestone occurred a few years later, in 2008, with t-distributed SNE, a variant of SNE that gained popularity thanks to its relative simplicitly and disruptively good experimental results [vdMH08]. Stochastic neighbour embedding aims at preserving probabilistic neighbourhoods from one space to the other: if some point $\boldsymbol{\xi}_\mathbf{j}$ is deemed to be a neighbour of $\boldsymbol{\xi}_i$ with good probability in the HD space, then its LD counterpart $\mathbf{x}_j$ should also be a neighbour of $\mathbf{x}_i$ with approximately the same probability. The properties of these stochastic neighbourhoods have been investigated [HR03, LV11, VCPn13, LV14], as well as objective functions to assess their HD-LD discrepancy [VPN+10, BHBV12, LRB+13]. Variants with simplified neighbourhoods [CPn10], multi-scale neighbourhoods [LPOnV15] and missing data robustness [dBMVL18b] exist, as well as scalable approximations of SNE with tree structure [HR03, vdM14, YPK13, dBMVL18a]. Section 5.5 presents an overview of probabilistic neighbourhood preservations, with details on SNE and its variants.

In parallel to the development of new DR methods, the issue of DR quality assessment (QA) has been investigated at some occasions. It is obviously important but difficult, DR being an unsupervised learning task. The most obvious way to evaluate quality of DR results is to look at how well the criterion of the considered DR method has been optimised. Popularity of stress-based MDS has led to the common adoption of stress functions, like Sammon's stress, as quality indicators [Sam69]. This approach is, however, intrinsically biased towards methods working with the very same objective. For instance, specific quality criteria have been devised for SOMs [VDHM97, BHV99, GS96, GS97, BP92]. Indirect ways to assess DR quality are also widespread, like classification errors, provided data are class-labeled. In the recent years, generic QA frameworks have been proposed, most of them quantifying neighbourhood preservation, as this principle makes only minimal assumptions about the data and method at hand [VK01, AC06, FC07, VK07, VPN+10, LV09, LRB+13, LPOnV15].

3 Categories

Dimensionality reduction is generic objective that can be particularized in many ways. Also there are different models and techniques to achieve these objectives, leading to methods with specific features.

3.1 Linear versus nonlinear model

Dimensionality reduction amounts to finding a subspace that can 'explain' observed data and dependencies between dimensions or variables. These dependencies can be either simply linear or more complicated, namely, nonlinear. In other words, data are distributed on some linear or nonlinear subspace (or close to these if there is some noise). Technically, a nonlinear subspace is called a *manifold*.

Some DR methods rely on a linear model: they can only identify a linear subspace and project data onto it. More complex DR methods can disentangle nonlinear dependencies between variables and are thus capable of identifying manifolds. Nonlinear DR methods are more powerful than linear ones, but also more prone to overfitting. Nonlinear DR can somehow unfold manifolds before projecting them in a lower-dimensional space; this is sometimes abusively called nonlinear projection.

The assumption of an underlying manifold, also known as the manifold hypothesis, is sometimes deemed central in some DR methods, called manifold learners. Most of the time, however, these methods are applied like others to all sort of data, be there a manifold, or clusters.

The manifold hypothesis calls for another concept, called the *intrinsic dimensionality* of the manifold [Fuk82]. If data is provided as coordinates in some space, the dimensionality of this space is just the number of features or variables. On the other hand, the dimensionality of the manifold is a priori unknown. For instance, data distributed on a sphere has an ambient dimensionality of three and an intrinsic dimensionality of two (the sphere is a surface in the physical space). There are various methods of estimating the intrinsic dimensionality of data [CV02, Fuk82].

3.2 Parametric versus nonparametric model

The distinction between these two types of models concerns mostly how the mapping from high- to low-dimensional coordinates is formalized and how it can be reused. In non-parametric methods, the model reduces to its simplest form: the DR method gives each available datum low-dimensional coordinates. This means that if new data come afterwards, the model will not be able to compute their associated low-dimensional. A complementary method, called an out-of-sample extension [BPV+04], can possibly carry out this additional task.

On the other hand, parametric models usually assume some analytical function that formalizes the mapping from high- to low-dimensional coordinates. Typically this function is made as generic as possible with the use of many tunable parameters and DR then just consists in adjusting those. Parametric models can be directly reused to new, out-of-sample data. The price to pay is that no matter how generic they are designed, parametric models always rely on some assumption about the data mapping, that can be restricted to be linear like in PCA [Jol86] or constrained to be continuous like in auto-encoders [HS06].

3.3 Local versus global DR

Very often, DR is categorized into local and global methods [dST03, RSH02]. However, these opposed categories are rather unclear and fuzzy. Most of the time, local methods are those who consider only short-range distances or neighborhoods, with respect to the total number of data, whereas global ones take into account all distances and neighborhoods indifferently. Not considering explicitly long distances (or non-neighbors) does not necessarily mean that the model or method does not implicitly give them some default value, like infinity. In spectral methods like Laplacian eigenmaps [BN02] or locally linear embedding [RS00], usually seen as local, reformulating them as classical metric MDS applied to commute-time distances transforms them into equivalent methods that would typically be considered global. Moreover, the local/global distinction appears mostly informal and can difficultly account for hybrid method like MDS and Sammon's mapping, where the importance of short and long distances can be modulated. A better way of determining whether a DR method or embedding is local or global would be to look at how it performs for small and large neighborhoods with the quality assessment tools detailed further below in Section 6. [LPOnV15].

3.4 Discriminative versus generative DR

This distinction deals with the direction the model or method maps data. A discriminative method typically maps high-dimensional data to a lower-dimensional embedding. Most methods belong

to this category. A generative method works the other way around and starts from the low-dimensional space: a candidate embedding and the mapping to the high-dimensional space are refined in order to be able reproduce the observed data.

Discriminative DR can be non-parametric, whereas generative DR is typically either parametric or non-parametric but with a user-fixed low-dimensional space, like in SOMs. Some methods like auto-encoders provide parametric mappings in both directions (which are not necessarily the inverse of one another).

3.5 Mild versus about-right versus harsh DR

Dimensionality reduction can process data with largely varying numbers of features or coordinates, between three and thousands, and aim typically at spaces with one, two, three, or a few more dimensions. Hence the dimensionality gap between the initial and final spaces can vary a lot [CPn97].

About-right DR would target an embedding dimensionality that is guided by the estimated intrinsic dimensionality of data [Tak85], under the manifold hypothesis detailed in Section 3.1 above. If there are for instance five latent parameters or degrees of freedom in the data generation process, then at first glance it would be counterproductive to reduce dimensionality below five, at the expense of squeezing data and possibly damaging part of its structure. Nevertheless, five dimensions remain difficult to explore visually and harshly crushing data down to three or two dimensions, whatever the actual intrinsic dimensionality is, might still be a useful tradeoff and provides some insights. Eventually, the converse is possible as well: if visualization is not intended, if preserving the data structure is essential, or if DR is just some form of preprocessing to compress data, then mild DR can target a dimensionality that is slightly higher than the intrinsic one [Tak85].

3.6 Spectral versus nonspectral DR

Dimensionality reduction methods can also be categorized according to the optimization techniques they involves. Two main categories can be distinguished. The first one includes all methods that solve a convex optimization problem, typically with an spectral decomposition into pairs of eigenvalues and eigenvectors, like in PCA, Laplacian eigenmaps, LLE, and many other so-called spectral embedding methods [BH03, XSB06, SWS+06].

The main advantage of solving a convex problem is the theoretical guarantee to get the global optimum. In the particular case of eigendecomposition for DR, the embedding dimensionality can be adapted very easily without reoptimizing: eigenvectors and eigenvalues are associated with dimensions and can be easily considered or discarded. On the other hand, the main shortcoming of convex optimization is that it usually forces working within a tight frame with little flexibility. Problem statement can be much more flexible when the convexity constraint is relaxed. The price to pay is that non-convex optimization can get stuck in (or close to) a local optimum. The hope is then that the local optimum of a problem stated without convexity constraint proves to be better than the global optimum of a problem where convexity is key in the statement. This remains an open and somewhat subjective debate but, currently, non-spectral (an thus non-convex) nonlinear DR methods seem to outperform spectral embedding [vdMH08]. Non-spectral DR methods typically involve generic optimization techniques that are based on derivatives, like (stochastic) gradient descent.

3.7 Paradigms of DR

Yet another even more obvious way to categorize DR methods consists in distiguishing various principles or paradigms that guide their conception. For instance, PCA is often considered as a method that best preserves variance from the data space to the projection space. But PCA can also be seen as minimizing the error in a reconstruction problem, where data dimensionality is first reduced and then recovered in a sort of encoding and decoding sequence. The same idea drives auto-associative networks (also known as auto-encoders), except that they substitute linear projection and reconstruction operators with nonlinear mappings.

Another important paradigm in DR is the preservation of pairwise proximities, typically in terms dissimilarities or similarities. These can be scalar products, like in classical MDS [Tor52], or distances, such as in nonlinear variants of MDS [Kru64]. Distance preservation can aim at perfect isometry for all distances, focus chiefly on small distances thanks a weighting scheme

[Sam69, DH97, WS06], or keep track of ordinal distance relationships, like in nonmetric MDS [Kru64]. The measured distance can be Euclidean, geodesic (or a graph approximation thereof), some implicitly defined distance defined by a kernel [SSM98, BN02, WS06], like the commute-time distance, or even some other metric.

To put the emphasis on local data structure, the use of similarities seems however more natural than distances. Distances grow and have significantly large values when points get far apart, whereas similarities are negligible in that case and become large when points get close to each other. Most modern successful DR methods, be they spectral or non-convex, rely rather on similarities than dissimilarities, like Laplacian eigenmaps, LLE, SNE, and their variants. Preservation of proximities, particularly in the form of (soft) neighborhoods, is thus the principle that proves to be the most efficient in DR currently.

4 Definitions

This sections defines some mathematical notation and a few concepts that will be used afterwards.

4.1 Scalars, vectors, and matrices

In this chapter, symbols in italic denote scalar numbers, like variable x, index i, parameter α, or constant N. Bold lower-case symbols denote column vectors, like $\mathbf{x} = [x_k]_{1 \leq k \leq M} \in \mathbb{R}^{\mathrm{M}}$, where M is the dimensionality of $\mathbf{x}$, namely, its number of components or entries. The transpose of $\mathbf{x}$ is a row vector noted $\mathbf{x}^T$. Bold upper-case symbols denote matrices, like $\mathbf{A} = [a_{ij}]_{1 \leq i \leq M, 1 \leq j \leq N} \in \mathbb{R}^{\mathrm{M} \times \mathrm{N}}$. Matrix $\mathbf{A}$ can be seen as collection of indexed column vectors, as in $\mathbf{A} = [\mathbf{a}_i]_{1 \leq i \leq N}$. The transpose of $\mathbf{A}$ is $\mathbf{A}^T = [a_{ji}]_{1 \leq j \leq N, 1 \leq i \leq M}$. For two equal-size vectors $\mathbf{u}$ and $\mathbf{v}$, their scalar product is computed as $\mathbf{u}^T\mathbf{v} = \mathbf{v}^T\mathbf{u} = \sum_{k=1}^{M} u_k v_k$. Similarly, the matrix-vector and matrix-matrix products are computed as $\mathbf{y} = \mathbf{A}\mathbf{x} = \left[\sum_{k=1}^{K} a_{ik} x_k\right]_{1 \leq i \leq M}$ and $\mathbf{C} = \mathbf{A}\mathbf{B} = \left[\sum_{k=1}^{K} a_{ik} b_{kj}\right]_{1 \leq i \leq M, 1 \leq j \leq N}$.

4.2 Data sets and coordinates in space

Let $\mathbf{\Xi} = [\boldsymbol{\xi}_i]_{1 \leq i \leq N}$ denote a set of N points in some M-dimensional data space, with M potentially very high. Each $\boldsymbol{\xi}_i$ is a column vector from $\mathbb{R}^{\mathrm{M}}$ holding the coordinates. Similarly, let $\mathbf{X} = [\mathbf{x}_i]_{1 \leq i \leq N}$ be the data set representation in a P-dimensional space, with $P \leq M$. All along this chapter, Greek symbols usually refer to quantities that are relative to the high-dimensional (HD) space, whereas usual latin symbols denote their counterpart in the low-dimensional (LD) representation space.

4.3 Objective function and optimisation

The goal of DR is to arrange points $\mathbf{x}_i \in \mathbb{R}^{\mathrm{P}}$, with P potentially much smaller than M, such that $\mathbf{X}$ is representative of $\mathbf{\Xi}$, namely, its preserves most of salient or relevant structural features. Representativeness of $\mathbf{X}$ with respect to $\mathbf{\Xi}$ is quantified formally by some mathematical function $E(\mathbf{X}; \mathbf{\Xi}, \boldsymbol{\theta})$, termed objective function, cost function, energy function, or stress function. Vector $\boldsymbol{\theta}$ denotes some additional parameters that could modulate the function behavior. Usually, representativeness is made optimal by minimizing the value of E. This is achieved by a clever trial-and-error procedure that progressively updates $\mathbf{X}$, like (stochatistic) gradient descent or spectral decomposition.

4.4 Covariance and Gram matrices

Assuming that the variables in $\mathbf{\Xi}$ and $\mathbf{X}$ are centered, the sample covariance matrices in the HD and LD spaces are $\mathbf{C}_{\mathbf{\Xi}} = \frac{1}{N}\mathbf{\Xi}\mathbf{\Xi}^T$ and $\mathbf{C}_{\mathbf{X}} = \frac{1}{N}\mathbf{X}\mathbf{X}^T$, with sizes M-by-M and P-by-P, respectively. Superscript ^{T} denotes the transpose of a matrix. In product $\mathbf{C} = \mathbf{A}^T\mathbf{B}$, element c_{ij} from $\mathbf{C}$ is computed as $c_{ij} = \sum_k a_{ki} b_{kj}$.

The N-by-N Gram matrices of all pairwise scalar products in the HD and LD spaces are $\mathbf{G}_{\mathbf{\Xi}} = \mathbf{\Xi}^T\mathbf{\Xi}$ and $\mathbf{G}_{\mathbf{X}} = \mathbf{X}^T\mathbf{X}$, respectively. Covariance assesses the linear relationships between two rows of $\mathbf{\Xi}$ or $\mathbf{X}$, i.e., two variables, whereas the Gram matrix assesses pairwise relationships between columns of $\mathbf{\Xi}$ or $\mathbf{X}$, i.e., observations.

4.5 Spectral decomposition

Square symmetric matrix $\mathbf{A}$ can be factorized into $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$, where $\mathbf{V}$ is orthogonal ($\mathbf{V}^T\mathbf{V} = \mathbf{I}$) and $\mathbf{\Lambda}$ is diagonal. The columns $\mathbf{v}_k$ of $\mathbf{V}$ are the eigenvectors of $\mathbf{A}$, while λ_{kk} are their associated eigenvalues.

Let us consider some known, square, symmetric matrix $\mathbf{Y}$ that results from the products $\mathbf{X}\mathbf{X}^T$ or $\mathbf{X}^T\mathbf{X}$, where $\mathbf{X}$ is unknown but has the same size as $\mathbf{Y}$. For such products, all eigenvalues are guaranteed to be non-negative ($0 \leq \lambda_{kk}$). The spectral decomposition would allow us to retrieve $\mathbf{X}$ by computing $\mathbf{X} = \mathbf{V}\mathbf{\Lambda}^{1/2}$ or $\mathbf{X} = \mathbf{\Lambda}^{1/2}\mathbf{V}^T$, respectively, where $\mathbf{\Lambda}$ is diagonal with entries $\lambda_{kk}^{-1/2}$. To some extent, $\mathbf{X}$ can thus be interpreted as the square root of $\mathbf{Y}$.

4.6 Norms and distances

The norm of some vector $\mathbf{u}$ is its length, noted $\|\mathbf{u}\|$. In particular, the Euclidean norm is defined as $\|\mathbf{u}\|_2 = \sqrt{\mathbf{u}^T\mathbf{u}} = \sqrt{\sum_k u_k^2}$. The Frobenius norm extends the Euclidean norm to matrices, namely, $\|\mathbf{A}\|_{\mathrm{F}} = \sqrt{\sum_{ij} a_{ij}^2}$.

The distances between the ith and jth points are defined as $\delta_{ij} = \|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|$ and $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ in the HD and LD spaces, respectively. Most of the time the distances are Euclidean, namely,

$$\delta_{ij} = \|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|_2 = \sqrt{\sum_{m=1}^{M} (\xi_{mi} - \xi_{mj})^2} \quad \text{and} \quad d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2 = \sqrt{\sum_{p=1}^{P} (x_{pi} - x_{pj})^2} \ . \tag{1}$$

Distances are obvious ways to assess proximity between two vectors. For matrices $\mathbf{A}$ and $\mathbf{B}$, the Frobenius distance $\|\mathbf{A} - \mathbf{B}\|_{\mathrm{F}}$ can be used.

5 DR paradigms and emblematic methods

5.1 Variance preservation

The main representent in this category is principal component analysis (PCA), detailed here below. Many variants of PCA exist, like minor component analysis (focusing on axes with the least variance) or linear discriminant analysis (favoring axes that best separate classes in labeled data).

5.1.1 Principal component analysis

The intuition behind PCA can be traced back to a multivariate regression problem. Regression is a supervised learning task in which a model is trained to predict outputs $\mathbf{y}$ from inputs $\mathbf{x}$. While scalar outputs y_i are often considered, nothing prevents us to have multiple outputs like here in vector $\mathbf{y}$. The model can be denoted by some parametric function $\hat{\mathbf{y}} = f_\theta(\mathbf{x})$, where $\hat{\mathbf{y}}$ is the predicted output. Performance is typically measured in terms of squared residues (or least squares) through $\|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2$. For a complete data set ($\mathbf{X} = [\mathbf{x}_i]_{1 \leq i \leq N}, \mathbf{Y} = [\mathbf{y}_i]_{1 \leq i \leq N}$), the mean squared error is then

$$E(\theta; \mathbf{X}, \mathbf{Y}) = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2 = \|\mathbf{Y} - \hat{\mathbf{Y}}\|_{\mathrm{F}}^2 \ . \tag{2}$$

Let us now consider an apparently trivial problem where only input coordinates are available. Instead of having both $\mathbf{X}$ and $\mathbf{y}$, we only have $\mathbf{\Xi}$ and no counterpart for $\mathbf{y}$. In this case, we would like to approximate $\mathbf{\Xi}$ itself, which is expressed as

$$E(\theta; \mathbf{\Xi}) = \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{\xi}_i - \hat{\boldsymbol{\xi}}_i\|_2^2 = \|\mathbf{\Xi} - \hat{\mathbf{\Xi}}\|_{\mathrm{F}}^2 \ , \tag{3}$$

where $\hat{\boldsymbol{\xi}} = f_\theta(\boldsymbol{\xi})$. A trivial solution to this reconstruction problem consists in choosing the identity function for f_θ. To avoid triviality, let us assume that f_θ consists actually of two successive functions e_θ and d_θ, namely $\hat{\boldsymbol{\xi}} = d_\theta(e_\theta(\boldsymbol{\xi}))$, with $\mathbf{x} = e_\theta(\boldsymbol{\xi})$ and $\hat{\boldsymbol{\xi}} = d_\theta(\mathbf{x})$, with the constraint that dimensionality of $\mathbf{x}$ is lower than that of $\boldsymbol{\xi}$.

Figure 3 illustrates this setup. In this figure, the input vector $\boldsymbol{\xi}$ here has three dimensions. The intermediate layer $\mathbf{x}$ is two-dimensional. Each straight line joining two squares is endowed with a

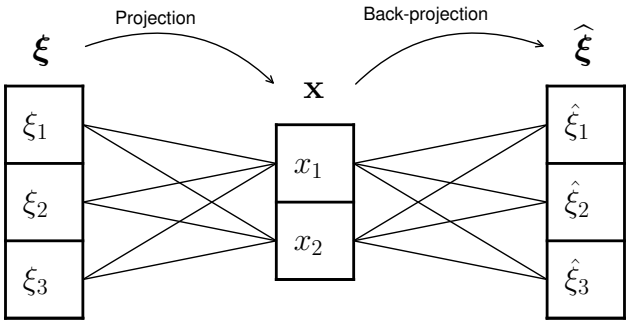


Figure 3: Illustration of PCA seen as a multivariate regression process.

particular coefficient (not shown). When several straight lines join together at the left side of some squares, it means that the corresponding component is obtained by adding up the components linked by the lines, weighted by their coefficients. Hence x_1 and x_2 are linear combinations of ξ_1, ξ_2 and ξ_3, and $\hat{\xi}_1$ to $\hat{\xi}_3$ are linear combination of x_1 and x_2. The set of straight lines joining $\boldsymbol{\xi}$ (resp. $\mathbf{x}$) to $\mathbf{x}$ (resp. $\hat{\boldsymbol{\xi}}$) encode e_θ (resp. d_θ) as a linear transformation. The whole network tries to reproduce the identity function, under the constraint of the bottleneck $\mathbf{x}$. The dimensionality reduction is performed by projecting $\boldsymbol{\xi}$ to $\mathbf{x}$.

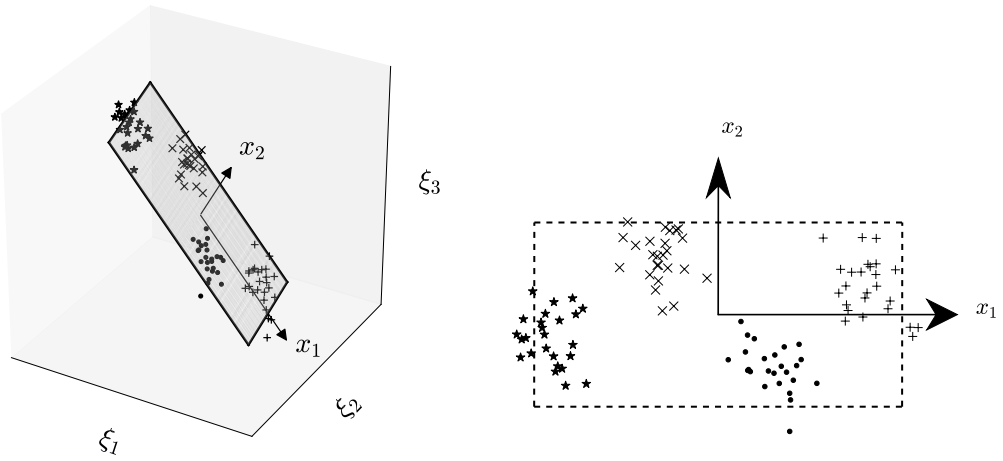
In PCA, functions e_θ and d_θ are thus chosen to be simple linear transformations, which somehow correspond to changes of coordinate systems. More precisely, only shifts, rotations, and mirroring operations are considered. This amounts to considering some linear subspace of the data space, like in Fig. 4. The axes $[x_1, x_2]$ define a 2D subspace of the 3D data space $[\xi_1, \xi_2, \xi_3]$. They are positioned such that projection residues are minimal in (3). In other words, PCA finds a linear projection subspace that maximizes proximity between original data and their reconstruction after projection. If rotations (and mirroring) are centered on the data set average, then the sum of squared residues can also be interpreted as a variance, computed *perpendicularly* to the $\mathbf{x}$ subspace, following the residue lines. Let us call it *lost variance*. Under the same assumption, variance can of course also be computed *along* the $\mathbf{x}$ axes, within the considered subspace. Let it be the *preserved variance*. Because the projection and backprojection then consist of rotations only, scales remain unchanged in all directions and the total variance, which is computed as the sum of the lost and preserved variances, remains constant. Rotating the subspace only transfer variance from lost to preserved or the other way round, like communicating vessels. PCA chooses the subspace with both minimal reconstruction residues and maximum projected variances.

In practice, PCA can be computed in several ways, with a first common step, which is always data centering. Subtracting the data set mean can indeed be proved necessary to minimize residues. The mean can always be added back after reconstruction. To compute the optimal rotation, one can either use the singular value decomposition of the centered data coordinates or the EVD of the covariance matrix. As a reminder, the sample covariance is written as $\frac{1}{N}\boldsymbol{\Xi}\boldsymbol{\Xi}^T$ if $\boldsymbol{\Xi}$ is centered (i.e., $\boldsymbol{\Xi}\mathbf{1}^T = \mathbf{0}$). The SVD of the centered data set is $\boldsymbol{\Xi} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T$. The P-dimensional PCA projection is then given by $\mathbf{X} = \mathbf{U}_P^T$, namely, the first P columns of $\mathbf{U}^T$. The EVD of the sample covariance is $\frac{1}{N}\boldsymbol{\Xi}\boldsymbol{\Xi}^T = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$. The P-dimensional PCA projection is then given by $\mathbf{X} = \mathbf{V}_P^T\boldsymbol{\Xi}$, where $\mathbf{V}_P^T$ gathers the eigenvectors associated with the largest P eigenvalues. Reconstruction is carried out by $\hat{\boldsymbol{\Xi}} = \mathbf{V}_P\mathbf{X}$.

PCA's main limitation is the linearity of the projection and back-projection it involves, which is suited only to linear dependencies among data features. Nonlinear extensions can rely on a discrete, nonlinear subspace, like the self-organizing maps in Section 5.3.1 or nonlinear encoders and decoders implemented with artificial neural networks, like auto-encoder networks in Section 5.3.2

5.2 Distance preservation

Hereafter are presented various flavors of distance-preserving methods, after a short detour by classical metric multidimensional scaling. This method makes the transition between PCA and



(a) (b)

Figure 4: Illustration of PCA as a dimensionality reduction method finding the linear subspace preserving the most variance of the original data.
Figure 4a: Three-dimensional data space $[\xi_1, \xi_2, \xi_3]$. The data points are illustrated by dots. x_1 and x_2 are the first two principal components of the data set. This means that the grey shaded plane, defined by axes x_1 and x_2, is the linear subspace of dimension two preserving the most variance of the data points in three dimensions, once orthogonally projected on the plane.
Figure 4b: Two-dimensional embedding obtained, which simply consists in the grey shaded plane of Figure 4a, on which the data points were projected.

distance preservation. It relies on pairwise scalar products, which are mathematically related to both covariances and Euclidean distances.

5.2.1 Classical metric multidimensional scaling

Compared to PCA, the idea behind classical metric multidimensional scaling (CMDS) is that data should be described by quantities characterizing the proximity relationship for pairs of data, rather than the proximity between pairs of variables, like covariances.

The simplest pairwise quantity that can be defined is the scalar product (or dot product, or inner product). For data $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$, the usual scalar product can be written as $\boldsymbol{\xi}_i^T \boldsymbol{\xi}_j = \boldsymbol{\xi}_j^T \boldsymbol{\xi}_i = \sum_k \xi_{ki} \xi_{kj}$. The interpretation of scalar product is not very intuitive and relies mainly on angles. If the scalar product between $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$ is null, then they are 90 degrees from each other with respect to the origin $\mathbf{0}$. If the scalar product is positive (resp. negative), then the angle is greater (resp. lower) than 90 degrees. For N data vectors in $\boldsymbol{\Xi}$, all pairwise scalar products can be gathered in a symmetric N-by-N matrix called Gram matrix: $\mathbf{G}_{\boldsymbol{\Xi}} = \boldsymbol{\Xi}^T \boldsymbol{\Xi}$.

As in PCA, the data set $\boldsymbol{\Xi}$ can be assumed to be centered. If it is not, then its mean $\frac{1}{N}\boldsymbol{\Xi}\mathbf{1} = \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{\xi}_j$ can be subtracted from each $\boldsymbol{\xi}_i$. After centering, we have $\sum_{i=1}^{N} \boldsymbol{\xi}_i = \mathbf{0}$.

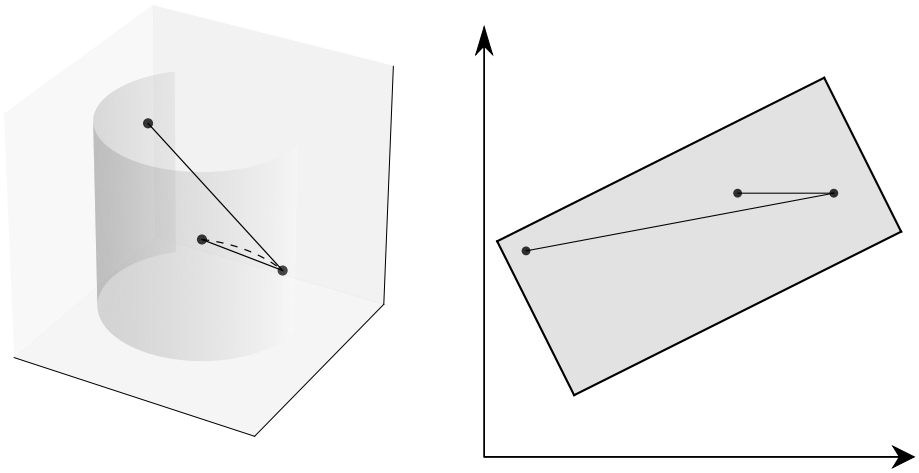
The idea of CMDS is to reproduce as well as possible in the LD space the dot products that are observed in the HD space. Formally, this can be written as $\min_{\mathbf{X}} \|\mathbf{G}_{\boldsymbol{\Xi}} - \mathbf{G}_{\mathbf{X}}\|_{\mathrm{F}}^2 = \min_{\mathbf{X}} \|\mathbf{G}_{\boldsymbol{\Xi}} - \mathbf{X}^T\mathbf{X}\|_{\mathrm{F}}^2$, which is usually called the *strain* function. The solution relies on positive semidefiniteness of Gram matrices and the eigenvalue decomposition $\mathbf{G}_{\boldsymbol{\Xi}} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$, where $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ and $\boldsymbol{\Lambda}$ is diagonal. We have $\min_{\mathbf{X}} \|\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T - \mathbf{X}^T\mathbf{X}\|_{\mathrm{F}}^2$. For target dimensionality P, the solution is then $\mathbf{X} = \boldsymbol{\Lambda}_P^{1/2}\mathbf{V}_P^T$. Subscript P indicates that that we keep only the first P columns of $\mathbf{V}$ and the first P rows and columns of $\boldsymbol{\Lambda}$.

What can be done if $\boldsymbol{\Xi}$ is not centered? Then, the centering matrix $\mathbf{C} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$ can be used. It is easy to see that $\mathbf{X}\mathbf{C} = \mathbf{X}(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T) = \mathbf{X} - (\frac{1}{N}\mathbf{X}\mathbf{1})\mathbf{1}^T$ is centered, since the data set mean $\frac{1}{N}\mathbf{X}\mathbf{1}$ is explicity subtracted from all columns. The main interest of the centering matrix is that it can be applied directly to the Gram matrix. Double centering consists in computing $\mathbf{C}\mathbf{G}_{\boldsymbol{\Xi}}\mathbf{C} = \mathbf{C}^T\boldsymbol{\Xi}^T\boldsymbol{\Xi}\mathbf{C}$ (since $\mathbf{C}$ is symmetric and invariant under transpose).

Centering is essential here since dot products are seen as a form of angular proximity measure with respect to the origin $\mathbf{0}$. If the data set is not centered, the origin might lie far away and

then all angles would be small. On the other hand, if data is centered, then angles are likely to vary between 0 and 180 degrees, leading therefore to a more discriminative similarity. Centering also reinforces the duality between PCA and CMDS. For centered data, one might demonstrate that the eigenvalues and eigenvectors of either the covariance matrix $\frac{1}{N}\boldsymbol{\Xi}\boldsymbol{\Xi}^T$ in PCA or the Gram matrix $\boldsymbol{\Xi}^T\boldsymbol{\Xi}$ in CMDS are closely related; the key to this duality is to look at the SVD of $\boldsymbol{\Xi}$ after centering.

Eventually, it is worth noticing that CMDS can also be applied to distance matrices as depicted in Figure 5, in addition to Gram matrices or raw centered coordinates. Let $\boldsymbol{\Delta}^2 = [\delta_{ij}^2]_{1\leq i,j\leq N}$ denote a matrix of squared pairwise distances. Then, double centering on $\boldsymbol{\Delta}^2$, namely, $\mathbf{C}\boldsymbol{\Delta}^2\mathbf{C}$, can be shown to yield a valid Gram matrix (provided δ_{ij} has all properties of a genuine distance). This can be easily seen in the case of the squared Euclidean distance, precisely defined with respect to scalar products: $\delta_{ij}^2 = \|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|_2^2 = \boldsymbol{\xi}_i^T\boldsymbol{\xi}_i - 2\boldsymbol{\xi}_i^T\boldsymbol{\xi}_j + \boldsymbol{\xi}_j^T\boldsymbol{\xi}_j$. (Notice that distances are shift invariant and thus remain the same whether data are centered or not.) The effect of double centering is then to substract the left and right terms, leaving only $-2\boldsymbol{\xi}_i^T\boldsymbol{\xi}_j$. Multiplying further by $-\frac{1}{2}$ yields the elements of the Gram matrix.



(a) (b)

Figure 5: Illustration of classical metric multidimensional scaling. Figure 5a: Three-dimensional space, in which a two-dimensional subspace corresponding to a half cylinder is embedded. Three data points (black dots) are lying on this subspace. The two solid black lines depict the Euclidean distances between two dots and the third one. The dashed line is the shortest path on the half cylinder between the two joined data points. Figure 5b: Two-dimensional embedding. The low-dimensional representation of the data points aims to preserve as much as possible the pairwise distances computed between their high-dimensional counterparts.

5.2.2 Stress-based metric multidimensional scaling

In its classical form, multidimensional scaling involves scalar products to characterize the (angular) proximities between data. Usual scalar products have a strong relationship with squared Euclidean distances, namely, $\delta_{ij}^2 = \|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|_2^2 = \boldsymbol{\xi}_i^T\boldsymbol{\xi}_i - 2\boldsymbol{\xi}_i^T\boldsymbol{\xi}_j + \boldsymbol{\xi}_j^T\boldsymbol{\xi}_j$. From that perspective, CMDS can thus be seen not only as direct way to preserve scalar product ($\min_{\mathbf{X}} \|\boldsymbol{\Xi}^T\boldsymbol{\Xi} - \mathbf{X}^T\mathbf{X}\|_{\mathrm{F}}$), but also as an indirect, approximate way to preserve squared Euclidean distances. If $\boldsymbol{\Delta}^2 = [\delta_{ij}^2]_{1\leq i,j\leq N}$ and $\mathbf{D}^2 = [d_{ij}^2]_{1\leq i,j\leq N}$ denote the matrices of pairwise squared Euclidean distances, then squared distance preservation would be written as $\min_{\mathbf{X}} \|\boldsymbol{\Delta}^2 - \mathbf{D}^2\|_{\mathrm{F}}$. Notice that this problem can no longer be solved with spectral techniques like eigenvector decomposition. Instead, more generic optimisation techniques are needed, typically based on derivatives, like gradient descent. Another side effect of this change, perhaps even more important, is that the problem is no longer linear. One cannot guarantee anymore that the solution in $\mathbf{X}$ results from a linear transformation of $\boldsymbol{\Xi}$ as in PCA and CMDS.

Considering distances instead of angular proximity proves to be easier for the intuition, short

distances translating immediately the notion of proximity. Starting from there, MDS could be extended to address the following cases:

- How raw (non-squared) Euclidean distances or even non-Euclidean distances can be preserved?
- How to put more emphasis on some pairwise relationships than on others?

The goals here are to broaden the applicability of MDS to other distances and to refine its objective, when for instance short distances (local structure) are deemed more relevant than longer ones (global structure). For this purpose, let us get rid of matrix notation and rewrite squared distance preservation as $\min_{\mathbf{X}} \|\mathbf{\Delta}^2 - \mathbf{D}^2\|_{\mathrm{F}} = \sum_{i,j=1}^{N} (\delta_{ij}^2 - d_{ij}^2)^2$. This cost function is called *sstress*, standing for squared stress. This formula involves distances to the 4th power, meaning that large distances have much heavier weight than short ones. At the same time, short distances are those that describe local proximities and neighbourhoods. To make them gain some importance, a first workaround consists in using raw, non-squared distances. The resulting *stress* function can be written as $\min_{\mathbf{X}} \sum_{i,j} (\delta_{ij} - d_{ij})^2$. Another workaround is to introduce explicit weights, as in $\min_{\mathbf{X}} \sum_{i,j} w_{ij} (\delta_{ij} - d_{ij})^2$, where $0 \leq w_{ij}$. An emblematic choice for the weights is $w_{ij} = 1/\delta_{ij}$, like in Sammon's nonlinear mapping [Sam69]. Such weights explicitly grant short distances more importance. For example, twice longer a distance has half the weight. Notice that weights and importance are determined here by δ_{ij}, the distance in the HD data space. The weighting scheme can be changed to take into account the distance in the LD representation space instead, leading to a method called curvilinear component analysis (CCA) [DH97]. The weight is then typically written as $w_{ij} = H(\lambda - d_{ij})$, where $H(u)$ is Heaviside's step function (null if $u < 0$, one elsewhere) and λ is parameter controlling the radius of the neighbourhood around each $\mathbf{x}_i$. CCA is more flexible than Sammon's mapping, in the sense that if d_{ij} gets larger than λ, then the weight gets null and the corresponding term in the cost function vanishes. In other words, CCA tolerates some distance stretching, whereas Sammon cannot. Distance stretching is useful to unfold a curved manifold or even to tear apart some piece of the manifold when necessary (typically to break loop structures [LV04] like circles and sphere). A planisphere of the Earth is the typical example where this property comes in very handy.

Eventually, distance preservation has limitations when it comes to reduce very high-dimensional data (tens of dimensions) to very low-dimensional representations (typically, 2D), because distances have completely different statistical properties in those high- and low-dimensional spaces. Distances are said to concentrate in HD spaces. This concentration gap makes the direct comparison of distances, as involved in stress-based MDS, hardly relevant. This issue is addressed partly by non-metric MDS, hereafter, and more thoroughly by methods of stochastic neighbour embedding, as described further in Section 5.5.

5.2.3 Non-metric multidimensional scaling

An even more evolved form of MDS is called non-metric or ordinal MDS. In this variant, distance preservation is expressed as $\min_{\mathbf{X},f} \sum_{i,j} (f(\delta_{ij}) - d_{ij})^2$, where f is a growing function (or monotonic, with a positive derivative). In this problem, d_{ij} is not told to match δ_{ij} as such. Instead, d_{ij} should reproduce δ_{ij} up to transofrmation f. Since f is increasing, we have that $\delta_{ij} \leq \delta_{kl}$ implies that $f(\delta_{ij}) \leq f(\delta_{kl})$. Introducing f in the cost function of MDS somehow relaxes distance preservation: instead of requiring strict one-to-one equalities $d_{ij} = \delta_{ij}$ and $d_{ik} = \delta_{ik}$, nonmetric MDS merely enforces ordinal relationships in triplets (i, j, k), namely, $d_{ij} = f(\delta_{ij})$, $d_{ik} = f(\delta_{ik})$, together with $\delta_{ij} \leq \delta_{ik}$ imply that $d_{ij} \leq d_{ik}$.

In nonmetric MDS, both $\mathbf{X}$ and f need to be optimized and identified. In particular, f is obtained with a technique called isotonic regression.

5.3 Neural approaches of DR with data reconstruction

In this subsection, we present DR methods based on artificial neural networks.

This type of methods are inspired by biological neural networks. The main principles of neural networks are detailed as following.

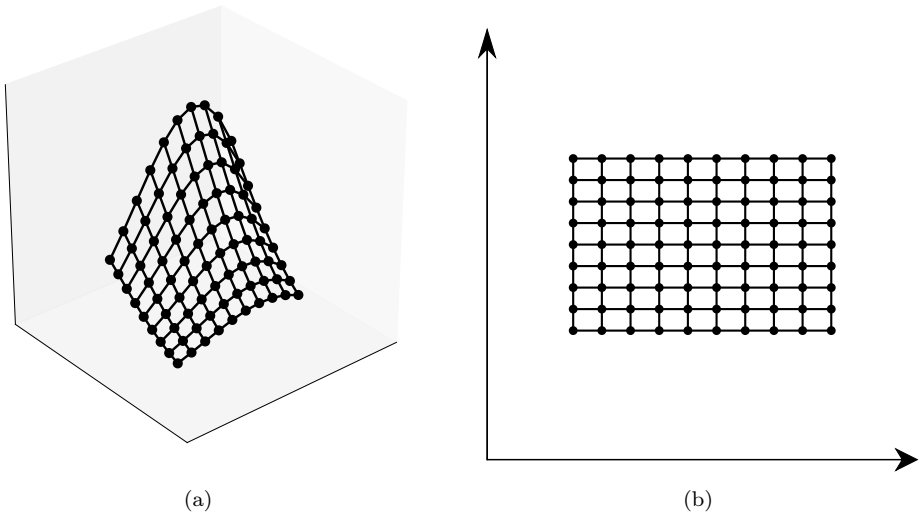
An artificial neural network (ANN) is a set of nodes, which share connections. In classic ANN, a node is constituted with two functions:

- A combination function, which calculates a value with the input nodes and the weights of the connections.
- An activation function, which is applied to the value calculated with the combination function and transfered to the following layer of nodes.

A learning phase identifies the optimal weights of the synaptic connections with optimization methods. In a first part, we detail the self-organizing maps (Section 5.3.1). Secondly, we present the auto-associative neural networks (Section 5.3.2).

5.3.1 Self-organizing maps

Linear projection with PCA or classical MDS consists in fitting a plane through the data cloud and then projecting data onto it. If data is distributed near a nonlinear manifold, a possible extension of PCA consists in replacing the plane with a nonlinear structure. In the case of the SOM, the plane is usually restricted to a rectangular, articulated grid of neurons, which can deform more or less freely, like in Fig. 6. Each neuron has low-dimensional (LD) coordinates as a grid node



(a) (b)

Figure 6: Illustration of the concept of self-organizing maps. Figure 6a depicts a three-dimensional space in which a rectangular self-organizing map is folded according to a curved two-dimensional manifold. Figure 6b presents the corresponding low-dimensional grid.

and high-dimensional (HD) coordinates in the data space. The low-dimensional space is given by grid coordinates $\mathbf{G} = [\mathbf{g}_i]_{1 \leq i \leq N}$ and distances $d(\mathbf{g}_i, \mathbf{g}_j)$. The user fix them a priori, typically on a regular rectangular grid. In the high-dimensional space, the grid coordinates are denoted by $\mathbf{\Gamma} = [\boldsymbol{\gamma}_i]_{1 \leq i \leq N}$; they are adjusted as follows. The available data set is traversed several times, each traversal being called an epoch. At each epoch, data vectors $\boldsymbol{\xi}_k$ are presented one by one. For each of those, the index of the closest grid node is denoted by $j = \arg\min_i \|\boldsymbol{\xi}_k - \boldsymbol{\gamma}_i\|_2$. The HD coordinates of all grid nodes are then updated according to $\boldsymbol{\gamma}_i \leftarrow \boldsymbol{\gamma}_i + \alpha K\left(\frac{d(\mathbf{g}_i, \mathbf{g}_j)}{\lambda}\right)(\boldsymbol{\xi}_k - \boldsymbol{\gamma}_i)$, where K is a decreasing function from $\mathbb{R}^+$ to $\mathbb{R}^+$. Function K can be a flipped step function (one below some threshold λ, zero beyond) or a Gaussian function with standard deviation λ. Parameter λ can be interpreted as a neighborhood radius in the grid space. Parameter α is a so-called learning rate and determines the neural plasticity, namely, the strength of the influence of any data vector $\boldsymbol{\xi}_k$ on a neuron. In other words, high α gives low inertia to neurons, which are then easily attracted by data vectors, in a kind of Brownian motion in time. Parameter λ controls the influence of the LD grid. High λ means that the winning neuron together with its neighbors will be attracted by the data points. Reducing λ decreases this effect and neurons then become loosely connected and almost independent of each other, nearly overlooking their respective location in the grid. In practice, both α and λ decrease in time from epoch to epoch. The schedule of their respective decrease must be carefully adjusted for strong attractions and cohesions to occur in the

beginning, to properly fit and unfold the SOM in the data space, while weaker attractions and cohesions allow for finer, smaller-scale adjustment later on.

Self-organizing maps have been developed mostly in a heuristic way and only a few attempts have been made to formulate their underlying cost function [Hes01]. The main intuition behind SOMs is to ensure consistency between neighborhoods in both spaces (LD grid and HD data), through the joint attraction of the closest neuron to a data vector and its neighbors. Such cohesion makes that grid neighbors of a neuron are also very likely to be its neighbors in the data space. In that sense, a SOM allows for visualizing proximities among data. The use of neurons, typically in smaller number than the data points, also relates SOMs to vector quantization and methods like competitive learning [RZ85]. Neurons will distribute themselves within the data cloud so as to best represent them.

The use of neurons with a preset position in a grid makes the SOM an outlier in the field of dimensionality reduction. Instead of building a LD scatterplot of N points, representative of the N HD data vectors, the SOM goes the other way around, by fitting a predefined LD structure within the HD data distribution. From a visualization standpoint, the SOM proceeds by fetching from the HD data space some local properties of interest and displays it on the grid representation. Such local property can be a class or cluster label encountered near the neuron, the local density, or simply some component of $\boldsymbol{\xi}_k$. For an illustration of SOM application on the well known iris dataset, see [VHA+99].

5.3.2 Auto-associative neural networks

Auto-associative networks [Kra91, Oja91, UNN91, DC93, HS06] are another neural approach to the problem of dimensionality reduction and, like the SOMs, they can be seen as a nonlinear extension of PCA. By fitting a plane through the data cloud, PCA seeks variance preservation and, hence, minimal loss of variance after projection on the plane. This also means minimal residues between the data points and their projected counterparts. PCA identifies the linear transformation (and its pseudo-inverse) that minimize these residues. Auto-associative networks replace the linear transformation with a nonlinear one. More specifically, artificial neural networks can approximate any function provided their architecture be adapted to the problem complexity. Complicated functions typically require more neurons and layers [Ben09, LBH15]. Figure 7 show a typical auto-associative network, with a succession of layers, among which the central narrower one forms a bottleneck and gives thus the whole network an hourglass shape. Auto-association means that input data ($\boldsymbol{\xi}$) must be closely reproduced in the outputs ($\hat{\boldsymbol{\xi}}$), with minimal residues, with the difficulty of passing through a central layer ($\mathbf{x}$) with fewer dimensions than the inputs and outputs. Data flowing through the network must therefore be first compactly encoded and then decoded back to its initial form, hence the alternative name auto-encoder. Dots in the figure represent the neurons. They combine scalar products (like the linear projection in PCA), bias terms (extending mean subtraction in PCA), and a nonlinear function (unlike PCA). Nonlinearities are typically sigmoid functions or rectified linear units (ReLU), namely piecewise linear functions. The difficulty of using neural networks in an auto-associative setting is that it requires a 'deep' network, that is, with many intermediate layers, so that both halves, the encoder and decoder, are complex enough to approximate nonlinear functions. Parameters in those layers (synaptic weights) are adjusted by training the network, which is achieved by presenting small batches of input and output pairs. The reconstruction error associated with these batches is then computed, as well as its gradient with respect to the synaptic weights. Whereas inputs flow forward along the network to compute the corresponding outputs, gradient information goes back from output layers to input layers, in a process called gradient back-propagation, where synaptic weight are tuned to improve auto-association. Depth makes networks more difficult to train. In particular, gradient back-propagation becomes critical because the derivative chain-rule then involves many factors, with significant probability of observing small factors or, conversely, very large ones, leading to the dual phenomena of gradient vanishing and gradient explosion. New learning techniques and network architectures can partly mitigate these risks. Nevertheless, to date, auto-encoders are still outperformed by DR methods like stochastic neighbour embedding, at least in visualisation tasks where the representation dimension must be very low. Auto-encoders keep the advantage of being fully parametric methods, in both ways (dimension reduction and expansion back). Also, proximity is entailed in both the reconstruction error and neural network smoothness (similar HD inputs yield similar LD representations, like with the topographic mapping in SOMs).

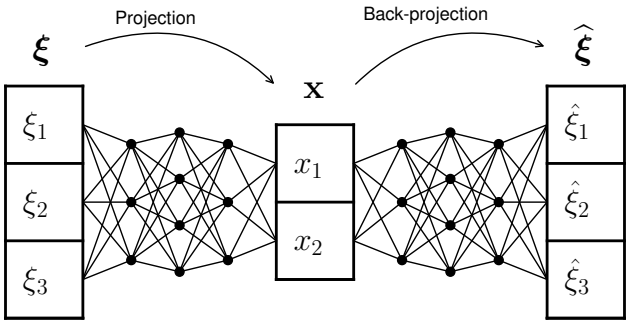


Figure 7: Illustration of auto-associative neural networks. The network is very similar to Figure 3 except that instead of directly joining two layers, the straight lines meet at some black dots, which represents nonlinear activation functions of the linear combinations at their inputs. Hence auto-associative neural networks are similar to PCA but model the transformations e_θ and d_θ as nonlinear functions.

5.4 Spectral methods

5.4.1 Kernel PCA

Until here, various generalisations of PCA take inspiration from geometry (scalar product or distance preservation in MDS, articulated grid instead of linear subspace in SOMs) or function approximation (neural network as universal approximators instead of linear projections). Kernel PCA extends PCA from a more theoretical standpoint, by generalizing the notion of scalar product. Strictly speaking, kernel PCA [SSM98, Wil01] is thus a nonlinear extension of classical MDS, the dual of PCA, acting on the Gram matrix of scalar products instead of the covariance matrix.

Intuitively, kernel PCA replaces the usual scalar products across all pairs of raw data in the Gram matrix with a scalar product in a so-called feature space, that is, after a nonlinear transformation of data. The surprising trick here is that there is no need to carry out these two steps (mapping to the feature space, then scalar product computation) explicitly. Instead, a theoretical result allows one to compute the new Gram matrix at once, starting from the regular one, without specifying the nonlinear mapping to the feature space.

Function k from $\mathbb{R}^{\mathrm{M}} \times \mathbb{R}^{\mathrm{M}}$ to $\mathbb{R}$ is said to be a kernel if it is symmetric ($k(\mathbf{a}, \mathbf{b}) = k(\mathbf{b}, \mathbf{a})$) and positive semidefinite, that is, fulfilling Mercer's condition:

$$\int \int g(\mathbf{a}) k(\mathbf{a}, \mathbf{b}) g(\mathbf{b}) \, \mathrm{d}\mathbf{a} \, \mathrm{d}\mathbf{b} \geq 0 \ , \tag{4}$$

where g is any square-integrable function. Then, kernel k has the following property:

$$k(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j) = \boldsymbol{\phi}(\boldsymbol{\xi}_i)^T \boldsymbol{\phi}(\boldsymbol{\xi}_j) \ , \tag{5}$$

where $\boldsymbol{\phi} : \ \mathbb{R}^{\mathrm{M}} \to \mathcal{F}, \ \boldsymbol{\xi} \mapsto \boldsymbol{\phi}(\boldsymbol{\xi})$ is an unknown mapping from the initial data space to some feature space $\mathcal{F}$. The unknown mapped coordinates in $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\boldsymbol{\xi}_i)]_{1 \leq i \leq N}$ are involved in matrix $\mathbf{K} = [k(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j)]_{1 \leq i,j \leq N} = \boldsymbol{\Phi}^T \boldsymbol{\Phi}$, which is then a symmetric and positive semidefinite Gram matrix by construction. Here, the specificity is that $\mathbf{K}$ is built directly, using kernel k, without computing the scalar products $\boldsymbol{\phi}(\boldsymbol{\xi}_i)^T \boldsymbol{\phi}(\boldsymbol{\xi}_j)$. In other words, we use the left-hand side of the kernel definition, without bothering about the right-hand side. The choice of k, beforehand, implicitly fixes ϕ, whereas the usual approach would have been to select ϕ first and then get the kernel afterwards.

As soon as a Gram matrix is available, classical MDS can be applied in order to get a P-dimensional projection, from the unknown feature space to some low-dimensional linear subspace. To carry out MDS, data must be centered. The apparent difficulty here is that centering must be performed on $\boldsymbol{\Phi}$ in $\mathcal{F}$, which is unknown. At this point, the magic of double centering enters into play. Computing $\mathbf{C}^T \mathbf{K} \mathbf{C}$, where $\mathbf{C} = \mathbf{I} - \frac{1}{N} \mathbf{1}\mathbf{1}^T$ is the centering matrix, allows us to center $\boldsymbol{\Phi}$ without knowing it, by acting on $\mathbf{K}$ instead of $\boldsymbol{\Phi}$. Eventually, eigenvalue decomposition $\mathbf{C}^T \mathbf{K} \mathbf{C} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^T$ yields $\boldsymbol{\Phi} = \boldsymbol{\Lambda}^{1/2} \mathbf{U}^T$. Keeping then only the eigenvectors associated with the largest

P eigenvalues gives us a projection from the feature space $\mathcal{F}$ to a P-dimensional representation space. Although this projection is linear, the use of kernel function k induces a nonlinear mapping ϕ. Kernel PCA can then be seen as two-step procedure, with first a nonlinear mapping from the data space to the feature space and then a linear projection from the feature space to the LD representation.

In practice, the central question raised but unanswered by kernel PCA is the choice of a particular kernel. Theory determines which kernels are admissible, like polynomial functions of the raw scalar product or a radial Gaussian function, but it does not state which ones are useful in a specific task. Hence, the main theoretical advantage of kernel methods, namely, the fact that they induce a nonlinear mapping implicitly, can also be their main drawback, since the properties or behaviour of the induced mapping are difficult to investigate. For instance, some kernels induce a mapping towards an infinite-dimensional space, which turns out to be counter-productive in the context of dimensionality reduction, whereas it is useful in clustering or classification. This is the reason why kernel PCA is hardly used as an effective DR method. On the other hand, it has provided a theoretical background for the development of other spectral DR methods, where the kernel is purposefully chosen or even data-driven. Isomap [TdSL00a] can be cast within this framework. Instead of considering Euclidean distances and the associated usual scalar product, it relies on the concept of geodesic distances, which somehow can be interpreted as a manifold-aware metric. The geodesic distance is measured not as the crow flies, but rather following the underlying manifold, as the ant would crawl. In practice, the geodesic distances are approximated with shortest paths in a graph of K-ary neighbourhoods. As a consequence, this is no longer a true kernel, although the method works fine with minor workarounds. In contrast, the kernel property is carefully preserved in a closely related method, namely, maximum variance embedding (MVU, formerly known as semidefinite embedding, or SDE) [WSS04, WS06]. In this method, the kernel is not fixed arbitrarily in advance, nor precomputed from a graph as in Isomap, it gets continuously updated and optimised under constraints with a technique called semidefinite embedding. This technique updates the Gram matrix and ensures that it remains valid. All these DR methods end by applying CMDS to a modified, 'kernelized' Gram matrix, whose dominant eigenvectors yield an embedding.

5.4.2 Laplacian eigenmaps

Proximity preservation, spectral embedding, and kernelised Gram matrix are the key elements of another DR method, called Laplacian eigenmaps. The name Laplacian comes from a matrix operator in the domain of graph theory. The idea is that edges or connections in a graph indicate vertices or data that are similar, whereas non-connected items would be more dissimilar. Following the intuition of previous DR methods, one might want to find a low-dimensional representation of the graph where similar, connected vertices are close to each other, whereas non-connected ones are lying farther away.

Formally, let us consider a non-directed graph, with N vertices, and connections among those. The connections can be encoded in an adjacency matrix

$$\mathbf{A} = [a_{ij}]_{1 \leq i,j \leq N} \text{ where } a_{ij} = \begin{cases} 1 & \text{if } j \text{ and } i \text{ are connected} \\ 0 & \text{otherwise} \end{cases} . \tag{6}$$

Usual ways to define the adjacency matrix when starting from data vectors $\boldsymbol{\xi}_i$ in $\boldsymbol{\Xi}$ are to connect either the closest K neighbours of $\boldsymbol{\xi}_i$ or all neighbours lying within an ϵ ball around $\boldsymbol{\xi}_i$.

If the graph edges are weighted, then a weight matrix can be defined, too. For instance, we could have $\mathbf{W} = \mathbf{A}$ or

$$\mathbf{W} = [w_{ij}]_{1 \leq i,j \leq N} \text{ where } w_{ij} = \begin{cases} \exp(-\delta_{ij}^2/(2\sigma^2)) & \text{if } j \text{ and } i \text{ are connected} \\ 0 & \text{otherwise} \end{cases} , \tag{7}$$

where δ_{ij} is the distance between data vectors $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$, and σ is some bandwidth. The value of σ should be such that the Gaussian is already close to zero for the farthest connected neighbours.

Having a weight matrix for the neighbours, an embedding $\mathbf{X}$ of $\boldsymbol{\Xi}$, in which neighbours are represented close to each other, can be found by minimizing

$$E(\mathbf{X}; \mathbf{W}) = \sum_{i,j=1}^{N} w_{ij} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|_2^2 . \tag{8}$$

A trivial solution to this problem is to map all data to a single point, that is, to have $\mathbf{x}_i = \mathbf{x}_j$ for all i and j, regardless whether they are connected or not. To take into account the fact that non-neighours should be represented separately, let us assume that $\mathbf{X}$ is centered ($\mathbf{X1} = \mathbf{0}$) and constrain its covariance matrix $\frac{1}{N}\mathbf{X}\mathbf{X}^T$ to be diagonal and full rank. This way, dimensions in $\mathbf{X}$ are decorrelated with nonzero variance. In other words, some spread of the embedding gets guaranteed.

Another observation is that the above problem can be reformulated using the Laplacian matrix, defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, with $\mathbf{D} = \mathrm{diag}(\mathbf{W1})$. The Laplacian matrix allows us to write

$$E(\mathbf{X}; \mathbf{W}) = \sum_{i,j=1}^{N} w_{ij} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 = \mathrm{Tr}(\mathbf{X}^T\mathbf{L}\mathbf{X}) \ . \tag{9}$$

Then, the solution that minimizes the weighted distances between neighbours while preserving some spread is $\arg\min_{\mathbf{X}} E(\mathbf{X}; \mathbf{W})$ such that $\mathbf{X1} = \mathbf{0}$ and $\mathbf{X}^T\mathbf{D}\mathbf{X} = \mathbf{I}$. The advantage of reformulating the problem in this way is that the Laplacian matrix is square, symmetric, and positive semidefinite. Its eigenvalue decomposition can be written $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$.

The solution $\mathbf{X} = \mathbf{U}\mathbf{D}^{-1/2}$ fulfills the constraint, that is, $\mathbf{X}\mathbf{D}\mathbf{X}^T = \mathbf{U}\mathbf{D}^{-1/2}\mathbf{D}\mathbf{D}^{-1/2}\mathbf{U}^T = \mathbf{I}$, since $\mathbf{D}$ is diagonal and $\mathbf{U}$ is orthogonal ($\mathbf{U}\mathbf{U}^T = \mathbf{U}^T\mathbf{U} = \mathbf{I}$). So far the solution gathers all N eigenvectors of $\mathbf{L}$. To get a P-dimensional embedding, and in order to minimize the cost function, the trailing P eigenvectors must be kept, namely, those with the smallest nonzero eigenvalues. Notice that $\mathbf{L}$ is rank-deficient with at least one zero eigenvalue and a trivial uniform eigenvector. Taking the trailing eigenvectors instead of the dominant ones is major difference with kernel PCA. Also, the considered Laplacian matrix is sparse (assuming only a few edges per vertex in the graph), in contrast to the dense Gram matrices used in CMDS, and thus in KPCA, Isomap, MVU, etc. Nevertheless, a form of duality exists between these two categories of methods. In particular, the pseudo-inverse of the Laplacian ($\mathbf{U}\mathbf{\Lambda}^+\mathbf{U}^T$, where $\lambda_{ii}^+ = 1/\lambda_{ii}$ if $\lambda_{ii} \neq 0$) can be considered to be a Gram matrix. The associated distance is the commute-time distance [SFYD04], involving random walks in the neighbourhood graph, which can be seen as a 'kernelised' distance like the geodesic distance in Isomap.

Other methods using sparse matrices are, for instance, locally linear embedding (LLE) [RS00] and Hessian LLE [DG03]. Laplacian eigenmaps is also closely related to spectral clustering [NJW02, BVP+03], a method that uses the Laplacian matrix mostly in the same way to better separate clusters by mapping them to a feature space before projecting them onto a lower-dimensional space. Whereas spectral clustering is well studied and quite successful in practice, spectral embedding as a DR tool suffers from the same limitation as kernel PCA, namely, the a priori mapping induced by the graph Laplacian can actually increase data dimensionality, instead of reducing it, which is detrimental to DR but beneficial to clustering.

5.5 Probabilistic neighbourhood preservation

As described in Section 5.2, many DR methods consider that the proximities between data points can be well captured by measuring their distances. As a consequence, the computation of low-dimensional coordinates of the data is naturally driven by a distance preservation criterion. The low-dimensional space should indeed well render the high-dimensional proximities between the data, as exemplified by Figure 2.

Many recent DR developments however question the last assumption that distances well summarize proximity relationships [LV11]. Distances in spaces with varying dimensions indeed behave differently. Due to the norm concentration effect induced by the curse of dimensionality [LV07], distances in high-dimensional spaces tend to concentrate toward much higher values than in low-dimensional ones. This misleads criteria aiming to exactly preserve distances between spaces with different dimensions.

The last observations motivated the development of proximity indicators being more robust when computed in different spaces. In particular, these indicators should have the ability to be invariant with respect to distance shifts. This would allow them to at least partly alleviate the undesirable consequences of the norm concentration.

In this context, the *Stochastic Neighbor Embedding* (SNE) method [HR03] constituted a real paradigm shift in DR. It is based on the definition of similarities between pairs of data points in both HD and LD spaces. These similarities more faithfully characterize the proximity relationships

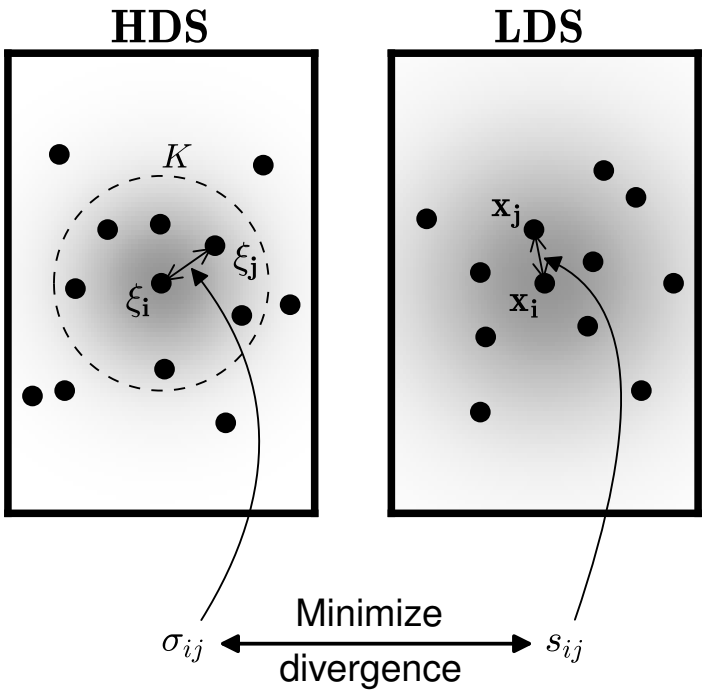


Figure 8: Illustration of similarity-based dimensionality reduction. In both high- and low-dimensional spaces (HDS and LDS), the data points are depicted using black dots. The HDS and LDS similarities between each pair of points are computed. This defines fields of influences around each data point in both spaces. The ones around $\boldsymbol{\xi}_i$ and $\mathbf{x}_i$ are represented by the grays shades with decreasing intensities as the distances to $\boldsymbol{\xi}_i$ and $\mathbf{x}_i$ increase. This leads to soft K-ary neighbourhoods around each datum, as represented around $\boldsymbol{\xi}_i$. As the HDS configuration is fixed, the low-dimensional positions of the data points are computed by minimizing the divergence between the HDS and LDS similarities.

among the data, precisely thanks to their invariance with respect to distance shifts. From the pairwise distances, the latter similarities are defined for $i \neq j$ in the high- and low-dimensional spaces as

$$\sigma_{ij} = \frac{\exp(-\pi_i \delta_{ij}^2/2)}{\sum_{k,k\neq i} \exp(-\pi_i \delta_{ik}^2/2)} \quad \text{and} \quad s_{ij} = \frac{\exp(-p_i d_{ij}^2/2)}{\sum_{k,k\neq i} \exp(-p_i d_{ik}^2/2)} \ . \tag{10}$$

Since proximities are only relevant to evaluate between different data items, σ_{ii} and s_{ii} are set to 0. The precisions of the normalized Gaussian functions centered on $\boldsymbol{\xi}_i$ and $\mathbf{x}_i$ are denoted by π_i and p_i.

σ_{ij} (resp. s_{ij}) can be interpreted as the probability of $\boldsymbol{\xi}_j$ (resp. $\mathbf{x}_j$) to lie in a soft Gaussian neighbourhood around $\boldsymbol{\xi}_i$ (resp. $\mathbf{x}_i$). The precisions of the Gaussian functions allow to characterize these neighbourhoods in terms of their sizes. Smaller (resp. larger) precisions indeed flatten (resp. narrow) the Gaussian functions, defining larger (resp. smaller) neighbourhoods as more (resp. less) data points have a non-negligible probability to belong to them. Hence the precision value and the soft neighbourhood size uniquely determine each other. As it is more natural to reason in terms of sizes than precisions, one usually fix the typical number of neighbors around each data point, which is termed as the perplexity. The precisions of the Gaussian functions are then deduced to equalize the soft Gaussian neighbourhoods sizes and the perplexity.

More formally, the perplexity K_i of the soft neighbourhood around $\boldsymbol{\xi}_i$ is defined as

$$K_i = \exp(H_i) \quad \text{and} \quad H_i = -\sum_{j=1}^{N} \sigma_{ij} \ln \sigma_{ij} \ , \tag{11}$$

where H_i denotes the entropy of the distribution of σ_{ij}. As intuitively explained above, the neighbourhood width, the size K_i and the entropy H_i increase as the precision π_i decreases. In all (single-scale) variants of SNE, a unique perplexity value $K_\star$ is provided by the user in order to adjust each HD precision π_i by solving (11) with $K_i = K_\star$ for $1 \leq i \leq N$.

In order to identify the LD precisions p_i, the same procedure can not be applied as the coordinates $\mathbf{x}_i$ are unknown. To circumvent this issue, SNE fixes $p_i = 1$ for all i.

Having defined the similarities σ_{ij} and s_{ij}, $\mathbf{X}$ can be computed in order to match σ_{ij} and s_{ij} for all $1 \leq i, j \leq N$. As these similarities sum up to one ($\sum_j \sigma_{ij} = \sum_j s_{ij} = 1$), $\boldsymbol{\sigma}_i = [\sigma_{ij}]_{1\leq j\leq N}$ and $\mathbf{s}_i = [s_{ij}]_{1\leq j\leq N}$ define discrete probability distributions. Divergences can therefore evaluate their mismatch. This process is illustrated by Figure 8.

The Kullback-Leibler divergence is used in SNE. It can be developed as

$$D_{\mathrm{KL}}(\boldsymbol{\sigma}_i \| \mathbf{s}_i) = \sum_j \sigma_{ij} \ln(\sigma_{ij}/s_{ij}) \ . \tag{12}$$

The cost function of SNE [HR03] then consists in the sum of (12) for each i:

$$E(\mathbf{X}; \boldsymbol{\Xi}, \mathbf{p}, \boldsymbol{\pi}) = \sum_i D_{\mathrm{KL}}(\boldsymbol{\sigma}_i \| \mathbf{s}_i) \ , \tag{13}$$

where $\mathbf{p} = [p_i]_{1\leq i\leq N}$ and $\boldsymbol{\pi} = [\pi_i]_{1\leq i\leq N}$. By optimizing this cost function, one can find the low-dimensional coordinates of the data points. The SNE method is hence fully defined.

Over the years, enhancements of the basic SNE method were proposed in order to tackle problems with increasing difficulties. One can first cite the t-SNE method [vdMH08]. It modifies SNE in two ways:

- The HD similarities are made symmetric in order to simplify the expression of the cost function gradient:
$$\sigma_{ij} \leftarrow \frac{\sigma_{ij} + \sigma_{ji}}{2N} \ . \tag{14}$$
- The LD similarities are defined using a Student t-distribution with one degree of freedom:
$$s_{ij} = \frac{\left(1 + d_{ij}^2\right)^{-1}}{\sum_{k\neq l} \left(1 + d_{kl}^2\right)^{-1}} \ . \tag{15}$$

At the exception of these two modifications, t-SNE remains identical to SNE, in particular concerning its cost function. Although symmetrizing the HD similarities is not clear to have a significant effect on the performances [LRB+13], modifying the LD similarities is crucial. Using a heavy-tailed distribution in the LD space has been shown to allow to partly alleviate the so-called crowding problem. When reducing the dimensionality of very high-dimensional spaces, one can not hope to preserve exact neighborhood relationships in the LD space as in the HD one as there is much more "place" around each datum in high dimensions. Hence data points lying at a moderate distance from each other must be pushed much farther away in the LD space in order to best preserve close neighborhoods. The use of a heavy-tailed distribution in the LD space therefore allows data points at a moderate distance in the HD space to lie far apart in the LD one while preserving a LD similarity close to their HD one. This avoids misleading the cost function. On the other side, the Student t-distribution in the LD space induces an exponential transformation between the HD and LD distances [LRB+13], which prevents t-SNE to yield isometric embeddings of linear manifolds.

Another variant of SNE is the *Neighborhood Retrieval Visualizer* (NeRV) [VPN+10]. It uses the exact same HD and LD similarities as SNE, at the exception that it fixes $p_i = \pi_i$. On the other hand, it modifies its cost function by blending two dual KL divergences in a so-called type 1 mixture [CA10]:

$$D^{\kappa}_{\mathrm{KLt1}}(\boldsymbol{\sigma}_i \| \mathbf{s}_i) = (1-\kappa) D_{\mathrm{KL}}(\boldsymbol{\sigma}_i \| \mathbf{s}_i) + \kappa D_{\mathrm{KL}}(\mathbf{s}_i \| \boldsymbol{\sigma}_i) \ , \tag{16}$$

where parameter $0 \leq \kappa \leq 1$ trades off both terms. This idea of mixing dual objective functions was previously investigated in the distance preservation paradigm [VK05]. NeRV cost function is then written as

$$E(\mathbf{X}; \boldsymbol{\Xi}, \mathbf{p}, \boldsymbol{\pi}, \kappa) = \sum_i D^{\kappa}_{\mathrm{KLt1}}(\boldsymbol{\sigma}_i \| \mathbf{s}_i) \ . \tag{17}$$

This divergence mixture can be interpreted as a way to balance the precisions and recalls of the retrieved LD neighborhoods around the data points [VPN+10].

Introducing the notation $\mathbf{z}_i = \kappa \boldsymbol{\sigma}_i + (1-\kappa)\mathbf{s}_i$, another KL divergences combination can be defined as

$$D^{\kappa}_{\mathrm{KLt2}}(\boldsymbol{\sigma}_i \| \mathbf{s}_i) = \kappa D_{\mathrm{KL}}(\boldsymbol{\sigma}_i \| \mathbf{z}_i) + (1-\kappa) D_{\mathrm{KL}}(\mathbf{s}_i \| \mathbf{z}_i) \ , \tag{18}$$

$D^{1/2}_{\mathrm{KLt2}}(\boldsymbol{\sigma}_i \| \mathbf{s}_i)$ is termed as the type 2 symmetric KL divergence [CA10], or symmetric Jensen-Shannon divergence [BR82, Lin91]. The cost function becomes

$$E(\mathbf{X}; \boldsymbol{\Xi}, \mathbf{p}, \boldsymbol{\pi}, \kappa) = \frac{1}{\kappa(1-\kappa)} \sum_i D^{\kappa}_{\mathrm{KLt2}}(\boldsymbol{\sigma}_i \| \mathbf{s}_i) \ , \tag{19}$$

which is effectively used in the Jensen-Shannon embedding (JSE, or 'Jessie') method [LRB+13], relying on the same HD and LD similarities as SNE. Other divergences can be defined and are discussed in [BHBV12].

As the obtained low-dimensional coordinates result from the optimization of the cost function, the success of similarity-based methods is crucially dependent on this last step. Yet cost functions of SNE-like methods are known to be difficult to optimize [HR03, vdMH08, LRB+13, LPOnV15]. They indeed most probably present many local minima, in particular when $K_\star$ is small compared to N since the localness of the similarities is then amplified. The squared distances in the normalised Gaussian functions furthermore poorly scale the gradient. These observations suggested to rely on optimization schemes benefiting from second order information about the cost function [YWO10, vdM10]. Limited-memory BFGS [Noc80] therefore replaces gradient-descent in NeRV [VPN+10] and JSE [LRB+13]. Moreover by modifying the values of the HD similarities in the few first iterations, either directly [vdMH08] or with an increased value of $K_\star$ [VPN+10, LRB+13], one can typically lessen the influence of local minima.

Until this point, all the detailed SNE variants considered a single, fixed perplexity value $K_\star$ in order to define the HD similarities. As detailed above, the latter perplexity implicitly determine the soft neighborhood size around each datum. The relationships between pairs of data points inside these neighborhoods will hence be considered while designing the low-dimensional embedding, but not the ones outside these neighborhoods. As $K_\star$ is usually small with respect to N, SNE-like methods thus best preserve close neighborhoods around each datum. However they usually do not capture the global structures of data sets as they only focus on local interactions. Large-scale neighborhoods are hence often poorly reproduced [LPOnV15]. Using only a single perplexity value

$K_\star$ indeed enforces to choose between preserving local or global structures. As one typically wish to retrieve both, combining several single-scale similarities, defined using different perplexities, into multi-scale ones could allow to focus both on local and global interactions [LPOnV15].

With this idea in mind, an index h can be introduced in the single-scale similarities:

$$\sigma_{hij} = \frac{\exp(-\pi_{hi}\delta_{ij}/2)}{\sum_{k,k\neq i}\exp(-\pi_{hi}\delta_{ik}/2)} \quad \text{and} \quad s_{hij} = \frac{\exp(-p_{hi}d_{ij}/2)}{\sum_{k,k\neq i}\exp(-p_{hi}d_{ik}/2)} \ . \tag{20}$$

The HD and LD precisions of the ith soft neighbourhoods on scale h are denoted by π_{hi} and p_{hi}. Again, σ_{hii} and s_{hii} are set to 0. Multi-scale similarities can then be defined as

$$\sigma_{ij} = \frac{1}{L}\sum_{h=L_{\min}}^{L_{\max}} \sigma_{hij} \quad \text{and} \quad s_{ij} = \frac{1}{L}\sum_{h=L_{\min}}^{L_{\max}} s_{hij} \ , \tag{21}$$

with $1 \leq L_{\min} \leq h \leq L_{\max}$ and where the considered number of scales is $L = L_{\max} - L_{\min} + 1$. Obviously if $\sum_{j=1}^{N}\sigma_{hij} = 1$ for all h and i, then $\sum_{j=1}^{N}\sigma_{ij} = 1$ for all i. It should also be noted that the multi-scale similarities reduces to single-scale ones if $L_{\min} = L_{\max}$.

Multi-scale SNE, t-SNE, NeRV and JSE methods can hence be defined [LPOnV15, dBMVL18c], using multi-scale similarities instead of single-scale ones in their respective cost functions. In order to determine the precisions π_{hi} at the various scales, perplexity values distributed over the whole range from 1 to N can be fixed without any user intervention. One could for example start from a small one such as $K_\star = 2$, and increase it by powers of two: $K_{h\star} = 2^{h-1}K_\star$ with $1 \leq L_{\min} \leq h \leq L_{\max} = \lfloor \log_2(N/K_\star) \rceil$, $\lfloor\cdot\rceil$ denoting the rounding operator. Defining $L_{\max}$ in this way allows to consider large neighbourhoods with respect to N while preventing σ_{hij} to become nearly uniform. Optionally, setting $L_{\min} > 1$ allows to more focus on large neighbourhoods.

On the other side, the determination of p_{hi} is related to the estimation of the correlation dimension [LPOnV15]. Finally multi-scale optimisation can be carried out by introducing the smaller scales one at the time during the optimisation. This procedure allows to ease the multi-scale methods tuning [LPOnV15].

6 Quality assessment

Dimensionality reduction is by nature an unsupervised learning task, namely, there is no ground truth in most practical cases and, therefore, no obvious performance assessment. It is the user's responsiblity to devise a performance criterion, which is often directly formalised into the objective function of the methods like those detailed above. This situation, however, makes it difficult to compare methods to each other.

For example, one could rely on variance preservation to assess DR quality, but doing so would bias evaluation favorably for PCA and CMDS, although nonlinear DR methods are designed and expected to outperform them in many cases. Similarly, stress functions have sometimes been deemed generic enough to be adopted as a universal quality measure. Again, this choice would not be fair as stress functions favor MDS-like methods that precisely optimize them. Moreover, distance preservation is known to be suboptimal in many DR problems where distances heavily concentrate in high-dimensional data spaces. Measures of data reconstruction, like those used in PCA and AEs, could be used as well and appear again as sufficiently generic. In this case, however, they apply only to parametric and inversible DR.

Yet another possibility to assess DR quality is to rely on some external ground truth that is not directly involved in the DR problem. The most typical example occurs when multivariate data is primarily collected to solve a supervised task. Then, data consists of both of inputs $\boldsymbol{\xi}_i$ and associated desired outputs y_i, like class labels or some regression output. The main issue with this approach is that it is indirect and requires such additional labels or variables to be systematically available even if they are not strictly necessary for DR itself, and possibly not relevant. For instance, class labels would typically be used to assess DR quality by measuring the classification performance in the embedding space. Better classification results would then mean better embedding. Notice also that DR quality is assessed only near class boundaries, since embedding errors within the bulk of a class would pass undetected.

With the recent cornucopian development of many new DR methods, sometimes very complicated, several attempts have been made to assess the quality of DR results in a generic and method-independent way. Instead of variance or distance preservation, these tools focus on neighborhood

preservation. Intuitively, this means that distances must not be preserved in value, but just in ranking. In other words, let us consider three points $\boldsymbol{\xi}_i$, $\boldsymbol{\xi}_j$, and $\boldsymbol{\xi}_k$ and their embeddings $\mathbf{x}_i$, $\mathbf{x}_j$, and $\mathbf{x}_k$; if $\boldsymbol{\xi}_i$ is taken as vantage point, with $\delta_{ij} < \delta_{ik}$, then distance preservation aims at $\delta_{ij} = d_{ij}$ and $\delta_{ik} = d_{ik}$, whereas neighborhood preservation would relax this into $\delta_{ij} \leq \delta_{ik} \Rightarrow d_{ij} \leq d_{ik}$. The main motivation is that distance rankings are expected to be relatively immune to norm concentration, contrarily to raw distances.

Formally, distance rankings call for the definition of ranks and K-ary neighborhoods. Points can be ranked in rising order of their distance with respect to a given point of interest. The rank of $\boldsymbol{\xi}_j$ with respect to $\boldsymbol{\xi}_i$ in the HD space is written as $\rho_{ij} = |\{k : \delta_{ik} < \delta_{ij} \text{ or } (\delta_{ik} = \delta_{ij} \text{ and } 1 \leq k < j \leq N)\}|$, where $|A|$ denotes the cardinality of set A. Similarly, the rank of $\mathbf{x}_j$ with respect to $\mathbf{x}_i$ in the LD space is $r_{ij} = |\{k : d_{ik} < d_{ij} \text{ or } (d_{ik} = d_{ij} \text{ and } 1 \leq k < j \leq N)\}|$. Notice that reflexive ranks are set to zero ($\rho_{ii} = r_{ii} = 0$) and that ranks are unique, i.e., there are no equal-valued ranks: $\rho_{ij} \neq \rho_{ik}$ for $k \neq j$, even if $\delta_{ij} = \delta_{ik}$. This means that nonreflexive ranks belong to $\{1, \ldots, N-1\}$. The non-reflexive K-ary neighbourhoods of $\boldsymbol{\xi}_i$ and $\mathbf{x}_i$ are the sets defined by $\nu_i^K = \{j : 1 \leq \rho_{ij} \leq K\}$ and $n_i^K = \{j : 1 \leq r_{ij} \leq K\}$, respectively.

With this notation, a measure of K-ary neighbourhood preservation [LV09] can be written as

$$Q_{\mathrm{NX}}(K) = \frac{1}{KN} \sum_{i=1}^{N} |\nu_i^K \cap n_i^K| \ , \tag{22}$$

which is the average proportion of preserved K-ary neighbours, for $1 \leq K \leq N-1$. It varies between 0 (empty intersection) and 1 (perfect agreement). Knowing that random coordinates in $\mathbf{X}$ lead on average to $Q_{\mathrm{NX}}(K) \approx K/(N-1)$ [CB09], the useful range of $Q_{\mathrm{NX}}(K)$ is $N-1-K$, which depends on K. Therefore, in order to fairly compare or combine values of $Q_{\mathrm{NX}}(K)$ for different neighbourhood sizes, the criterion can be rescaled, like in [Kun07, LRB+13], to get

$$R_{\mathrm{NX}}(K) = \frac{(N-1)Q_{\mathrm{NX}}(K) - K}{N-1-K} \ , \tag{23}$$

for $1 \leq K \leq N-2$. This modified criterion indicates the improvement over a random embedding and has the same useful range between 0 (random) and 1 (perfect) for all K.

7 Examples of DR results

In order to illustrate the results of various DR methods and their quality assessment, Brendan Frey's faces are used. This database is a collection of images of a single person's face, with various facial expressions, extracted as consecutive frames of a video clip. The data base includes $N = 1965$ images, with size 28 times 20 pixels, some of them being illustrated in Fig. 9. The images are simply vectorized, namely, the 28 rows of 20 pixels each are concatenated to form long 560-dimensional vectors $\boldsymbol{\xi}_i$. Starting from these high-dimensional vectors, dimensionality reduction can determine low-dimensional representations. If the target dimensionality is two, then the representations can be displayed as scatter plots, where each point corresponds to a facial expression. In the scatter plots, the same color code as in Fig. 9 is used.

Figure 10 shows the embeddings of Frey's faces with various DR methods. The considered ones are:

- A linear projection with classical MDS (CMDS) [YH38], which is equivalent to PCA [Pea01].
- Variants of stress-based MDS, namely non-metric (NMDS) [She62, Kru64], Sammon's non-linear mapping (NLM) [Sam69], and curvilinear component analysis (CCA) [DH97].
- Spectral embedding with Laplacian eigenmaps (LE).
- Probabilistic neighbour embedding, with SNE [HR03], NeRV [VPN+10], JSE [LRB+13], their multi-scale counterpart [LPOnV15], and t-SNE [vdMH08].

This short list covers many types of methods, from linear to nonlinear, from old to quite recent, and with various approaches. The focus is set on neighbour embedding techniques, which are currently the state of the art in the domain. For these methods, the perplexity parameter is set to 32 by default, except for the multi-scale extensions, which consider automatically several neighbourhood sizes.

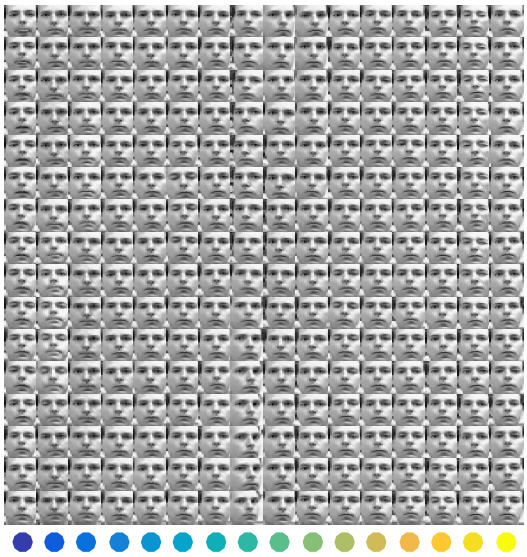


Figure 9: Brendan Frey's faces. Some examples of facial expression images from the database, as well as the color code used further in the embedding scatter plots. As consecutive frames from a video clip, the thumbnail images should be read from top to bottom and then from left to right. The shades of colors in the bottom give an indication of time, per column.

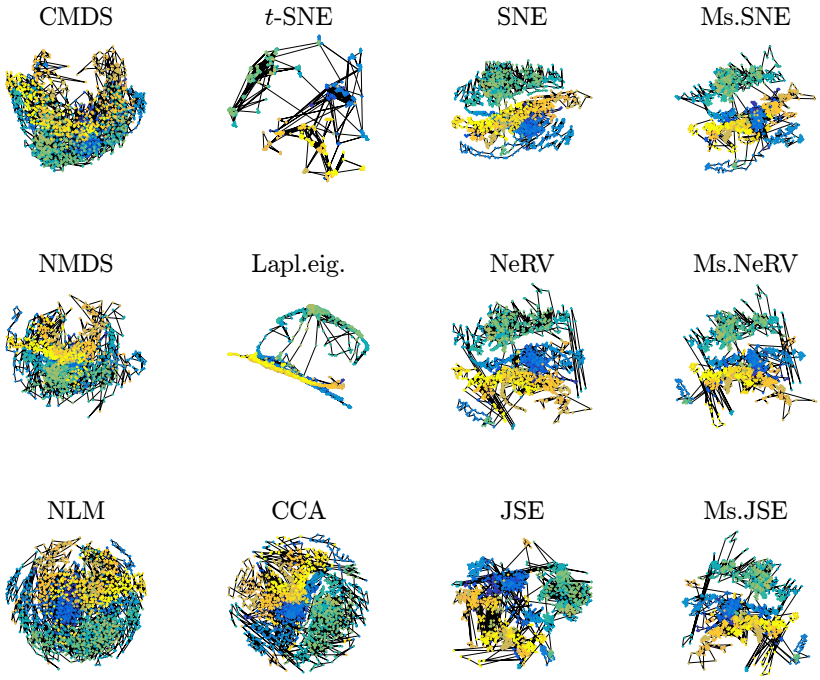


Figure 10: Embeddings of Brendan Frey's faces. In those scatter plots, each point corresponds to an image of a facial expression. Since the images are frames of a video clip, consecutive ones are linked with black lines. The color code follows time as well, as in Fig. 9.

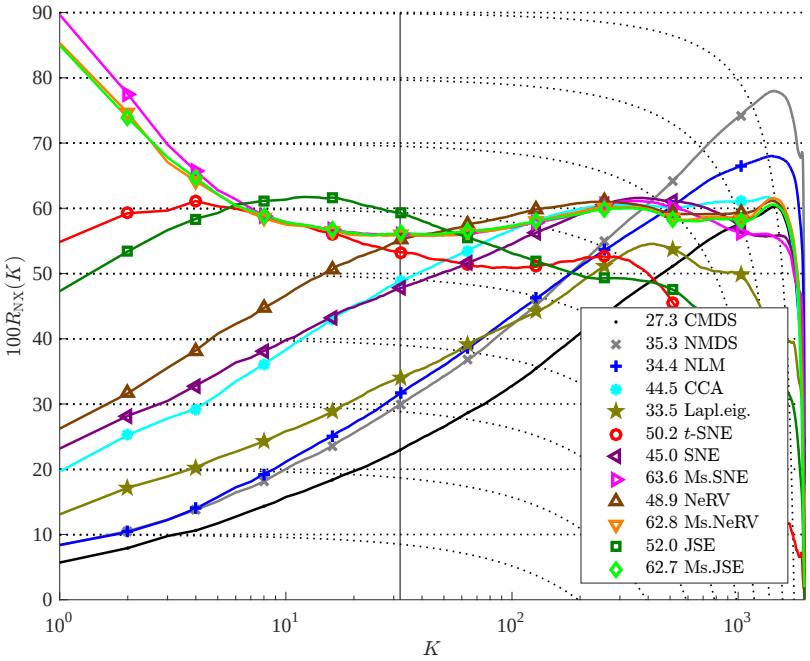


Figure 11: Quality assessment for the embeddings of Brendan Frey's faces in Fig. 10. Quality is evaluated through K-ary neighbourhood preservation. The neighbourhood size K varies along the horizontal axis.

Figure 11 shows the quality assessment of the scatter plots in Fig. 10. In this plot, each colored curve corresponds to an embedding. The horizontal axis is a log-scale of neighbourhood size K. The vertical axis indicates $R_{\mathrm{NX}}(K)$, that is, the relative improvement over an arbitrary embedding with random coordinates. A null value means no improvement, whereas 100% is synonymous of perfect neighbourhood preservation in the low-dimensional representation. The areas under the curves, providing global scores over all K, with more focus on the small neighbourhoods due to the log-scale, are reported in the legend, next to the methods' names.

A method like PCA (or CMDS) provides a linear projection of data, which typically reveals the global shape of the point cloud. The corresponding curve culminates indeed for rather large neighbourhood sizes. Small neighbourhoods are not well reproduced due to false neighbours, that is, intruders caused by the inevitable squashing entailed by linear projections. Showing a similar behaviour, nonmetric MDS performs even better, with the best rendition of large neighbourhoods. Another stress-based MDS method like Sammon's NLM, does not perform as well as NMDS, but is much simpler; the embedding is more cluttered and indistinctly circular. With some added complexity, CCA improves on small neighbourhoods; also some clusters become discernible. As a method closely related to spectral clustering through the use of the graph Laplacian matrix, Laplacian eigenmaps clearly separates the few main clusters, at the expense of collapsing them. Variants of SNE, like t-SNE, NeRV, JSE, and their multi-scale extensions, through the use of normalised similarities, can better cope with high-dimensional data than stress-based MDS, which tries to match distances directly, disregarding norm concentration effects [FWV07, LV11]. Due to slightly discrepant similarity definitions (Student versus Gaussian), t-SNE tends to over-emphasize cluster separation, compared to SNE. With respect to neighbourhood preservation, SNE outperforms CCA, and it provides an embedding where clusters are less packed. NeRV behaves even better than SNE, but does not match the performance of t-SNE for very small neighbourhoods. Another SNE variant, JSE performs similarly as t-SNE, being slightly more faithful overall, at the cost of small neighbourhoods. Eventually, considering neighbourhoods over a large range of sizes allows multi-scale variants of SNE (Ms.SNE, Ms.NeRV, Ms.JSE) to perform the best, almost identically. Clusters are clearly visible, without magnifying their separation, leaving some room for intra-cluster details. The smallest neighbourhoods reach nearly 90%, whereas the largest ones remain above 50%.

These results show the improvements achieved over more than a century of research in the domain, both quantitatively and visually.

8 Alternative metrics and distances

8.1 Mahalanobis

Introduced by [C+36], the Mahalanobis Distance (M_D), as for Euclidean distance, can be calculated in the original HD space and in the principal component (PC) space. M_D is based on the correlation between variables by which different models can be identified and analyzed. This is a useful way to determine the similarity between a series of known and unknown data. It differs from the Euclidean distance in that it takes into account the variance and correlation of the data series. Thus, unlike the Euclidean distance where all the components of the vectors are treated independently and in the same way, the M_D gives a smaller weight to the most dispersed components. In the case of signal analysis, and assuming that each component is a Gaussian random variable, this amounts to minimizing the influence of the most noisy components (those with the largest variance). The M_D is often used for the detection of outliers in a dataset, or to determine the consistency of data provided by a sensor for example: this distance is calculated between the received data and those predicted by a model. In practice, the MD of a multivariate vector $\mathbf{x} = (x_1, x_2, x_3, ..., x_M)^T$ to a set of mean value vectors $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3, ..., \mu_M)^T$ and having a covariance matrix $\mathbf{C_X}$ is defined by $D_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T C_{\mathbf{x}}^{-1} (\mathbf{x} - \boldsymbol{\mu})}$.

Finaly, it is important to note that the squared Mahalanobis distance is equal to the sum of squares of the scores of all non-zero standardised principal components (see [Bre15] for more details).

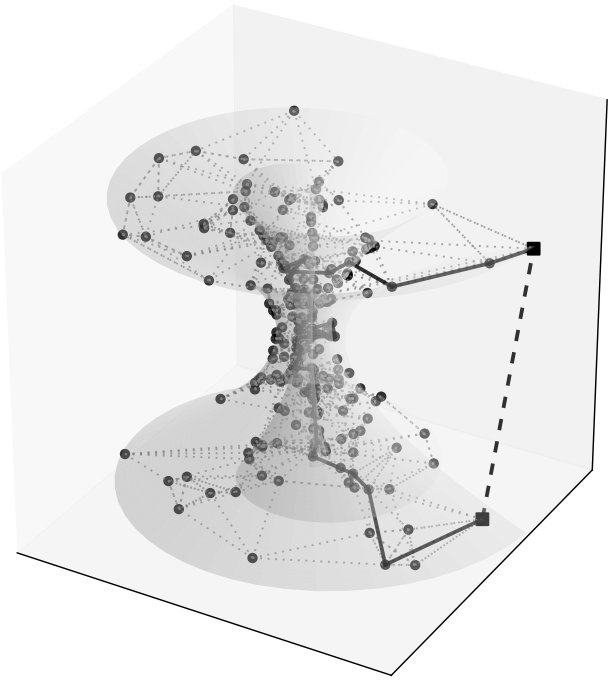


Figure 12: Illustration of the concept of geodesic distance. The data points are distributed on a three-dimensional molted swiss roll manifold. A K-neighborhood graph is constructed on the data set, its edges being depicted as gray dotted lines linking data points. An edge can be weighted by the Euclidean distance between the two dots it binds. Some short-circuits across the manifold can be observed due to the fact that the data set is finite. The geodesic distance between the two squared data points is then computed as the length of the shortest path between them in the graph. This shortest path is depicted by solid black lines. On the other side, the dashed black line represents their Euclidean distance. One can observe that the geodesic distance more faithfully describes the relationship between data points as it tries to approximate the manifold on which they are distributed.

8.2 Geodesic distances

Used in the well known ISOMAP method [TDSL00b], geodesic distance corresponds to a useful nonlinear metric for estimating the intrinsic geometry of a data manifold. Compared to traditional Euclidean distance (Figure 12), Geodesic distance consists in the sum of edge weights along the shortest path between two nodes in a K-neighborhood graph of the observed data (computed using Dijkstra's algorithm, for example). In the context of DR, this consists in estimating the shortest paths in the graph for all pairs of data points in high dimension and/or low dimension.

8.3 Commute-time distances

See Section 5.4.2.

8.4 Gower's similarity

The Gower's similarity is a similarity which comes from ecologica analysis [Gow71]. This similarity is designed for heterogeneous data sets. We define an heterogeneous data set as a data set wherein the objects are described with categorical and numerical variables. Moreover, the Gower's similarity can deal with missing data.

The Gower's similarity is calculated as following. Let two objects $\mathbf{x}_i$ and $\mathbf{x}_j$, described by M

Table 1: An example of heterogeneous data set

	V_1	V_2	V_3	V_4	V_5	V_6
x_1	1,5	A	2,3			
x_2				7	10	W
x_3		A		8	9	W

variables, which can be categorical or numerical. The Gower's similarity S_{ij} between $\mathbf{x}_i$ and $\mathbf{x}_j$ is:

$$S_{ij} = \frac{\sum_{k=1}^{M} w_{ijk} S_{ijk}}{\sum_{k=1}^{M} w_{ijk}} \tag{24}$$

In this formula, S_{ijk} is a "local" similarity, calculated between $\mathbf{x}_i$ and $\mathbf{x}_j$ for the variable k, such as:

- if k is a categorical variable, then

$$S_{ijk} = \{ \begin{array}{l} 0 \text{ if } x_{ik} \neq x_{jk} \\ 1 \text{ if } x_{ik} = x_{jk} \end{array} \tag{25}$$

- if k is a numerical variable, then

$$S_{ijk} = 1 - \frac{\|x_{ik} - x_{jk}\|}{max(x_k) - min(x_k)} \tag{26}$$

In this formula, w_{ijk} is a weight associated to the variable k for the pair (x_i, x_j), such as:

- if x_{ik} or x_{jk} are missing data, then $w_{ijk} = 0$
- else $w_{ijk} = 1$

The Gower's similarity has values between 0 and 1: 1 indicates a maximal similarity.

The Gower's similarity can be easily transformed into a dissimilarity: $D_{ij} = 1 - S_{ij}$.

The Gower's similarity is an interesting similarity to manage heterogeneous data sets with missing data. But this similarity should be used carefully, due to these properties.

In the table 1, we propose an example. In this example, $S_{1,2}$ cannot be calculated: x_1 and x_2 have no variable in common, because of missing data. But, $S_{1,3}$ and $S_{2,3}$ can be calculated. Moreover, $S_{1,3} = 1$, the maximal possible value for a Gower's similarity.

In conclusion on missing data, the Gower's similarity can be used on data set with missing data, but the user of this similarity have to ensure than missing data do not compromise the representation of the data set.

Another point which schould be discussed, before using Gower's similarity, is the weighting of pair (x_i, x_j). As mentionned below, these weights are originally used to manage missing data. But, we assume that these weights can be used to improve the calculation of the similarity. Two approaches can be considered. On the one hand, the weights can be chosen by the user of the similarity, depending on his expertise on the data set. On the other hand, the weights can be calculated, optimized in order to improve the representation of similarities between vectors in a data set.

Concerning this second approach, some works offer algorithms that set the weights in order to optimize a clustering process [vdH16] . But, for the moment, the optimization of variable weights in a reduction dimension perspective is actually not adressed.

9 Conclusions

Dimensionality reduction has been studied for over a century. Different paradigms and techniques have been investigated to create a plethora of methods. The oldest one is PCA, based on a variance-preserving linear projection. Classical MDS seeks a projection that preserves pairwise scalar products. Both work by eigenvalue decomposition. Other variants of MDS yield nonlinear mapping with ad hoc optimisation of so-called stress functions that quantify distance preservation

(or (dis)similarity ranks in nonmetric MDS). Various neural networks, like self-organising maps and auto-encoders, can achieve DR as well, also with ad hoc optimisation techniques. Spectral optimisation, initially associated with linear projection only, has raised interest again in combination with the kernel trick, leading to methods like kernel PCA, Isomap, Laplacian eigenmaps, or locally linear embedding. Eventually, the current state of the art relies on stochastic neighbourhood embedding (SNE, t-SNE), where distances are wrapped in similarities with invariance properties that allow mitigating distance concentration, which affects stress-based MDS.

In parallel to DR method development, new tools to assess DR quality have been devised, trying to capture to essence of modern, nonlinear DR, which focuses more and more on the preservation of local neighbourhoods from the HD data space to the LD representation space.

Dimensionality reduction remains a branch of non-supervised machine learning, used mainly for exploratory data analysis and visualisation. Quality assessment and the choice of the metric to evaluate actual proximities between data are still open questions.

References

[AC06] U. Akkucuk and J.D. Carroll. PARAMAP vs. Isomap: A comparison of two nonlinear mapping algorithms. *Journal of Classification*, 23(2):221–254, 2006.

[Ben09] Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning archive*, 2(1):1–127, 2009.

[BH03] M. Brand and K. Huang. A unifying theorem for spectral embedding and clustering. In C.M. Bishop and B.J. Frey, editors, *Proceedings of International Workshop on Artificial Intelligence and Statistics (AISTATS'03)*. January 2003.

[BHBV12] K. Bunte, S. Haase, M. Biehl, and T. Villmann. Stochastic neighbor embedding (sne) for dimension reduction and visualization using arbitrary divergences. *Neurocomputing*, 90:23–45, 2012.

[BHV99] H.-U. Bauer, M. Herrmann, and T. Villmann. Neural maps and topographic vector quantization. *Neural Networks*, 12:659–676, 1999.

[BN02] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems (NIPS 2001)*, volume 14. MIT Press, 2002.

[BP92] H.-U. Bauer and K.R. Pawelzik. Quantifying the neighborhood preservation of self-organizing maps. *IEEE Transactions on Neural Networks*, 3:570–579, 1992.

[BPV+04] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet. Out-of-sample extensions for LLE, isomap, MDS, eigenmaps, and spectral clustering. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems (NIPS 2003)*, volume 16. MIT Press, Cambridge, MA, 2004.

[BR82] J. Burbea and C.R. Rao. On the convexity of some divergence measures based on entropy functions. *IEEE Transactions on Information Theory*, 28(3):489–495, 1982.

[Bre15] Richard G Brereton. The mahalanobis distance and its relationship to principal component scores. *Journal of Chemometrics*, 29(3):143–145, 2015.

[BVP+03] Y. Bengio, P. Vincent, J.-F. Paiement, O. Delalleau, M. Ouimet, and N. Le Roux. Spectral clustering and kernel PCA are learning eigenfunctions. Technical Report 1239, Département d'Informatique et Recherche Opérationnelle, Université de Montréal, Montréal, July 2003.

[C+36] Mahalanobis Prasanta Chandra et al. On the generalised distance in statistics. In *Proceedings of the National Institute of Sciences of India*, volume 2, pages 49–55, 1936.

[CA10] A. Cichocki and S.-i. Amari. Families of alpha- beta- and gamma-divergences: Flexible and robust measures of similarities. *Entropy*, 12:1532–1568, 2010.

[CB09] L. Chen and A. Buja. Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association*, 104(485):209–219, 2009.

[CLL+05] R.R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. Zucker. Geometric diffusion as a tool for harmonic analysis and structure definition of data, part i: Diffusion maps. *Proceedings of the National Academy of Sciences*, 102(21):7426–7431, 2005.

[CPn97] M.A. Carreira-Perpiñán. A review of dimension reduction techniques. Technical report, University of Sheffield, Sheffield, January 1997.

[CPn10] M.Á. Carreira-Perpiñán. The elastic embedding algorithm for dimensionality reduction. In *Proc. of the 27th International Conference on Machine Learning*, pages 167–174, 2010.

[CV02] F. Camastra and A. Vinciarelli. Estimating the intrinsic dimension of data with a fractal-based method. *IEEE Transactions on pattern analysis and machine intelligence*, 24(10):1404–1407, 2002.

[dBMVL18a] C. de Bodt, D. Mulders, M. Verleysen, and J. A. Lee. Extensive assessment of Barnes-Hut t-SNE. In *ESANN*, pages 135–140, 2018.

[dBMVL18b] C. de Bodt, D. Mulders, M. Verleysen, and J. A. Lee. Nonlinear Dimensionality Reduction with Missing Data using Parametric Multiple Imputations. *IEEE Transactions on Neural Networks and Learning Systems*, PP(99):1–14, August 2018. DOI: 10.1109/TNNLS.2018.2861891.

[dBMVL18c] C. de Bodt, D. Mulders, M. Verleysen, and J. A. Lee. Perplexity-free t-SNE and twice Student tt-SNE. In *ESANN*, pages 123–128, 2018.

[DC93] D. DeMers and G.W. Cottrell. Nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems (NIPS 1992)*, volume 5, pages 580–587. 1993.

[DG03] D.L. Donoho and C. Grimes. Hessian eigenmaps: New locally linear techniques for high-dimensional data. Technical Report TR03-08, Department of Statistics, Stanford University, Palo Alto, CA, 2003.

[DH97] P. Demartines and J. Hérault. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8(1):148–154, January 1997.

[Don00] D.L. Donoho. High-Dimensional Data Analysis: The Curse and Blessings of Dimensionality. Aide-mémoire for a lecture for the American Math. Society "Math. Challenges of the 21st Century", 2000.

[dST03] V. de Silva and J.B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 705–712. MIT Press, Cambridge, MA, 2003.

[FC07] S.L. France and J.D. Carroll. Development of an agreement metric based upon the RAND index for the evaluation of dimensionality reduction techniques, with applications to mapping customer data. In *Proceedings of MLDM 2007*, pages 499–517. Springer-Verlag, 2007.

[Fuk82] K. Fukunaga. Intrinsic dimensionality extraction. In P.R. Krishnaiah and L.N. Kanal, editors, *Classification, Pattern Recognition and Reduction of Dimensionality, Volume 2 of Handbook of Statistics*, pages 347–360. Elsevier, Amsterdam, 1982.

[FWV07] D. François, V. Wertz, and M. Verleysen. The concentration of fractional distances. *IEEE Transactions on Knowledge and Data Engineering*, 19(7):873–886, July 2007.

[Gow66] J. Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53:325–338, 1966.

[Gow71] John C Gower. A general coefficient of similarity and some of its properties. *Biometrics*, pages 857–871, 1971.

[GS96] G.J. Goodhill and T.J. Sejnowski. Quantifying neighbourhood preservation in topographic mappings. In *Proceedings of the Third Joint Symposium on Neural Computation*, pages 61–82. 1996.

[GS97] G.J. Goodhill and T.J. Sejnowski. A unifying measure for neighbourhood preservation in topographic mappings. In *Proceedings of the 2nd Joint Symposium on Neural Computation*, volume 5, pages 191–202. 1997.

[Hes01] Tom Heskes. Self-organizing maps, vector quantization, and mixture modeling. *IEEE transactions on neural networks*, 12(6):1299–1305, 2001.

[Hot33] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[HR03] G. Hinton and S.T. Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems (NIPS 2002)*, volume 15, pages 833–840. MIT Press, 2003.

[HS06] G.E. Hinton and R.R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006.

[Jol86] I.T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, NY, 1986.

[Kar46] K. Karhunen. Zur Spektraltheorie stochastischer Prozesse. *Ann. Acad. Sci. Fennicae*, 34, 1946.

[Koh82] T. Kohonen. Self-organization of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

[Kra91] M. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243, 1991.

[Kru64] J.B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–28, 1964.

[Kun07] L.I. Kuncheva. A stability index for feature selection. In *Proceedings of 25th Internation Multi-Conference Artificial Intelligence and Applications (IASTED)*, pages 390–395, 2007.

[LBH15] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436—444, 2015.

[Lin91] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

[Loe48] M. Loeve. Fonctions aléatoire du second ordre. In P. Lévy, editor, *Processus stochastiques et mouvement Brownien*, page 299. Gauthier-Villars, Paris, 1948.

[LPOnV15] J.A. Lee, D.H. Peluffo-Ordóñez, and M. Verleysen. Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure. *Neurocomputing*, 169:246–261, 2015.

[LRB+13] J.A. Lee, E. Renard, G. Bernard, P. Dupont, and M. Verleysen. Type 1 and 2 mixtures of Kullback-Leibler divergences as cost functions in dimensionality reduction based on similarity preservation. *Neurocomputing*, 112:92–108, 2013.

[LV04] J.A. Lee and M. Vereleysen. How to project "circular" manifolds using geodesic distances. In M. Verleysen, editor, *Proceedings of ESANN 2004, 12th European Symposium on Artificial Neural Networks*, pages 223–230. d-side, April 2004.

[LV07] J.A. Lee and M. Verleysen. *Nonlinear dimensionality reduction*. Springer, 2007.

[LV09] J.A. Lee and M. Verleysen. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing*, 72(7–9):1431–1443, 2009.

[LV11] J.A. Lee and M. Verleysen. Shift-invariant similarities circumvent distance concentration in stochastic neighbor embedding and variants. In *Proc. International Conference on Computational Science (ICCS 2011)*, pages 538–547, Singapore, 2011.

[LV14] J.A. Lee and M. Verleysen. Two key properties of dimensionality reduction methods. In *IEEE Symposium Series on Computational Intelligence (SSCI) – Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 163–170, 2014.

[NJW02] A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems (NIPS 2001)*, volume 14. MIT Press, Cambridge, MA, 2002.

[Noc80] J. Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980.

[Oja91] E. Oja. Data compression, feature extraction, and autoassociation in feedforward neural networks. In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors, *Artificial Neural Networks*, volume 1, pages 737–745. Elsevier Science Publishers, B.V., North-Holland, 1991.

[Pea01] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

[RHW86] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[RS00] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[RSH02] S.T. Roweis, L.K. Saul, and G.E. Hinton. Global coordination of local linear models. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems (NIPS 2001)*, volume 14. MIT Press, Cambridge, MA, 2002.

[RZ85] D.E. Rumelhart and D. Zipser. Feature discovery by competitive learning. *Cognitive Science*, 9:75–112, 1985.

[Sam69] J.W. Sammon. A nonlinear mapping algorithm for data structure analysis. *IEEE Transactions on Computers*, CC-18(5):401–409, 1969.

[SFYD04] M. Saerens, F. Fouss, L. Yen, and P. Dupont. The principal components analysis of a graph, and its relationships to spectral clustering. In *Proceddings of the 15th European Conference on Machine Learning (ECML 2004)*, pages 371–383, 2004.

[She62] R.N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function (parts 1 and 2). *Psychometrika*, 27:125–140, 219–249, 1962.

[SSM98] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[SWS+06] L.K. Saul, K.Q. Weinberger, F. Sha, J.H. Ham, and D.D. Lee. Spectral methods for dimensionality reduction. In O. Chapelle, B. Schoelkopf, and A. Zien, editors, *Semisupervised Learning*. MIT Press, 2006.

[Tak85] F. Takens. On the numerical determination of the dimension of an attractor, in dynamical systems and bifurcations. In B. Braaksma, H. Broer, and F. Takens, editors, *Lecture Notes in Mathematics*, volume 1125, pages 99–106. Springer-Verlag, Berlin, 1985.

[TdSL00a] J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.

[TDSL00b] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[Tor52] W.S. Torgerson. Multidimensional scaling, (i): Theory and method. *Psychometrika*, 17:401–419, 1952.

[TYdL77] Y. Takane, F.W. Young, and J. de Leeuw. Nonmetric individual differences multidimensional scaling: an alternating least squares method with optimal scaling features. *Psychometrika*, 42:7–67, 1977.

[UNN91] S. Usui, S. Nakauchi, and M. Nakano. Internal colour representation acquired by a five-layer neural network. In T. Kohonen, K. Makisara, O. Simula, and J. Kangas, editors, *Artificial Neural Networks*. Elsevier Science Publishers, B.V., North-Holland, 1991.

[VCPn13] M. Vladymyrov and M.Á. Carreira-Perpiñán. Entropic affinities: Properties and efficient numerical computation. In *Proc. 30th International Conference on Machine Learning (ICML)*, volume 28 of *JMLR: W&CP*. Atlanta, Georgia, 2013.

[vdH16] Jeroen van den Hoven. Clustering with optimised weights for Gower's metric. Technical report, University Amsterdam, University Amsterdam, 2016.

[VDHM97] T. Villmann, R. Der, M. Herrmann, and T. Martinetz. Topology preservation in self-organizing feature maps: Exact definition and measurement. *IEEE Transactions on Neural Networks*, 8(2):256–266, 1997.

[vdM10] L. van der Maaten. Fast optimization for t-sne. In *Advances in Neural Information Processing Systems (NIPS2009), Workshop on Challenges in Data Visualization*, volume 23, Vancouver, 2010.

[vdM14] L. van der Maaten. Accelerating t-SNE using tree-based algorithms. *Journal of Machine Learning Research*, 15:3221–3245, 2014.

[vdMH08] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[VHA+99] Juha Vesanto, Johan Himberg, Esa Alhoniemi, Juha Parhankangas, et al. Self-organizing map in matlab: the som toolbox. In *Proceedings of the Matlab DSP conference*, volume 99, pages 16–17, 1999.

[VK01] J. Venna and S. Kaski. Neighborhood preservation in nonlinear projection methods: An experimental study. In *Proceedings of ICANN 2001*, pages 485–491. Springer, 2001.

[VK05] J. Venna and S. Kaski. Local multidimensional scaling with controlled tradeoff between trustworthiness and continuity. In *Proceedings of the 5th Workshop on Self-Organizing Maps (WSOM'05)*, pages 695–702. Paris, September 2005.

[VK07] J. Venna and S. Kaski. Nonlinear dimensionality reduction as information retrieval. *Journal of Machine Learning Research: Workshop and Conference Proceedings*, 2:572–579, 2007.

[VPN+10] J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research*, 11:451–490, 2010.

[Wil01] C.K.I. Williams. On a connection between Kernel PCA and metric multidimensional scaling. In T.K. Leen, T.G. Diettrich, and V. Tresp, editors, *Advances in Neural Information Processing Systems (NIPS 2000)*, volume 13, pages 675–681. MIT Press, Cambridge, MA, 2001.

[WS06] K.Q. Weinberger and L.K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*, 70(1):77–90, 2006.

[WSS04] K.Q. Weinberger, F. Sha, and L.K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the Twenty-First International Conference on Machine Learning (ICML-04)*, pages 839–846, Banff, Canada, 2004.

[XSB06] L. Xiao, J. Sun, and S. Boyd. A duality view of spectral methods for dimensionality reduction. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 1041–1048. Pittsburg, PA, 2006.

[YH38] G. Young and A.S. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3:19–22, 1938.

[YPK13] Z. Yang, J. Peltonen, and S. Kaski. Scalable optimization of neighbor embedding for visualization. In *Proc. 30th International Conference on Machine Learning (ICML)*, volume 28 of *JMLR: W&CP*, pages 786–794, Atlanta, Georgia, 2013.

[YVW+05] L. Yen, D. Vanvyve, F. Wouters, F. Fouss, M. Verleysen, and M. Saerens. Clustering using a random-walk based distance measure. In M. Verleysen, editor, *Proceedings of ESANN 2005, 13th European Symposium on Artificial Neural Networks*, pages 317–324, Bruges, Belgium, April 2005. d-side.

[YWO10] Z. Yang, C. Wang, and E. Oja. Multiplicative updates for t-sne. In *Proc. 20th IEEE International Workshop on Machine Learning For Signal Processing (MLSP2010)*, pages 19–23, Kittilä (Finland), 2010.