7

Conclusions

Overview. In addition to summarizing the key points of the book, this chapter attempts to establish a generic procedure (or "data flow") for the analysis of high-dimensional data. All methods used in the previous chapters are also classified from several points of view. Next, some guidelines are given for using the various methods. The chapter ends by presenting some perspectives for future developments in the field of nonlinear dimensionality reduction.

7.1 Summary of the book

The main motivations of this book are the analysis and comparison of various DR methods, with particular attention paid to nonlinear ones. Dimensionality reduction often plays an important role in the analysis, interpretation, and understanding of numerical data. In practice, dimensionality reduction can help one to extract some information from arrays of numbers that would otherwise remain useless because of their large size. To some extent, the goal consists of enhancing the readability of data. This can be achieved by visualizing data in charts, diagrams, plots, and other graphical representations.

7.1.1 The problem

As illustrated in Chapter 1, visualization becomes problematic once the dimensionality — the number of coordinates or simultaneous observations — goes beyond three or four. Usual projections and perspective techniques already reach their limits for only three dimensions ! This suggests using other methods to build low-dimensional representations of data. Beyond visualization, dimensionality reduction is also justified from a theoretical point of view by unexpected properties of high-dimensional spaces. In high dimensions, usual mathematical objects like spheres and cubes behave strangely and do not share the same nice properties as in the two- or three-dimensional

cases. Other examples are the Euclidean norm, which is nearly useless in high-dimensional spaces, and the intrinsic sparsity of high-dimensional spaces (the "empty space phenomenon"). All those issues are usually called the "curse of dimensionality" and must be taken into account when processing high-dimensional data.

7.1.2 A basic solution

Historically, one of the first methods intended for the analysis of high-dimensional data was principal component analysis (PCA), introduced in Chapter 2. Starting from a data set in matrix form, and under some conditions, this method is able to perform three essential tasks:

- **Intrinsic dimensionality estimation.** This consists in estimating the (small) number of hidden parameters, called latent variables, that generated data.
- **Dimensionality reduction.** This consists in building a low-dimensional representation of data (a projection), according to the estimated dimensionality.
- **Latent variable separation.** This consists of a further transformation of the low-dimensional representation, such that the latent variables appear as mutually "independent" as possible.

Obviously, these are very desirable functionalities. Unfortunately, PCA remains a rather basic method and suffers from many shortcomings. For example, PCA assumes that observed variables are linear combinations of the latent ones. According to this data model, PCA just yields a linear projection of the observed variables. Additionally, the latent variable separation is achieved by simple decorrelation, explaining the quotes around the adjective "independent" in the above list.

For more than seven decades, the limitations of PCA have motivated the development of more powerful methods. Mainly two directions have been explored: namely, dimensionality reduction and latent variable separation.

7.1.3 Dimensionality reduction

Much work has been devoted to designing methods that are able to reduce the data dimensionality in a nonlinear way, instead of merely projecting data with a linear transformation. The first step in that direction was made by reformulating the PCA as a distance-preserving method. This yielded the classical metric multidimensional scaling (MDS) in the late 1930s (see Table 7.1). Although this method remains linear, like PCA, it is the basis of numerous nonlinear variants described in Chapter 4. The most widely known ones are undoubtedly nonmetric MDS and Sammon's nonlinear mapping (published in the late 1960s). Further optimizations are possible, by using stochastic techniques, for example, as in curvilinear component analysis (CCA), published

in the early 1990s. Besides this evolution toward more and more complex algorithms, recent progress has been accomplished in the family of distance-preserving methods by replacing the usual Euclidean distance with another metric: the geodesic distance, introduced in the late 1990s. This particular distance measure is especially well suited for dimensionality reduction. The unfolding of nonlinear manifolds is made much easier with geodesic distances than with Euclidean ones.

Geodesic distances, however, cannot be used as such, because they hide a complex mathematical machinery that would create a heavy computational burden in practical cases. Fortunately, geodesic distances may be approximated in a very elegant way by graph distances. To this end, it suffices to connect neighboring points in the data set, in order to obtain a graph, and then to compute the graph distances with Dijkstra's algorithm [53], for instance.

A simple change allows us to use graph distances instead of Euclidean ones in classical distance-preserving methods. Doing so transforms metric MDS, Sammon's NLM and CCA in Isomap (1998), geodesic NLM (2002), and curvilinear distance analysis (2000), respectively. Comparisons on various examples in Chapter 6 clearly show that the graph distance outperforms the traditional Euclidean metric. Yet, in many cases and in spite of all its advantages, the graph distance is not the panacea: it broadens the set of manifolds that can easily be projected by distance preservation, but it does not help in all cases. For that reason, the algorithm that manages the preservation of distances fully keeps its importance in the dimensionality reduction process. This explains why the flexibility of GNLM and CDA is welcome in difficult cases where Isomap can fail.

Distance preservation is not the sole paradigm used for dimensionality reduction. Topology preservation, introduced in Chapter 5, is certainly more powerful and appealing but also more difficult to implement. Actually, in order to be usable, the concept of "topology" must be clearly defined; its translation from theory to practice does not prove as straightforward as measuring a distance. Because of that difficulty, topology-preserving methods like Kohonen's self-organizing maps appeared later (in the early 1980s) than distance-based methods. Other methods, like the generative topographic mapping (1995), may be viewed as principled reformulations of the SOM, within a probabilistic framework. More recent methods, like locally linear embedding (2000) and Isotop (2002), attempt to overcome some limitations of the SOM.

In Chapter 5, methods are classified according to the way they model the topology of the data set. Typically, this topology is encoded as neighborhood relations between points, using a graph that connects the points, for instance. The simplest solution consists of predefining those relations, without regards to the available data, as it is done in an SOM and GTM. If data are taken into account, the topology is said to be data-driven, like with LLE and Isotop. While data-driven methods generally outperform SOMs for dimensionality reduction purposes, the latter remains a reference tool for 2D visualization.

ANN	DR	Method	Author(s) & reference(s)
	1901	PCA	Pearson [149]
	1933	PCA	Hotelling [92]
	1938	classical metric MDS	Young & Householder [208]
1943		formal neuron	McCulloch & Pitts [137]
	1946	PCA	Karhunen [102]
	1948	PCA	Loève [128]
	1952	MDS	Torgerson [182]
1958		Perceptron	Rosenblatt [157]
1959		Shortest paths in a graph	Dijkstra [53]
	1962	nonmetric MDS	Shepard [171]
	1964	nonmetric MDS	Kruskal [108]
1965		K-means (VQ)	Forgy [61]
1967		K-means (VQ)	MacQueen [61]
		ISODATA (VQ)	Ball & Hall [8]
	1969	PP	Kruskal [109]
		NLM (nonlinear MDS)	Sammon [109]
1969		Perceptron	Minsky & Papert's paper [138]
	1972	PP	Kruskal [110]
	1973	SOM	von der Malsburg [191]
	1974	PP	Friedman & Tukey [67]
1974		Back-propagation	Werbos [201]
1980		LBG (VQ)	Linde, Buzo & Gray [124]
	1982	SOM (VQ & NLDR)	Kohonen [104]
1982		Hopfield network	Hopfield [91]
		LLoyd (VQ)	Lloyd [127]
	1984	Principal curves	Hastie & Stuetzle [79, 80]
1985		Competitive learning (VQ)	Rumelhart & Zipser [162, 163]
1986		Back-propagation & MLP	Rumelhart, Hinton & Williams [161, 160]
		BSS/ICA	Jutten [99, 98, 100]
	1991	Autoassociative MLP	Kramer [107, 144, 183]
	1992	"Neural" PCA	Oja [145]
	1993	VQP (NLM)	Demartines & Hérault [46]
		Autoassociative ANN	DeMers & Cottrell [49]
	1994	Local PCA	Kambhatla & Leen [101]
	1995	CCA (VQP)	Demartines & Hérault [47, 48]
		NLM with ANN	Mao & Jain [134]
	1996	KPCA	Schölkopf, Smola & Müller [167]
		GTM	Bishop, Svensén & Williams [22, 23, 24]
	1997	Normalized cut (spectral clustering)	Shi & Malik [172, 199]
	1998	Isomap	Tenenbaum [179, 180]
	2000	CDA (CCA)	Lee & Verleysen [116, 120]
		LLE	Roweis & Saul [158]
	2002	Isotop (MDS)	Lee [119, 114]
		LE	Belkin & Niyogi [12, 13]
		Spectral clustering	Ng, Jordan & Weiss [143]
		Coordination of local linear models	Roweis, Saul & Hinton [159]
	2003	HLLE	Donoho & Grimes [56, 55]
	2004	LPP	He & Niyogi [81]
		SDE (MDS)	Weinberger & Saul [196]
	2005	LMDS (CCA)	Venna & Kaski [186, 187]
	2006	Autoassociative ANN	Hinton & Salakhutdinov [89]

Table 7.1. Timeline of DR methods. Major steps in ANN history are given as milestones. Spectral clustering has been added because of its tight relationship with spectral DR methods.

7.1.4 Latent variable separation

Starting from PCA, the other direction that can be explored is latent variable separation. The first step in that direction was made with projection pursuit (PP; see Table 7.1) [109, 110, 67]. This technique, which is widely used in exploratory data analysis, aims at finding "interesting" (linear) one- or two-dimensional projections of a data set. Axes of these projections can then be interpreted as latent variables. A more recent approach, initiated in the late 1980s by Jutten and Hérault [99, 98, 100], led to the flourishing development of blind source separation (BSS) and independent component analysis (ICA). These fields propose more recent but also more principled ways to tackle the problem of latent variable separation. In contrast with PCA, BSS and ICA can

go beyond variable decorrelation: most methods involve an objective function that can be related to statistical independence.

In spite of its appealing elegance, latent variable separation does not fit in the scope of this book. The reason is that most methods remain limited to linear data models. Only GTM can be cast within that framework: it is one of the rare NLDR methods that propose a latent variable model, i.e. one that considers the observed variables to be functions of the latent ones. Most other methods follow a more pragmatic strategy and work in the opposite direction, by finding any set of variables that give a suitable low-dimensional representation of the observed variables, regardless of the true latent variables. It is noteworthy, however, that GTM involves a nonlinear mapping and therefore offers no guarantee of recovering the true latent variables either, despite its more complex data model.

More information on projection pursuit can be found in [94, 66, 97]. For BSS and ICA, many details and references can be found in the excellent book by Hyvärinen, Karhunen, and Oja [95].

7.1.5 Intrinsic dimensionality estimation

Finally, an important key to the success of both dimensionality reduction and latent variable separation resides in the right estimation of the intrinsic dimensionality of data. This dimensionality indicates the minimal number of variables or free parameters that are needed to describe the data set without losing the information it conveys. The word "information" can be understood in many ways: it can be the variance in the context of PCA, for instance. Within the framework of manifold learning, it can also be the manifold "structure" or topology; finding the intrinsic dimensionality then amounts to determining the underlying manifold dimensionality. Chapter 3 reviews a couple of classical methods that can estimate the intrinsic dimensionality of a data set. A widely used approach consists of measuring the fractal dimension of the data set. Several fractal dimensions exist: the most-known ones are the correlation dimension and the box-counting dimension. These measures come from subdomains of physics, where they are used to study dynamical systems. Although they are often criticized in the physics literature, their claimed shortcomings do not really matter within the framework of dimensionality reduction. It is just useful to know that fractal dimensions tend to underestimate the true dimensionality [174] and that noise may pollute the estimation. But if the measure of the fractal dimension fails, then it is very likely that the data set is insufficient or too noisy and that any attempt to reduce the dimensionality will fail, too.

Other methods to estimate the intrinsic dimensionality are also reviewed in Chapter 3. For example, some DR methods can also be used to estimate the intrinsic dimensionality: they are run iteratively, with a decreasing target dimension, until they fail. The intrinsic dimensionality may then be assumed

to be equal to the smallest target dimension before failure. This is a "trial-and-error" approach. Obviously, this way of estimating the dimensionality largely depends on the method used to reduce the dimensionality. The result may vary significantly just by changing the type of dimensionality reducer (distance- or topology-preserving method, Euclidean or graph distance, etc.). Moreover, the computational cost of repeating the dimensionality reduction to obtain *merely* the dimensionality may rapidly become prohibitive. From this point of view, methods that can build projections in an incremental way (see Subsection 2.5.7), such as PCA, Local PCA, Isomap, or SDE, appear as the best compromise because a single run suffices to determine the projections of all possible dimensionalities at once. In contrast with fractal dimensions, the trial-and-error technique tends to overestimate the true intrinsic dimensionality.

Section 3.4 compares various methods for the estimation of the intrinsic dimensionality. The correlation dimension (or another fractal dimension) and local PCA give the best results on the proposed data sets. Indeed, these methods are able to estimate the dimensionality on different scales (or resolutions) and thus yield more informative results.

7.2 Data flow

This section proposes a generic data flow for the analysis of high-dimensional data. Of course, the accent is put on the word "generic": the proposed pattern must obviously be particularized to each application. However, it provides a basis that has been proven effective in many cases.

As a starting point, it is assumed that data consist of an unordered set of vectors. All vector entries are real numbers; there are no missing data.

7.2.1 Variable Selection

The aim of this first step is to make sure that all variables or signals in the data set convey useful information about the phenomenon of interest. Hence, if some variables or signals are zero or are related to another phenomenon, a variable selection must be achieved beforehand, in order to discard them. To some extent, this selection is a "binary" dimensionality reduction: each observed variable is kept or thrown away. Variable selection methods are beyond the scope of this book; this topic is covered in, e.g., [2, 96, 139].

7.2.2 Calibration

This second step aims at "standardizing" the variables. When this is required, the average of each variable is subtracted. Variables can also be scaled if needed. The division by the standard deviation is useful when the variables

come from various origins. For example, meters do not compare with kilograms, and kilometers do not with grams. Scaling the variables helps to make them more comparable.

Sometimes, however, the standardization can make things worse. For example, an almost-silent signal becomes pure noise after standardization. Obviously, the knowledge that it was silent is important and should not be lost. In the ideal case, silent signals and other useless variables are eliminated by the above-mentioned variable selection. Otherwise, if no standardization has been performed, further processing methods can still remove almost-zero variables. (See Subsection 2.4.1 for a more thorough discussion.)

7.2.3 Linear dimensionality reduction

When data dimensionality is very high, linear dimensionality reduction by PCA may be very usesul to suppress a large number of useless dimensions. Indeed, PCA clearly remains one of the best techniques for "hard" dimensionality reduction. For this step, the strategy consists in elimating the largest number of variables while maintaining the reconstruction error very close to zero. This is achieved in order to make the operation as "transparent" as possible, i.e., nearly reversible. This also eases the work to be achieved by subsequent nonlinear methods (e.g., for a further dimensionality reduction). If the dimensionality is not too high, or if linear dimensionality causes a large reconstruction error, then PCA may be skipped.

In some cases, whitening can also be used [95]. Whitening, also known as sphering, is closely related to PCA. In the latter, the data space is merely rotated, using an orthogonal matrix, and the decorrelated variables having a variance close to zero are discarded. In whitening an additional step is used for scaling the decorrelated variables, in order to end up with unit-variance variables. This amounts to performing a standardization, just as described above, after PCA instead of before. Whereas the rotation involved in PCA does not change pairwise distances in the data set, the additional transformation achieved by whitening does, like the standardization. For zero-mean variables, Euclidean distances measured after whitening are equivalent to Mahalanobis distances measured in the raw data set (with the Mahalanobis matrix being the inverse of the data set covariance matrix).

7.2.4 Nonlinear dimensionality reduction

Nonlinear methods of dimensionality reduction may take over from PCA once the dimensionality is no longer too high, between a few tens and a few hundreds, depending on the chosen method. The use of PCA as preprocessing is justified by the fact that most nonlinear methods remain more sensitive to the curse of dimensionality than PCA due to their more complex model, which involves many parameters to identify.

Typically, nonlinear dimensionality reduction is the last step in the data flow. Indeed, the use of nonlinear methods transforms the data set in such a way that latent variable separation becomes difficult or impossible.

7.2.5 Latent variable separation

In the current state of the art, most methods for latent variable separation are incompatible with nonlinear dimensionality reduction. Hence, latent variable separation appears more as an alternative step to nonlinear dimensionality reduction than a subsequent one. The explanation is that most methods for latent variable separation like ICA assume in fact that the observed variables are linear combinations of the latent ones [95]. Without that assumption, these methods may not use statistical independence as a criterion to separate the variables. Only a few methods can cope with restricted forms of nonlinear mixtures like, such as, for example, postnonlinear mixtures [205, 177].

7.2.6 Further processing

Once dimensionality reduction or latent variable separation is achieved, the transformed data may be further processed, depending on the targeted application. This can range from simple visualization to automated classification or function approximation. In the two last cases, unsupervised learning is followed by supervised learning.

In summary, the proposed generic data flow for the analysis of high-dimensional data goes through the following steps:

1. **(Variable selection.)** This step allows the suppression of useless variables.
2. **Calibration.** This step gathers all preprocessings that must or may be applied to data (mean subtraction, scaling, or standardization, etc.).
3. **Linear dimensionality reduction.** This step usually consists of performing PCA (data may be whitened at the same time, if necessary).
4. **Nonlinear dimensionality reduction and/or latent variable separation.** These (often incompatible) steps are the main ones; they allow us to find "interesting" representations of data, by optimizing either the number of required variables (nonlinear dimensionality reduction) or their independence (latent variable separation).
5. **(Further processing.)** Visualization, classification, function approximation, etc.

Steps between paratheses are topics that are not covered in this book.

7.3 Model complexity

It is noteworthy that in the above-mentioned data flow, the model complexity grows at each step. For example, if N observations of D variables or signals are

available, the calibration determines D means and D standard deviations; the time complexity to compute them is then $\mathcal{O}(DN)$. Next, for PCA, the covariance matrix contains $D(D-1)/2$ independent entries and the time complexity to compute them is $\mathcal{O}(D^2N)$. Obtaining a P-dimensional projection requires $\mathcal{O}(PDN)$ additional operations.

Things become worse for nonlinear dimensionality reduction. For example, a typical distance-preserving method requires $N(N-1)/2$ memory entries to store all pairwise distances. The time complexity to compute them is $\mathcal{O}(DN^2)$, at least for Euclidean distances. For graph distances indeed, the time complexity grows further to $\mathcal{O}(PN^2 \log N)$. In order to obtain a P-dimensional embedding, an NLDR method relying on a gradient descent such as— Sammon's NLM —requires $\mathcal{O}(PN^2)$ operations for a single iteration. On the other hand, a spectral method requires the same amount of operations per iteration, but the eigensolver has the advantage of converging much faster.

To some extent, progress of NLDR models and methods seems to be related not only to science breakthroughs but also to the continually increasing power of computers, which allows us to investigate directions that were previously out of reach from a practical point of view.

7.4 Taxonomy

Figure 7.1 presents a nonexhaustive hierarchy tree of some unsupervised data analysis methods, according to their purpose (latent variable separation or dimensionality reduction). This figure also gives an overview of all methods described in this book, which focuses on nonlinear dimensionality reduction based mainly on "geometrical" concepts (distances, topology, neighborhoods, manifolds, etc.).

Two classes of NLDR methods are distinguished in this book: those trying to preserve pairwise distances measured in the data set and those attempting to reproduce the data set topology. This distinction may seem quite arbitrary, and other ways to classify the methods exist. For instance, methods can be distinguished according to their algorithmic structure. In the latter case, spectral methods can be separated from those relying on iterative optimization schemes like (stochastic) gradient ascent/descent. Nevertheless, this last distinction seems to be less fundamental.

Actually, it can be observed that all distance-preserving methods involve pairwise distances either directly (metric MDS, Isomap) or with some kind of weighting (NLM, GNLM, CCA, CDA, SDE). In (G)NLM, this weighting is proportional to the inverse of the Euclidean (or geodesic) distances measured in the data space, whereas a decreasing function of the Euclidean distances in the embedding space is used in CCA and CDA. For SDE, only Euclidean distances to the K nearest neighbors are taken into account, while others are simply forgotten and replaced by those determined during the semidefinite programming step.

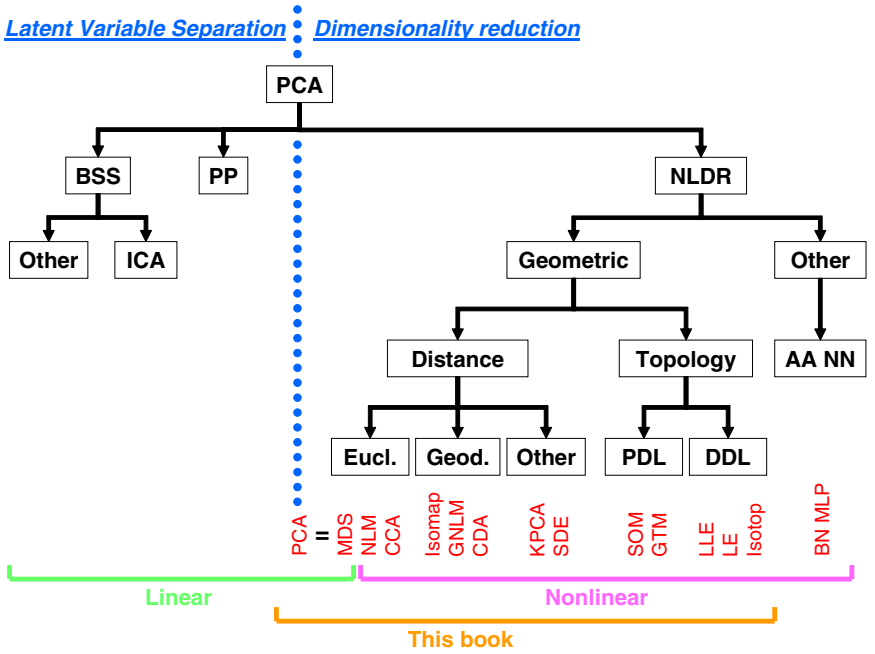

Fig. 7.1. Methods for latent variable separation and dimensionality reduction: a nonexhaustive hierarchy tree. Acronyms: PCA, principal component analysis; BSS, blind source separation; PP, projection pursuit; NLDR, nonlinear dimensionality reduction; ICA, independent component analysis; AA NN, auto-associative neural network; PDL, predefined lattice; DDL, data-driven lattice. Methods are shown as tree leaves.

On the other hand, in topology-preserving methods pairwise distances are never used directly. Instead they are replaced with some kind of similarity measure, which most of the time is a decreasing function of the pairwise distances. For instance, in LE only distances to the K nearest neighbors are involved; next the heat kernel is applied to them (possibly with an infinite temperature) and the Laplacian matrix is computed. This matrix is such that off-diagonal entries in a row or column are always lower than the corresponding entry on the diagonal. A similar reasoning leads to the same conclusion for LLE. In an SOM or Isotop, either grid distances (in the embedding space) or graph distances (in the data space) are used as the argument of a Gaussian kernel.

In the case of spectral methods, this distinction between distance and topology preservation has practical consequences. In all distance-preserving methods, the eigensolver is applied to a dense matrix, whose entries are either Euclidean distances (metric MDS), graph distances (Isomap), or distances

optimized by means of semidefinite programming (SDE). In all these methods, the top eigenvectors are sought (those associated with the largest eigenvalues). To some extent, these eigenvectors form the solution of a maximization problem; in the considered case, the problem primarily consists of maximizing the variance in the embedding space, which is directly related to the associated eigenvalues.[1]

The situation gets reversed for topology-preserving methods. Most of the time, in this case, the eigensolver is applied to a sparse matrix and the bottom eigenvectors are sought, those associated with the eigenvalues of lowest (but nonzero) magnitude. These eigenvectors identify the solution of a minimization problem. The objective function generally corresponds to a local reconstruction error or distortion measure (see Subsections 5.3.1 and 5.3.2 about LLE and LE, respectively).

A duality relationship can be established between those maximizations and minimizations involving, respectively, dense and sparse Gram-like matrices, as sketched in [164] and more clearly stated in [204]. Hence to some extent distance and topology preservation are different aspects or ways to formulate the same problem. Obviously, it should be erroneous to conclude from the unifying theoretical framework described in [204] that all spectral methods are equivalent in practice ! This is not the case; experimental results in Chapter 6 show the large variety that can be observed among their results. For instance, following the reasoning in [164, 78], it can easily be demonstrated that under some conditions, the bottom eigenvectors of a Laplacian matrix (except the last one associated with a null eigenvalue) correspond to the leading eigenvectors obtained from a double-centered matrix of pairwise commute-time distances (CTDs). It can be shown quite easily that CTDs respect all axioms of a distance measure (Subsection 4.2.1). But unlike Euclidean distances, CTDs cannot be computed simply by knowing coordinates of two points. Instead, CTD distances are more closely related to graph distances, in the sense that other known points are involved the computation. Actually, it can be shown that the easiest way to obtain the matrix of pairwise CTDs consists of computing the pseudo-inverse of a Laplacian matrix. As a direct consequence, the leading eigenvectors of the CTD matrix precisely correspond to the bottom eigenvectors of the Laplacian matrix, just as stated above. Similarly, eigenvalues of the CTD matrix are inversely proportional to those of the Laplacian matrix. Therefore, even if distance- and topology-preserving spectral methods are equivalent from a theoretical viewpoint, it remains useful to keep both frameworks in practice, as the formulation of a particular method can be made easier or more natural in the one or the other.

[1] Of course, this maximization of the variance can easily be reformulated into a minimization of the reconstruction error, as observed in Subsection 2.4.2.

7.4.1 Distance preservation

Distance-preserving methods can be classified according to the distance and the algorithm they use. Several kinds of distances and kernels can be used. Similarly, the embedding can be obtained by three different types of algorithms. The first one is the spectral decomposition found in metric MDS. The second one is the quasi-Newton optimization procedure implemented in Sammon's NLM. The third one is the stochastic optimization procedure proposed in curvilinear component analysis. Table 7.2 shows the different combinations that have been described in the literature. The table also proposes an alterna-

Table 7.2. Classification of distance-preserving NLDR methods, according to the distance/kernel function and the algorithm. A unifying naming convention is proposed (the real name of each method is given between parentheses).

	MDS algorithm	NLM algorithm	CCA algorithm
Euclidean	EMDS (metric MDS)	ENLM (NLM)	ECCA (CCA)
Geodesic (Dijkstra)	GMDS (Isomap)	GNLM	GCCA (CDA)
Commute time	CMDS (LE)		
Fixed kernel	KMDS (KPCA)		
Optimized kernel (SDP)	OMDS (SDE/MVU)		

tive naming convention. The first letter indicates the distance or kernel type (E for Euclidean, G for geodesic/graph, C for commute-time distance, K for fixed kernel, and O for optimized kernel), whereas the three next letters refer to the algorithm (MDS for spectral decomposition, NLM for quasi-Newton optimization, and CCA for CCA-like stochastic gradient descent).

It is noteworthy that most methods have an intricate name that often gives few or no clues about their principle. For instance, KPCA is by far closer to metric MDS than to PCA. While SDE stands for semidefinite embedding, it should be remarked that all spectral methods compute an embedding from a positive semidefinite Gram-like matrix. The author of SDE renamed his method MVU, standing for maximum variance unfolding [197]; this new name does not shed any new light on the method: all MDS-based methods (like Isomap and KPCA, for instance) yield an embedding having maximal variance. The series can be continued, for instance, with PCA (pricipal component analysis), CCA (curvilinear component analysis), and CDA (curvilinear distances analysis), whose names seem to be designed to ensure a kind of filiation while remaining rather unclear about their principle. The

name Isomap, which stands for Isometric feature mapping [179], is quite unclear too, since all distance-preserving NLDR methods attempt to yield an isometric embedding. Unfortunately, in most practical cases, perfect isometry is not reached.

Looking back at Table 7.2, the third and fourth rows contain methods that were initially not designed as distance-preserving methods. Regarding KPCA, the kernels listed in [167] are given for their theoretical properties, without any geometrical justification. However, the application of the kernel is equivalent to mapping data to a feature space in which a distance-preserving embedding is found by metric MDS. In the case of LE, the duality described in [204] and the connection with commute-time distances detailed in [164, 78] allow it to occupy a table entry.

Finally, it should be remarked that the bottom right corner of Table 7.2 contains many empty cells that could give rise to new methods with potentially good performances.

7.4.2 Topology preservation

As proposed in Chapter 5, topology-preserving methods fall into two classes. On one hand, methods like Kohonen's self-organizing maps and Svensén's GTM map data to a discrete lattice that is predefined by the user. On the other hand, more recent techniques like LLE and Isotop automatically build a data-driven lattice, meaning that the shape of the lattice depends on the data and is entirely determined by them. In most cases this lattice is a graph induced by k-ary or ϵ-ball neighborhoods, which provides a good discrete approximation of the underlying manifold topology. From that point of view, the corresponding methods can be qualified to be graph- or manifold-driven.

Table 7.3 is an attempt to classify topology-preserving methods as it was done in Table 7.2 for distance-preserving methods. If PCA is interpreted as

Table 7.3. Classification of topology-preserving NLDR methods, according to the kind of lattice and algorithm. The acronyms ANN, MLE and EM in the first row stand for artificial neural network, maximum likelihood estimation and expectation-maximization respectively.

	ANN-like	MLE by EM	Spectral
Predefined lattice	SOM	GTM	
Data-driven lattice	Isotop		LLE, LE

a method that fits a plane in the data space in order to capture as much

variance as possible after projection on this plane, then to some extent running an SOM can then be seen as a way to fit a nonrigid (or articulated) piece of plane within the data cloud. Isotop, on the contrary, follows a similar strategy in the opposite direction: an SOM-like update rule is used for embedding a graph deduced from the data set in a low-dimensional space.

By essence, GTM solves the problem in a similar way as an SOM would. The approach is more principled, however, and the resulting algorithm works in a totally different way. A generative model is used, in which the latent space is fixed a priori and whose mapping parameters are identified by statistical inference.

Finally, topology-preserving spectral methods, like LLE and LE, develop a third approach to the problem. They build what can be called an "affinity" matrix [31], which is generally sparse; after double-centering or application of the Laplacian operator, some of its bottom eigenvectors form the embedding. A relationship between LLE and LE is established in [13], while the duality described in [204] allows us to relate both methods to distance-preserving spectral methods.

7.5 Spectral methods

Since 2000 most recent NLDR methods have been spectral, whereas methods based on sophisticated (stochastic) gradient ascent/descent were very popular during the previous decades. Most of these older methods were designed and developed in the community of artificial neural networks (ANNs). Methods like Kohonen's SOM and Demartines' VQP [46] (the precursor of CCA) share the same distributed structure and algorithmic scheme as other well-known ANNs like the multilayer perceptron.

Since the late 1990s, the ANN community has evolved and split into two groups. The first one has joined the fast-growing and descriptive field of neurosciences, whereas the second has been included in the wide field of machine learning, which can be seen as a fundamental and theory-oriented emanation of data mining. The way to tackle problems is indeed more formal in machine learning than it was in the ANN community and often resorts to concepts coming from statistics, optimization and graph theory, for example. The growing interest in spectral methods stems at least partly from this different perspective. Spectral algebra provides an appealing and elegant framework where NLDR but also clustering or other optimization problems encountered in data analysis can be cast within. From the theoretical viewpoint, the central problem of spectral algebra, i.e., finding eigenvalues and eigenvectors of a given matrix, can be translated into a convex objective function to be optimized. This guarantees the existence of a unique and global maximum; in addition, the solutions to the problem are also "orthogonal" in many cases, giving the opportunity to divide the problem into subproblems that can be solved successively, yielding the eigenvectors one by one. Another advantage

of spectral algebra is to associate practice with theory: rather robust and efficient eigensolvers, with well-studied properties, are widely available.

This pretty picture hides some drawback however. The first to be mentioned is the "rigidity" of the framework provided by spectral algebra. In the case of NLDR, all spectral methods can be interpreted as performing metric MDS on a double-centered kernel matrix instead of a Gram matrix [31, 167, 203, 78, 15, 198, 17]. This kernel matrix is generally built in one of the following two ways:

- Apply a kernel function to the matrix of squared Euclidean pairwise distances or directly to the Gram matrix of dot products.
- Replace the Euclidean distance with some other distance function in the matrix of squared pairwise distances.

The first approach is followed in KPCA and SDE, and the second one in Isomap, for which graph distances are used instead of Euclidean ones. Eventually, the above-mentioned duality (Section 7.4) allows us to relate methods working with bottom eigenvectors of sparse matrices, like LLE, LE, and their variants, to the same scheme. Knowing also that metric MDS applied to a matrix of pairwise Euclidean distances yields a linear projection of the data set, it appears that the ability of all recent spectral methods to provide a nonlinear embedding actually relies merely on the chosen distance or kernel function. In other words, the first step of a spectral NLDR method consists of building a kernel matrix (under some conditions) or a distance matrix (with a non-Euclidean distance), which amounts to implicitly mapping data to a feature space in a nonlinear way. In the second step, metric MDS enables us to compute a linear projection from the feature space to an embedding space having the desired dimensionality. Hence, although the second step is purely linear, the succession of the two steps yields a nonlinear embedding.

A consequence of this two-step data processing is that for most methods the optimization (performed by the eigensolver) occurs in the second step only. This means that the nonlinear mapping involved in the first step (or equivalently the corresponding kernel) results from a more or less "arbitrary" user's choice. This can explain the large variety of spectral methods described in the literature: each of them describes a particular way to build a Gram-like matrix before computing its eigenvectors. A gradation of the "arbitrariness" of the data transformation can be established as follows.

The most arbitrary transformation is undoubtedly the one involved in KPCA. Based on kernel theory, KPCA requires the data transformation to be induced by a kernel function. Several kernel functions in agreement with the theory are proposed in [167], but there is no indication about which one to choose. Moreover, most of these kernels involve one or several metaparameters; again, no way to determine their optimal value is given.

In LLE, LE, Isomap, and their variants, the data transformation is based on geometric information extracted from data. In all cases, K-ary neighborhoods or ϵ-balls are used to induce a graph whose edges connect the data

points. This graph, in turn, brings a discrete approximation of the underlying manifold topology/structure. In that perspective, it can be said that LLE, LE and Isomap are graph- or manifold-driven. They induce a "data-dependent" kernel function [15]. In practice, this means that the kernel value stored at row i and column j in the Gram-like matrix does not depend only on the ith and jth data points, but also on other points in the data set [78]. Since the Gram-like matrix built by those methods is an attempt to take into account the underlying manifold topology, it can be said the induced data mapping is less "arbitrary" than in the case of KPCA. Nevertheless, the Gram-like matrix still depends on K or ϵ; changing the value of these parameters modifies the induced data mapping, which may can lead to a completely different embedding. LLE seems to be particularly sensitive to these parameters, as witnessed in the literature [166, 90]. In this respect, the numerical stability of the eigenvectors computed by LLE can also be questioned [30] (the tail of the eigenspectrum is flat, i.e., the ratio of successive lowest-amplitude eigenvalues is close to one).

The highest level in the proposed gradation is eventually reached by SDE. In contrast with all other spectral methods, SDE is (currently) the only one that optimizes the Gram-like matrix before the metric MDS step. In Isomap, Euclidean distances within the K-ary neighborhoods are kept (by construction, graph distances in that case reduce to Euclidean distances), whereas distances between non-neighbors are replaced with graph distances, assuming that these distances subsequently lead to a better embedding. In SDE, distances between nonneighbors can be seen as parameters whose value is optimized by semidefinite programming. This optimization scheme is specifically designed to work in matrix spaces, accepts equality or inequality constraints on the matrix entries, and ensures that their properties are maintained (e.g., positive or negative semidefiniteness). An additional advantage of semidefinite programming is that the objective function is convex, ensuring the existence of a unique global maximum. In the case of a convex developable manifold, if the embedding provided by SDE is qualified to be optimal, then Isomap yields a good approximation of it although it is generally suboptimal [204] (see an example in Fig. 6.6). In the case of a nonconvex or nondevelopable manifold, the two methods behave very differently.

In summary, the advantages of spectral NLDR methods are as follows: they benefit from a strong and sound theoretical framework; eigensolvers are also efficient, leading to methods that are generally fast. Once the NLDR problem is cast within the framework, a global optimum of the objective function can be reached without implementing any "exotic" optimization procedure: it suffices to call an eigensolver, which can be found in many software toolboxes or libraries. However, it has been reported that Gram-like matrices built from sparse graphs (e.g., K-ary neighborhoods) can lead to ill-conditioned and/or ill-posed eigenproblems [30]. Moreover, the claim that spectral methods provide a global optimum is true but hides the fact that the actual nonlinear

transformation of data is not optimized, except in SDE. In the last case, the kernel optimization unfortunately induces a heavy computational burden.

7.6 Nonspectral methods

Experimentally, nonspectral methods reach a better tradeoff between flexibility and computation time than spectral methods. The construction of methods like NLM or CCA/CDA consists of defining an objective function and then choosing an adequate optimization technique. This goes in the opposite direction of the thought process behind spectral methods, for which the objective function is chosen with the a priori idea to translate it easily into an eigenproblem.

Hence, more freedom is left in the design of nonspectral methods than in that of spectral ones. The price to pay is that although they are efficient in practice, they often give few theoretical guarantees. For instance, methods relying on a (stochastic) gradient ascent/decent can fall in local optima, unlike spectral methods. The optimization techniques can also involve metaparameters like step sizes, learning rates, stopping criteria, tolerances, and numbers of iterations. Although most of these parameters can be left to their default values, some of them can have a nonnegligible influence on the final embedding.

This is the case of the so-called neighborhood width involved in CCA/CDA, SOMs and Isotop. Coming from the field of artificial neural networks, these methods all rely on stochastic gradient ascent/descent or resort to update rules that are applied in the same way. As usual in stochastic update rules, the step size or learning rate is scheduled to slowly decrease from one iteration to the other, in order to reach convergence. In the above-mentioned algorithms, the neighborhood width is not kept constant and follows a similar schedule. This makes their dynamic behavior quite difficult to analyze, since the optimization process is applied to a varying objective function that depends on the current value of the neighborhood width. Things are even worse in the cases of SOMs and Isotop, since the update rules are empirically built: there exists no objective function from which the update rules can be derived. Since the objective function, when it exists, is nonconstant and depends on the neighborhood width, it can be expected that embeddings obtained with these methods will depend on the same parameter, independently from the fact that the method can get stuck in local optima.

The presence of metaparameters like the neighborhood width can also be considered an advantage, to some extent. It does provide additional flexibility to nonspectral methods in the sense that for a given method, the experienced user can obtain different behaviors just by changing the values of the metaparameters.

Finally, it is noteworthy that the respective strategies of spectral and nonspectral NLDR methods completely differ. Most spectral methods usually

transform data (nonlinearly when building the Gram-like matrix, and then linearly when solving the eigenproblem) before pruning the unnecessary dimensions (eigenvectors are discarded). In contrast, most nonspectral methods start by initializing mapped data vectors in a low-dimensional space and then rearrange them to optimize some objective function.

7.7 Tentative methodology

Throughout this book, some examples and applications have demonstrated that the proposed analysis methods efficiently tackle the problems raised by high-dimensional data. This section is an attempt to guide the user through the large variety of NLDR methods described in the literature, according to characteristics of the available data.

A first list of guidelines can be given according to the shape, nature, or properties of the manifold to embed. In the case of ...

- **slightly curved manifolds.** Use a linear method like PCA or metric MDS; alternatively, NLM offers a good tradeoff between robustness and reproducibility and gives the ability to provide a nonlinear embedding.
- **convex developable manifolds.** Use methods relying on geodesic/graph distances (Isomap, GNLM, CDA) or SDE. Conditions to observe convex and developable manifolds in computer vision are discussed in [54].
- **nonconvex developable manifolds.** Do not use Isomap; use GNLM or CDA instead; SDE works well, too.
- **nearly developable manifolds.** Do not use Isomap or SDE; it is better use GNLM or CDA instead.
- **other manifolds.** Use GNLM or preferably CDA. Topology-preserving methods can be used too (LLE, LE, Isotop).
- **manifolds with essential loops.** Use CCA or CDA; these methods are able to tear the manifold, i.e., break the loop. The tearing procedure proposed in [121] can also break essential loops and make data easier to embed with graph-based methods (Isomap, SDE, GNLM, CDA, LLE, LE, Isotop).
- **manifolds with essential spheres.** Use CCA or CDA. The abovementioned tearing procedure is not able to "open" essential spheres.
- **disconnected manifolds.** This remains an open question. Most methods do not explicitly handle this case. The easiest solution is to build an adjacency graph, detect the disconnected components, and embed them separately. Of course, "neighborhood relationships" between the components are lost in the process.
- **clustered data.** In this case the existence of one or several underlying manifolds must be questioned. If the clusters do not have a low intrinsic dimension, the manifold assumption is probably wrong (or useless). Then use clustering algorithms, like spectral clustering or preferably other techniques like hierarchical clustering.

Guidelines can also be given according to the data set's size. In the case of ...

- **Large data set.** If several thousands of data points are available ($N > 2000$), most NLDR methods will generate a heavy computational burden because of their time and space complexities, which are generally proportional to N^2 (or even higher for the computation time). It is then useful to reduce the data set's size, at least to perform some preliminary steps. The easiest way to obtain a smaller data set consists of resampling the available one, i.e., drawing a subset of points at random. Obviously, this is not an optimal way, since it is possible, as ill luck would have it, for the drawn subsample not to be representative of the whole data set. Some examples throughout this book have shown that a representative subset can be determined using vector quantization techniques, like K-means and similar methods.
- **Medium-sized set.** If several hundreds of data points are available ($200 < N \leq 2000$), most NLDR methods can be applied directly to the data set, without any size reduction.
- **Small data set.** When fewer than 200 data points are available, the use of most NLDR methods becomes questionable, as the limited amount of data could be insufficient to identify the large number of parameters involved in many of these methods. Using PCA or classical metric MDS often proves to be a better option.

The dimensionality of data, along with the target dimension, can also be taken into account. In case of a ...

- **very high data dimensionality.** For more than 50 dimensions ($D > 50$), NLDR methods can suffer from the curse of dimensionality, get confused, and provide meaningless results. It can then be wise first to apply PCA or metric MDS in order to perform a hard dimensionality reduction. These two methods can considerably decrease the data dimensionality without losing much information (in terms of measured variance, for instance). Depending on the data set's characteristics, PCA or metric MDS can also help attenuate statistical noise in data. After PCA/MDS, a nonlinear method can be used with more confidence (see the two next cases) in order to further reduce the dimensionality.
- **high data dimensionality.** For a few tens of dimensions ($5 < D \leq 50$), NLDR methods should be used with care. The curse of dimensionality is already no longer negligible.
- **low data dimensionality.** For up to five dimensions, any NLDR method can be applied with full confidence.

Obviously, the choice of the target dimensionality should take into account the intrinsic dimensionality of data if it is known or can be estimated.

- If the target dimensionality is (much) higher than the intrinsic one, PCA or MDS performs very well. These two methods have numerous advantages:

they are simple, fast, do not fall in local optima, and involve no parameters. In this case, even the fact that they transform data in a linear way can be considered an advantage in many respects.

- If the target dimensionality is equal to or hardly higher than the intrinsic one, NLDR methods can yield very good results. Most spectral or nonspectral methods work quite well in this case. For highly curved manifolds, one or two supernumerary dimensions can improve the embedding quality. Most NLDR methods (and especially those based on distance preservation) have limited abilities to deform/distort manifolds. Some extra dimensions can then compensate for this lack of "flexibility." The same strategy can be followed to embed manifolds with essential loops or spheres.
- If the target dimensionality is lower than the intrinsic one, such as for visualization purposes, use NLDR methods at your own risk. It is likely that results will be meaningless since the embedding dimensionality is "forced." In this case, most nonspectral NLDR methods should be avoided. They simply fail to converge in an embedding space of insufficient dimensionality. On the other hand, spectral methods do not share this drawback since they solve an eigenproblem independently from the target dimensionality. This last parameter is involved only in the final selection of eigenvectors. Obviously, although an embedding dimensionality that is deliberately too low does not jeopardize the method convergence, this option does not guarantee that the obtained embedding is meaningful either. Its interpretation and/or subsequent use must be questioned.

Here is a list of additional advices related to the application's purpose and other considerations.

- Collect information about your data set prior to NLDR: estimate the intrinsic dimensionality and compute an adjacency graph in order to deduce the manifold connectivity.
- Never use any NLDR method without knowing the role and influence of all its parameters (true for any method, with a special emphasis on nonspectral methods).
- For 2D visualization and exploratory data analysis, Kohonen's SOM remains a reference tool.
- Never use KPCA for embedding purposes. The theoretical framework hidden behind KPCA is elegant and appealing; it paved the way toward a unified view of all spectral methods. However, in practice, the method lacks a geometrical interpretation that could help the user choose useful kernel functions. Use SDE instead; this method resembles KPCA in many respects, and the SDP step implicitly determines the optimal kernel function for distance preservation.
- Never use SDE with large data sets; this method generates a heavy computational burden and needs to run on much more powerful computers than alternative methods do.

- Avoid using GTM as much as possible; the method involves too many parameters and is restricted to 1D or 2D rectangular latent spaces; the mapping model proves to be not flexible enough to deal with highly curved manifolds.
- LLE is very sensitive to its parameter values (K or ϵ, and the regularization parameter Δ). Use it carefully, and do not hesitate to try different values, as is done in the literature [166].
- Most nonspectral methods can get stuck in local optima: depending on the initialization, different embeddings can be obtained.
- Finally, do not forget to assess the embedding quality using appropriate criteria [186, 185, 9, 74, 10, 190, 103] (see an example in Subsection 6.3.1).

The above recommendations leave the following question unanswered: given a data set, does one choose between distance and topology preservation? If the data set is small, the methods with the simplest models often suit the best (e.g., PCA, MDS, or NLM). With mid-sized data sets, more complex distance-preserving methods like Isomap or CCA/CDA often provide more meaningful results. Topology-preserving methods like LLE, LE, and Isotop should be applied to large data sets only. Actually, the final decision between distance and topology preservation should then be guided by the shape of the underlying manifold. Heavily crumpled manifolds are more easily embedded using topology preservation rather than distance preservation. The key point to know is that both strategies extract neither the same kind nor the same amount of information from data. Topology-preserving methods focus on local information (neighborhood relationships), whereas distance-preserving ones exploit both the local and global manifold structure.

7.8 Perspectives

During the 1900s, dimensionality reduction went through several eras. The first era mainly relied on spectral methods like PCA and then classical metric MDS. Next, the second era consisted of the generalization of MDS into nonlinear variants, many of them being based on distance or rank preservation and among which Sammon's NLM is probably the most emblematic representative. At the end of the century, the field of NLDR was deeply influenced by "neural" approaches; the autoassociative MLP and Kohonen's SOM are the most prominent examples of this stream. The beginning of the new century witnessed the rebirth of spectral approaches, starting with the discovery of KPCA.

So in which directions will the researchers orient their investigations in the coming years ? The paradigm of distance preservation can be counted among the classical NLDR tools, whereas no real breakthrough has happened in topology preservation since the SOM invention. It seems that the vein of spectral methods has now been largely exploited. Many recent papers dealing with that topic do not present new methods but are instead surveys that

summarize the domain and explore fundamental aspects of the methods, like their connections or duality within a unifying frameworks. A recent publication in *Science* [89] describing a new training technique for auto-associative MLP could reorient the NLDR research toward artificial neural networks once again, in the same way as the publication of Isomap and LLE in the same journal in 2000 lead to the rapid development of many spectral methods. This renewed interest in ANNs could focus on issues that were barely addressed by spectral methods and distance preservation: large-scale NLDR problems (training samples with several thousands of items), "out-of-sample" generalization, bidirectional mapping, etc.

A last open question regards the curse of dimensionality. An important motivation behind (NL)DR aims at avoiding its harmful effects. Paradoxically, however, many NLDR methods do not bring a complete solution to the problem, but only dodge it. Many NLDR methods give poor results when the intrinsic dimensionality of the underlying manifold exceeds four or five. In such cases, the dimension of the embedding space becomes high enough to observe undesired effects related to the curse of dimensionality, such as the empty space phenomenon. The future will tell whether new techniques will be able to take up this ultimate challenge.