Overview

Distance-preserving dimensionality reduction

Li Yang*

This paper presents an overview of basic concepts and principles that deal with the problem of mapping high-dimensional data to low-dimensional space such that distances between all or some pairs of data points are preserved. It introduces related techniques and systematizes today's methods into linear methods, methods using iterative optimization, methods preserving exact distances, methods using geodesic distances, and methods using alignments of local models. It discusses these methods by focusing on their basic ideas, by summarizing their common features and differences, and by comparing their strengths and weaknesses. This paper assumes no familiarity with dimensionality reduction. The main text should be readable by people with little technical background. Technical information of important algorithms is briefly presented in sidebars, the reading of which assumes basics in statistics and matrix computation. © 2011 John Wiley & Sons, Inc. *WIREs Data Mining Knowl Discov* 2011 1 369–380 DOI: 10.1002/widm.39

INTRODUCTION

Dimensionality reduction is an important step for data preprocessing in data mining and knowledge discovery. The problem is defined as follows: given a set of high-dimensional data points, project them to a low-dimensional space so that the resulting transformed data perform more efficiently and more effectively than the original data in further processing such as classification, clustering, indexing, and searching. Dimensionality reduction contributes to the preprocessing of high-dimensional data in two important ways: first, it gives a low-dimensional representation of high-dimensional data for the purpose of effective and efficient data analysis; second, it provides a way to visualize the data, enables human intervention into the data mining process, and helps to select appropriate data mining techniques and parameters for further processing. Dimensionality reduction is also believed to be fundamental to human perception: an image can be thought as a point in high-dimensional space; although the input dimensionality is very high (e.g., 4096 for a small image with size of 64 × 64), the perceptually meaningful structure of a sequence of such images has probably much fewer independent degrees of freedom.

In principle, the existence of a data set is independent of a coordinate system within which the data set is represented. As a representation of the data set, data coordinates and data dimensionality do not directly reflect the information the data set conveys. We may visualize the situation by using the analogy of a physical structure, which has a strut between every pair of points. The structure is rigid and should not be distorted whatever the data are represented by high-dimensional coordinates or by low-dimensional coordinates. The structure is what we are interested in while the data coordinate is merely a representation. As measures of dissimilarities between data observations, distances between pairs of data points reflect fundamental information in the data set and should, in principle, be preserved by dimensionality reduction techniques. Therefore, it is the job of a distance-preserving dimensionality reduction method to represent the data in a low-dimensional space while preserving the distances between pairs of data points.

In reality, it is rarely possible to project a high-dimensional data set to a low-dimensional space without change of any distances. Therefore, the problem of dimensionality reduction becomes how to project the data into a low-dimensional space with the goal of best preservation of distances. Distance-preserving methods for dimensionality reduction can be roughly classified into linear methods and nonlinear methods. Linear methods refer to processes that derive new dimensions of the data based on linear transformations of the original dimensions. A celebrated linear method

*Correspondence to: li.yang@wmich.edu

Department of Computer Science, Western Michigan University, Kalamazoo, MI, USA.

DOI: 10.1002/widm.39

is principal component analysis (PCA),[1] which maps data to a lower dimensional space such that variance of the data is maximized. Nonlinear methods often have the explicit goal of preserving distances. One commonly used idea is to construct a cost function, which measures difference between the distances in the input space and the corresponding distances in the output space, and project the data to a low-dimensional space by minimizing the cost function. Recently, research on manifold learning for dimensionality reduction has become active. One way to understand this is to assume that the data are located on a manifold in high-dimensional space and geodesic distance along the manifold provides better measurement of dissimilarity between pairs of the data points than Euclidean distances. Using the analogy of a folded newspaper, the paper needs to be unfolded to reveal its contents.

The above constitutes major ideas used for distance-preserving dimensionality reduction. This paper introduces related techniques and systematizes major methods using these ideas. Throughout the paper, we assume that all dimensions (variables, features) are numerical and do not consider the practical issue of how to define distance measures for categorical variables and how to reduce the dimensionality of data with mixed categorical and numerical variables. In addition to using data coordinates as input, algorithms should also be able to directly accept distances as input, in which case all we need to do is to find a data configuration in Euclidean space with a fixed dimensionality such that the input distances are best preserved. These algorithms are often in duality with techniques using data coordinates as input and are also considered as dimensionality reduction techniques. They have applications in areas such as psychometrics and perceptual mapping where the collected data are interpoint distances.

BOX 1: THE INPUT DATA

We are given n data observations $\{x_1, \ldots, x_n\}$, each being a column vector $x_i = [x_{i1}, \ldots, x_{ip}]^{\mathrm{T}}$, representing a point in p-dimensional Euclidean space. The data set has a mean $\mu = E(x_i)$ and a covariance matrix $\Sigma_{p\times p} = E\{(x_i - \mu)(x_i - \mu)^{\mathrm{T}}\}$. For easy representation, we denote the input data in a matrix form $X_{n\times p} = [x_i, \ldots, x_n]^{\mathrm{T}}$. Assume the data is centered, that is, $\mu = 0$, the covariance matrix can be expressed as $\Sigma = \frac{1}{n}X^{\mathrm{T}}X$. A popular measure of distance between x_i and x_j is the Minkowski distance, which is defined as $d_{ij} = (|x_{i1} - x_{j1}|^c + \cdots + |x_{ip} - x_{jp}|^c)^{1/c}$, where c is a constant. When $c = 2$, it is the usual Euclidean distance. When $c = 1$, it is called Manhattan distance or city distance.

TRADITIONAL METHODS

The two commonly used strategies for dimensionality reduction are feature selection and feature extraction. Feature selection reduces the dimensionality by selecting a subset of dimensions. Clearly, distance preservation is not an objective of feature selection. Feature extraction goes a step further by extracting new dimensions from input data based on transformations or combinations of the original dimensions. Distance-preserving methods for dimensionality reduction are feature extraction methods. Traditional distance-preserving methods can be classified into two categories: linear methods and nonlinear methods. A linear method is characterized by a projection matrix that linearly transforms input data points to a low-dimensional space. As an example, PCA has the distinction of being the optimal linear transformation such that projected data in the subspace have the largest variances. From a distance-preservation perspective, it offers the best linear mapping such that the sum of changes of squared distances is minimized. Nonlinear methods are often derived either through extensions of linear methods or by minimizing cost functions that are defined explicitly in terms of change of distances. These methods often try to preserve all distances and usually end with the case where no distance is exactly preserved. Nonlinear methods are also developed to explicitly retain some exact distances.

Linear Methods

Linear methods include PCA, classical multidimensional scaling (classical MDS)[2] that take interpoint distances as inputs and are equivalent to PCA, higher-order generalizations such as independent component analysis (ICA),[3] and random projection where the projection matrix is randomly chosen. In various fields, PCA is also known as Karhunen–Loève transform or Hotelling transform.

PCA offers an orthogonal linear transformation that projects the data to a low-dimensional space. It seeks to reduce the dimensionality of the data by finding a few orthogonal linear combinations of the original dimensions such that the greatest variance by any projection of the data comes to lie on the first coordinate, the second greatest variance on the second coordinate, and so on, as shown in Figure 1. PCA achieves this through eigenvalue decomposition of the covariance matrix. For many data sets, a few principal coordinates would explain most of the data

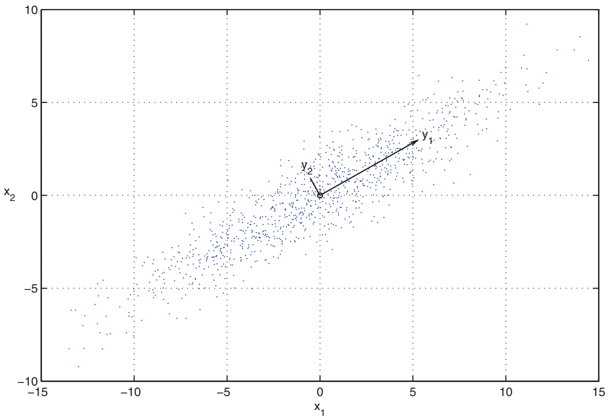

FIGURE 1 | Principal components of a Gaussian distribution.

variances so that the rest of the coordinates can be disregarded with minimal information loss.

Classical MDS is a technique that takes interpoint distances as input and derives data coordinates through eigenvalue decomposition of the inner product matrix (Gram matrix) of all data points. Because distances are invariant to translations, we assume that the data set is centered at the origin in order to derive a unique inner product matrix. The inner product matrix is obtained from the input distances through a bidirectional centering process. Classical MDS is equivalent to PCA. In fact, when Euclidean distances are used as input distances, PCA and classical MDS produce identical results.

Being based on eigenvalue decomposition of the covariance matrix, PCA is a second-order method which projects the data to uncorrelated dimensions. One step further, ICA[4] has the goal to produce statistically independent dimensions. This is a stronger requirement involving higher-order statistics. In order for the problem to have a solution, the input data set has to be indeed generated by a linear combination of independent components. In practice, ICA is usually implemented as an extra unitary transformation after the application of PCA.

Random projection[5] is a simple and computationally efficient approximation to the above linear methods. In random projection, the linear transformation is randomly chosen. For distance preservation, we can constrain the projection matrix such that its columns have unit lengths. The key idea of random projection comes from the Johnson–Lindenstrauss Lemma: if data points in a vector space are projected onto a randomly selected subspace of suitable high dimension, the distances between the points are approximately preserved. Strictly speaking, random projection is not a projection because the projection matrix may not be orthogonal. When the dimensionality is high, however, vectors having random directions might be sufficiently close to being orthogonal with each other and a random projection would well approximate an orthogonal projection.

Kernel PCA

Kernel principal component analysis (kernel PCA)[6] is a nonlinear extension of PCA using a technique called kernel method. It is equivalent to mapping the data to a very high (up to infinite) dimensional space, namely, reproducing kernel Hilbert space (RKHS), and applying PCA in the RKHS. According to Mercer's Theorem, the inner product of two mapped points can be expressed as a kernel function of the two points. Therefore, any algorithm that solely depends on the inner products between pairs of input vectors can be transformed to a kernelized version by using a kernel function to replace the otherwise intractable inner product operation. The kernelized version is equivalent to the algorithm operating in an RKHS. Because the kernel function is used, however, mapping to the RKHS is never explicitly computed. As we have discussed, classical MDS is such an algorithm that involves only inner products of the input data. By replacing the inner product with a kernel operation, classical MDS is extended to a nonlinear version, which effectively performs PCA after a nonlinear mapping of the input data to the RKHS. Because of the nonlinear mapping, distance preservation is not an objective of kernel PCA although PCA offers distance preservation in the RKHS.

Distance Preservation by Iterative Optimization

Being linear methods, PCA and classical MDS preserve distances while reducing data dimensionality under the assumption that the data are distributed on or close to a hyperplane in high-dimensional space. Such an assumption is usually too restrictive in many applications. Therefore, methods have been developed to generate nonlinear maps with the explicit objective of distance preservation.

Many of these methods belong to an area called multidimensional scaling. Multidimensional scaling originated from the earlier study in psychometrics where researchers were interested in giving quantitative scores to subjects and dimensionalities to abstract concepts. It searches for a configuration of data points in a low-dimensional space such that the Euclidean distances between the points in the space match the original dissimilarities as much as possible. Thus it is most often used to project data when only interpoint

BOX 2: PCA AND CLASSICAL MDS

Assume the input data X is centered at the origin, that is, it has a zero mean $\mu = 0$, its covariance matrix $\Sigma = \frac{1}{n}X^{\mathrm{T}}X$ can be decomposed as $\Sigma = \frac{1}{n}V\Lambda V^{\mathrm{T}}$, where $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_p)$ is a diagonal matrix of eigenvalues of Σ and $V = [\upsilon_1, \dots, \upsilon_p]$ is a matrix whose column vectors are the corresponding normalized eigenvectors. X can then be transformed into $Y = XV$. It is easy to verify that Y has zero mean $\mu_Y = 0$ and a diagonal covariance matrix $\Sigma_Y = \frac{1}{n}Y^{\mathrm{T}}Y = \frac{1}{n}\Lambda$. Without loss of generality, we list the eigenvalues of Σ in a nonincreasing order, that is, $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$, columns of Y have nonincreasing variances. To project the data to a q-dimensional space, we can simply keep the first q coordinates, that is, the first q columns in Y, and discard the rest coordinates. It can be shown that the subspace spanned by the first q eigenvectors has the smallest mean square derivation from X among all subspaces of dimension q. This procedure is well known as principal component analysis.

Classical MDS takes distances between pairs of data points as input and produces identical results as PCA if the input distances are Euclidean distances. Suppose we are given a matrix $D_{n\times n} = \{\delta_{ij}^2\}$, where $\delta_{ij}^2 = (x_i - x_j)^{\mathrm{T}}(x_i - x_j)$ is the squared Euclidean distance between x_i and x_j. Assume that the input data X is centered at the origin, that is, $\frac{1}{n}e^{\mathrm{T}}X = 0$, where $e = [1, \dots, 1]^{\mathrm{T}}$ is a column vector of n ones. The inner product matrix (Gram matrix) $G_{n\times n} = XX^{\mathrm{T}}$ can then be obtained from D using a bidirectional centering process

$$G = -\frac{1}{2}\left(1 - \frac{1}{n}ee^{\mathrm{T}}\right) D \left(1 - \frac{1}{n}ee^{\mathrm{T}}\right).$$

By definition, G is symmetric and positive semidefinite. It has a rank of at most p. In a similar way as decomposition of the covariance matrix used in PCA, G can be decomposed as $G = UKU^{\mathrm{T}}$ where $K = \mathrm{diag}(k_1, \dots, k_p)$ is a diagonal matrix of eigenvalues of G and $U = [u_1, \dots, u_p]$ is a matrix whose column vectors are the corresponding normalized eigenvectors. The input data X can then be recovered as $Y = UK^{1/2}$. Without loss of generality, we can list the eigenvalues of G in a nonincreasing order, $k_1 \geq k_2 \geq \cdots \geq k_p$. To project the data to a q-dimensional space, we can simply keep the first q columns in Y, which are produced by $[u_1, \dots, u_q]\mathrm{diag}(k_1^{1/2}, \dots, k_q^{1/2})$. It can be shown that classical MDS gives an optimal linear solution in the sense that the sum of differences between the resulting squared Euclidean distances and $\{\delta_{ij}^2\}$'s is minimum. In fact, the squared distance d_{ij}^2 between x_i and x_j after the projection can be expressed as $d_{ij}^2 = \sum_{l=1}^{p} \kappa_l (u_{li} - u_{lj})^2$, where u_{li} and u_{lj} are the ith and the jth coordinates of the eigenvector u_l, respectively. If the eigenvalues $\kappa_{q+1}, \dots, \kappa_p$ are very small, their contributions to d_{ij}^2 can be neglected. If the input data are indeed distributed on a q-dimensional hyperplane, in particular, $\kappa_{q+1}, \dots, \kappa_n$ are zero and the input distances are exactly preserved.

Classical MDS is equivalent to PCA and produces identical results when the input distances are Euclidean distances. This is because that nonzero eigenvalues of XX^{T} are the same as nonzero eigenvalues of $X^{\mathrm{T}}X$. In fact, if λ is an eigenvalue of $X^{\mathrm{T}}X$ and υ is the corresponding eigenvector, that is, $X^{\mathrm{T}}X\upsilon = \lambda\upsilon$, then we have $XX^{\mathrm{T}}X\upsilon = \lambda X\upsilon$, which means that λ is also an eigenvalue of XX^{T} and $X\upsilon$ is the corresponding eigenvector. Therefore, $n\Sigma$ and G share the same set of nonzero eigenvalues, that is, $\Lambda = K$. U can be obtained by normalizing columns of XV, that is, $U = XV\Lambda^{-1/2}$. Inversely, V can be obtained by normalizing columns of $X^{\mathrm{T}}U$, that is, $V = X^{\mathrm{T}}UK^{-1/2}$. In fact, we have $Y = XV = UK^{1/2}$.

ICA goes one step further than PCA and demands that features are statistically independent. In addition to zero cross-correlation achieved by PCA, this demand is equivalent to the requirement that all higher-order cross-cummulants are also zero. In real world applications, it is often sufficient to check up to the fourth-order cummulants. We may further assume that the data distribution is symmetric, making all odd-order cummulants zero. Therefore, a common method for ICA has two steps: the first step performs PCA on the input data; the second step computes a unitary transformation matrix so that the fourth-order cross-cummulants of the transformed data are close to zero. This problem is to diagonalize a four-dimensional array of the fourth-order cummulants. It is equivalent to an optimization problem that searches for a unitary transformation that maximizes the sum of squares of the fourth-order auto-cummulants.

dissimilarities are available. When the data coordinates are available, multidimensional scaling can also be used by calculating interpoint distances and by creating a low-dimensional configuration of points to best preserve the distances.

To minimize distance changes, a simple idea is to define an error measure of distance changes and then design a procedure to minimize the measure. For the purpose of distance preservation, the error measure is usually defined as a weighted sum of differences between distances in the input space and the corresponding distances in the output space. Because of the complexity of the error measure, there is usually no closed form solution to the minimization problem. A method using this idea usually employs an iterative procedure to minimize the error measure. With regard

BOX 3: KERNEL PCA

Given a data set $X = [x_1, \ldots, x_n]^{\mathrm{T}}$. For each x_i, we make an implicit mapping $\phi(x_i)$ into an RKHS. Denote $\Phi(X) = [\phi(x_1), \ldots, \phi(x_n)]^{\mathrm{T}}$. Following classical MDS, the inner product matrix $\Phi(X)\Phi^{\mathrm{T}}(X)$ in the RKHS can be decomposed as $\Phi(X)\Phi^{\mathrm{T}}(X) = U K U^{\mathrm{T}}$, where $K = \mathrm{diag}(\kappa_1, \ldots, \kappa_n)$ is a diagonal matrix of eigenvalues and $U = [u_1, \ldots, u_n]$ is a matrix whose column vectors are the corresponding normalized eigenvectors. A nonlinear configuration of X can be given as $Y = U K^{1/2}$. Without loss of generality, let $\kappa_1 \geq \kappa_2 \geq \ldots \geq \kappa_n$ and $\kappa_1, \ldots, \kappa_q$ be the first q dominant eigenvalues, a q-dimensional configuration of X is given as $[u_1, \ldots, u_q]\,\mathrm{diag}(\kappa_1^{1/2}, \ldots, \kappa_q^{1/2})$. Furthermore, let the covariance matrix $\Sigma = \frac{1}{n}\Phi^{\mathrm{T}}(X)\Phi(X)$ be decomposed into $V \Lambda V^{\mathrm{T}}$ in the RKHS. From our previous discussion on the equivalence between PCA and classical MDS, V can be expressed as $V = \Phi^{T}(X) U K^{-1/2}$. Therefore, an arbitrary input data point x is projected to

$$\begin{aligned} V^{\mathrm{T}}\phi(x) &= K^{-1/2}U^{\mathrm{T}}\Phi(X)\phi(x) \\ &= K^{-1/2}U^{\mathrm{T}}[\phi^{\mathrm{T}}(x)\phi(x_1), \ldots, \phi^{\mathrm{T}}(x)\phi(x_n)]^{\mathrm{T}}. \end{aligned}$$

Please note that $\phi(.)$ appears in pairs and only inner products of the mapped data points are involved. By Mercer's theorem, $\phi^{\mathrm{T}}(x)\phi(x_i)$ can be replaced by a kernel function $K(x, x_i)$. In this way, the inner product matrix $\Phi(X)\Phi^{\mathrm{T}}(X)$ and the projection of an arbitrary input data point can be computed without explicitly calculating $\phi(x)$.

to this method, one has to make decisions on how to define the error measure and how to minimize it. The basic questions are which distances to preserve and how techniques and algorithms work to preserve them.

Example algorithms include Kruskal's metric multidimensional scaling (MDS),[7] Sammon's nonlinear mapping (NLM),[8] and curvilinear component analysis (CCA).[9] These algorithms work in similar ways: Each algorithm starts with an estimated configuration of points in the output space. It uses an error measure that is defined as a function of differences between input distances and the corresponding output distances. Several optimization techniques can be used to iteratively reconfigure the coordinates of points to minimize the error measure. These techniques include the gradient descent algorithm and its improvements such as Newton's algorithm and conjugate-gradient algorithm. They are all iterative refinement algorithms. Each algorithm usually stops when the error measure falls below a user-defined threshold or the number of iterations exceeds a user-specified limit. The difference between them is the amount of adjustment to be made in each iteration and thus the speed of convergence to the final result. These iterative optimization techniques share a well-known deficiency: they may stop at local optima. To find the global optima, they are often combined with simulated annealing or genetic algorithms to escape from the local optima. In many cases, they produce better preservation of distances (especially short distances) than linear methods.

Exact Preservation of Some Distances

In a q-dimensional space, a point can be mapped to a location such that its Euclidean distances to other q points are exactly preserved. There are two candidate locations for the point. We can choose one location which better preserves the distance to an extra reference point. For example, a data set can be projected to two-dimensional so that each point preserves distances to two other points and minimizes the change of distances to an extra point. This idea has been used in a triangulation mapping technique[11] to map data to two-dimensional and in a method[12] for mapping data to an arbitrary low-dimensional space. Whenever a new point is mapped, its distances to a few points previously mapped are exactly preserved. The question here is which distances to preserve and in which sequence to map data points.

Among the distances to be preserved from a point, one could select the distance to the nearest neighbor among the mapped points. This can be done by constructing a minimal spanning tree, which is a spanning tree that connects all the points and whose total length is a minimum. The other distances to preserve can be chosen from the distances to the second/third nearest neighbors and/or distances to some reference points. These choices provide us a few alternatives to display data.

As an example, Figure 2 shows snapshots of three-dimensional visualization of the United Nation statistical data of its member states. The input data set comes from the United Nations InfoNation Database and has eight dimensions: population, GDP per capita, life expectancy, illiteracy rate, spending on education, and numbers of TV sets, phones, and newspaper circulations per 1000 inhabitants. Only 24 states whose populations exceed 50 million are considered. Again, Euclidean distance was used. In Figure 2(a), the initial point was chosen to be USA and each data point is projected so that distances to its three nearest neighbors are preserved. It looks that the distance between USA and China is the greatest. However, this is an illusion because global distances are not preserved at all. Figure 2(b) shows the

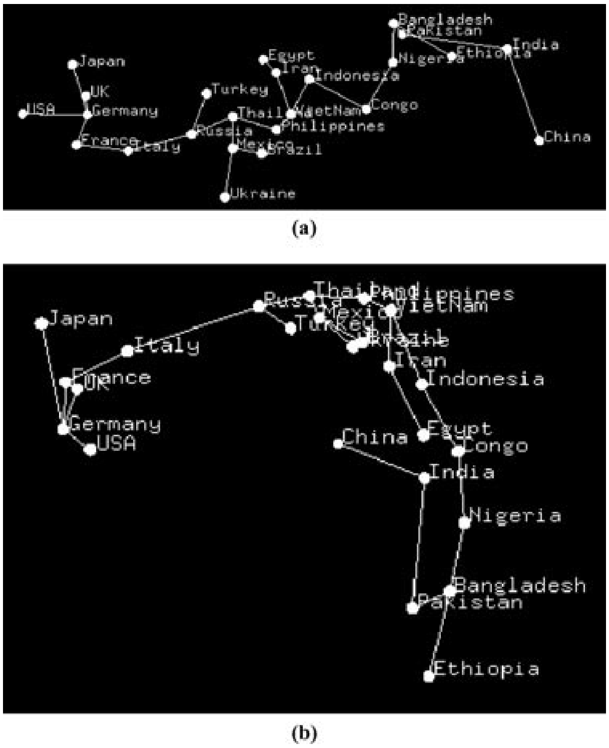

FIGURE 2 | Plots of the UN statistical data using (a) nearest neighbors; (b) two reference points.

result of using USA and China as two reference points and gives a global view of distances to them from all other states such that each point preserves distance to its mapped nearest neighbor, and distances to points representing USA and China. One can thus use this map to analyze the dissimilarities between USA, China and to other countries.

USING GEODESIC DISTANCES

In many applications, nonlinear procedures produce better data projections. One way to understand this is to assume that real world data sets are usually located on a q-dimensional, possibly nonlinear, manifold in p-dimensional feature space ($q < p$), in which case Euclidean distance is not a good measure of dissimilarity between a pair of data points. A significant recent development in dimensionality reduction is the use of geodesic distances. Intuitively, geodesic distance between a pair of points on a manifold is the distance measured along the manifold. It provides the ground truth of dissimilarity between a pair of data points. Therefore, distance-preserving techniques for dimensionality reduction should take geodesic distances as input and project the data to a q-dimensional space such that the geodesic distances are preserved by the Euclidean distances in the q-dimensional space.

In implementation, the geodesic distance between a pair of data points is usually estimated by the length of the shortest path between the pair on a neighborhood graph that connects every point to its neighbor points. As the number of data points increases, the length of the shortest path is expected to asymptotically converge to the true geodesic distance. A typical algorithm using geodesic distances consists of three steps: the first step constructs a neighborhood graph that spans all data points by connecting neighbor points; the second step estimates the geodesic distance between every pair of data points by calculating the length of the shortest path between the pair of vertices on the neighborhood graph, which is usually accomplished by applying Dijkstra's algorithm or Floyd's algorithm to the neighborhood graph; the third step applies a traditional algorithm to map data to a low-dimensional space such that the estimated geodesic distances are preserved by Euclidean distances in the low-dimensional space. These three steps are illustrated in Figure 3. Example algorithms include Isomap,[13] GeoNLM[14] and curvilinear distance analysis (CDA).[15] They use the same approach to estimate geodesic distances. Differences between them reside at the third step: Isomap uses classical MDS, GeoNLM uses NLM, and CDA uses CCA to derive the final data configuration.

As long as good estimates of geodesic distances are obtained, any traditional distance-preserving algorithm would perform well to project the data and preserve the geodesic distances. The success of these algorithms depends on the first step, that is, to build a quality neighborhood graph so that geodesic distances can be well estimated. There are two simple ways to define whether two points are neighbors and are connected by an edge on a neighborhood graph: the first approach is called the k-nearest neighbor (k-NN) approach, it defines that two points are neighbors if one is within the k nearest neighbors of the other; the second (ε-neighbor) approach defines that

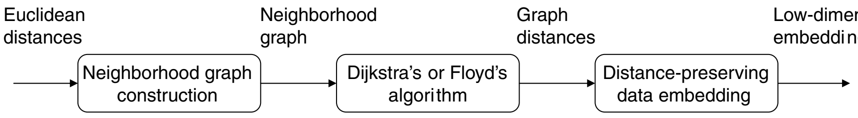
Euclidean distances
Neighborhood graph
Graph distances
Low-dimensional embedding
Neighborhood graph construction
Dijkstra's or Floyd's algorithm
Distance-preserving data embedding

FIGURE 3 | Three steps for dimensionality reduction using geodesic distances.

BOX 4: DISTANCE PRESERVATION VIA ITERATIVE OPTIMIZATION

Let δ_{ij} denote the original distance between x_i and x_j and let d_{ij} denote the corresponding distance after the mapping. An error measure can be defined to measure the change of distances. In fact, classical MDS is an approach that minimize the error measure $\sum_{i<j}(\delta_{ij}^2 - d_{ij}^2)$. It belongs to a group of methods for metric MDS, where each method defines its own error measure. Many error measures take the form $E = \sum_{i<j} w_{ij}(\delta_{ij} - d_{ij})^2$ with various weight w_{ij}. A fundamental difference between E and the error measure of classical MDS is that E is no longer a quadratic form of the underlying x_i and cannot be minimized by solving an eigenvalue problem. In general, we cannot obtain a closed-form solution and have to turn to iterative procedures to minimize E.

In Kruskal's MDS, one definition (normalized stress) of E is given as the normalized sum of squared differences of distances:

$$E_{MDS} = \frac{1}{\Sigma_{i<j}\delta_{ij}^2} \sum_{i<j} (\delta_{ij} - d_{ij})^2.$$

Sammon's NLM defines E as

$$E_{NLM} = \frac{1}{\Sigma_{i<j}\delta_{ij}} \sum_{i<j} \frac{(\delta_{ij} - d_{ij})^2}{\delta_{ij}}.$$

It is commonly referred as Sammon's stress.

The difference between E_{MDS} and E_{NLM} is that each term of E_{NLM} is normalized by δ_{ij}. While changes of long distances dominate E_{MDS}, changes of shorter distances dominate E_{NLM}. In NLM, short distances have a higher priority to be preserved than long distances. That is the reason why NLM has an effect of unfolding a data manifold. Niemann and Weiss[10] have further given a unified error measure $\frac{1}{\Sigma_{i<j}\delta_{ij}^{(c+2)}}\Sigma_{i<j}(\delta_{ij} - d_{ij})^2\delta_{ij}^c$ and a way to improve the convergence of NLM. The error measure becomes E_{NLM} when $c = -0$ and becomes E_{MDS} when $c = 0$. A value $c < 0$ prefers the preservation of short distances. A value $c \geq 0$ prefers the preservation of long distances. To unfold strongly twisted patterns, CCA goes one step further by completely ignoring the preservation of long distances. Its error measure is defined as $E_{CCA} = \sum_{i<j}(\delta_{ij} - d_{ij})^2 F(d_{ij}, \lambda)$, where $F(d_{ij}, \lambda)$ is a weight function with a user-defined parameter λ. F has to be a monotone decreasing function to d_{ij} to make CCA favor the preservation of short distances. Usually, F is defined as a binary gate function that makes CCA to completely ignore distances longer than λ. It has been reported successful to unfold highly twisted data manifolds. CCA also employs a new procedure for fast minimization and to escape from local minima of E_{CCA}.

An error measure E is often minimized iteratively. Let Y denote the low-dimensional projection of X. The iteration starts from an initial estimate of Y. Each iteration has the form $Y_{(\mathrm{new})} = Y_{(\mathrm{old})} = \Delta Y$, where ΔY denotes the adjustment in each step. In the gradient descent algorithm, ΔY is calculated as $\Delta Y = -\mu \frac{\partial E}{\partial Y}|_{Y=Y_{(\mathrm{old})}}$, where $\mu > 0$ is a user-defined constant parameter. In Newton's algorithm, μ is literally replaced by the inverse of the Hessian matrix of E computed at $Y_{(\mathrm{old})}$. Newton's algorithm requires more computation and converges faster than the gradient descent algorithm.

two points are neighbors if the distance between them is smaller than a user-defined threshold ε.

For illustration, Figure 4(a) shows a synthetic data of 1000 points randomly distributed on a 4 × 1 rectangle, which is then wrapped into a three-dimensional Swiss roll. Figure 4(b) shows its 5-NN ($k = 5$) neighborhood graph superimposed on the data. Figure 4(c) displays the neighborhood graph unwrapped in two-dimensional. Figures 4(b) and (c) illustrate the shortest path between an example pair of data points A and B. Geodesic distance between A and B is estimated as the length of the path. Applying distance-preserving methods (such as classical MDS) to the estimated geodesic distances will unfold the data set.

k-NN and ε-neighbor have problems in constructing neighborhood graphs for geodesic distance estimation. Neither approach guarantees the connectivity of the constructed neighborhood graph, especially when the data are not evenly distributed. How to choose a proper value of k or ε becomes a difficult problem. If the parameter were chosen too small, the neighborhood graph would be disconnected. If it were chosen too large, a so-called 'short-circuit' problem would occur and the constructed neighborhood graph would not follow the manifold. This is shown in Figure 4(b) where the points A and B may be directly connected by an edge if the parameter is chosen too large.

In principle, graph distances approach the corresponding geodesic distances as the number of data points increases. Data projection using geodesic distances is guaranteed to converge asymptotically to the true intrinsic structure of the data. In practice, however, neighborhood graph may not offer good estimation of a geodesic distance, especially a short one that spans a few edges on the neighborhood graph. For example, the constructed neighborhood graph shown in Figure 4(c) contains many holes and looks like a Swiss cheese. As illustrated by a path between points

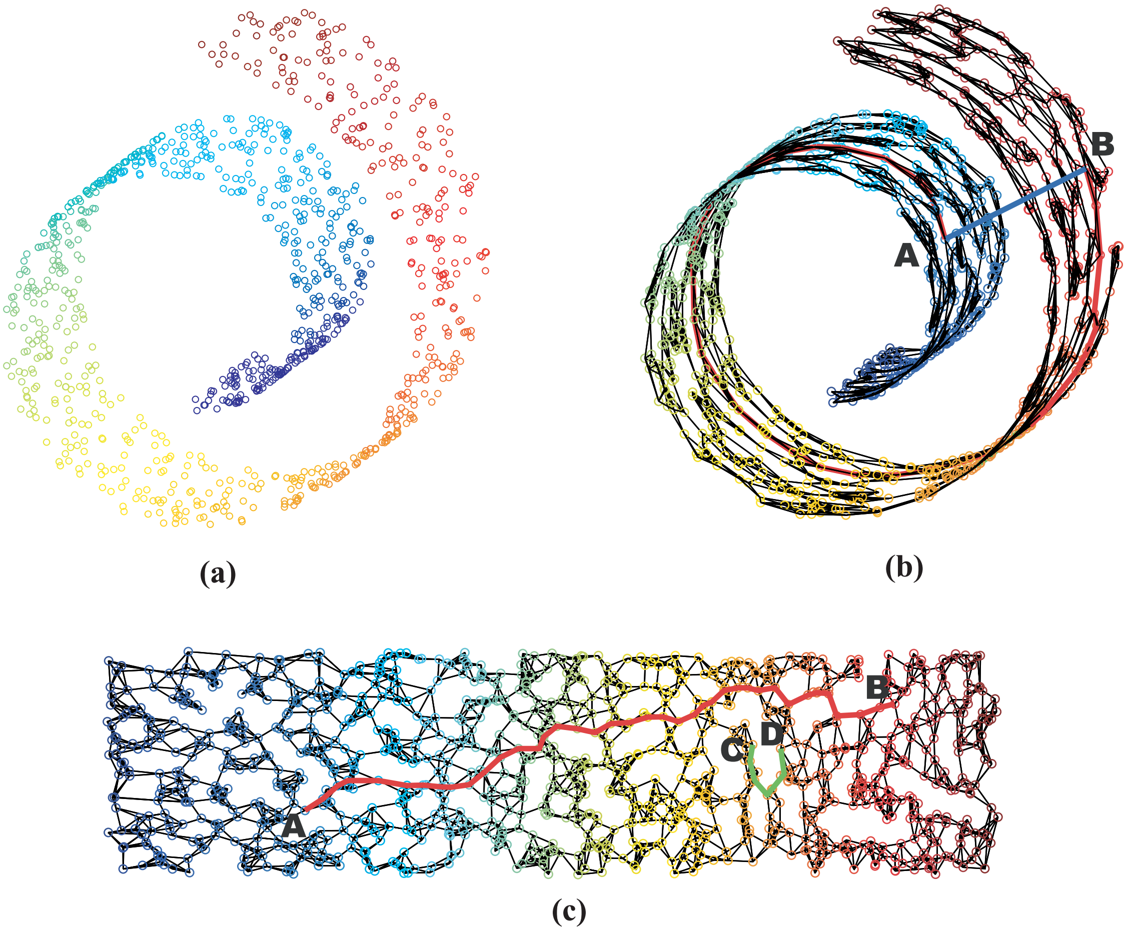

FIGURE 4 | (a) A Swiss-roll data set, (b) its 5-NN neighborhood graph, and (c) its two-dimensional projection.

C and D in the figure, geodesic distance between a pair of points on opposite sides of a hole is overestimated by length of the path along the bank of the hole. This means that the calculated graph distances, especially the short ones each of which spans a few edges on the graph, may not be good estimates of the corresponding geodesic distances. This may explain the different behavior of Isomap, GeoNLM, and CDA. The underlying methods, NLM and CCA, used in GeoNLM and CDA emphasize the preservation of short distances, some of which may be overestimated by detoured paths on a neighborhood graph with holes.

ALIGNMENT APPROACHES

Another way for dimensionality reduction of data distributed on a q-dimensional manifold is through direct alignment of local linear models. In principle, a q-dimensional smooth manifold locally resembles a q-dimensional Euclidean space. A neighborhood can then be defined for each data point on the manifold, in which all points resume the behavior in q-dimensional Euclidean space. Therefore, q-dimensional local models can be derived at each neighborhood through linear methods. On the global scale, we are interested in data projection through an alignment of the local models. The local models are coordinated and aligned by affine transformations, one for each model from its own variable space to a single global coordinate system. If the local models preserve distances, we expect that the resulting global configuration preserves local distances. The alignment is inherently an optimization problem to minimize a pre-defined global alignment error. An exciting result is that, as long as the error can be expressed in a quadratic form of the resulting coordinates, it can be minimized by solving a sparse eigenvalue problem subject to constraints that make the problem well posed. Results of the alignment are globally optimized and there is no problem with local optima. The underlying manifold may be highly nonlinear although both the local models and the alignment process are linear.

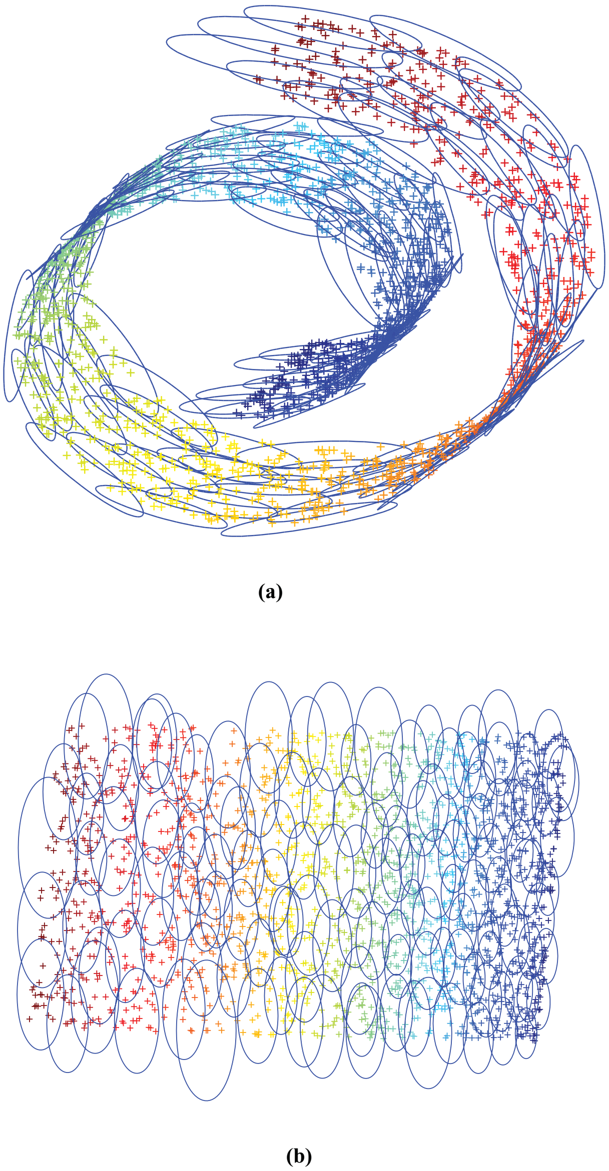

FIGURE 5 | (a) Swiss-roll data superimposed with neighborhoods; (b) two-dimensional projection by alignment of local models.

For illustration, Figure 5(a) shows the Swiss-roll data set superimposed with a set of local neighborhoods, each of which is marked by a circle. Local model derived from each neighborhood could be transformed and aligned with local models derived from other neighborhoods in a two-dimensional space, as shown in Figure 5(b).

A celebrated approach for model alignment is locally linear embedding (LLE).[16] Its local model approximates a data point by a weighted linear combination of its neighbor points. The local approximation problem, which is decoupled across data points, is a typical least square fitting problem. On the global scale, LLE finds a low-dimensional data configuration so that the weights used in the linear approximations of all data points are best preserved. Clearly, the local model used in LLE is not isometric and it is not the goal of LLE to preserve any distance. However, LLE sets a foundation for other methods that use isometric local models. Computationally, the global optimization problem is to minimize a residual measure, which has a normalized quadratic form of resulting coordinates and thus can be solved analytically by using an eigensolver.

The idea of using an eigensolver to derive globally optimized analytical solution is so appealing that researchers have developed techniques by using different local models. Laplacian eigenmap[17] tries to solve a constrained optimization problem in order to minimize a weighted sum of squared local distances between neighbor points. The physical interpretation of the error measure is that it reflects the average of the Laplacian operator on tangent spaces over the manifold. Another technique, Hessian eigenmaps,[18] uses a similar idea but with a Hessian matrix replacing the Laplacian operator. The local models used in these approaches are isometric. Since these approaches are based on the intrinsic geometric structure of the manifold, they exhibit stability with respect to the manifold and offer less deformed results than those produced by the LLE.

Recall that PCA and classical MDS are traditional linear methods for dimensionality reduction and can thus be used to derive local models. Because a smooth manifold locally resembles a Euclidean space, models derived by PCA or classical MDS are locally distance preserving and reflect local geometry of the manifold. To align the local models, the alignment error can be defined as the sum of squared distances between resulting points and the corresponding points on the transformed local models. The alignment error takes a quadratic form to the resulting coordinates and can be minimized by using an eigensolver. Although such an alignment may not be globally distance preserving, it maps the data to a single global low-dimensional coordinate system with the property of local isometry.

Local tangent space alignment (LTSA)[19] is an approach taking this idea by using PCA to derive a low-dimensional local model at every data point and then aligning these models for dimensionality reduction. Another method, Locally Multi-dimensional Scaling (LMDS)[20] uses classical MDS to derive local models. For model alignment, a simple observation of PCA and classical MDS is that they project all data points within a neighborhood in the same way and thus the local model does not discriminate

BOX 5: LOCALLY LINEAR EMBEDDING

Although LLE does not preserve distances, we present its basic idea since it inspired other alignment methods with isometric local models. LLE uses local linearity of the manifold to find a linear representation of each point in terms of its neighbors. Let $W = \{w_{ij}\}$ be a matrix of weights where each w_{ij} summarizes the contribution of x_j to x_i's reconstruction. The first step of LLE is to find W to minimize the reconstruction error $\sum_i (x_i - \sum_j w_{ij} x_j)^2$ with constraints $\sum_j w_{ij} = 1$, $w_{ii} = 0$ and $w_{ij} = 0$ if x_j is not within the neighborhood of x_i. Please note that each column of W is decoupled and can be minimized independently. The second step of LLE finds a new set of points $\{y_1, \ldots, y_n\}$ in q-dimensional space such that the global alignment error $\Sigma_i (y_i - \Sigma_j w_{ij} y_j)^2$ is minimized. Both error measures have the similar form, but variables in the first error measure are w_{ij}'s and variables in the second error measure are y_i's. The second error measure defines a quadratic form of y_i's. Subject to constraints that make the problem well posed, it can be minimized by solving a sparse $n \times n$ eigenvector problem, whose smallest q nonzero eigenvectors provide an ordered set of orthogonal coordinates centered at the origin. It should be noted, similar to PCA and classical MDS, that a solution can be found for all values of q simultaneously: the best one-dimensional projection is simply the first coordinate of the best two-dimensional projection, and so on. Other methods extend LLE with different local models and alignment functions.

a data point from other data points in its neighborhood. LMDS takes advantage of this observation by deriving and aligning local models on a minimum set of overlapping neighborhoods. The set of overlapping neighborhoods can be obtained through a greedy approximation algorithm for the classical minimum set cover problem. Consequently, this makes LMDS scalable to the number of overlapping neighborhoods instead of the number of data points. Another major difference between LTSA and LMDS associates with their requirements on input data. Unlike PCA, which requires input data coordinates, classical MDS requires only local distances as input. This makes LMDS applicable to applications where each entity knows nothing beyond its neighborhoods and there are demands to derive global information from local neighborhoods.

SUMMARY AND COMPARISON

The objective of dimensionality reduction is to project a set of high-dimensional data observations to a low-dimensional space such that the projection preserves essential information (such as distances between data observations) and performs better than the original data in further processing. We summarize the major approaches used for distance-preserving dimensionality reduction as the following:

- A simple approach is linear mapping, which is characterized by a projection matrix that linearly transforms high-dimensional points to low-dimensional space. Example methods are PCA and classical MDS. The projection matrix is designed so that variations of the result data points are maximized. It offers the optimal linear mapping for distance preservation in the sense that the sum of changes of squared distances is minimized. In addition, random projection offers a simple and efficient approach for linear mapping. Random projection can project a small set of high-dimensional data points to a low-dimensional space in such a way that distances between data points are approximately preserved.
- Methods can be developed to explicitly minimize the change of distances before and after the projection. Usually, the minimization of a cost function has no closed-form solution. Therefore, the cost function is iteratively minimized. Example methods include Kruskal's MDS, NLM, and CCA. A problem with iterative minimization is that it may terminate at a local minimum.
- Many methods take interpoint distances as input. In addition to the usual Euclidean distances, the input distances can be geodesic distances between data points along the manifold. The geodesic distance between a pair of data points is estimated by the length of the shortest path between the pair of points on a neighborhood graph, which connects every data point to its neighbor data points. Example methods include Isomap, GeoNLM, and CCA. Isomap applies classical MDS, GeoNLM applies NLM, and CDA applies CCA to the estimated geodesic distances.
- A manifold locally resembles a Euclidean space. Therefore, linear models can be built locally within local neighborhoods and are then aligned on a global scale in the target space. The alignment process is to minimize an error, which is usually defined as having a quadratic form to the resulting coordinates and can be elegantly minimized using

an eigensolver. These methods include LLE and its derivations using various local models and alignment mechanisms. If the local models preserve distances, for example, in LTSA and LMDS, the result of global alignment offers local isometry.

- There are also other distance-preserving methods that do not fit into the above framework. These include the triangular mapping method and the tetrahedral mapping method that preserve some exact distances.

The linear methods (PCA and classical MDS), Isomap (which uses classical MDS), and the methods using local model alignments share an important feature that they provide closed-form solutions by solving eigenvalue problems of specially constructed matrices. This feature comes from the simple fact that their error measures are defined as quadratic forms of the unknown data configurations, the minimization of which can be solved using an eigensolver. These methods considered together are called spectral methods. Their computations are based on tractable optimization techniques such as shortest paths, least squares fits, and matrix eigenvalue decomposition. The solutions they give are globally optimized. In contrast, this situation changes dramatically if the error measure cannot be expressed in a quadratic form, in which case we have to appeal to iterative procedures to minimize the error measure. Indeed, the power of mathematical tool at our disposal decides the scope of what we can do in this and many other research areas.

Dimensionality reduction using geodesic distances and dimensionality reduction using model alignments represent active areas of research. In both areas, methods take bottom-up approaches by assigning a local neighborhood to each data point and combining local information to obtain global data configuration. However, they are fundamentally different in terms of the distances they preserve. By applying classical MDS to the estimated geodesic distance between every pair of data points, Isomap attempts to offer global isometry. As we discussed, however, Isomap does not work well at preserving local distances. On the other hand, variations of LLE, such as LTSA and LMDS, try to maintain local isometry by applying distance-preserving local models within each neighborhood at the cost of giving up global isometry. In recent years, there are attempts, such as maximum variance unfolding,[21] that try to maximize the overall variance of the embedding while preserving distances within local neighborhoods. In fact, a data projection preserving global distances, such as the one attempted by Isomap, is possible only when the manifold is flat. A simple counterexample is data distributed on the surface of a sphere, which cannot be projected to two-dimensional space with global distances preserved.

The success of both Isomap-like and LLE-like methods depends on how to assign a neighborhood to each data point. The size of the neighborhood is a compromise: it must be large enough to build a connected neighborhood graph for Isomap or to allow good construction of local models for model alignment, but small enough for the data manifold on the neighborhood to have little or no curvature. Several approaches have been proposed[22] to overcome these problems and build connected neighborhood graphs. The difficulty of choosing a proper neighborhood size and the contradiction between the preservation of global and local distances reflect a fundamental problem: what is global and what is local, a persistent problem in pattern analysis and machine learning that may not be easily answered without domain-specific knowledge.

CONCLUSION

In this paper, we have introduced major ideas and important techniques for distance-preserving dimensionality reduction. These techniques and algorithms are classified into linear methods, methods using iterative optimization, methods using geodesic distances, and methods through alignments of local models. We discussed these methods by focusing on their basic ideas and by summarizing their common features and differences.

Dimensionality reduction using geodesic distances and through model alignment represents active areas of research. In both areas, the assignment of local neighborhoods is of ultimate importance to the success of dimensionality reduction. There exists a risk of overfitting or disconnected neighborhood graph if the neighborhood is too small and a risk of mismatch of local geometry if the neighborhood is too large. Therefore, one direct question is how to make the neighborhood sizes adaptive to local curvatures and data densities. In general, studying data transformation and manipulation on manifold opens a new arena of research for data processing, which invites ideas and theories from differential geometry. From an algorithmic point of view, other interesting topics include bidirectional mapping and incremental mapping of new data points. In fact, there exists incremental extension[23] of Isomap. We expect to see more extensions and improvements along these directions in the near future.

REFERENCES

1. Jolliffe I. *Principal Component Analysis*. 2nd ed. New York: Springer; 2002.
2. Cox T, Cox M. *Multidimensional Scaling*. 2nd ed. London: Chapman and Hall/CRC; 2000.
3. Stone JV. *Independent Component Analysis: A Tutorial Introduction*. Cambridge: The MIT Press; 2004.
4. Hyvärinen A, Karhunen J, Oja E. *Independent Component Analysis*. New York: Wiley-Interscience; 2001.
5. Bingham E, Mannila H. Random projection in dimensionality reduction: applications to image and text data. In: *Proceedings of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '01)*. New York: ACM Press; 2001, 245–250.
6. Schölkopf B, Smola A, Müller K-R. Nonlinear component analysis as a kernel eigenvalue problem. 1998, 10:1299–1319.
7. Kruskal JB, Wish M. *Multidimensional Scaling*. Beverly Hills, CA: Sage Publications; 1978.
8. Sammon JJ. A nonlinear mapping for data structure analysis. *IEEE Trans Comput* 1969, C-18:401–409.
9. Demartines P, Herault J. Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. *IEEE Trans Neural Netw* 1997, 8:148–154.
10. Niemann H, Weiss J. A fast converging algorithm for nonlinear mapping of high-dimensional data onto a plane. *IEEE Trans Comput* 1979, C-28:142–147.
11. Lee RC, Slagle JR, Blum H. A triangulation method for the sequential mapping of points from N-space to two-space. *IEEE Trans Comput* 1977, 26:288–292.
12. Yang L. Distance-preserving projection of high-dimensional data for nonlinear dimensionality reduction. *IEEE Trans Pattern Anal Machine Intell* 2004, 26: 1243–1246.
13. Tenenbaum JB, Silva Vd, Langford JC. A global geometric framework for nonlinear dimensionality reduction. *Science* 2000, 290:2319–2323.
14. Yang L. Sammon's nonlinear mapping using geodesic distances. In: *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*. Vol. 2. IEEE Computer Society, Cambridge; 2004.
15. Lee JA, Lendasse A, Donckers N, Verleysen M. A robust nonlinear projection method. In: *Proceedings of the 8th European Symposium on Artificial Neural Networks (ESANN2000)* Bruges; 2000, 13–20.
16. Roweis ST, Saul LK. Nonlineaar dimensionality reduction by locally linear embedding. *Science* 2000, 290: 2323–2326.
17. Belkin M, Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput* 2003, 15:1373–1396.
18. Donoho DL, Grimes C. Hessian eigenmaps: locally linear embedding techniques for high-dimensional data. *Proc Natl Acad Sci* 2003, 100:5591–5596.
19. Zhang Z, Zha H. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM J Sci Comput* 2005, 26:313–338.
20. Yang L. Alignment of overlapping locally scaled patches for multidimensional scaling and dimensionality reduction. *IEEE Tran Pattern Anal Machine Intell* 2008, 30: 438–450.
21. Weinberger KQ, Sha F, Saul LK. Learning a kernel matrix for nonlinear dimensionality reduction. In: *Proceedings of the 21st International Conference on Machine Learning (ICML-04)*. Banff: ACM Press; 2004, 839–846.
22. Yang L. Building connected neighborhood graphs for isometric data embedding. In: *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'2005)*. ACM Press, Chicago, IL; 2005.
23. Law MH, Jain AK. Incremental nonlinear dimensionality reduction by manifold learning. *IEEE Trans Pattern Anal Machine Intell* 2006, 28:377–391.

FURTHER READING

Borg I, Groenen P. *Modern Multidimensional Scaling: Theory and Applications*. 2nd ed. New York: Springer; 2005.

Gorban AN, Kégl B, Wunsch DC, Zinovyev A, eds. *Principal Manifolds for Data Visualization and Dimension Reduction*. Berlin: Springer; 2007.

Lee JA, Verleysen M. *Nonlinear Dimensionality Reduction*. New York: Springer; 2007.