International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 5 Issue: 7 195 – 200

# A Review on Non Linear Dimensionality Reduction Techniques for Face Recognition

Ms. Preeti Sharma
Mewar university
*Preeti.sh80@gmail.com*

Dr. R. P. Saxena
Research Guide Mewar University
*rp_saxena@yahoo.com*

***Abstract:*** Principal component Analysis (PCA) has gained much attention among researchers to address the pboblem of high dimensional data sets.during last decade a non-linear variantof PCA has been used to reduce the dimensions on a non linear hyperplane.This paper reviews the various Non linear techniques ,applied on real and artificial data .It is observed that Non-Linear PCA outperform in the counterpart in most cases .However exceptions are noted.

***Keywords:*** *Nonlinear Dimensionality reduction, manifold learning, feature extraction.*

*****

## I. INTRODUCTION

Data collection and Data storage capacity is improving day by day, which led to exponential rise in data and information .Many scientists and researchers are working on different field like engineering, astronomy, medical, remote sensing, e-commerce ,social networking and media etc are facing high volume of data transaction on daily basis .It become challenge in data analysis to deal with high volume data. Traditional statistical methods are not performing well because of increase in observation but mostly because of high dimension data set used for the observation.

Real world data usually has high dimensionality,its dimensionality needs to be reduced. Dimensionality reduction is the process of reducing the dimension by transforming the high dimensional data into low dimension without loosing the parameter require to analyses the property of data .The reduced representation should have a dimensionality that corresponds to the intrinsic dimensionality of the data. The intrinsic dimensionality of data is the minimum number of parameters needed to account for the observed properties of the data [1].In many Domains Dimensionality reduction is important , since it reduces the curse of dimensionality and other unwanted properties of high-dimensional spaces [2].Dimensionality reduction will smooth the classification, visualization and compression of high dimension data. Traditionally Principal component analysis and (PCA) and factor Analysis was implemented for dimensionality reduction [3].However complex Nonlinear problems cannot solved by these linear techniques. Therefore from the last few years several non linear techniques was proposed . Non linear techniques have advantage over linear techniques solve the problems of real world .Real world data are nonlinear in nature .From the previous research it is observed that the nonlinear techniques are performing better then linear techniques on artificial task. For instance, the Swiss roll dataset comprises a set of points that lie on a spiral-like two-dimensional manifold within a three-dimensional space. A vast number of nonlinear techniques are perfectly able to find this embedding, whereas linear techniques fail to do so.In contrast to that success of non linear techniques over natural dataset is poor. Above all it is not clear that upto what exent the performance of non linear technique differ from the linear technique[7].The aims of the paper is to identify the weaknesses of nonlinear techniques and suggest how the perfornmance of system are improved . The investigation is performed by both a theoretical and an empirical evaluation of the dimensionality reduction techniques.

their linear counterparts on complex artificial tasks. For instance, the Swiss roll dataset comprises a set of points that lie on a spiral-like two-dimensional manifold within a three-dimensional space. A vast number of nonlinear techniques are perfectly able to find this embedding, whereas linear techniques fail to do so. In contrast to these successes on artificial datasets, successful applications of nonlinear dimensionality reduction techniques on natural datasets are scarce. Beyond this observation, it is not clear

## II. DIMENSIONALITY REDUCTION

High-dimensional data, meaning data that requires more than two or three dimensions to represent, can be difficult to interpret. One approach to simplification is to assume that the data of interest lie on an embedded non-linear manifold within the higher-dimensional space. If the manifold is of low enough dimension, the data can be visualized in the low dimensional space.The problem of (nonlinear) dimensionality reduction can be defined as follows[8]. Assume we have a dataset represented in a n×D matrix X consisting of n data vectors xi (iϵ {1, 2, . . . , n}) with dimensionality D. Assume further that this dataset has intrinsic dimensionality d (where d < D, and often d _ D). Here, in mathematical terms, intrinsic dimensionality means that the points in dataset X are lying on or near a manifold with dimensionality d that is embedded in the D-dimensional space. Dimensionality reduction techniques transform dataset X with dimensionality D into a new dataset Y with dimensionality d, while retaining the geometry of the data as much as possible. In general, neither the geometry of the data manifold, nor the intrinsic dimensionality d of the dataset X

are known. Therefore, dimensionality reduction is an ill-posed problem that can only be solved by assuming certain properties of the data (such as its intrinsic dimensionality). Throughout the paper, we denote a high-dimensional datapoint by xi, where xi is the ith row of the D-dimensional data matrix X. The low-dimensional counterpart of xi is denoted by yi, where yi is the ith row of the d-dimensional data matrix Y . In the remainder of the paper, we adopt the notation presented above.

Figure 1 shows a taxonomy of techniques for dimensionality reduction. The main distinction between techniques for dimensionality reduction is the distinction between linear and nonlinear techniques. Linear techniques assume that the data lie on or near a linear subspace of the high-dimensional space. Nonlinear techniques for dimensionality reduction do not rely on the linearity assumption as a result of which more complex embedding of the data in the high-dimensional space can

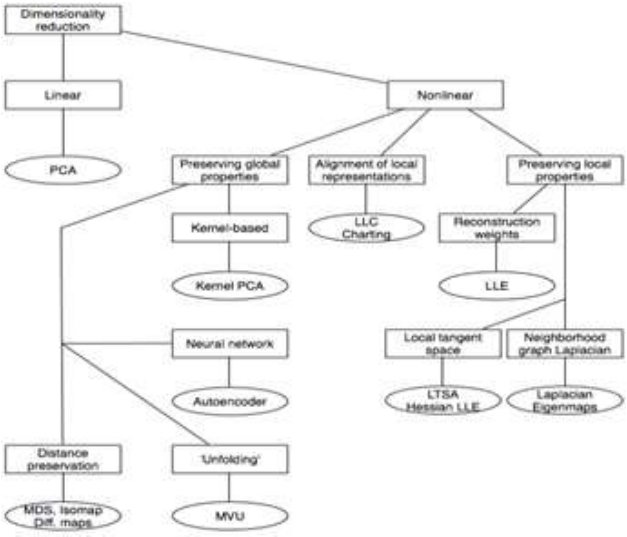


Fig. 1 Taxonomy for Dimensionality Reduction[8]

In machine learning and statistics, dimensionality reduction is the process of reducing the number of random be identified. variables under consideration, [9] and can be divided into feature selection and feature extraction. [10]

A. Feature selection

Feature selection approaches try to find a subset of the original variables (also called features or attributes). Two strategies are filter (e.g. information gain) and wrapper (e.g. search guided by the accuracy) approaches. In some cases, data analysis such as regression or classification can be done in the reduced space more accurately than in the original space.

B. Feature extraction

Feature extraction transforms the data in the high-dimensional space to a space of fewer dimensions. The data transformation may be linear, as in principal component analysis (PCA), but many nonlinear dimensionality reduction techniques also exist. [11, 12] For multidimensional data, tensor representation can be used in dimensionality reduction through multilinear subspace learning. [13]

## III. NONLINEAR TECHNIQUES FOR DIMENSIONALITY REDUCTION

Consider a dataset represented as a matrix (or a database table), such that each row represents a set of attributes (or features or dimensions) that describe a particular instance of something. If the number of attributes is large, then the space of unique possible rows is exponentially large. Thus, the larger the dimensionality, the more difficult it becomes to sample the space. This causes many problems. Algorithms that operate on high-dimensional data tend to have a very high time complexity. Many machine learning algorithms, for example, struggle with high-dimensional data. This has become known as the curse of dimensionality. Reducing data into fewer dimensions often makes analysis algorithms more efficient, and can help machine learning algorithms make more accurate predictions.

In this section, we discuss nonlinear techniques for dimensionality reduction, as well as their weaknesses and applications as reported in the literature. Nonlinear techniques for dimensionality reduction can be subdivided into three main types :( 1) techniques that attempt to preserve global properties of the original data in the low dimensional representation, (2) techniques that attempt to preserve local properties of the original data in the low-dimensional representation, and (3) techniques that perform global alignment of a mixture of linear models.

A. Global Techniques for Non Linear Dimensionality Reduction

Global techniques preserve the global properties of data. There are six global nonlinear techniques for dimensionality reduction: (1) MDS, (2) Isomap, (3) MVU, (4) Kernel PCA, (5) auto encoders and (6) diffusion map.

1) MDS:

Multidimensional scaling (MDS) represents a collection of nonlinear techniques that maps the high dimensional data representation to a low-dimensional representation while retaining the pairwise distances be-tween the data points as much as possible. The quality of the mapping is expressed in the stress function, a measure of the error between the pairwise distances in the low-dimensional and high-dimensional representation of the data. Two important examples of stress functions (for metric MDS) are the raw stress function and the Summon cost function.

2) Isomap:

In statistics, Isomap is one of the widely used low-dimensional embedding methods, where geodesic distances on a weighted graph are incorporated with the classical scaling (metric multidimensional scaling). Isomap is used for computing a quasi-isometric, low-dimensional embedding of a set of high-dimensional data points. The algorithm provides a simple method for estimating the intrinsic geometry of a data manifold based on a rough estimate of each data point's

neighbors on the manifold. Isomap is highly efficient and generally applicable to a broad range of data sources and dimensionality.

Isomap is one representative of isometric mapping methods, and extends metric multidimensional scaling (MDS) by incorporating the geodesic distances imposed by a weighted graph. To be specific, the classical scaling of metric MDS performs low-dimensional embedding based on the pairwise distance between data points, which is generally measured using straight-line Euclidean distance. Isomap is distinguished by its use of the geodesic distance induced by a neighborhood graph embedded in the classical scaling. This is done to incorporate manifold structure in the resulting embedding. Isomap defines the geodesic distance to be the sum of edge weights along the shortest path between two nodes (computed using Dijkstra's algorithm, for example). The top n eigenvectors of the geodesic distance matrix, represent the coordinates in the new n-dimensional Euclidean space.

The connectivity of each data point in the neighborhood graph is defined as its nearest k Euclidean neighbors in the high-dimensional space. This step is vulnerable to "short-circuit errors" if k is too large with respect to the manifold structure or if noise in the data moves the points slightly off the manifold. Even a single short-circuit error can alter many entries in the geodesic distance matrix, which in turn can lead to a drastically different (and incorrect) low-dimensional embedding. Conversely, if k is too small, the neighborhood graph may become too sparse to approximate geodesic paths accurately.

Following the connection between the classical scaling and PCA, metric MDS can be interpreted as kernel PCA. In a similar manner, the geodesic distance matrix in Isomap can be viewed as a kernel matrix. The doubly centered geodesic distance matrix K in Isomap is of the form

$$K = \frac{1}{2} H D^2 H$$

Where $D^2 = D_{ij}^2 := (D_{ij})^2$ is the elementwise square of the geodesic distance matrix D = [Dij], H is the centering matrix, given by

$$H = I_n - \frac{1}{N} e_N e_N^T, \quad \text{where } e_N = [1 \ \dots \ 1]^T$$

However, the kernel matrix K is not always positive semi definite. The main idea for kernel Isomap is to make this K as a Mercer kernel matrix (that is positive semi definite) using a constant-shifting method, in order to relate it to kernel PCA such that the generalization property naturally emerges.

3) Maximum Variance unfolding:

Semidefinite embedding (SDE) or maximum variance unfolding (MVU) is an algorithm in computer science that uses semi programming to perform non-linear dimensionality reduction of high-dimensional Victoria input data. MVU can be viewed as a non-linear generalization of Principal component analysis. Non-linear dimensionality reduction algorithms attempt to map high-dimensional data onto a low-dimensional Euclidean vector space. Maximum variance Unfolding is a member of the manifold learning family, which also include algorithms such as isomap and locally linear embedding. In manifold learning, the input data is assumed to be sampled from a low dimensional manifold that is embedded inside of a higher dimensional vector space. The main intuition behind MVU is to exploit the local linearity of manifolds and create a mapping that preserves local neighborhoods at every point of the underlying manifold.MVU creates a mapping from the high dimensional input vectors to some low dimensional Euclidean vector space in the following steps: A neighborhood graph is created. Each input is connected with its k-nearest input vectors (according to Euclidean distance metric) and all k-nearest neighbors are connected with each other. If the data is sampled well enough, the resulting graph is a discrete approximation of the underlying manifold. The neighborhood graph is "unfolded" with the help of semi definite programming. Instead of learning the output vectors directly, the semi definite programming aims to find an inner product matrix that maximizes the pairwise distances between any two inputs that are not connected in the neighborhood graph while preserving the nearest neighbor's distances. The low-dimensional embedding is finally obtained by application of multidimensional scaling on the learned inner product matrix. The steps of applying semi definite programming followed by a linear dimensionality reduction step to recover a low-dimensional embedding into a Euclidean space were first proposed by Linial, London, and Rabinovich.

4) Kernel Principal Component Analysis:

Perhaps the most widely used algorithm for manifold learning is kernel PCA [14].It is a combination of Principal component analysis and the kernel trick. PCA begins by computing the covariance matrix of the $m \times n$ matrix $\mathbf{X}$

$$C = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i \mathbf{x}_i^{\mathsf{T}}.$$

It then projects the data onto the first k eigenvectors of that matrix. By comparison, KPCA begins by computing the covariance matrix of the data after being transformed into a higher-dimensional space,

$$C = \frac{1}{m} \sum_{i=1}^{m} \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_i)^{\mathsf{T}}.$$

It then projects the transformed data onto the first k eigenvectors of that matrix, just like PCA. It uses the kernel trick to factor away much of the computation, such that the

entire process can be performed without actually computing $\Phi(\mathbf{x})$. Of course $\Phi$must be chosen such that it has a known corresponding kernel. Unfortunately, it is not trivial to find a good kernel for a given problem, so KPCA does not yield good results with some problems. For example, it is known to perform poorly with the Swiss roll manifold.

KPCA has an internal model, so it can be used to map points onto its embedding that were not available at training time.

5) Auto encoders

An auto encoder is a feed-forward neural network which is trained to approximate the identity function. That is, it is trained to map from a vector of values to the same vector. When used for dimensionality reduction purposes, one of the hidden layers in the network is limited to contain only a small number of network units. Thus, the network must learn to encode the vector into a small number of dimensions and then decode it back into the original space. Thus, the first half of the network is a model which maps from high to low-dimensional space, and the second half maps from low to high-dimensional space. Although the idea of auto encoders is quite old, training of deep auto encoders has only recently become possible through the use of restricted Boltzmann machines and stacked demising auto encoders.

6) Diffusion Maps

Diffusion maps is a machine learning algorithm introduced by R. R. Coif man and S. Landon.[15, 16] It computes a family of embedding of a data set into Euclidean space (often low-dimensional) whose coordinates can be computed from the eigenvectors and eigenvalues of a diffusion operator on the data. The Euclidean distance between points in the embedded space is equal to the "diffusion distance" between probability distributions centered at those points. Different from other dimensionality reduction methods such as principal component analysis (PCA) and multi-dimensional scaling (MDS), diffusion maps is a non-linear method that focuses on discovering the underlying manifold that the data has been sampled from. By integrating local similarities at different scales, diffusion maps gives a global description of the data-set. Compared with other methods, the diffusion maps algorithm is robust to noise perturbation and is computationally inexpensive.

B) Local Techniques

Local Techniques for dimensionality reduction are based preserving properties of small neighborhoods around the data points. By preservation of local properties of the data, the global layout of the data manifold is retained as well. This subsection presents four local nonlinear techniques for dimensionality reduction: (1) LLE, (2) Laplacian Eigen maps, (3) Hessian LLE, and (4) Modified LLE (5) LTSA

1) Locally-Linear Embedding

Locally-Linear Embedding (LLE) [17] was presented at approximately the same time as Isomap. It has several advantages over Isomap, including faster optimization when implemented to take advantage of sparse matrix algorithms, and better results with many problems. LLE also begins by finding a set of the nearest neighbors of each point. It then computes a set of weights for each point that best describe the point as a linear combination of its neighbors. Finally, it uses an eigenvector-based optimization technique to find the low-dimensional embedding of points, such that each point is still described with the same linear combination of its neighbors. LLE tends to handle non-uniform sample densities poorly because there is no fixed unit to prevent the weights from drifting as various regions differ in sample densities. LLE has no internal model.

LLE computes the barycentric coordinates of a point Xi based on its neighbors Xj. The original point is reconstructed by a linear combination, given by the weight matrix Wij, of its neighbors. The reconstruction error is given by the cost function E (W).

$$E(W) = \sum_i |\mathbf{X}_i - \sum_j \mathbf{W}_{ij}\mathbf{X}_j|^2$$

The weights Wij refer to the amount of contribution the point Xj has while reconstructing the point Xi. The cost function is minimized under two constraints: (a) Each data point Xi is reconstructed only from its neighbors, thus enforcing Wij to be zero if point Xj is not a neighbor of the point Xi and (b) The sum of every row of the weight matrix equals 1.

$$\sum_j \mathbf{W}_{ij} = 1$$

The original data points are collected in a D dimensional space and the goal of the algorithm is to reduce the dimensionality to d such that D >> d. The same weights Wij that reconstructs the ith data point in the D dimensional space will be used to reconstruct the same point in the lower d dimensional space. A neighborhood preserving map is created based on this idea. Each point Xi in the D dimensional space is mapped onto a point Yi in the d dimensional space by minimizing the cost function

$$C(Y) = \sum_i |\mathbf{Y}_i - \sum_j \mathbf{W}_{ij}\mathbf{Y}_j|^2$$

In this cost function, unlike the previous one, the weights Wij are kept fixed and the minimization is done on the points Yi to optimize the coordinates. This minimization problem can be solved by solving a sparse N X N Eigen value problem (N being the number of data points), whose bottom d nonzero Eigen vectors provide an orthogonal set of coordinates. Generally the data points are reconstructed from K nearest neighbors, as measured by Euclidean distance. For such an implementation the algorithm has only one free parameter K, which can be chosen by cross validation.

2) Laplacian Eigen maps

Laplacian Eigen maps [18] uses spectral techniques to perform dimensionality reduction. This technique relies on the basic assumption that the data lies in a low dimensional manifold in a high dimensional space.[19] This algorithm cannot embed out of sample points, but techniques based on Reproducing kernel Hilbert space regularization exist for adding this capability.[19] Such techniques can be applied to other nonlinear dimensionality reduction algorithms as well.
Traditional techniques like principal component analysis do not consider the intrinsic geometry of the data. Laplacian Eigen maps builds a graph from neighborhood information of the data set. Each data point serves as a node on the graph and connectivity between nodes is governed by the proximity of neighboring points (using e.g. the k-nearest neighbor algorithm). The graph thus generated can be considered as a discrete approximation of the low dimensional manifold in the high dimensional space. Minimization of a cost function based on the graph ensures that points close to each other on the manifold are mapped close to each other in the low dimensional space, preserving local distances. The Eigen functions of the Laplace–Beltrami operator on the manifold serve as the embedding dimensions, since under mild conditions this operator has a countable spectrum that is a basis for square integral functions on the manifold (compare to Fourier series on the unit circle manifold). Attempts to place Laplacian Eigen maps on solid theoretical ground have met with some success, as under certain nonrestrictive assumptions, the graph Laplacian matrix has been shown to converge to the Laplace–Beltrami operator as the number of points goes to infinity. [21] Mat lab code for Laplacian Eigen maps can be found in algorithms [22] and the PhD thesis of Belkin can be found at the Ohio State University [21].

3) Hessian LLE
Like LLE, Hessian LLE [23] is also based on sparse matrix techniques. It tends to yield results of a much higher quality than LLE. Unfortunately, it has a very costly computational complexity, so it is not well-suited for heavily-sampled manifolds. It has no internal model.

4) Modified LLE
Modified LLE (MLLE) [24] is another LLE variant which uses multiple weights in each neighborhood to address the local weight matrix conditioning problem which leads to distortions in LLE maps. MLLE produces robust projections similar to Hessian LLE, but without the significant additional computational cost.

5) Local Tangent Space Alignment
Local tangent space alignment (LTSA) [25] is a method for manifold learning, which can efficiently learn a nonlinear embedding into low-dimensional coordinates from high-dimensional data, and can also reconstruct high-dimensional coordinates from embedding coordinates. It is based on the intuition that when a manifold is correctly unfolded, all of the tangent hyperplanes to the manifold will become aligned. It begins by computing the k-nearest neighbors of every point. It computes the tangent space at every point by computing the d-first principal components in each local neighborhood. It then optimizes to find an embedding that aligns the tangent spaces, but it ignores the label information conveyed by data samples, and thus cannot be used for classification directly.

C) Global Alignment of Linear Models

Techniques that perform global alignment of linear models combine these two types: they compute a number of locally linear models and perform a global alignment of these linear models. There are two such techniques (1) LLC (2) manifold charting.

1) LLC

Locally Linear Coordination (LLC) [25] computes a number of locally linear models and subsequently performs a global alignment of the linear models. This process consists of two steps: (1) computing a mixture of local linear models on the data by means of an Expectation Maximization (EM) algorithm and (2) aligning the local linear models in order to obtain the low dimensional data representation using a variant of LLE.

2) Manifold Charting

Similar to LLC, manifold charting constructs a low dimensional data representation by aligning a MoFA or MoPPCA model [26]. In contrast to LLC, manifold charting does not minimize a cost function that corresponds to another dimensionality reduction technique (such as the LLE cost function). Manifold charting minimizes convex cost function that measures the amount of disagreement between the linear models on the global coordinates of the data points. The minimization of this cost function can be performed by solving a generalized Eigen problem.

## IV. CONCLUSIONS

This paper presents a review of techniques for Non Linear dimensionality reduction. From the analysis we may conclude that nonlinear techniques for dimensionality reduction are, despite their large variance, not yet capable of outperforming traditional PCA. In the future, we look forward to implement new nonlinear techniques for dimensionality reduction that do not rely on local properties of the data manifold.

## REFERENCES

[1] K. Fukunaga. Introduction to Statistical Pattern Recognition. Academic Press Professional, Inc., San Diego, CA, USA, 1990.
[2] L.O. Jimenez and D.A. Landgrebe. Supervised classification in high-dimensional space: geometrical, statistical, and asymptotical properties of multivariate data. IEEE Transactions on Systems, Man and Cybernetics, 28(1):39–54, 1997.
[3] C.M. Bishop. Pattern recognition and machine learning. Springer, 2006.

[4] C.J.C. Burges. Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers, chapter Geometric Methods for Feature Selection and Dimensional Reduction: A Guided Tour. Kluwer Academic Publishers, 2005.

[5] L.K. Saul, K.Q. Weinberger, J.H. Ham, F. Sha, and D.D. Lee. Spectral methods for dimensionality reduction. In Semisupervised Learning, Cambridge, MA, USA, 2006. The MIT Press.

[6] J. Venna. Dimensionality reduction for visual exploration of similarity structures. PhD thesis, Helsinki University of Technology, 2007.

[7] M. Niskanen and O. Silv´en. Comparison of dimensionality reduction methods for wood surface inspection. In Proceedings of the 6th International Conference on Quality Control by Artificial Vision, pages 178–188, Gatlinburg, TN, USA, 2003. International Society for Optical Engineering.

[8] Laurens Van Der Maaten,Eric Postma and H.Jaap Van Den Herik.Dimensionality Reduction: A Comparative Review.Journal of Machine Learning Research 10(1) · January 2007

[9] Roweis, S. T.; Saul, L. K. (2000). "Nonlinear Dimensionality Reduction by Locally Linear Embedding". Science 290 (5500): 2323–2326. doi:10.1126/science.290.5500.2323. PMID 11125150.

[10] Pudil, P.; Novovičová, J. (1998). "Novel Methods for Feature Subset Selection with Respect to Problem Knowledge". In Liu, Huan; Motoda, Hiroshi. Feature Extraction, Construction and Selection. p. 101. doi:10.1007/978-1-4615-5725-8_7. ISBN 978-1-4613-7622-4.

[11] Samet, H. (2006) Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann. ISBN 0-12-369446-9

[12] C. Ding, , X. He , H. Zha , H.D. Simon, Adaptive Dimension Reduction for Clustering High Dimensional Data,Proceedings of International Conference on Data Mining, 2002

[13] Lu, Haiping; Plataniotis, K.N.; Venetsanopoulos, A.N. (2011). "A Survey of Multilinear Subspace Learning for Tensor Data". Pattern Recognition 44 (7): 1540–1551. doi:10.1016/j.patcog.2011.01.004.

[14] B. Schölkopf, A. Smola, K.-R. Muller, Kernel Principal Component Analysis, In: Bernhard Schölkopf, Christopher J. C. Burges, Alexander J. Smola (Eds.), Advances in Kernel Methods-Support Vector Learning, 1999, MIT Press Cambridge, MA, USA, 327–352. ISBN 0-262-19416-3

[15] Coifman, R.R.; S. Lafon. (2006). "Diffusion maps". Applied and Computational Harmonic Analysis.

[16] Lafon, S.S. (2004). "Diffusion Maps and Geometric Harmonics". PhD thesis, Yale University.

[17] S. T. Roweis and L. K. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, Science Vol 290, 22 December 2000, 2323–2326.

[18] Mikhail Belkin and Partha Niyogi, Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering, Advances in Neural Information Processing Systems 14, 2001, p. 586–691, MIT Press

[19] Mikhail Belkin Problems of Learning on Manifolds, PhD Thesis, Department of Mathematics, The University Of Chicago, August 2003

[20] Bengio et al. "Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering" in Advances in Neural Information Processing Systems (2004)

[21] Mikhail Belkin Problems of Learning on Manifolds, PhD Thesis, Department of Mathematics, The University Of Chicago, August 2003

[22] Ohio-state.edu

[23] D. Donoho and C. Grimes, "Hessian Eigen maps: Locally linear embedding techniques for high-dimensional data" Proc Natl Acad Sci U S A. 2003 May 13; 100(10): 5591–5596

[24] Z. Zhang and J. Wang, "MLLE: Modified Locally Linear Embedding Using Multiple Weights" http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.70.382

[25] Zhang, Zhenyue; Hongyuan Zha (2004). "Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment". SIAM Journal on Scientific Computing 26 (1): 313–338.

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