



Journal of Artificial Intelligence

ISSN 1994-5450

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

Comparative Study of Dimensionality Reduction Techniques for Data Visualization

F.S. Tsai

School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore

Abstract: This study analyzed current linear and nonlinear dimensionality reduction techniques in the context of data visualization. A summary of current linear and nonlinear dimensionality reduction techniques was presented. Linear techniques such as Principal Component Analysis (PCA) and Multidimensional Scaling (MDS) are good for handling data that is inherently linear in nature. Nonlinear techniques such as Locally Linear Embedding (LLE), Hessian LLE (HLLE), Isometric Feature Mapping (Isomap), Local Tangent Space Alignment (LTSA), Kernel PCA, diffusion maps and multilayer autoencoders can perform well on nonlinear data. Experiments were conducted on varying the neighborhood, density and noise of data. Results on two real-world datasets (ORL face and business blogs) indicate that dimensionality reduction techniques generally performed better on the synthetic data. In our experiments, the best performing algorithm overall for both real-world and artificial data was Isomap.

Key words: Feature extraction, visual representation, multidimensional scaling, isomap, locally linear embedding

INTRODUCTION

Dimensionality reduction is the search for a small set of features to describe a large set of observed dimensions. Dimensionality reduction is useful in visualizing data, discovering a compact representation, decreasing computational processing time and addressing the curse of dimensionality of high-dimensional spaces. In addition, reducing the number of dimensions can separate the important features or variables from the less important ones, thus providing additional insight into the nature of the data that may otherwise be left undiscovered.

When analyzing large data of multiple dimensions, it may be necessary to perform dimensionality reduction (i.e., projection or feature extraction) techniques to transform the data into a smaller, more manageable set. By reducing the data set, we hope to uncover hidden structure that aids in the understanding as well as visualization of the data. Visualization helps to graphically depict the underlying knowledge in the data and includes techniques such as category information visualization, ontology visualization and summary visualization (Mala and Geetha, 2008). Dimensionality reduction techniques such as Principal Component Analysis (PCA) (Pearson, 1901) and Multidimensional Scaling (MDS) (Cox and Cox, 2000; Davison, 2000; Kruskal and Wish, 1978) have existed for quite some time, but most are capable only of handling data that is inherently linear in nature. Recently, some unsupervised nonlinear techniques for dimensionality reduction such as Locally Linear Embedding (LLE) (Roweis and Saul, 2000), Hessian LLE (HLLE) (Donoho and Grimes, 2003),

Isometric Feature Mapping (Isomap) (Tenenbaum *et al.*, 2000), Local Tangent Space Alignment (LTSA) (Zhang and Zha, 2005), Kernel PCA (Scholkopf *et al.*, 1998), diffusion maps (Lafon and Lee, 2006) and multilayer autoencoders (Hinton and Salakhutdinov, 2006) have achieved remarkable results for data that fit certain types of topological manifolds. However, the nonlinear techniques tend to be extremely sensitive to noise, sample size and choice of neighborhood or other parameters. This study reviews dimensionality reduction techniques for visualization and examines the reasons why certain techniques do well on some types but not other classes of data by performing a detailed analysis of existing techniques on various classes of nonlinear data sets. Although, we do not attempt to review all possible dimensionality reduction techniques, we try to cover a broad spectrum of techniques, so that results can be evaluated and general conclusions can be made. As new dimensionality reduction algorithms are constantly being developed, we think that it is important to analyze current techniques to lead to a better understanding for the development of new techniques, which can benefit the entire pattern recognition community. Even though there are previous review articles on dimensionality reduction techniques (Lee and Verleysen, 2007; Van der Maaten, 2007; Tsai and Chan, 2007), we provide both a comparative study with common data sets and quantitative metrics for evaluation, which is lacking in previous studies. Thus, we feel that this study provides up-to-date quantitative evaluation that has not been sufficiently addressed in earlier articles.

LINEAR DIMENSIONALITY REDUCTION TECHNIQUES

Linear dimensionality reduction transforms the data to a reduced dimension space using a linear combination of the original variables. The aim is to replace the original variables by a smaller set of underlying variables. Unsupervised linear dimensionality reduction techniques include Principal Component Analysis (PCA) and Multidimensional Scaling (MDS).

Principal Component Analysis (PCA) (Pearson, 1901), also known as Karhunen-Loeve transform, is a linear transformation method, so it is simple to compute and is guaranteed to work. It is useful in reducing dimensionality and finding new, more informative, uncorrelated features. It is a popular technique for dimensionality reduction and classification tasks (Quanhua *et al.*, 2008). However, since PCA is a linear dimensionality reduction technique, it may not be able to accurately represent nonlinear data. Also, one does not know how many principal components to keep, although as a general rule, the number may be chosen such that the variance of the components is roughly 90-95% of the original variance. In addition, PCA may not lead to an interesting viewpoint for data clustering because it is not good at discriminating data.

Whereas PCA takes a set of points in \mathbb{R}^n and gives a mapping of the points in \mathbb{R}^d , Multidimensional Scaling (MDS) (Cox and Cox, 2000) takes a matrix of pairwise distances and gives a mapping into \mathbb{R}^d . Some advantages of MDS are that it is relatively simple to implement, very useful for visualization and thus, able to uncover hidden structure in the data. Some drawbacks of using MDS include the difficulty in selecting the appropriate dimension of the map, difficulty in representing small distances corresponding to the local structure and unlike PCA, cannot obtain an $(n-1)$ -dimensional map out of an n -dimensional one by dropping a coordinate (Belkin and Niyogi, 2003). Additional weaknesses of MDS are its high computational and memory complexity, $O(n^3)$ and $O(n^2)$, respectively.

NONLINEAR DIMENSIONALITY REDUCTION TECHNIQUES

Linear dimensionality reduction techniques such as PCA and classical MDS are unsuitable if the data set contains nonlinear relationships among the variables. General MDS techniques are appropriate when the data is highly nonmetric or sparse. If the original high-dimensional data set contains nonlinear relationships, then nonlinear dimensionality reduction techniques may be more appropriate.

Some methods such as LLE and Isomap rely on applying linear techniques on a set of local neighborhoods, which are assumed to be locally linear in nature. As such, they fall into the category of local linear dimensionality reduction techniques.

Isometric Feature Mapping (Isomap) (Tenenbaum *et al.*, 2000) is a nonlinear dimensionality reduction technique that uses MDS techniques with geodesic interpoint distances instead of Euclidean distances. Geodesic distances represent the shortest paths along the curved surface of the manifold (a subspace of \mathbb{R}^n , measured as if the surface were flat). Unlike the linear techniques, Isomap can discover the nonlinear degrees of freedom that underlie complex natural observations (Tenenbaum *et al.*, 2000). Isomap is a very useful noniterative, polynomial-time algorithm for nonlinear dimensionality reduction if the data is severely nonlinear. Isomap is able to compute a globally optimal solution and for a certain class of data manifolds, is guaranteed to converge asymptotically to the true structure (Tenenbaum *et al.*, 2000). However, Isomap may not easily handle more complex domains such as non-trivial curvature or topology and it has the same (high) complexities as MDS.

Locally Linear Embedding (LLE) (Roweis and Saul, 2000) is a nonlinear dimensionality reduction technique that computes low-dimensional, neighborhood preserving embeddings of high-dimensional inputs. Unlike Isomap, LLE eliminates the need to estimate pairwise distances between widely separated data points and recovers global nonlinear structure from locally linear fits (Roweis and Saul, 2000). LLE assumes that the manifold is linear when viewed locally. Compared to Isomap, LLE is more efficient. However, LLE finds an embedding that only preserves the local structure, is not guaranteed to asymptotically converge and may introduce unpredictable distortions. LLE should be able to perform well for open planar manifolds, but only if the surface is a smooth curve and the neighborhood chosen is small enough. LLE falls into the general category of local linear transformations (Kirby, 2000). Both Isomap and LLE algorithms require dense data points on the manifold for good estimation and are strongly dependent on a good local neighborhood for success.

Other variations of LLE, such as *Hessian Eigenmaps* (Donoho and Grimes, 2003), have been developed that combines LLE with Laplacian Eigenmaps (Belkin and Niyogi, 2003). Hessian Eigenmap modifies the Laplacian Eigenmap framework, where they substitute a quadratic form based on the Hessian in place of one based on the Laplacian (Donoho and Grimes, 2003). The Laplacian matrix is a matrix representation of a graph and the Hessian matrix is the square matrix of second-order partial derivatives of a function. The computational demands of LLE algorithms are very different than those of the Isomap distance-processing step. LLE and HLLE are both capable of handling large n problems, because initial computations are performed only on smaller neighborhoods, whereas Isomap has to compute a full matrix of graph distances for the initial distance-processing step. However, both LLE and HLLE are more sensitive to the dimensionality of the data space, n , because they must estimate a local tangent space at each point. Although an orthogonalization step was introduced in HLLE that makes the local fits more robust to pathological neighborhoods than LLE, HLLE still requires effectively a numerical second

differencing at each point that can be very noisy at low sampling density (Donoho and Grimes, 2003). An additional weakness of Hessian LLE is that it cannot embed data into a dimensionality $d > k$.

Local Tangent Space Alignment (LTSA) (Zhang and Zha, 2005) is a method for nonlinear dimensionality reduction that constructs approximations of tangent spaces in order to represent local geometry of the manifold and the global alignment of the tangent spaces to obtain the global coordinate system (Zhang and Zha, 2005). Based on a set of unorganized data points sampled with noise from a parameterized manifold, the local geometry of the manifold is learned by constructing an approximation for the tangent space at each data point and those tangent spaces are then aligned to give the global coordinates of the data points with respect to the underlying manifold (Zhang and Zha, 2005). In general, LTSA is less sensitive to the choice of k neighborhoods, as compared to LLE. Although LTSA as well as the other nonlinear dimensionality reduction algorithms are able to handle nonlinearities in data, they are generally not as robust as the linear dimensionality reduction techniques and not able to handle certain types of nonlinear manifolds. Similar to HLLLE, LTSA also cannot embed data into a dimensionality $d > k$.

Kernel principal component analysis (Kernel PCA) (Scholkopf *et al.*, 1998) is an extension of Principal Component Analysis (PCA) using techniques of kernel methods. Using a kernel, the originally linear operations of PCA are done in a reproducing kernel Hilbert space with a nonlinear mapping. Since kernel PCA is a kernel-based method, the mapping performed by Kernel PCA relies on the choice of the kernel function κ . Possible choices for the kernel function include the linear kernel (which is the same as traditional PCA), the polynomial kernel and the Gaussian kernel. One important weakness of kernel PCA is that the size of the kernel matrix is proportional to the square of the number of instances in the dataset (Van der Maaten *et al.*, 2008).

Autoencoders (Hinton and Salakhutdinov, 2006) are artificial neural networks used for learning efficient codings. The aim of an autoencoder is to learn a compressed representation (encoding) for a set of data. An autoencoder is often trained using one of the many backpropagation variants (Conjugate Gradient Method, Steepest Descent, etc.) Though often reasonably effective, there are fundamental problems with using backpropagation to train networks with many hidden layers. Once the errors get backpropagated to the first few layers, they are minuscule and quite ineffectual. This causes the network to almost always learn to reconstruct the average of all the training data. Though more advanced backpropagation methods (such as the Conjugate Gradient Method) help with this to some degree, it still results in very slow learning and poor solutions. This problem is remedied by using initial weights that approximate the final solution. The process to find these initial weights is often called pretraining. A pretraining technique developed by Geoffrey Hinton for training many-layered deep autoencoders involves treating each neighboring set of two layers like a Restricted Boltzmann Machine for pretraining to approximate a good solution and then using a backpropagation technique to fine-tune. The computational complexity of autoencoders is dependent on n (the matrix size), w (the number of weights in the neural network) and i (the number of iterations). The memory usage is dependent on w weights. Therefore, if the weights and iterations are large, autoencoders can have very high complexity and memory usage.

Diffusion Maps (Lafon and Lee, 2006; Nadler *et al.*, 2006) are based on defining a Markov random walk on the graph of the data. The technique defines a system of coordinates with an explicit metric that reflects the connectivity of a given data set and that is robust to noise. This diffusion metric is a transition probabilities of a Markov chain that evolves forward in time and is very robust to noise, unlike the geodesic or Euclidean

Table 1: Summary of dimensionality reduction techniques (Van der Maaten *et al.*, 2008)

Technique	Parameter	Complexity	Memory
PCA	none	$O(n^3)$	$O(n^2)$
MDS	none	$O(n^3)$	$O(n^2)$
Isomap	k	$O(n^2)$	$O(n^2)$
LLE	k	$O(pn^2)$	$O(pn^2)$
HLLS	k	$O(pn^2)$	$O(pn^2)$
LTSA	k	$O(pn^2)$	$O(pn^2)$
Autoencoders	net size	$O(inw)$	$O(w)$
Kernel PCA	$\kappa(\bullet, \bullet)$	$O(n^3)$	$O(n^2)$
Diffusion Maps	σ, t	$O(n^2)$	$O(n^2)$

distance. Though diffusion maps perform exceptionally well with clean, well-sampled data, problems arise with the addition of noise, or when the data exists in multiple submanifolds (Van der Maaten *et al.*, 2008). Although diffusion maps are similar to Isomap, they integrate over all paths through the graph instead of only considering shortest paths. This makes them less sensitive to short-circuiting than Isomap.

Summary of Techniques

Table 1 summarizes the main free parameters that have to be optimized, the computational complexity of each technique and the memory complexity of the technique (Van der Maaten *et al.*, 2008). In Table 1, n is the matrix size, k is the nearest neighbors, p is the ratio of nonzero elements in a sparse matrix to the total number of elements, w is the number of weights in a neural network and i is the number of iterations.

LLE, HLLS and LTSA are more computationally efficient than MDS and Isomap, with roughly similar memory usage. Of all the techniques, autoencoders has the highest complexity as it takes a long time to converge if the number of weights are high.

EXPERIMENTS AND RESULTS

In order to evaluate and compare the various techniques, experiments were conducted using dimensionality reduction techniques on three synthetic datasets (S-Curve, toroidal helix and twin peaks). We examined the results of the various dimensionality reduction techniques under different conditions, such as different levels of noise, different n data points and different k as well as adaptive neighborhoods. We also compared the various techniques for two real datasets, Olivetti face data (Samaria and Harter, 1994) and business blogs (Chen *et al.*, 2008). The programs were executed using the Matlab Toolbox for Dimensionality Reduction (Van der Maaten, 2007).

Manifold Visualization Metric

In order to quantitatively evaluate the visualization results, we used the manifold visualization metric based on the correlation coefficient that computes the pairwise geodesic distance vector between the original manifold and the lower-dimensional embedding results. The calculation of the metric is similar to the correlation coefficient used by Geng *et al.* (2005), except that the pairwise geodesic distance vector is calculated for the original data instead of the Euclidean distance vector. A previous study showed that metric was more suitable for representing the visualization results if the original data lies on a manifold (Tsai and Chan, 2009).

Results of Dimensionality Reduction for Artificial Data

Experiments were conducted using different dimensionality reduction techniques for various categories of data sets. Figure 1 shows the results obtained using classical MDS,

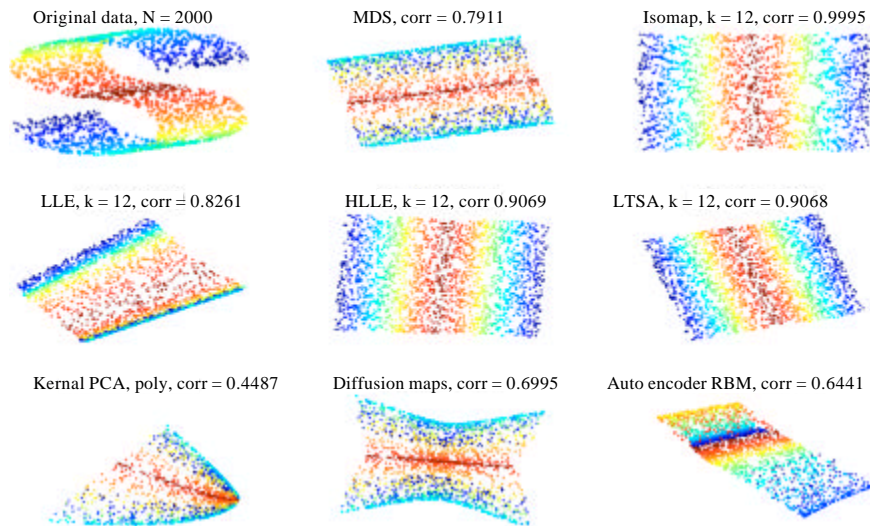


Fig. 1: Results of reducing dimension on a S-Curve, N = 2000

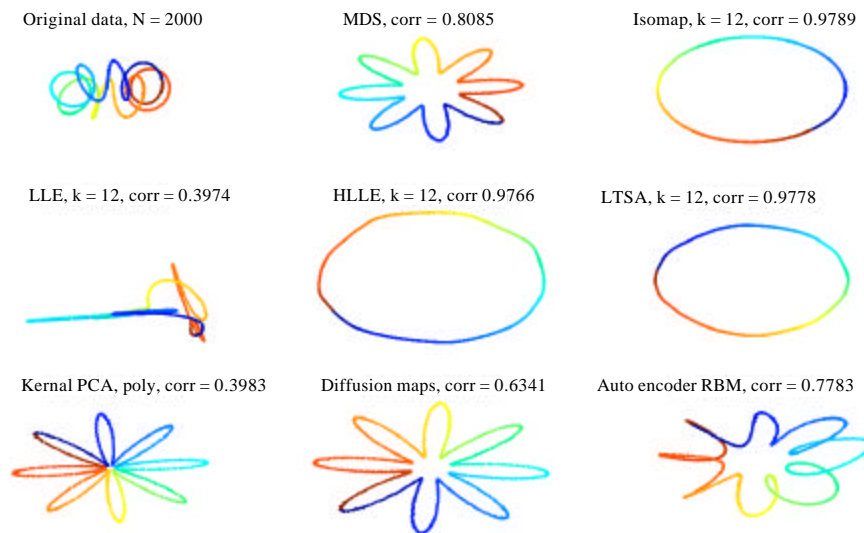


Fig. 2: Results of reducing dimension on toroidal helix, N = 2000

Isomap, LLE, HLLC, LTSA, Kernel PCA, diffusion maps and autoencoders on S-Curve, with 2000 data points. The different colored lines in each figure represent different classes. For this dataset, Isomap performed the best, with a correlation score of 0.9995, followed by HLLC (0.9069) and LTSA (0.9068). LLE and MDS performed better than autoencoders and diffusion maps. Kernel PCA did not produce a good embedding, with a correlation score of 0.4487.

Figure 2 shows the results obtained using classical MDS, Isomap, LLE, HLLC, LTSA, Kernel PCA, diffusion maps and autoencoders on a toroidal helix, with 2000 data points. The best performance is achieved using Isomap, LTSA and HLLC, with a correlation score of

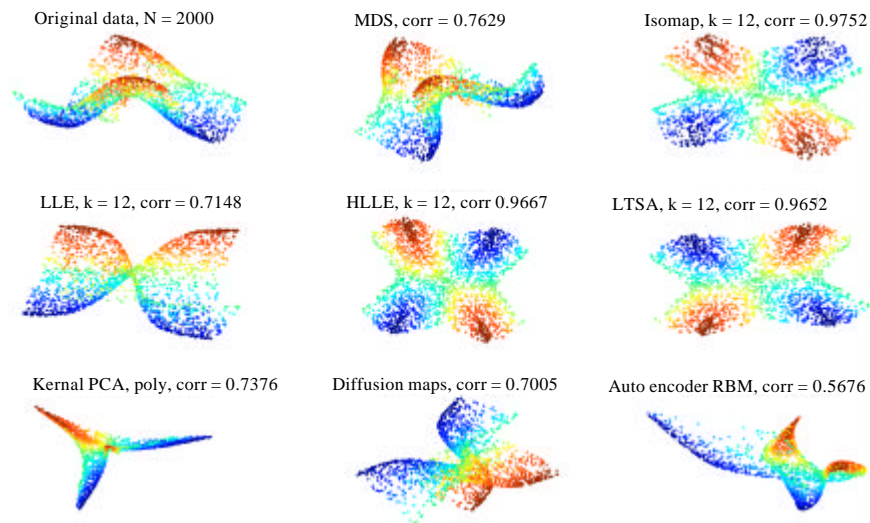


Fig. 3: Results of reducing dimension on twin peaks, N = 2000

0.9789, 0.9778 and 0.9766, respectively, with the low dimensional results reflect this score, as the correct embedding results should be in the form of a circle. MDS and auto encoders performed reasonably well overall, with a score of 0.8085 and 0.7783, respectively. LLE and Kernel PCA achieved the lowest scores for this dataset.

Figure 3 shows the results obtained using classical MDS, Isomap, LLE, HLLE, LTSA, Kernel PCA, diffusion maps and autoencoders on twin peaks, with 2000 data points. The results are quite different than those obtained previously with the toroidal helix. For this dataset, Isomap performed the best, with a correlation score of 0.9752, followed closely by HLLE (0.9667) and LTSA (0.9652). For this dataset, MDS (0.7629) performed slightly better than Kernel PCA (0.7376), LLE (0.7148) and diffusion maps (0.7005). The autoencoders did not produce a good embedding, with a correlation score of 0.5676.

Results for Nonlinear Manifolds in Presence of Noise

To see the performance of the various algorithms in the presence of noise, experiments were conducted on adding Gaussian noise to a nonlinear manifold in the shape of a toroidal helix. Figure 4 and 5 show the results of the various algorithms in the presence of noise. As the noise increased from 5 to 10%, the quality of the lower-dimensional embedding mostly went down.

Results for Different N Data Points

Densely sampled data points are also required for good embedding results. If there are not enough data points in the manifold, then the results of the dimensionality reduction algorithms may not be good. To illustrate this point, experiments were conducted on different values of n data points, sampled on a nonlinear manifold.

Figure 6 and 7 show the results of applying the dimensionality reduction algorithms on a nonlinear manifold in the form of a S-Curve, with different values of N sampled data points. As the number of points increased to 500, the results of applying the dimensionality reduction algorithms all increased.

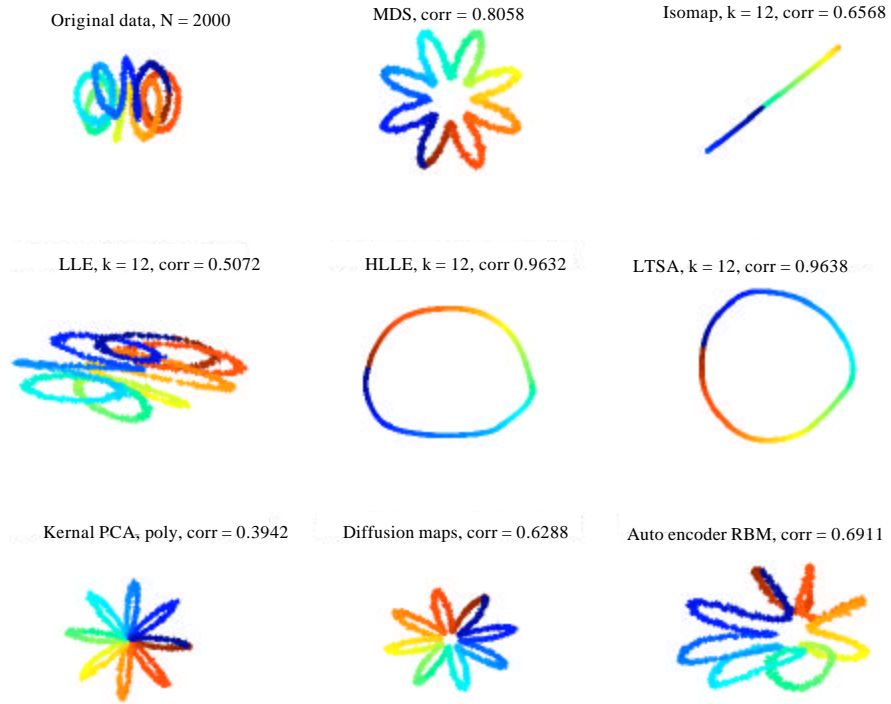


Fig. 4: Results on adding 5% noise to toroidal helix

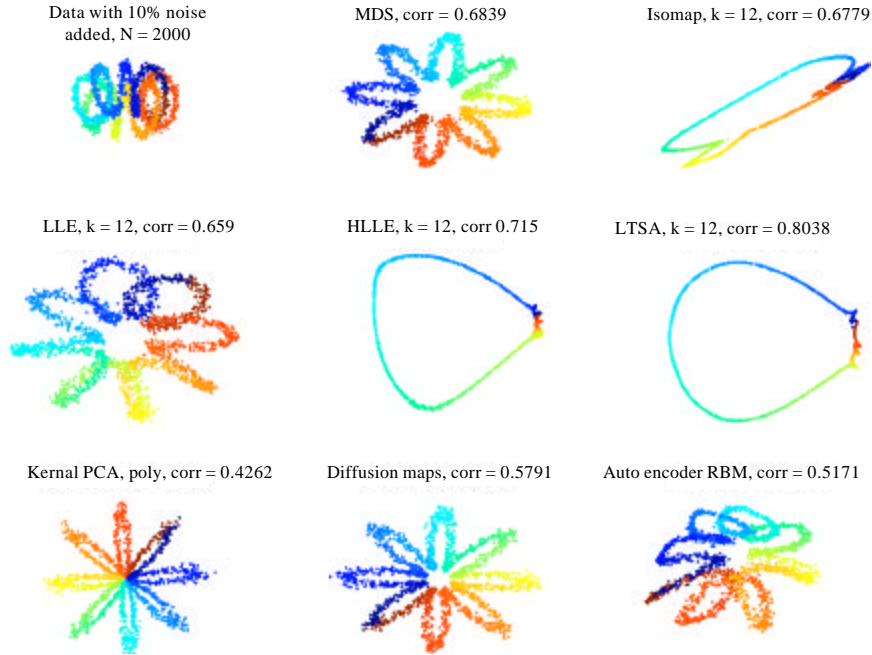


Fig. 5: Results on adding 10% noise to toroidal helix

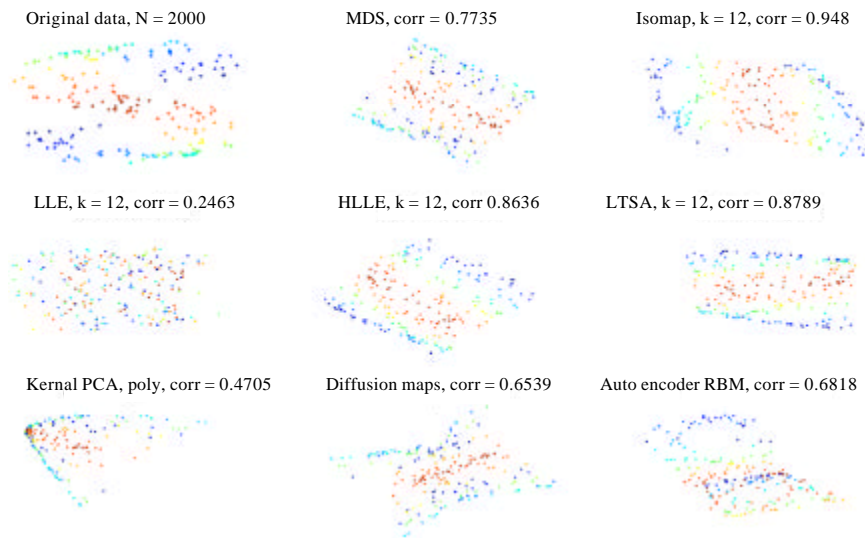


Fig. 6: Results of reducing dimension on a S-Curve, N = 200

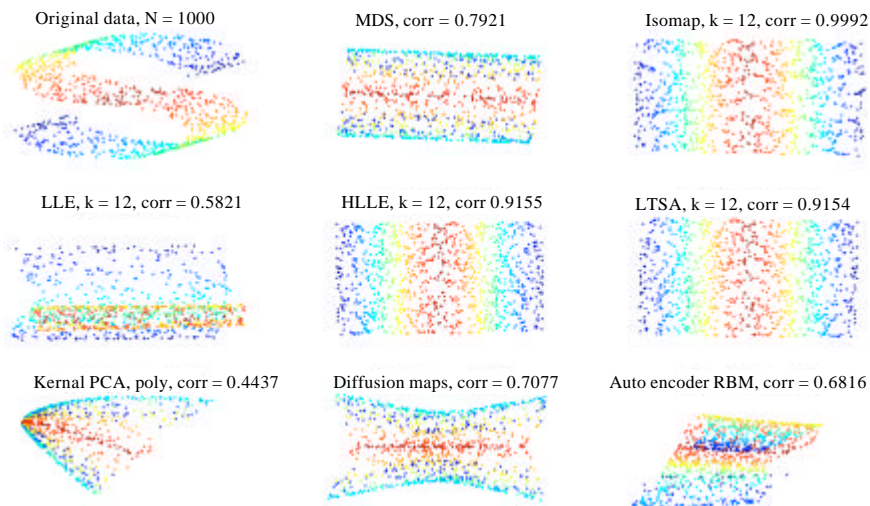


Fig. 7: Results of reducing dimension on a S-Curve, N = 1000

Figure 8 and 9 show the results of applying the dimensionality reduction algorithms on a nonlinear manifold in the form of a toroidal helix, with different values of n sampled data points.

Results of LLE, Isomap, HLLC and LTSA for Different k Neighborhoods

Experiments were conducted on varying the neighborhood parameter for LLE, Isomap, HLLC and LTSA. Figure 10 for the results of applying the techniques with different values of k neighborhoods, on a nonlinear manifold in the form of a toroidal helix with 1000 sampled data points. The results of using the adaptive option in the MATLAB Toolbox for

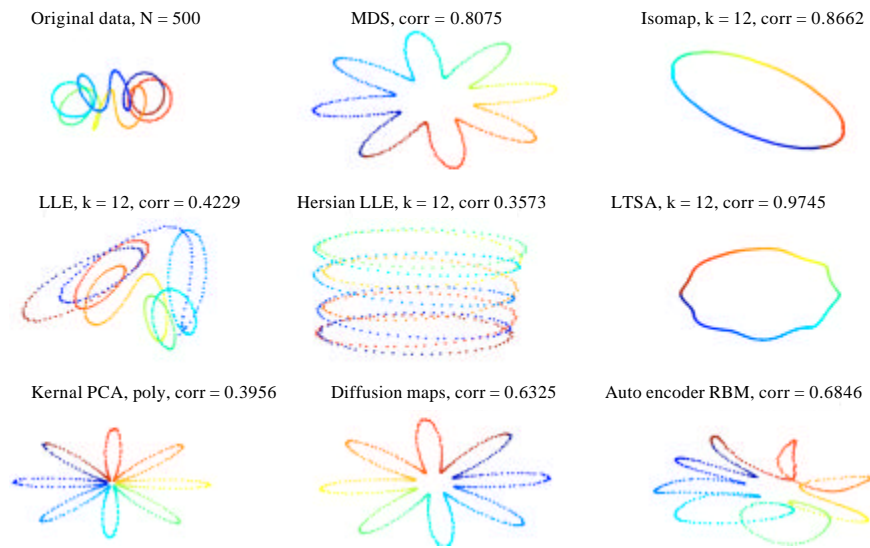


Fig. 8: Results of reducing dimension on a helix, $N = 500$

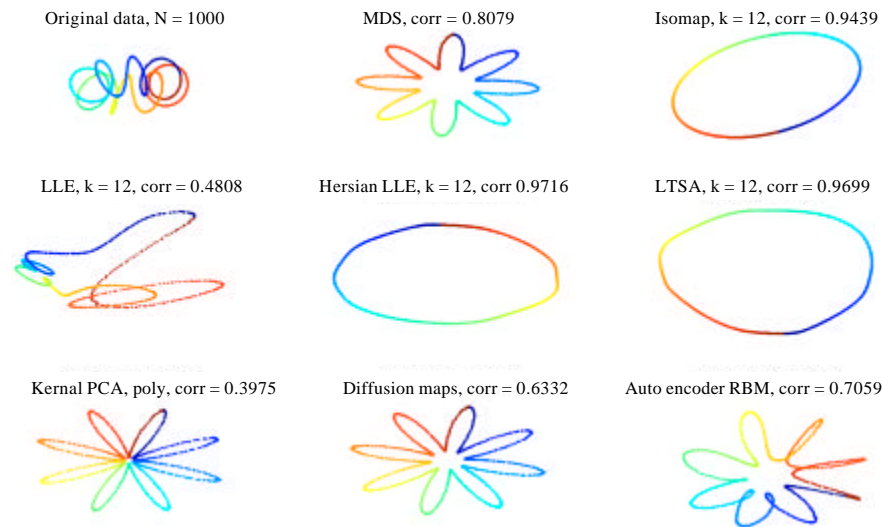


Fig. 9: Results of reducing dimension on a helix, $N = 1000$

Dimensionality Reduction (Van der Maaten, 2007), which implemented the adaptive neighborhood selection described by Mekuz and Tsotsos (2006). For the adaptive neighborhood selection, only Isomap was able to run, as running on LLE, HLLS and LTSA resulted in singularities. As seen from the results, the adaptive neighborhood selection did not necessarily correspond to the ideal neighborhood for Isomap. All the techniques shown here depend on a good local neighborhood for success. This is one of the restrictions of the techniques that rely on the parameter of k neighborhoods. The adaptive neighborhood

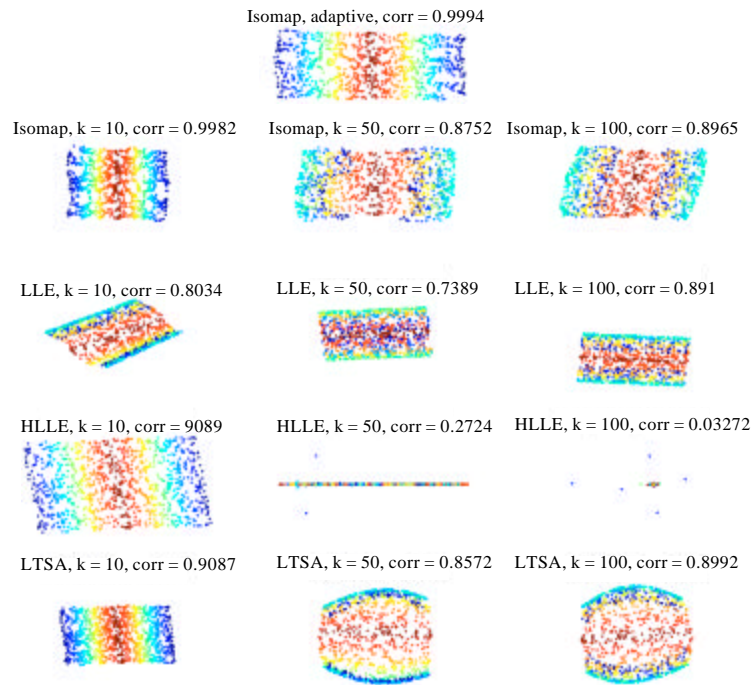


Fig. 10: Results of varying k neighborhoods for S-Curve

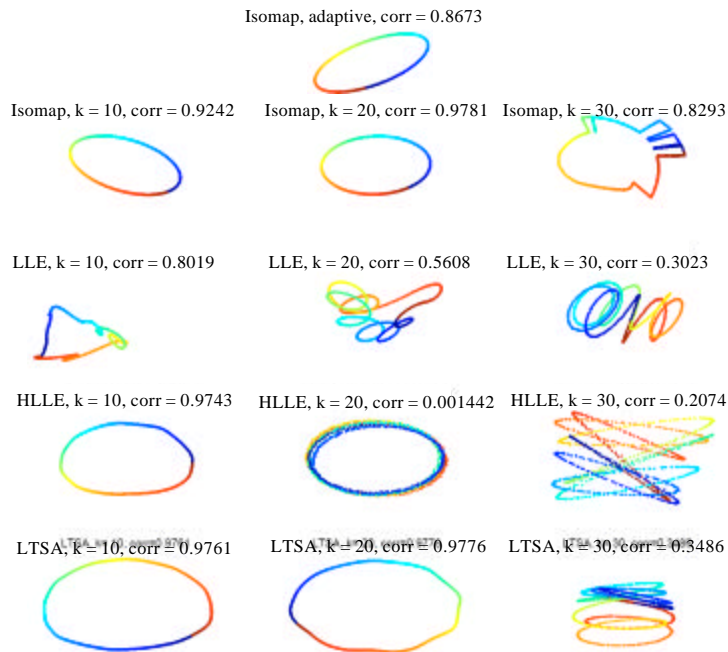


Fig. 11: Results of varying k neighborhoods for toroidal helix

selection may not always be a viable option, as many of the techniques were not able to make use of the adaptive neighborhoods.

Figure 11 for the results of applying LLE, Isomap, HLLC and LTSA with different values of k neighborhoods, on a nonlinear manifold in the form of a toroidal helix with 1000 sampled data points. Again, for the adaptive neighborhood selection, only Isomap was able to run. Similarly to the S-Curve dataset, the results were quite different depending on the neighborhoods.

Figure 12 for the results of applying LLE, Isomap, HLLC and LTSA with different values of k neighborhoods, on a nonlinear manifold in the form of twin peaks with 1000 sampled data points. For this dataset, both Isomap and LLE were able to run with the adaptive neighborhood selection, but the adaptive neighborhood was not always the ideal neighborhood. However, if there is no prior knowledge about the dataset, the adaptive option can be a viable option, provided that the algorithms are able to find an adaptive neighborhood. The twin peaks dataset seemed more robust to different neighborhoods when using the Isomap and LTSA algorithms, compared to the previous two datasets.

Summary of Results on Artificial Data

From the results on artificial data, we observe that Isomap, LLE, HLLC and LTSA depend on a good local neighborhood for success. In addition to a good neighborhood, densely sampled data points are also required. These findings are consistent with the results from previous studies (Tenenbaum *et al.*, 2000; Roweis and Saul, 2000; Donoho and Grimes, 2003; Zhang and Zha, 2005; Tsai and Chan, 2007). Data with a relatively low noise level is also required in order to fit the manifold to the topological structure. This is where topology learning can help to determine if the nature of the data set is suitable for such techniques. Therefore, although these manifold learning algorithms can achieve remarkable results with certain classes of manifolds, they can benefit further by using topology learning to recommend the optimal algorithms and parameters for various topologies.

Results on Real Data

To see the performance of the various algorithms on real-world data, we used the following data sets: Olivetti Research Laboratory (ORL) face data (Samaria and Harter, 1994) and BizBlogs07 blog data (Chen *et al.*, 2008). The ORL dataset (Samaria and Harter, 1994) is a face recognition dataset that contains 400 grayscale images of 112×92 pixels that depict 40 faces under various conditions (10 images per face). Figure 13 shows the results of applying the dimensionality reduction algorithms on the ORL face data.

The best algorithm for this data was LTSA (0.7131), followed by HLLC and Isomap. Autoencoders, MDS and LLE had similar results, with scores ranging from 0.4708 down to 0.4198. Kernel PCA and diffusion maps did not perform well on this dataset. As previous studies (Donoho and Grimes, 2003; Hinton and Salakhutdinov, 2006; Lafon and Lee, 2006; Roweis and Saul, 2000; Schölkopf *et al.*, 1998; Tenenbaum *et al.*, 2000; Zhang and Zha, 2005) did not evaluate the results quantitatively, the conclusions presented here are new and unexpected.

We next tried the algorithms on another real-world dataset of business blogs, BizBlogs07 (Chen *et al.*, 2008). BizBlogs07 contained 1269 business blog entries from various CEOs's blog sites and business blog sites. There were a total of 86 companies represented in the blog entries and the blogs were classified into four categories based on the contents or the main description of the blog: Product, Company, Marketing and Finance (Chen *et al.*, 2008). In order to prepare the dataset, we first removed stopwords, performed word stemming

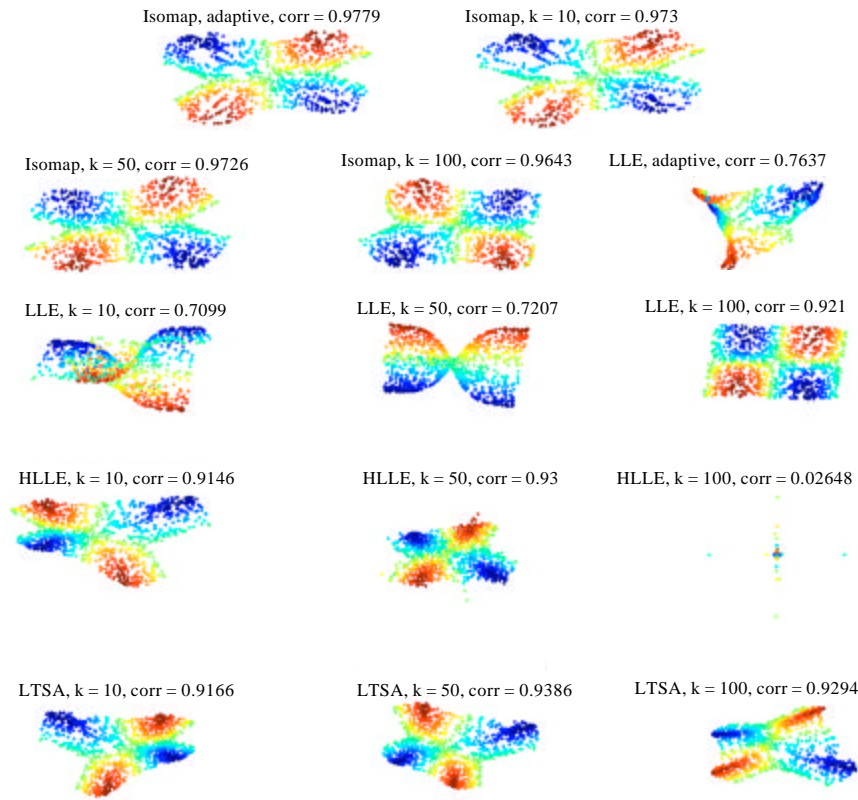


Fig. 12: Results of varying k neighborhoods for twin peaks data

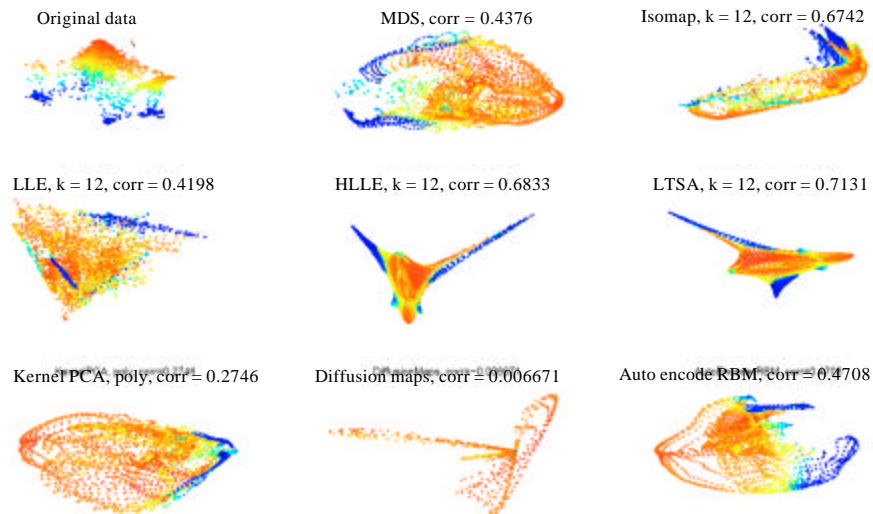


Fig. 13: Results for dimensionality reduction on Olivetti face data

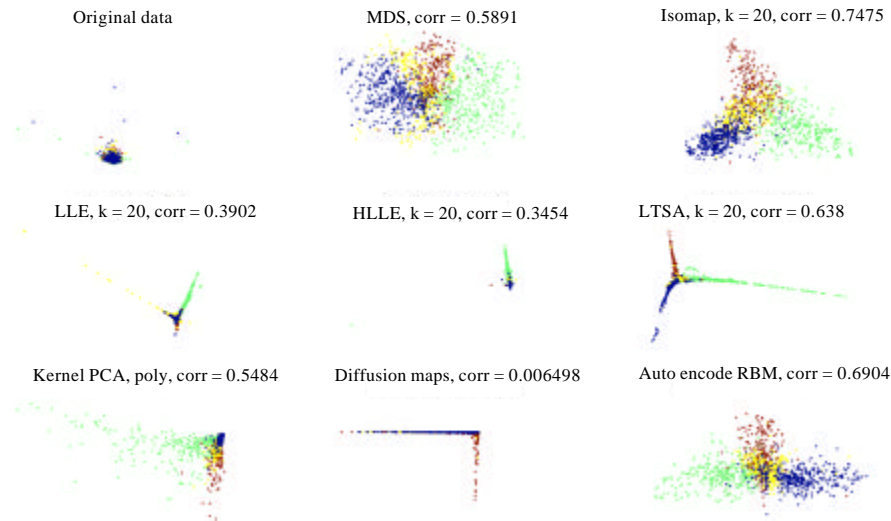


Fig. 14: Results for dimensionality reduction on blog data

and created a normalized term-document matrix with term frequency (TF) local term weighting and Inverse Document Frequency (IDF) global term weighting. From this matrix, we created the 1269×1269 document-document cosine similarity matrix and used this as input to the dimensionality reduction algorithms. Figure 14 shows the results of applying the dimensionality reduction algorithms on the BizBlogs07 business blog data. Each of the colors correspond to one of the four categories of the blog data and only the first three dimensions of the original data is shown.

Isomap had the highest correlation score (0.7475) of all the techniques. Autoencoders, LTSA, MDS and Kernel PCA had correlation scores ranging from 0.6904 down to 0.5484. LLE, HLLE and diffusion maps all had low embedding results. As this is the first study evaluating dimensionality reduction techniques on business blogs, the results presented here are new and incomparable to past studies on other data. The performance of the dimensionality reduction techniques on real-world data was generally worse than on the artificial data. In our experiments, the best performing algorithm overall for both real-world and artificial data was Isomap.

CONCLUSIONS

The growth of high-dimensional data creates a need for dimensionality reduction techniques to transform the data into a smaller, more manageable set which can be easily visualized. In this study, we have surveyed research on the use of dimensionality reduction techniques for data visualization. A summary of some current linear and nonlinear dimensionality reduction techniques was presented.

For basic types of nonlinear manifolds, an evaluation was performed on various dimensionality reduction techniques. Because the nonlinear dimensionality reduction techniques depend on a good neighborhood, the results when using different neighborhoods are significantly different. A good neighborhood will depend on the particular characteristics of the data set. If the neighborhood is too small, the global structure

may not be captured effectively. If the neighborhood is too big, the nonlinearities of the data set may not be mapped appropriately. Because a “good” neighborhood is dependent on the data set, it is difficult to generalize an acceptable value that will work under all circumstances. Adaptive algorithms may not always be a viable option as they often results in singularities and are not able to run.

In addition, the algorithms require that the manifold consists of well-sampled data points, otherwise the nonlinearities in the manifold structure may not be effectively captured. Thus, the algorithms may not work properly if the data set is very sparse, as seen from the results.

From the results, the techniques generally do not perform well in the presence of noise. Thus, in order to apply dimensionality reduction techniques effectively, the neighborhood, the density and noise levels need to be taken into account.

In our experiments, the best performing algorithm overall for both real-world and artificial data was Isomap. In general, the dimensionality reduction techniques performed better on the synthetic data; however, some algorithms such as Isomap performed reasonably well on the real-world data.

Future work can benefit by using topology learning to recommend the optimal algorithms and parameters for various topologies and thus, extend the applicability of dimensionality reduction techniques for real-world data.

REFERENCES

- Belkin, M. and P. Niyogi, 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15: 1373-1396.
- Chen, Y., F.S. Tsai and K.L. Chan, 2008. Machine learning techniques for business blog search and mining. *Expert Syst. Appl.*, 35: 581-590.
- Cox, T.F. and M.A. Cox, 2000. *Multidimensional Scaling*. 2nd Edn., Chapman and Hall/CRC, New York, ISBN: 978-1584880943.
- Davison, M., 2000. *Multidimensional Scaling*. Krieger Publishing Company, Florida.
- Donoho, D.L. and C. Grimes, 2003. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *PNAS*, 100: 5591-5596.
- Geng, X., D.C. Zhan and Z.H. Zhou, 2005. Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Trans. Syst. Man Cybernetics Part B*, 35: 1098-1107.
- Hinton, G.E. and R.R. Salakhutdinov, 2006. Reducing the dimensionality of data with neural networks. *Science*, 313: 504-507.
- Kirby, M., 2000. *Geometric Data Analysis: An Empirical Approach to Dimensionality Reduction and the Study of Patterns*. John Wiley and Sons Inc., New York, USA.
- Kruskal, J. and M. Wish, 1978. *Multidimensional Scaling*. Sage Publications, London, ISBN: 978-0803909403.
- Lafon, S. and A.B. Lee, 2006. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning and data set parameterization. *IEEE Trans. Pattern Anal. Machine Intel.*, 28: 1393-1403.
- Lee, J. and M. Verleysen, 2007. *Nonlinear Dimensionality Reduction*. Springer, New York, ISBN: 978-0387393506.
- Mala, T. and T.V. Geetha, 2008. Story summary visualizer using L systems. *J. Artif. Intell.*, 1: 53-60.

- Mekuz, N. and J.K. Tsotsos, 2006. Parameterless Isomap with Adaptive Neighborhood Selection. In: *Lecture Notes in Computer Science*, Pranke, K. *et al.* (Eds.). Springer, Berlin, Heidelberg, ISBN: 978-3-540-44412-1, pp: 364-373.
- Nadler, B., S. Lafon, R. Coifman and I. Kevrekidis, 2006. Diffusion maps, spectral clustering and the reaction coordinates of dynamical systems. *Applied Comput. Harmonic Anal.*, 21: 113-127.
- Pearson, K., 1901. On lines and planes of closest fit to systems of points in space. *Phil. Mag. Ser. B.*, 2: 559-572.
- Quanhua, C., L. Zunxiong and D. Guoqiang, 2008. Facial gender classification with eigenfaces and least squares support vector machine. *J. Artif. Intell.*, 1: 28-33.
- Roweis, S.T. and L.K. Saul, 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290: 2323-2326.
- Samaria, F.S. and A.C. Harter, 1994. Parameterisation of a stochastic model for human face identification. *Proceedings of the 2nd IEEE Workshop on Applications of Computer Vision*, Dec. 5-7, Sarasota, FL, USA., pp: 138-142.
- Scholkopf, B., A.J. Smola and K.R. Muller, 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10: 1299-1319.
- Tenenbaum, J., V.D. Silva and J. Langford, 2000. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290: 2319-2323.
- Tsai, F.S. and K.L. Chan, 2007. Dimensionality reduction techniques for data exploration. *Proceedings of the 6th International Conference on Information, Communications and Signal Processing*, Dec. 10-13, Nanyang Technol. Univ., Singapore, pp: 1-5.
- Tsai, F.S. and K.L. Chan, 2009. A manifold visualization metric for dimensionality reduction. Nanyang Technological University Technical Report.
- Van der Maaten, L.J.P., 2007. An introduction to dimensionality reduction using Matlab. Report MICC 07-07. Maastricht, http://tsam-fich.wdfiles.com/local--files/apuntes/Report_final.pdf.
- Van der Maaten, L.J.P., E.O. Postma and H.J. van den Herik, 2008. Dimensionality reduction: A comparative review. MICC, Maastricht University, Tech. Report. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.125.6716&rep=rep1&type=pdf>.
- Zhang, Z. and H. Zha, 2005. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM J. Sci. Comput.*, 26: 313-338.