Study of Dimensionality Reduction techniques for handwritten digit classification

Karthik Anantha Padmanabhan
Department of Computer Science
The University of Texas at Austin
akarthik@cs.utexas.edu

Abstract

Linear Dimensionality reduction techniques like Principal Component Analysis and Fisher's Linear Discriminant assume that the intrinsic dimensionality of data lies in a linear subspace. Although these methods work for data sets that are approximately linear, they fail when the data is highly complex and nonlinear. For such datasets, Non-Linear Dimensionality reduction techniques have shown to be very effective. In this project, Non-linear dimensionality reduction techniques have been used as a preprocessing step for pattern classification. The pattern classification task that is considered here is the classification of handwritten digits. The two classes of techniques (Linear and Non-Linear) are compared by their computational complexities and by measuring the generalization error of the k-nn classifier. Analysis of the results gives an insight into the nature of the dataset and reveals some of the weaknesses of the Non-Linear techniques.

1. Introduction

Real world data that is used in patter recognition tasks have very high dimensionality. Such high dimensional data may however consist of multiple indirect measurements of the same underlying cause, which cannot be measured. Dimensionality reduction is the process of converting the data into lower dimensions, which ideally corresponds to these underlying causes. The underlying causes that characterize the data will correspond to the intrinsic dimensionality of the data [2].Intrinsic dimensionality is the minimum number of parameters required to account for the observed properties of the data[9].Dimensionality reduction techniques are important in many fields as they facilitate, among others, data visualization, data compression, noise removal and data preprocessing for classification. The most popular dimensionality reduction technique is Principal Component Analysis (PCA). PCA assumes that data lies on a linear subspace embedded in the high dimensional input space and projects data onto a lower dimensional space that accounts for maximum variance in the data. Other popular linear dimensionality reduction techniques are Fisher's linear discriminant, which uses class labels to maximize the separation of mean between the two classes, Probabilistic PCA [9] and Independent component analysis. A fundamental weakness in all these techniques is the linearity assumption that they make about the data. Consider the artificial data set shown in Figure 1(a).The projection of the data points onto the first principal component is shown in Figure 1(b). Dimensionality reduction techniques are required to preserve the distances in the lower dimensional space for visualization tasks and minimize overlap of different classes for classification tasks. PCA on this non-linear data set fails in visualization and will not do well in classification if the class separation is along the manifold. Figure 1(c) shows the performance of a non-linear technique, Locally Linear Embedding on this dataset. The mapping obtained preserves the distances and will separate the various classes when the class separation is along the manifold.

In this project, the use of Non-linear dimensionality reduction as a preprocessing tool for classification is studied. The task that is considered here is the classification of hand written digits, where each image has to be classified as one of the 10 digits. The performance of the Non-linear techniques is compared with linear techniques, mainly PCA. The metrics for comparison used in this project are: (1) Classification accuracy on a test data set. (2) Classification accuracy when noise is added to the test data set. (3) Computational complexities of the various techniques. By analyzing the results, some insight into the nature of this data set is also obtained. This project uses the following dimensionality reduction techniques: (1) Principal Component Analysis, (2) Kernel Principal Component Analysis, (3) Locally Linear Embedding, (4) Isomap, (4) Multilayer Autoencoders and Probabilistic Principal Component Analysis.

The outline of the remainder of the paper is as follows. Section 2 formally defines dimensionality reduction and gives a brief description of the above dimensionality reduction techniques. Section 3 discusses the experimental setup for the various techniques and subsequently section 4 performs an analysis based on the results obtained. In section 5, the conclusion of the project is presented.

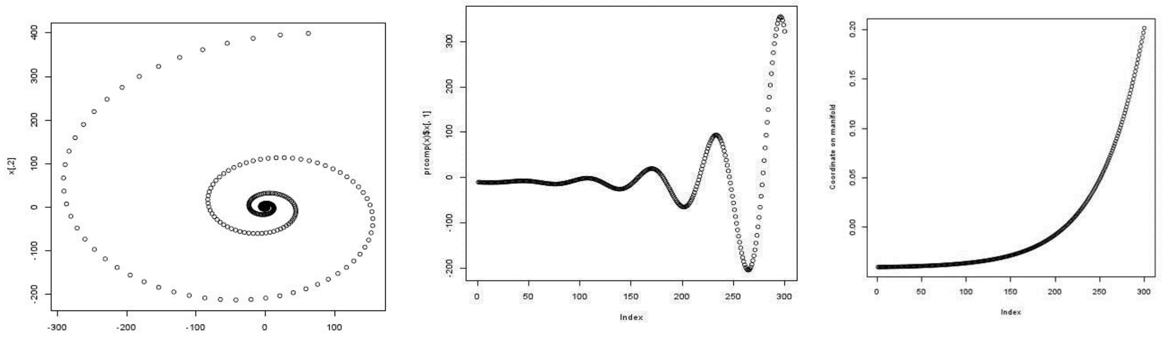

Figure 1: Linear and Non-linear dimensionality reduction on an artificial dataset.(a) Artificial spiral dataset-The index of the data points also spirals out from the center in an increasing order.(b) Projection of the data points on the first principal component obtained from PCA.(c)Projection obtained using Locally Linear embedding – a non-linear technique[16]

2. Dimensionality Reduction

Dimensionality reduction can be defined as a technique to effectively downsize high dimensional data by preserving only the most important dimensions. Consider a data set, $X = \{x_1, x_2, x_3 \ldots . x_n\}$ with n data points in a D dimensional space R^D which is represented by a $n \times D$ marix. Dimensionality reduction seeks to map each point $x_i \in R^D \rightarrow y_i \in R^m$, where $m \ll D$. In the ideal case, m should be the intrinsic dimensionality of the data, but this however is not easy to estimate [2]. The lower dimensional data can be viewed as a m dimensional manifold that is embedded on to a D dimensional space [2]. A manifold is defined as a smooth curved subset of a Euclidean space in which it is embedded [16]. Throughout this paper, a high dimensional data point will be denoted by x_i and its corresponding dimension will be denoted by D. The lower-dimensional mapping of x_i will be denoted by y_i and its dimension will be denoted by m. The two main classes of dimension reduction techniques are Linear dimensionality reduction techniques and Non-Linear dimensionality reduction techniques. As mentioned in section 1, linear techniques make the assumption that the data lie on linear subspace of the high dimensional space. Nonlinear techniques on the other hand make no such assumptions and are more flexible when compared to linear techniques, in the sense that they can handle complex nonlinear data.

2.1. Principal component analysis

Principal Component Analysis (PCA) is by far the most popular linear dimensionality reduction technique. PCA seeks to project data onto a lower dimension such that it accounts for as much of the variance as possible. It assumes that the most important modes of variation in data is linear and performs a reconstruction of the data onto that linear subspace [10]. So given n data points in R^n, PCA aims to find a linear subspace of dimension m such that $m \ll D$. The linear subspace is defined by m orthogonal basis vectors that form the new coordinate system in R^m.

Consider the dataset X defined by a $n \times D$ data matrix that consists of n rows of mean centered observations. Let C be the $D \times D$ covariance matrix of X. PCA attempts at finding a linear mapping W that maximizes $W^T C W$ .It can be shown that this linear mapping is obtained by taking the top m Eigen vectors of the covariance matrix C. The Eigen problem that PCA solves is given by

$$Cx = \lambda x \tag{1}$$

The matrix M consist of eigen vectors x that are arranged along its columns. The m dimensional reconstruction of x_i is obtained by projecting the mean centered observations onto the m principal components

$$y_i = x_i * M \tag{2}$$

PCA can be become computationally expensive when D becomes extremely large. But in situations where $n < D$, this problem is circumvented by finding the Eigen vectors of the $n \times n$ Euclidean distance matrix instead of the covariance matrix [11]. A probabilistic interpretation of PCA was given in [9]. Probabilistic PCA is useful for missing data and may be used for missing value estimation [10].

PCA has been successfully applied to a large spectrum of problem domains such as Image classification and recognition, Text classification and data visualization.

2.2. Isomap

Isomap [4] is a non-linear dimensionality reduction technique that is closely related to another technique called Multi-Dimensional scaling [12]. In Multi-dimensional scaling, high dimensional data is reduced to a lower dimension by preserving the pairwise Euclidean distance between any two points. The quality of the mapping is expressed as

$$f(Y) = \sum_{ij} \left(\left||x_i - x_j|\right| - \left||y_i - y_j|\right|\right)^2 \qquad (3)$$

Figure [2] shows the difficulty in using Euclidean distance for a nonlinear manifold. The two points in Figure [2], although quite distant on the manifold, have a smaller Euclidean distance and using Euclidean distances gives a poor mapping

In Isomap, the geodesic distance is considered instead of Euclidean distance. Geodesic distance is the distance of two points measured along a manifold [13] .The geodesic distance between two points is approximated using the shortest path method. First, a graph of the entire manifold is created. There is an edge between two points $x_i\ and\ x_j$ when xj falls within the neighborhood of x_i.This neighborhood can be defined by finding the k nearest neighbors. A distance matrix R that stores pairwise geodesic distances is defined. To find the low dimensional mapping in R^m , the top m eigen values(λ) and its corresponding eigen vectors $(v_1, v_2 \ldots v_m)$ of $\tau(R)$ are calculated, where $\tau(R) = -HSH/2$, $S_{ij} = \left(R_{ij}\right)^2$, $H_{ij} = I - \frac{1}{n} ee^T$and e is a column vector of 1's.The p^{th} dimension of y_i is computed as:

$$y_i^p = \sqrt{(}\lambda_p * v_p^i \qquad (4)$$

2.3. Kernel PCA

Kernel PCA [6] is the reformulation of PCA in a high dimensional feature space, that is related to the input space through the use of Kernel Functions [2].Kernel PCA finds the principal eigen vectors in a higher dimensional feature space and maps them to the input space using Kernel functions. But computing the eigen vectors in a high dimensional space is expensive as the complexity of PCA is proportional to its dimensions. By using kernel functions, it is not necessary to explicitly map the data onto the higher dimension. Kernel functions find the dot product in the higher dimensional R^H for any two points in R^D. The kernel function is defined as follows:

$$K(x_i, x_j) = <\phi(x_i).\phi(x_j)> \qquad (5)$$

PCA is reformulated by using an n× n kernel matrix, k that computes the "dot product" in the high dimensional space using equation (5). The type of kernel used defines the nature of the mapping. If a linear kernel is used, then Kernel PCA is similar to PCA. Some common choices of kernel functions are linear kernel, Gaussian kernel and polynomial kernel.

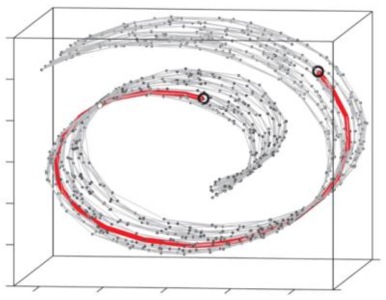

Figure 2: A Non-linear manifold. The red trace along the manifold shows the geodesic distance.

The entries of the kernel matrix are defined as

$$k_{ij} = K(x_i, x_j) \qquad (6)$$

The mean centered data points in R^H can be obtained by modifying the kernel matrix as

$$k_{ij} = k_{ij} - \frac{1}{n}\sum_l k_{il} - \frac{1}{n}\sum_l k_{jl} + \frac{1}{n^2}\sum_{ij} k_{ij} \qquad (7)$$

So the eigen problem that is solved is the following

$$kx = \lambda x \qquad (8)$$

The eigen vectors(v_i) of the covariance matrix C is related to the eigen vectors (u_i) of the kernel matrix(k) through

$$v_i = \frac{1}{\sqrt{\lambda_i}} X u_i \qquad (9)$$

The low dimensional reconstruction y_i is obtained by projecting x_i on to the eigen vectors $v_{i.}(p$ denotes the $p^{th} component)$

$$y_i^p = v_i^p * \phi(x) = \sum_i^N \alpha_i^p * K(x_i, x_j) \qquad (10)$$

A notable weakness of Kernel PCA is the size of Kernel matrix that depends on the number of data points in X

2.4. Locally Linear Embedding

Locally Linear embedding [3] like Isomap is a graph based technique for dimensionality reduction. The intuition behind LLE is that non-linear manifolds can be split up into several regions and each of these regions can be considered as approximately linear. A linear dimensionality reduction can be done on these regions to map the data to a lower dimensional space. In effect, LLE tries to fit a hyper plane through each of its local sub regions. As a result of the linearity assumptions the reconstruction weights for each data point x_i do not change even when they are projected to the lower dimension [16]. In other words, the reconstructions weights W_i that reconstruct the weight w_i can also be used to reconstruct y_i. The cost function that is to minimized is given by

$$f(Y) = \sum_i (y_i - \sum_j w_{i_j} y_{ij})\text{^}2 \tag{11}$$

The data points y_i that result in minimizing the above equation is given by finding the smallest eigen vectors of $(I - W)^T(I - W)$.

2.5. AutoEncoders

Autoencoders are feed forward neural networks that consist of an input layer, at least one hidden layer and an output layer. Typically with neural networks, the input is a set of input –output mappings $\{x_i, y_i\}$ and the output is a function $h_{w,b}$ that maps the input x_i to y_i. If it is a classification problem $y_i's$ correspond to class labels. Auto encoders learn a function $h_{w,b}$ where the target value y_i is set to be the input itself. In order to use autoencoders as a dimensionality reduction technique, the number of neurons in the hidden layer should be set to m , the target dimensionality to which the input must be reduced. In effect, by restricting the number of neurons in the hidden layer, the autoencoder is forced to learn a mapping that produces a compressed representation of the input (x_i). To ensure that the compressed representation is relevant to the original data, the reduced data representation is made to reconstruct the input. The activation function used in the neurons defines the mapping to the lower dimensions. If a linear activation function is used, then autoencoders work just like PCA. Typically sigmoid / Gaussian activation functions are used.

3. Experimental Setup

3.1. Handwritten Digit Dataset

The real world pattern recognition task considered is classifying handwritten digits and the MNIST [14] (Modified National Institute of Standard and Technology) data set is used. This classic data set consists of a database handwritten digits that has a training set of 60,000 examples and test set of 10,00 examples. For computational reasons that will be explained in section 4, only 5000 were used for training and 1000 for testing.

3.2. Out of Sample Extensions

Out of-sample extension is the ability of a dimensional reduction algorithm to project unseen data points onto an existing low dimensional space. For the methods that were discussed in above sections, PCA, Kernel PCA and autoencoders have straight forward ways to perform out-of sampling. For PCA a new data point can be projected into the low dimensional space by just multiplying with the eigen vector matrix. Kernel PCA also has a similar technique but requires extra kernel computation. In Autoencoders, mapping information is stored in the weights connecting the neurons. Finding low dimensional representation for autoencoders just requires multiplying the weight matrices at different layers.

Isomap and LLE, on the other hand, don't have straight forward methods for out-of-sample extensions This mapping however, can be learnt from the data points that already been embedded in the low dimensional space. It has been shown in [15] that neural networks with sigmoid units are a Universal approximator of functions. Therefore a neural network can be used to learn a function $M : R^D \rightarrow R^m$. The input to the neural network will be $\{x_i, y_i\}$ and the output will be an approximation of the function M. The neural network that was used consisted of a single hidden layer with 20 sigmoid units.

3.3. Parameter Setting

The parameter settings for the different dimensionality reduction techniques are as follows: (a) Kernel PCA used a polynomial kernel with $\boldsymbol{K(x_i, x_j) = (< x, y > +1)^2}$ (b) Locally Linear Embedding has its neighborhood defined choosing k=12 nearest neighbors (c) Isomap has its neighborhood defined by choosing k=10 nearest neighbors (d) Autoencoders use 3 hidden layers. Using these techniques, dimensionality of the data was reduced to m=2, 10, 20 and 30. A k-nearest neighbor classifier with k=3 was used to classify the data points after dimensionality reduction. The classifier's performance was also evaluated after adding Gaussian and Salt-and -Pepper noise.

4. Results and Discussions

Figure [4] presents the classification errors of the k-nearest neighbor classifier that was trained on lower dimensional data produced by the different Dimensionality reduction techniques. It can be seen that at lower dimensions, Locally Linear Embedding and Isomap perform better than all other techniques. However, as the dimensionality increases, PCA tends to outperform other methods. When m=2, the top two principal components

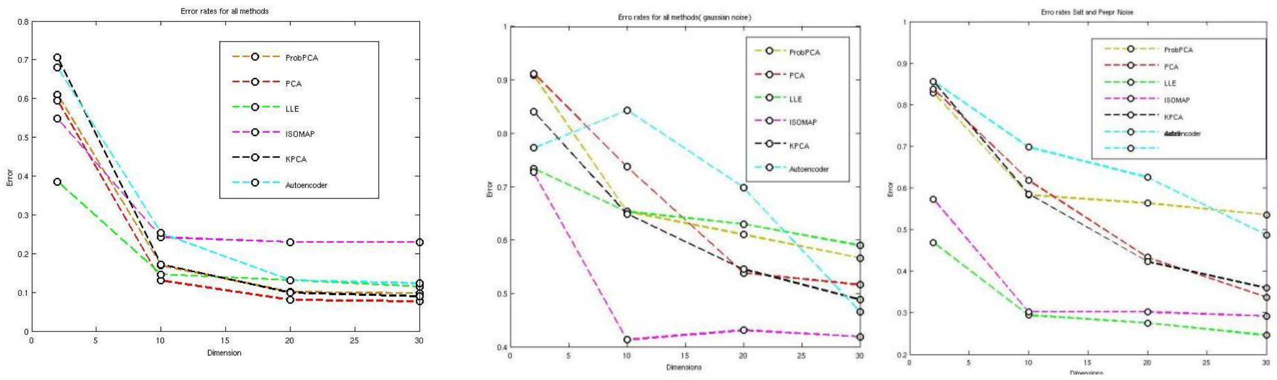

Figure 4: Generalization errors of k-nearest neighbors classifier that uses data preprocessed by dimensionality reduction techniques. Figure 4(a) shows the generalization accuracy when no noise is introduced in the test data set. Figures 4(b) and 4(c) correspond to test data having Gaussian noise and Salt-and-Pepper noise respectively.

account for only 17% of the variance of the data. However as m is increased to 10, 20, 30 it is able to account for 49.35%, 64.98%, and 73.63% of the variance respectively. As a result it is able to minimize the overlap and k-nn performs better. Isomap and LLE, on the other hand, do not show a significant increase in accuracy as the dimensionality is increased. This could be attributed to the error in function approximation of the neural network. As m increases, more number of free parameters have to be learnt in the neural network. This increase in the degrees of freedom, could lead to over fitting of data as m increases. The autoencoders, which optimize a non-convex function, are susceptible to local minima that could impair their performance. Kernel PCA also does better as the dimensionality is increased

When noise is added to the data, it can be seen from Figures 4(b) &(c) that non-linear techniques Isomap and LLE show the best performance and there is a significant decrease in the performance of PCA. When noise is added to the digits, pixels are turned on /off at random and as a result, they displace from the original position in the digit space. This introduces some degree of non-linearity in the data and as a result PCA, does not perform well in preserving the geometry of the data in low dimensions. Isomap, LLE and autoencoders are able to do better than PCA when noise is included. It was also observed that using Kernel PCA gives relatively low error rates across all experiments. It is very robust and its performance is always close to the best method. The results of Kernel PCA could help in analyzing the nature of the data set as Kernel PCA performs poorly when data extremely is complex and nonlinear [2].

Table [1] shows the computational complexities of the various methods. Isomap, LLE, Kernel PCA and PCA solve the eigen problem, which takes $O(n^3)$ for an n$\times$ n matrix when the "power method" is used. Except for PCA, all the other techniques find the eigen vectors of a $n \times n$ matrix. For the data set that was used, n=5000 and D=784. So PCA is computationally more efficient than other techniques. PCA scaled so well that it was possible to use the entire data set for training and testing. This resulted in an error rate of just 2.92% .When compared to all other techniques autoencoders were computationally the most expensive. The time complexity is proportional to the number of inputs n , number of weight vectors w and number of layers i. The data was scaled down from 60,000 training samples to 10,000 training samples because non-linear techniques especially autoencoders and Kernel PCA were computationally too expensive.

The experiments also revealed the nature of the MNIST data set. The performance of Kernel PCA is an indication that the data set is not highly non-linear and not very complex. Also the good performance of k-nn classifier when preprocessed with PCA is an indication that the MNIST digit data set is not highly non-linear. The experiments also affirm the claim that LLE and Isomap are good at data visualization since they give good classification performance at lower dimensions. The classification accuracy is also an indication of how well the geometry of the data is preserved in the lower dimensions [2].

Method	**Time complexity**
PCA	$O(D^3)$
Kernel PCA	$O(n^3)$
Locally Linear Embedding	$O(n^3)$
Isomap	$O(n^3)$
Autoencoders	$O(inw)$

Table1: Computational complexities

5. Future Work

Due to hardware constraints it was not possible to check the performance of Non-linear techniques when the entire data set is considered. In [5] the entire data set was used and preprocessing was done by stacking up multiple layers of auto encoders. Although time consuming, it resulted in very high levels of accuracy. So it would be interesting to see how other non-linear techniques like Locally Linear Embedding and Isomap perform when the entire data set in considered. The out-of-sample extensions were

Noise	No Noise				Gaussian Noise				Salt and Pepper Noise			
Dimensions	2d	10d	20d	30d	2d	10d	20d	30d	2d	10d	20d	30d
PCA	0.59	***0.13***	***0.08***	***0.07***	0.91	0.73	0.53	0.51	0.83	0.61	0.43	0.33
KPCA	0.70	0.17	0.09	0.09	0.84	0.64	0.54	0.48	0.85	0.58	0.42	0.35
Isomap	0.54	0.24	0.23	0.23	***0.72***	***0.41***	***0.43***	***0.41***	0.57	0.30	0.30	0.29
LLE	***0.38***	0.14	0.13	0.11	0.73	0.65	0.63	0.59	***0.46***	***0.29***	***0.27***	***0.24***
Autoencoders	0.68	0.25	0.13	0.12	0.73	0.84	0.69	0.46	0.85	0.69	0.62	0.48
Prob PCA	0.61	0.16	0.10	0.09	0.9	0.65	0.61	0.56	0.82	0.58	0.56	0.53

Table 2: Classification errors of k-nearest neighbor classifier on the digit classification dataset. Dimensions refers to lower dimensional spaces to which the points were mapped

approximated using neural networks for Isomap and LLE. Neural networks are however slow and it will be useful to find faster techniques that can perform out-of-sample extensions. It will be also be useful to modify the existing non-linear techniques [7] that can make use of labeled information.

6. Conclusions

The results of our experiments indicate that non-linear techniques despite their flexibility do not outperform PCA for this dataset. However, when noise is incorporated to the test data, the degree of non-linearity in the dataset becomes higher and Isomap and LLE do a good job. So, when it is known that data is non-linear and low dimensional projections are required, Isomap and LLE maybe better options when compared to Principal Component Analysis. When computational aspects were considered, PCA is computationally less demanding than the non-linear techniques. As a result, they can easily scale to work with large datasets and have the advantage that they can use more data. So the non-linear techniques still have some weaknesses that have to be overcome before they can be considered on par with PCA for dimensionality reduction on this digit classification dataset.

References

[1] Ghodsi, Ali. "Dimensionality reduction a short tutorial."

[2] Van der Maaten, L. J. P., E. O. Postma, and H. J. Van Den Herik. "Dimensionality reduction: A comparative review." *Journal of Machine Learning Research* 10 (2009): 1-41.

[3] Roweis, Sam T., and Lawrence K. Saul. "Nonlinear dimensionality reduction by locally linear embedding." *Science* 290.5500 (2000): 2323-2326.

[4] Tenenbaum, Joshua B., Vin De Silva, and John C. Langford. "A global geometric framework for nonlinear dimensionality reduction." *Science* 290.5500 (2000): 2319-2323.

[5] Vincent, Pascal, et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *The Journal of Machine Learning Research* 11 (2010): 3371-3408.

[6] Schölkopf, Bernhard, Alexander Smola, and Klaus-Robert Müller. "Kernel principal component analysis." *Artificial Neural Networks—ICANN'97* (1997): 583-588.

[7] Vlachos, Michail, et al. "Non-linear dimensionality reduction techniques for classification and visualization." *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002.

[8] Keinosuke Fukunaga. Introduction to statistical pattern recognition. Academic Pr, 1990.

[9] Tipping, Michael E., and Christopher M. Bishop. "Probabilistic principal component analysis." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3 (1999): 611-622.

[10] Weinberger, Kilian Q., and Lawrence K. Saul. "An introduction to nonlinear dimensionality reduction by maximum variance unfolding." *Proceedings of the National Conference on Artificial Intelligence*. Vol. 21. No. 2. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[11] Turk, Matthew, and Alex Pentland. "Eigenfaces for recognition." *Journal of cognitive neuroscience* 3.1 (1991): 71-86.

[12] Kruskal, Joseph B. "Nonmetric multidimensional scaling: a numerical method." *Psychometrika* 29.2 (1964): 115-129.

[13] Soille, Pierre. "Generalized geodesic distances applied to interpolation and shape description." (1994).

[14] LeCun, Yann, and Corinna Cortes. "MNIST handwritten digit database." *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist* (1998).

[15] Cybenko, George. "Approximation by superpositions of a sigmoidal function." *Mathematics of Control, Signals, and Systems (MCSS)* 2.4 (1989): 303-314.

[16] CMU Data Mining Lecture Notes :Non-Linear Dimensionality Reduction.