ISSN: 2321-7782 (Online)
Volume 3, Issue 5, May 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Mathematical Analysis of PCA, MDS and ISOMAP Techniques in Dimension Reduction

Dipa Patra[1]
School of Computer Engineering
KIIT University, India

Jnyanaranjan Dash[2]
School of Computer Engineering
KIIT University, India

Chittaranjan Pradhan[3]
School of Computer Engineering
KIIT University, India

R. N. Ramakant Parida[4]
School of Computer Engineering
KIIT University, India

Abstract: Dimensionality reduction techniques are the methods which can effectively reduce the number of dimensions of the dataset that gives best description to object and also efficient for so many data processing tasks. Many more research on dimensionality reduction has carried out in the last many decades. Also day by day its demand is growing due to higher-dimensional applications and proper visualization of higher dimensional data into lower dimensional space for better understanding is an important concern. Here, the field of dimensionality reduction (DR) is divided into two types: linear DR techniques such as PCA and non-linear DR techniques such as MDS and ISOMAP. Both try to reduce or deduct the dimensions of a dataset to efficiently handle higher dimensional data. This paper mathematically analyzes PCA, MDS and ISOMAP techniques.

Keywords: Dimensionality reduction; linear DR; nonlinear DR; PCA; MDS; ISOMAP.

I. INTRODUCTION

A dimension refers to a measurement of a certain aspect of an object. Dimensionality reduction is the study of methods for reducing the number of dimensions describing the object. Mathematically, the problem we investigate can be stated as follows: given the p-dimensional random variable $x = (x_1; \ldots ; x_p)^T$, find a lower dimensional representation of it, $s = (s_1; \ldots ; s_k)^T$,(where T is the transpose) with $k <= p$, that captures the content in the original data, according to some criterion[1, 2].

Dimensionality reduction methods can be categorized as linear dimensionality reduction techniques and nonlinear dimensionality reduction techniques. Data which has linear relationship among variables is called as linear data and others are called as nonlinear data. So the techniques deals with those linear data are called linear dimensionality reduction techniques such as Principal Component Analysis (PCA). And others are the nonlinear dimensionality reduction techniques such as multi-dimensional scaling (MDS), Isometric feature map (ISOMAP) etc.

II. PRINCIPAL COMPONENT ANALYSIS

Principal component analysis (PCA) is the best, in the mean-square error sense, linear dimension reduction technique. Based on the covariance matrix of the variables, it is a second-order method. It is also known as the singular value decomposition (SVD), the Karhunen-Loeve transform, the Hotelling transform, and the empirical orthogonal function (EOF) method [1, 3]. The idea was conceived by Pearson (1901) and independently developed by Hotelling (1933).

a) Steps in PCA

Let X is the data set with dimension M. We have to convert it to a dataset Y of smaller number of dimensions L. Where L<M. To obtain Y for a given set of X, the steps are:

1. Write N observations $X_1,\ldots,X_N$ as a column vectors into a single matrix X of dimensions M x N.
2. Calculate the empirical mean. For PCA to work properly, subtract the empirical mean vector from each column of the data Matrix X. This produces a data set whose mean is zero, i.e.

$$DataAdjust = X - mean \tag{1}$$

3. Calculate the M x M covariance matrix C.
4. Find the eigenvalues and eigenvectors of the covariance matrix C. Let D is the matrix of eigenvalues and V is the matrix of eigenvectors. The eigenvalues and the eigenvectors are ordered and paired. The mth eigenvalue corresponds to the mth eigenvector.
5. Rearrange the eigenvalues, highest to lowest. If originally there are m-dimensions in the dataset then there will be m-eigenvectors and eigenvalues, and then choose only the first one eigenvectors, then the final data set has only one dimensions. So form a *feature vector*, which is
6. constructed by taking the eigenvectors, and forming a matrix with these eigenvectors in the columns. Then choose one number of eigenvectors.

$$FeatureVector = (eigenvec_1, eigenvec_2, \ldots, eigenvec_N) \tag{2}$$

7. Now derive the final new set of data Y as follows:

$$FinalData\ (Y) = RowFeatureVector * RowDataAdjust \tag{3}$$

Where RowFeatureVector is the matrix with the eigenvectors in the columns transposed so that the eigenvectors are now in the rows, with the most significant eigenvector at the top, and RowDataAdjust is the mean-adjusted data transposed, i.e. the data items are in each column, with each row holding a separate dimension [4].

b) *Analysis of PCA*

Let the following data set X:

$$X = \left\{\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}2\\3\end{pmatrix},\begin{pmatrix}4\\1\end{pmatrix},\begin{pmatrix}5\\4\end{pmatrix},\begin{pmatrix}4\\5\end{pmatrix},\begin{pmatrix}6\\6\end{pmatrix}\right\}$$

Here, the number of data points are N=6 and the number of dimensions is M=2. Figure 1 shows the graphical representation of the dataset X.

The mean of these data points is (3.6667 3.3333). So the mean subtracted data are:

$$DataAdjust = \begin{pmatrix} -2.6667 & -2.3333 \\ -1.6667 & -0.3333 \\ 0.3333 & -2.3333 \\ 1.3333 & 0.6667 \\ 0.3333 & 1.6667 \\ 2.3333 & 2.6667 \end{pmatrix}$$

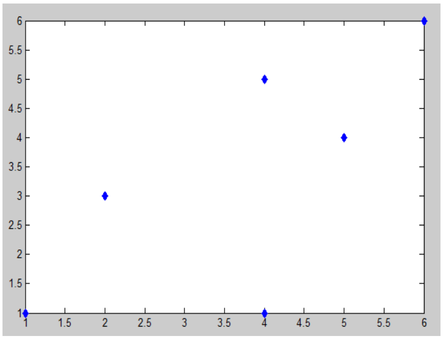

Figure.1 Graphical representation of the data set

Next calculate the covariance matrix for above data:

$$cov(DataAdjust) = \begin{matrix} 3.4667 & 2.7333 \\ 2.7333 & 4.2667 \end{matrix}$$

Next, calculate the eigenvectors and eigenvalues of the covariance matrix. As it is two dimensional data, it will give two eigen values and two pair of eigenvectors:

$$Eigen\ values = 6.6291\ and\ 1.1042$$
$$Eigen\ vectors = \begin{pmatrix} 0.6539 \\ 0.7566 \end{pmatrix}\ and\ \begin{pmatrix} 0.7566 \\ -0.6539 \end{pmatrix}$$

Notice that one eigenvalue is much larger than the other which shows the difference in variation of two dimensions, i.e.larger value shows more variance of data in that direction. So the feature vector will be:

$$Feature\ vector = \left(\begin{array}{c|c} 0.6539 & 0.7566 \\ 0.7566 & -0.6539 \end{array}\right)$$

So, the final data will be:

$$Final\ data = \begin{matrix} -3.5091 & -0.4917 \\ -1.3420 & -1.0430 \\ -1.5474 & 1.7780 \\ 1.3763 & 0.5728 \\ 1.4789 & -0.8377 \\ 3.5433 & 0.0216 \end{matrix}$$

Figure 2 shows the 2D representation of the dataset X after applying PCA.

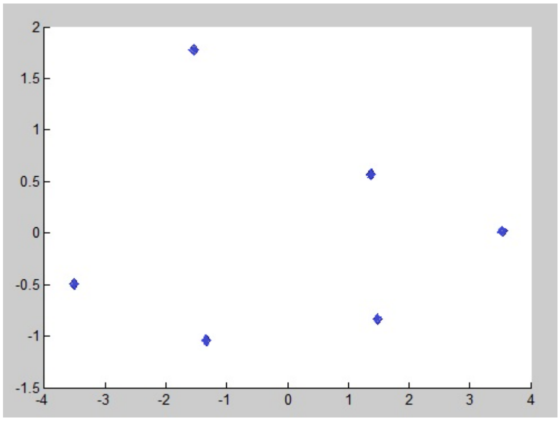

Figure2. Dataset X after applying PCA

But the major axis (for 1D representation) can be defined by the first largest eigenvector and the new (1D) coordinate system will be:

$$Final\ data = \begin{matrix} -3.5091 \\ -1.3420 \\ -1.5474 \\ 1.3763 \\ 1.4789 \\ 3.5433 \end{matrix}$$

III. MULTI-DIMENSIONAL SCALING

Multi-Dimensional Scaling (MDS) is an information visualization technique which has become more and more popular for multivariate data analysis. The raw data entering into an MDS analysis are typically a measure of the global similarity or dissimilarity. The primary outcome of an MDS analysis is a spatial configuration, in which the objects are represented as points [5]. For better understanding of MDS the basic mechanism is the classical MDS is might helpful. Here, we have considered the classical MDS.

a) Steps in MDS

1. Input the proximity matrix 'p'.
2. Set up the matrix of squared proximities $P^2 = [p^2]$.
3. Apply the double centering: $B=-1/2(JP^2J)$ using the matrix $J=I-n^{-1}11^T$, where n is the number of objects, I is the nxn identity matrix and 1 is the column vector of n ones & T is the transpose.
4. Extract the m largest positive eigen values $\lambda_1....\lambda_2$ of B and the corresponding m eigenvectors $e_1.....e_m$. Here m will be final number of reduced dimensions.
5. A m-dimensional spatial configuration of the n objects is derived from the coordinate matrix $X = E_m \Lambda_m^{1/2}$ whereE_m is the matrix of m eigenvectors and Λ_m is the diagonal matrix of m eigenvalues of B.

b) Analysis of MDS

Let consider the same data set X shown in Figure 1. Basically classical MDS tries to find the spatial location from a set of proximities. First start with proximities among the data points which are simply the Euclidean distances between them. As Classical MDS deals with the higher dimensional data, so here consider the space for data points are of N-dimensional, where N is the number of the data points or objects. In this example the number of data points N=6.

So the proximity matrix 'p' will be:

$$p= \begin{matrix} 0 & 2.2361 & 3.0000 & 5.0000 & 5.0000 & 7.0711 \\ 2.2361 & 0 & 2.8284 & 3.1623 & 2.8284 & 5.0000 \\ 3.0000 & 2.8284 & 0 & 3.1623 & 4.0000 & 5.3852 \\ 5.0000 & 3.1623 & 3.1623 & 0 & 1.4142 & 2.2361 \\ 5.0000 & 2.8284 & 4.0000 & 1.4142 & 0 & 2.2361 \\ 7.0711 & 5.0000 & 5.3852 & 2.2361 & 2.2361 & 0 \end{matrix}$$

Then, set up the square proximities P^2:

$$P^2= \begin{matrix} 0.0000 & 5.0000 & 9.0000 & 25.000 & 25.000 & 50.000 \\ 5.0000 & 0.0000 & 8.0000 & 10.000 & 8.0000 & 25.000 \\ 9.0000 & 8.0000 & 0.0000 & 10.000 & 16.000 & 29.000 \\ 25.000 & 10.000 & 10.000 & 0.0000 & 2.0000 & 5.0000 \\ 25.000 & 8.0000 & 16.000 & 2.0000 & 0.0000 & 5.0000 \\ 50.000 & 25.000 & 29.000 & 5.0000 & 5.0000 & 0.0000 \end{matrix}$$

Here the number of objects n=6, so the centering matrix J will be:

$$J= \begin{matrix} 0.7500 & -0.2500 & -0.2500 & -0.2500 & -0.2500 & -0.2500 \\ -0.2500 & 0.7500 & -0.2500 & -0.2500 & -0.2500 & -0.2500 \\ -0.2500 & -0.2500 & 0.7500 & -0.2500 & -0.2500 & -0.2500 \\ -0.2500 & -0.2500 & -0.2500 & 0.7500 & -0.2500 & -0.2500 \\ -0.2500 & -0.2500 & -0.2500 & -0.2500 & 0.7500 & -0.2500 \\ -0.2500 & -0.2500 & -0.2500 & -0.2500 & -0.2500 & 0.7500 \end{matrix}$$

Then calculate the double centering matrix B. Double centering is subtracting the row and the column means of the matrix from its element. Now B calculated this way error free scalar products to the origin. So the calculated double centering matrix B is:

$$B = \begin{matrix} 14.0000 & 4.2500 & 4.2500 & -6.2500 & -5.7500 & -11.0000 \\ 4.2500 & -0.5000 & -2.5000 & -6.0000 & -4.5000 & -5.7500 \\ 4.2500 & -2.5000 & 3.5000 & -4.0000 & -6.5000 & -5.7500 \\ -6.2500 & -6.0000 & -4.0000 & -1.5000 & -2.0000 & 3.7500 \\ -5.7500 & -4.5000 & -6.5000 & -2.0000 & -0.5000 & 4.2500 \\ -11.0000 & -5.7500 & -5.7500 & 3.7500 & 4.2500 & 14.0000 \end{matrix}$$

For deriving a 2-D representation of the data points, choose first two largest eigen values and its corresponding eigen vectors of matrix B:

$$Eigen\ values = 33.1516\ and\ 13.0584$$

$$Eigen\ vectors = \begin{matrix} -0.6157 & -0.1236 \\ -0.2372 & -0.5185 \\ -0.2734 & -0.4068 \\ 0.2350 & -0.5341 \\ 0.2527 & -0.5064 \\ 0.6090 & -0.0933 \end{matrix}$$

Now the 2D final data X will be:

$$X = \begin{matrix} -3.5448 & -0.4466 \\ -1.3657 & -1.8737 \\ -1.5743 & -1.4699 \\ 1.3531 & -1.9301 \\ 1.4548 & -1.8299 \\ 3.5065 & -0.3372 \end{matrix}$$

The two dimensional representation of the data points by MDS is shown in the Figure3.

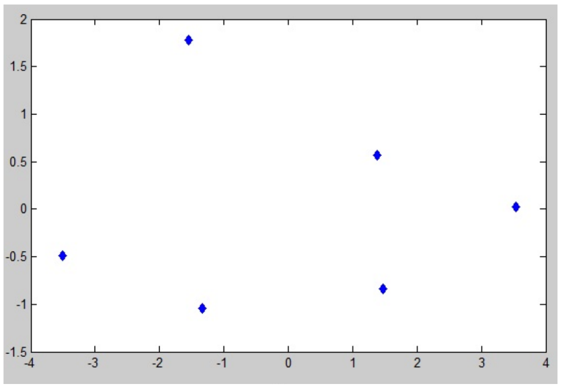

Figure3 . Representation of data points by MDS

IV. ISOMETRIC FEATURE MAP

Isometric Feature Map (ISOMAP) is originally developed by J. B. Tenenbaum, V. de Silva, J. C. Langford (December, 2000) [6, 7]. Isomap algorithm tries to recover the original embedding of the hidden data. This algorithm is the non-linear generalization of the classical MDS with the main idea is to perform the classical MDS in the geodesic distances of the data points of the manifold not in the input space of Euclidean distances of the data points. Isomap is better than other techniques and gives a global optimal solution because it uses geodesic distances as its input. Neighborhood size (number of nearest neighbor's k or neighborhood radius ε) is a key parameter of Isomap algorithm, which has to be specified manually.

a) Steps in ISOMAP

The algorithm given by Tenenbaum [6] proceeds with three steps:

1. *Construction of neighborhood graph*:-Two points i and j are connected if they are closer than **ε** or i is one of the k nearest neighbors of j. This process is repeated for all points and a graph called neighborhood graph is constructed.

Edge lengths are set equal to the distance between the two end points. If i and j are two points then d_x (i, j) represents the distance between i and j and set as the length of the edge connecting points i and j.

2. *Shortest path computation:*-d_G(i, j) is initialized as d_x(i, j) if i and j are connected by an edge otherwise d_G(i, j) is set to ∞. Then shortest path algorithm is applied to find shortest path distances between all pairs of points in G.

3. *Construction of d-dimensional embedding:*- Classical MDS [5] is used for constructing lower dimensional embedding.

b) Analysis of ISOMAP

The Consider the example of the Figure 1, with setting up the number of neighbors is 2. Here also we have to consider the original data space is the N-dimensional space, where 'N' is the number of data points. Figure 4 shows the neighbors of each data points.

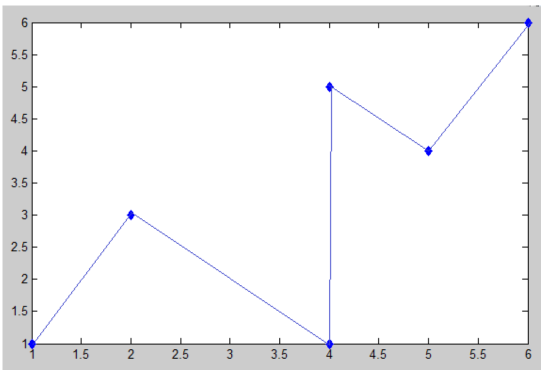

Figure4. Data points with the nearest neighbors

So the corresponding distances among the data points is given by the distance matrix D. This matrix is filled with the actual Euclidean distances of the neighbors only and the non-neighbor points with infinite (Inf) distances.

$$D=\begin{matrix} 0.0000 & 2.2361 & 5.0645 & \mathrm{Inf} & \mathrm{Inf} & \mathrm{Inf} \\ 2.2361 & 0.0000 & 2.8284 & \mathrm{Inf} & \mathrm{Inf} & \mathrm{Inf} \\ \mathrm{Inf} & 2.8284 & 0.0000 & \mathrm{Inf} & 4.0000 & \mathrm{Inf} \\ \mathrm{Inf} & \mathrm{Inf} & \mathrm{Inf} & 0.0000 & 1.4142 & 2.2361 \\ \mathrm{Inf} & \mathrm{Inf} & 4.0000 & 1.4142 & 0.0000 & \mathrm{Inf} \\ \mathrm{Inf} & \mathrm{Inf} & \mathrm{Inf} & 2.2361 & 3.6503 & 0.0000 \end{matrix}$$

Now, compute the shortest paths among all pair of data points to approximate the geodesic distances. So, to achieve that use any of the shortest path algorithms like Dijkstra’s shortest path algorithm, floyd-warshall's algorithm etc. These shortest paths are given by the matrix D_short:

$$D\_short=\begin{matrix} 0 & 2.2361 & 5.0645 & 10.4787 & 9.0645 & 12.7148 \\ 2.2361 & 0 & 2.8284 & 8.2426 & 6.8284 & 10.4787 \\ 5.0645 & 2.8284 & 0 & 5.4142 & 4.0000 & 7.6503 \\ 10.4787 & 8.2426 & 5.4142 & 0 & 1.4142 & 2.2361 \\ 9.0645 & 6.8284 & 4.0000 & 1.4142 & 0 & 3.6503 \\ 12.7148 & 10.4787 & 7.6503 & 2.2361 & 3.6503 & 0 \end{matrix}$$

Now on these geodesic distances among the points the Classical MDS will be applied to find the ISOMAP embedding. So the final output (in 2D)will be:

$$Final\ data=\begin{matrix} -6.7364 & -0.6943 \\ -4.4675 & -2.6463 \\ -1.6177 & -3.9280 \\ 3.7745 & -2.6819 \\ 2.3740 & -3.4763 \\ 5.9775 & -0.7492 \end{matrix}$$

Figure 5 shows the final result after applying the ISOMAP on the data set according Figure 4.

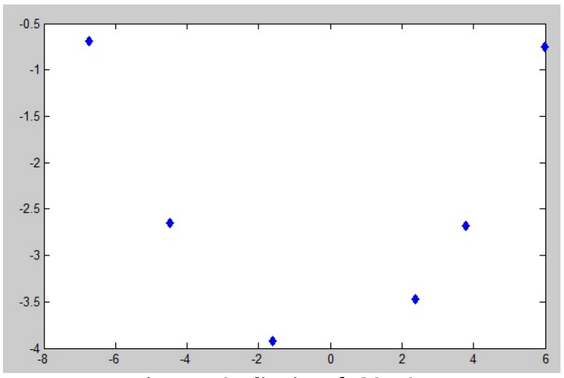

Figure 5. Application of ISOMAP

V. CONCLUSION

PCA works better on linear data, but it cannot handle the complex non-linear data. So nonlinear DR techniques like MDS, ISOMAP etc. are developed. Also MDS can't handle the curved or twisted nonlinear dataset because MDS uses the Euclidean distances among the data points which does not preserves the actual distances for a twisted dataset. So ISOMAP is designed which gives the better result because it uses the geodesic distances of the data points which cause to remains the actual distances.

References

1 I. K. Fodor, "A survey of Dimension Reduction Techniques", U. S. Department of Commerce, National Technical Information Service, Springfield, VA, 2002.

2 C. Pradhan, S. Mishra, " Optimized ISOMAP algorithm using Similarity Matrix", IEEE International Conference on Electronics Computer Technology, Vol. 5, 2011, pp. 212-215.

3 Principal component analysis, http://en.wikipedia.org/wiki/Principal_component_analysis.

4 L. I. Smith, "A Tutorial on Principal Components Analysis", Technical Report, Cornell University, USA, 2002.

5 F. Wickelmaier, "An Introduction to MDS", Reports from the Sound Quality Research Unit, Aalborg University, Denmark, No. 7, 2003.

6 J. B. Tenenbaum, V. de Silva, J. C. Langford, " A Global Geometric Framework for Non-linear Dimensionality Reduction", Science, Vol. 290, No. 5500, 2000, pp. 2319-2323.

7 L. Yang, " Sammon's Nonlinear Mapping Using Geodesic Distances, IEEE International Conference on Pattern Recognition, Vol. 2, 2004, pp. 303-306.