ISSN:0975-9646

A Review of Various Linear and Non Linear Dimensionality Reduction Techniques

Sumithra V.S
Department of Computer Science and Engineering
SCT College of Engineering
Trivandrum,India

Subu Surendran
Associate Prof., Dept. Computer Science & Engineering
SCT College of Engineering
Trivandrum,India

***Abstract*— Data dimensionality refers to the number of variables that are measured on each observation. Recent trends in technology and applications result in the generation of huge volume of high dimensional data. The analysis of these data is inevitable for various research and production activities. Data analysis focuses on understanding, manipulating and interpreting large scale data. Relevant information is hidden in this huge volume dataset which needs to be extracted for analysis. Several methods have been developed in the field of data mining for automated data processing. Owing to the huge dimension of data these methods fail to meet the requirement efficiently. Dimensionality reduction offers an optimal solution to this problem by reducing the data dimension. It transforms data into a meaningful and reduced dimension space with minimal information loss. This reduces the computational cost involved in data analysis and founds effective in data compression, visualization and big data analysis. Dimension reduction is applicable in many real world domains such as regression analysis, cluster analysis, computer vision, image processing, text categorization and so on. There are various classes of dimension reduction techniques such as supervised, unsupervised, linear, nonlinear etc. The paper presents a concise review of some relevant linear and nonlinear dimensionality reduction techniques.**

***Keywords*— Dimension reduction, PCA, Fastmap, LTSA, LaplacianEigenmap**

I. INTRODUCTION

The computational advancements in various domains give rise to new technologies and applications that involves huge amount of data. Application domains like bio-informatics, computer science, astronomy, statistics, remote sensing, social network etc. generate high volume of heterogeneous data. These data are in turn called as 'Big Data' which is difficult to process using traditional data processing methods. Only a certain amount of data generated can encapsulate useful information which may also contain noise, correlated features etc. Hence it is essential to discover the hidden portions of data that are significant. The increase in data dimensionality leads to increase in demand for processing and storage requirements. This problem is called the curse of dimensionality. For effective data processing amid of these constrains it is essential to have control on the number of useful variables. The field of data mining faces tremendous growth in order to meet such excessive computation requirements. This lead to the development of some novel research areas viz. machine learning, computational intelligence etc. that helps in automated data processing, thereby yielding relevant observations in various domains. Dimensionality reduction is a machine learning technique that reduces the data dimensionality with minimal information loss before proceeding with the analysis. Dimensionality reduction transforms high-dimensional data into a meaningful reduced dimension space. The reduced representation must have correspondence to the intrinsic data dimensionality, i.e. the minimum number of parameters that can define the observable properties of data. The primary focus of dimensionality reduction is redundancy reduction and intrinsic structure discovery. It is also applied for feature extraction, data visualization, computation and machine learning purposes. Dimensionality reduction is normally considered as a data pre-processing step. The main challenge here is that the transformation should be done with minimal information loss and also need to preserve the structure of the data. Various techniques have been proposed in this regard which either transforms the existing features into a new reduced set of features or selects a subset of the existing features. Dimensionality reduction techniques can be widely classified as linear and nonlinear techniques. Linear dimensionality reduction transforms the data to a low dimension space as a linear combination of the original variables. This is applicable when the data lies in a linear subspace and here the original variables are replaced by a smaller set of underlying variables. Nonlinear dimensionality reduction is applied when the original high dimensional data contains nonlinear relationships. Here the lower dimensional representation of the data is achieved while preserving the original distances between the data points. This paper discusses some of the linear and nonlinear dimensionality reduction techniques which are widely used in a variety of applications. These include Principal Component analysis (PCA), Independent Component Analysis (ICA), Canonical Correlation Analysis (CCA), Singular Value Decomposition (SVD), CUR Matrix Decomposition, Compact Matrix Decomposition (CMD), Non Negative Matrix Factorization (NMF), Linear Discriminant Analysis (LDA), Kernel PCA, Multidimensional Scaling (MDS), Isomap, Locally Linear Embedding (LLE), Laplacian Eigen map, Local Tangent Space Alignment (LTSA) and Fast map. The linear techniques discussed here mostly adopt concepts from linear algebra for performing dimensionality reduction. Linear techniques may fail in effectively handling data that has nonlinear relationships. Real world applications mostly generate nonlinear data, which can be dealt with nonlinear

reduction techniques. In fact dimension reduction reduces the computational complexity of the problem and also increases the accuracy of data analysis.

II. DIMENSION REDUCTION

Dimension reduction is defined as the mapping of data to a lower dimensional space such that uninformative variance in the data is discarded, or such that a subspace in which the data lives is detected [1]. It is mainly used as data analysis, compression and visualization methods. Some major techniques used for linear and nonlinear dimension reduction is discussed in the subsequent sections.

A. Principal Component Analysis (PCA)

PCA is a very established method of linear dimensionality reduction. The purpose of PCA is to derive new variables that are linear combinations of the original variables and are uncorrelated. It finds smaller group of underlying variables that describe the data. PCA projects *n*-dimensional data onto a lower *d*-dimensional subspace in a way that minimizes the sum of squared error, or maximizes the variance, and gives uncorrelated projected distributions [2]. In most cases the underlying structure of data will be sparse. But PCA often generates dense expressions which makes interpretation difficult. It is computed by performing eigendecomposition on data covariance matrix, $\sum$. Laplacian matrix and modularity matrix are best suited to be used as covariance matrix [3]. The covariance matrix $\sum$ can be decomposed as,

$$\sum = U \wedge U^T \quad (1)$$

where $\wedge$ is the diagonal matrix that contains the eigenvalues in diagonals and U is the matrix that contains the corresponding eigenvectors. The eigenvectors obtained resembles the principal axes of maximum variance subspace, eigenvalues represent the variance of projected inputs along principal axes and the number of significant eigenvalues denotes the estimated dimensionality. The size of the covariance matrix is proportional to the data dimensionality which makes eigendecomposition computationally expensive for very high dimensional data. PCA is simple to compute and guaranteed to produce accurate low dimensional representation. But it does not yield high accuracy on uncorrelated data [4].

B. Independent Component Analysis (ICA)

ICA assumes the latent variables to be mutually independent and they are called the independent components of the observed data. It is superficially related to principal component analysis and a more powerful technique. ICA is well suited for separating superimposed signals. It applies linear transformation to decompose the original data into components that are maximally independent from each other. It is not necessary that the independent components are orthogonal to each other. For dimension reduction, ICA finds k components that effectively capture variability of the original data [5]. It decomposes the data matrix A of size $t \times d$ into two matrices such that

$$A_{t\times d} = C_{t\times k}.F_{k\times d} \quad (2)$$

where C is the coefficient matrix and F contains the independent components. ICA guarantees accuracy in case of uncorrelated data but the independent components obtained may not be relevant.

C. Singular Value Decomposition (SVD)

SVD is used to reduce a large matrix into significantly small matrix. It produces the best rank k approximation of the matrix. Let X is an $m \times n$ rank r matrix. Let $\sigma_1 \ldots \sigma_r$ be the eigenvalues of a matrix $\sqrt{XX^T}$. Then there are orthogonal matrices, $U = (u_1, \ldots u_r)$ and $V = (v_1, \ldots v_r)$, whose column vectors are orthonormal and a diagonal matrix $S = diag(\sigma_1, \sigma_r)$[6]. The decomposition $X = USV^T$ is called singular value decomposition of a matrix X and numbers $\sigma_1 \ldots \sigma_r$ are singular values of matrix X. The columns of V^T defines the new axes, the rows of U represents the coordinates of the objects in the space spanned by the new axes and is the scaling factor indicating the relative importance of each new axis. The SVD of X have at most r non-zero singular numbers, where rank r is the smaller of the two matrix dimensions. From that only k greatest singular values are taken to create a k-reduced singular decomposition of X. SVD produces optimal low rank approximation with minimal reconstruction error. The individual components of the actual data are not interpretable in terms of the resultant matrices. The output of SVD is always dense even if the input provided is sparse.

D. CUR Matrix Decomposition

Modern datasets are often represented by large matrices which provides a natural structure for encoding information. Analysis of such data requires the construction of a compressed matrix representation that is easier to analyze and interpret. Principal components analysis and, the Singular Value Decomposition are fundamental data analysis tools that express a data matrix in terms of a sequence of orthogonal or uncorrelated vectors of decreasing importance. But, being linear combinations of up to all the data points, these vectors are difficult to interpret in terms of the data and processes generating the data.

CUR decompositions are low rank matrix decompositions that are explicitly expressed in terms of a small number of actual columns and/or actual rows of the data matrix. An $m \times n$ matrix A, is decomposed as a product of three matrices, C, U, and R, where C consists of a small number of actual columns of A, R consists of a small number of actual rows of A, and U is a small carefully constructed matrix that guarantees that the product CUR is close to A. The extent to which $A \approx CUR$ can be used in place of A or A_k (best rank-k approximation of the data matrix A) in data analysis tasks, depends on the choice of C and R, as well as on the construction of U [7].

Since they are constructed from actual data elements, CUR decompositions are interpretable by practitioners of the field from which the data are drawn. The chosen columns and rows are those that exhibit high statistical leverage. By selecting columns and rows in this manner, the relative error is improved and also they can be employed for exploratory data analysis. Moreover, the actual data elements are easily interpretable from the reduced data representation. CUR preserves the sparsity property, i.e. it produces a sparse result for a sparse input

data matrix. The problem with CUR is that since the columns and rows are generated based on random sampling, there is a chance for the duplicate entries to be present in the outcome. The existing CUR algorithms require many columns and rows to be chosen, which increases the computation complexity and hence makes it impractical for large scale matrices.

E. Compact Matrix Decomposition (CMD)

The author [8] proposes a novel matrix decomposition technique for large sparse graphs. Several important applications such as research citation network analysis, social network analysis, regulatory networks in genes etc. can be modeled as large sparse graphs. Low rank decompositions, such as SVD and CUR, are powerful techniques for revealing latent variables and associated patterns from high dimensional data. These methods often ignore the sparsity property of the graph, and hence usually incur too high memory and computational cost to be practical. Compact Matrix Decompositions (CMD) can analyze static as well as dynamic graphs and can be used for high speed applications. CMD approximates a matrix A of size $m \times n$ as the product of three matrices $C_s U R_s$, where C_s and R_s contains scaled columns (rows) sampled from A, and U is a small dense matrix which can be computed from Cs and Rs. CMD selects columns and rows from input matrix A as CUR does and the duplicate entries are carefully removed. Thus it reduces both the storage space required as well as the computational effort. It scales up the columns that are sampled multiple times while removing the duplicates. CMD computes sparse low rank approximations and provides equivalent decomposition as CUR, but requires less space and computation time and hence it is more efficient. Extension of CMD, with careful sampling, and fast estimation of the reconstruction error, can be used to spot anomalies. CMD is the best cost effective method with respect to time and space complexity.

F. Non Negative Matrix Factorization (NMF)

NMF produces non negative basis vectors which creates a parts-based representation. Basis vectors contain no negative entries and allow only additive combinations of the vectors to reproduce the original [9]. The perception of the whole becomes a combination of its parts represented by these basis vectors. It works under the assumption that the data and components are all non-negative. The representations produced by NMF are additive vectors, obtained by superimposing components which are efficient for image and text representation. NMF factorizes the data matrix A as $A = W.H$ where W is the $t \times k$ matrix whose columns contain the basis vector and H is the $k \times d$ matrix contains the weights used to approximate the columns in A with the corresponding basis vectors from W [5]. It retains more localized patterns but consumes huge amount of memory for large matrices.

G. Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis performs dimensionality reduction on multi-class data. It is a supervised learning technique which generates a single linear projection that maximizes the separation among classes. The input data is projected to a subspace consisting of the most discriminant directions. The objective of LDA is to perform dimensionality reduction while preserving as much of the class discriminatory information as possible [10]. It can be viewed as a pre-processing step for pattern classification and machine learning applications. LDA applies eigendecomposition on the dataset and the computed eigenvectors are stored in a set of scatter matrices viz. between-class scatter matrix and within-class scatter matrix. The corresponding eigenvalues denote the length or magnitude of the eigenvectors. If all eigenvalues are observed to have similar magnitude, then it can be inferred that the data is projected on a good feature space. In general the eigenvectors associated with the largest eigenvalues are selected, as they convey significant information about the data distribution.

LDA is performed as a 5 step process [11]:

1. Compute the d-dimensional mean vectors, m_i for the different classes from the dataset.
2. Compute the scatter matrices:

   (a) Within Class Scatter Matrix:

   $$S_W = \sum_{i=1}^{c} S_i \quad (3)$$

   where S_i is the scatter matrix for every class.

   (b) Between Class Scatter Matrix:

   $$S_B = \sum_{i=1}^{c} Ni(m - m)(mi - m)^T \quad (4)$$

   where m is the overall mean and N_i is the size of each class.
3. Solve the generalized eigenvalue problem for the matrix

   $S_w^{-1} S_B$

   The eigenvectors and eigenvalues convey information about the distortion of the linear transformation. Eigenvectors represent the direction and eigenvalues denote the magnitude of distortion. Resultant eigenvectors form the new axes of the new feature space.
4. Select the linear discriminants for the new feature space. This is done by sorting the eigenvectors with respect to descending order of eigenvalues and chooses k eigenvectors with largest eigenvalues, thereby construct eigenvector matrix $W_{d \times k.}$
5. Transforming the samples onto the new subspace via the equation $Y = X \times W$, where X is $n \times d$ matrix, i^{th} row representing i^{th} sample and Y is the $n \times k$ transformed matrix.

H. Multidimensional Scaling

If the pairwise distance between pairs of points is provided, MDS preserves the distance by projecting the points to a low dimension space, such that the pairwise distances in the reduced space are kept, maximum close to that of original space. This constructs a configuration of points in a Euclidean space from information about inter-point distances [12]. There are two variations for MDS, both are based on similar principles. The difference lies in the metrics used and calculations performed. The distance matrix representing the distances between pairs of objects acts as input for MDS. The distance matrix representation

in required dimension represents the distance between data points in the reduced dimension d_{ij} , approximately equal to the actual distance according to the distance matrix δ_{ij}. MDS constructed distances are called disparities. The relationship between actual data distances and disparities can be linear (classical/metric) or monotonic (non metric). Classic solution is optimal when the distance matrix represents the Euclidean distance between pairs of points. For performing nonlinear dimensionality reduction, non-metric MDS uses distances that can be interpreted in an ordinal sense. The effectiveness of the method is estimated based on the difference between actual distances and their predicted values. This measure is called stress.

$$Stress = \sqrt{\frac{\sum\left(d_{ij}-\hat{d_{ij}}\right)^2}{\sum d_{ij}^2}} \quad (5)$$

MDS starts by assuming an initial configuration of N objects by setting up t dimensional coordinates for each object. The following steps are performed and repeated until stress is reduced [13].

- The Euclidean distances between each pair of objects are computed, denoted as d_{ij}.
- Perform a linear, polynomial or monotonic regression of d_{ij} on actual distance provided, δ_{ij}. The predicted distances obtained after regression are called disparities, $\hat{d_{ij}}$
- Stress is computed to evaluate the goodness of fit between the predicted and actual distances.
- The coordinates of objects are reassigned so as to reduce the value of stress.

MDS is an easy to implement technique which produces relatively precise solution. Classical MDS is not an accurate distance preserving method; also it cannot achieve nonlinear dimension reduction. Non metric MDS designed for nonlinear dimension reduction may generate solutions that involve local optima.

I. Kernel PCA

Kernel PCA is an extension to PCA for performing nonlinear dimension reduction. While PCA works on the linear input space, Kernel PCA works on the linear feature space transformed from a nonlinear feature space using a kernel function [14]. The kernel function is used to create a kernel matrix which is equivalent to the inner product of the high dimensional data points. Any symmetric positive definite matrix can be regarded as kernel matrix. Kernel PCA achieves dimensionality reduction by performing eigendecomposition on the kernel matrix. It selects the most significant eigenvectors and eigenvalues of the kernel matrix to produce the low dimensional representation of the data objects. The idea here is to map the nonlinear data to a higher dimensional space where it becomes linearly separable. The nonlinear mapping function is called the kernel function [15] and the mapping of a sample $\boldsymbol{x}$ is in the form, $\boldsymbol{x} \rightarrow \phi(\boldsymbol{x})$. The kernel function calculates the dot product of the images of the samples $\boldsymbol{x}$ under ϕ.

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j) \quad (6)$$

The steps involved in Kernel PCA are [16]:

1. Compute the kernel matrix, $K_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$.
2. Center the kernel matrix, $K_c = K - 1_N K - K 1_N + 1_N K 1_N$ where 1_N is a N square matrix for which $(1_N)_{ij} = \frac{1}{N}$; $\forall_{ij} \in [1, ..., N]$.
3. Diagonalize K_c and normalize eigenvectors: $\lambda_k (\alpha^k . \alpha^k) = 1$
4. Extract the k first principal components:

$$\phi(\boldsymbol{x})_{kpc}^k = \sum_{i=1}^N \alpha_i^k (\phi(\boldsymbol{xi}). \phi(\boldsymbol{x}))$$

The main advantage here is that KPCA can select the kernel function used, which can normally be linear, polynomial and gaussian kernel. The principal components can be efficiently computed in high dimensional feature space using kernels.

J. FastMap

Fastmap is used to generate a low dimensional representation of high dimensional data. As with MDS, it takes the distance matrix of N objects as input, and applies the cosine law to compute the low dimensional coordinates of the N objects. Given a high dimensional data X of m dimensions and N objects, it uses a distance function to compute the distance matrix $S_{N\times N}$ [17]. The method assumes that the objects are points in some n-dimensional space that needs to be projected on k mutually orthogonal directions. The projections are computed using the distance matrix. For that it selects two objects O_a and O_b with larger distances as pivot objects and the objects are projected on a line that passes through the pivot objects in n-dimensional space. The projection of objects on that line is computed using the cosine law.

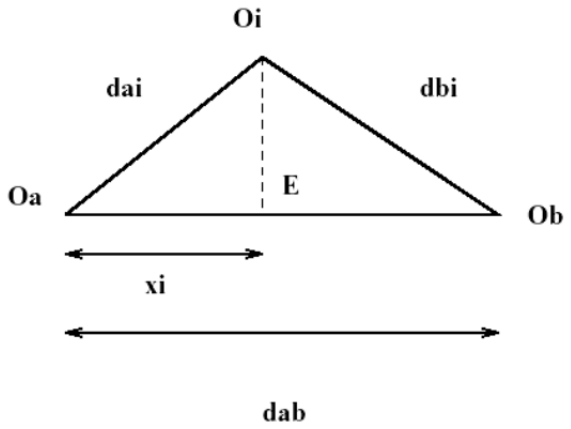

Figure 1: Projection on line $O_a O_b$

The first dimension coordinate of object O_i is computed using the cosine equation:

$$x_i = \frac{d_{a,i}^2 + d_{a,b}^2 - d_{b,i}^2}{da,b} \quad (7)$$

where $d_{a,i}$is the distance between pivot object O_a and object O_i, $d_{b,i}$ is the distance between pivot object O_b and object O_i and $d_{a,b}$ is the distance between pivot objects O_a

and O_b. Once the coordinates of *N* objects are obtained, a reduced distance matrix $S^{'}$ of *N* objects is computed as,

$$d^{'}(O_i, O_j)^2 = d(O_i, O_j)^2 - (x_i\text{-}x_j)^2 \quad (8)$$

where $d^{'}$ is the distance in reduced distance matrix $S'_{N\times N}$ and *d* is the distance in $S_{N\times N}$. From the reduced distance matrix a new set of pivot objects can be selected to compute the coordinates of objects in the second dimension using equation(7). This process is repeated *k* times until the *k* dimensional representation of the data is obtained. Thus Fastmap algorithm determines the coordinates of *N* objects on a new axis, after each of the *k* recursive calls [18]. It is an efficient way of dimensionality reduction which tries to find the axes where the range or spread of data is maximum. Fastmap retains the cluster structure of original data in the
reduced representation. It finds application in efficient retrieval of data and data visualization.

K. Isomap

Isomap is a variant of MDS which uses the concept of geodesic distances between data points rather than Euclidean distances. It is a global method that produces a low dimensional embedding by preserving the pairwise distances between data points. Given *n* data points and associated distance matrix *D*, Isomap generates the reduced representation by performing eigendecomposition of the matrix *D*. Isomap algorithm has three steps:

1. Build the *k* nearest neighbor graph of the manifold based on the distance between pair of points in the input space.

2. Construct the distance matrix by estimating the geodesic distance between all pairs of points using graph shortest path distance algorithm.

3. Find a low dimensional embedding by performing eigenvalue decomposition on the distance matrix.

As PCA and MDS guarantees to recover the true structure of linear manifolds if sufficient data is available, Isomap is guaranteed asymptotically to recover the true dimensionality and geometric structure of a strictly larger class of nonlinear manifolds [19]. It preserves the global structure and transforms the original data to the new coordinate system defined by the most significant eigenvectors.

L. Locally Linear Embedding

Locally Linear Embedding is a nonlinear dimensionality reduction technique that produces low dimensional locality preserving embedding of high dimensional data. It exploits the local symmetries of the linear reconstruction for discovering nonlinear structure in high dimensional data. LLE maps its input to a single global coordinate system of low dimensionality and it do not involve local minima [20]. If the input data is in the form of d dimensional vectors $\vec{X}_i$, LLE starts by computing the neighbors of each data point $\vec{X}_i$. The points along with the neighbors are assumed to lie on a locally linear are of the manifold, characterized by linear coefficients that can reconstruct each data point from its neighbors [20]. The reconstruction errors are measured using the following equation:

$$\epsilon(W) = \sum_i \left|\vec{X}_i - \sum_j W_{ij}\vec{X}_j\right|^2 \quad (9)$$

where W_{ij} is the weight that describe the contribution of the j^{th} data point to the i^{th} reconstruction. The weights that best reconstruct each data point are computed such that, each
data point is only reconstructed from its neighbors. They characterize the intrinsic geometric structure of the dataset. The data vector $\vec{X}_i$ is mapped to low dimensional space by computing d dimensional coordinates $\vec{Y}_i$ that minimizes the cost function,

$$\Phi(Y) = \sum_i \left|\vec{Y}_i - \sum_j W_{ij}Y_j\right|^2 \quad (10)$$

The low dimensional vectors $\vec{Y}_i$ are best reconstructed by the weights W_{ij}, by minimizing the cost function in equation (10).

The reconstruction weights are computed from the local neighborhoods of data points whereas the low dimensional embedding is computed using eigendecomposition. The various dimensions in the reduced dimension space are iteratively computed using the eigenvectors one at a time.

M. Laplacian Eigenmaps

Laplacian Eigenmaps is a geometric algorithm for high dimensional data representation. It relies on spectral techniques for performing dimensionality reduction. The method first constructs a graph that incorporates neighborhood information of the dataset [21]. Then the graph Laplacian is used to generate a reduced representation that preserves the local neighborhood characteristics. The graph Laplacian is an ideal representation of the network that can intuitively explains the network structure. It computes the eigenvalues and eigenvectors for the generalized eigenvalue problem: $Lf = \lambda Df$. Laplacian matrix is a symmetric, positive semi definite matrix which has real and non-negative eigenvalues. The smallest eigenvalue of *L* is *0* and the corresponding eigenvector is the constant *1* vector. The multiplicity of the eigenvalue *0* is equal to the number of connected components of the graph *G*. The eigenvector corresponding to the smallest eigenvalue of the Laplacian matrix results in a trivial partition of the network. The eigenpairs obtained as a result of eigendecomposition on graph Laplacian captures significant topological information about the network [22]. The eigenvectors, $f_0, \ldots f_{k-1}$ are ordered with respect to their eigenvalues. The eigenvector associated with eigenvalue *0* is neglected and the next *m* significant eigenvectors are selected for embedding in *m* dimensional Euclidean space. Laplacian Eigenmap provides a computationally efficient approach to nonlinear dimensionality reduction. It finds a low dimensional representation by preserving the local properties of the manifold.

N. Local Tangent Space Alignment

LTSA is a manifold learning method that transforms a nonlinear embedding of high dimensional data into a reduced dimensional space and also reconstruct the high dimensional
coordinates from the reduced representation. This is similar to LLE, but rather than projecting the points to a locally linear neighborhood, LTSA make uses of the tangent space of each data point and align those local tangent spaces to construct the embedding. It assumes that for each data point

in the high dimensional space, there exist a linear mapping to its local tangent space and vice versa. The steps for performing LTSA are similar to LLE which includes nearest neighbor search, weight matrix construction and partial eigendecomposition. For nonlinear dimensionality reduction, the local linear structures of data points are exploited to obtain a global nonlinear structure. LTSA considers tangent space, constructed from the neighborhood of a point as the local geometric information. The local tangent space provides a low-dimensional linear approximation of the local geometric structure of the nonlinear manifold [23]. The local tangent spaces are all aligned using appropriate transformation function to obtain a global non
linear embedding in the reduced dimension space. LTSA begins with the extraction of local information by determining *k* nearest neighbors of the data point [23]. The local information of a data point is calculated by performing eigendecomposition on the correlation matrix of the data neighborhood. The *d* largest eigenvectors obtained are centered to form the local information matrix. Then this local information has to be properly aligned to form a global solution. All local information matrices are iteratively added up to form the alignment matrix. The final step is the alignment of global coordinates. The low dimensional embedding is provided by the smallest *d+1* eigenvectors of the alignment
matrix, discarding the one associated with the smallest eigenvalue. LTSA focus to minimize the distance between points in the tangent space and the reduced dimension space. The solution to this minimization problem is provided by the d smallest eigenvectors of the alignment matrix [24]. LTSA performs eigendecomposition on a per point basis, rather than applying on the entire sparse matrix which highly reduces the computational complexity. It is fast as well as adaptive to complex nonlinear manifolds.

CONCLUSION

Dimension reduction is a popular research area which has gained focus due to its significance in high dimensional data analysis. It decreases the computational load and extract better quality features from data, hence find application in many areas. A variety of linear/nonlinear techniques are proposed for dimensionality reduction depending on the nature of the domain of interest. PCA is the oldest and most common approach used for dimensionality reduction. It is easy to compute and guaranteed to produce accurate low dimensional representation. For PCA to be effective the data elements should be related to each other, it performs poor in the case of uncorrelated data. Even though PCA produces accurate low rank embedding, the numerous principal components generated are difficult to interpret. ICA guarantees accuracy in the case of uncorrelated data since it operates on independent components of data. The problem may arise here is that the independent components generated may not be relevant to the context. With SVD, it produces the best rank-k approximation with minimum reconstruction error. The vectors formed in SVD may lack any meaning in terms of the field from which data is drawn. Also it does not take into account the sparse nature of input data. NMF is considered to be interpretable, as it produces parts based representation and retains more localized patterns. The drawback is that it might consume huge amount of memory for processing if the input matrix is large. CUR and CMD eliminated the interpretability problem by selecting columns and rows from actual data. They also preserve the sparsity property and yields accurate outcomes. CMD is considered to be the best cost effective method as it scales up the duplicate columns/rows produced by CUR. These are lossy methods that randomly select columns and rows to compute a low dimension representation. Hence there isn't any guarantee that it retains the intrinsic properties and structure of the original data in the reduced embedding. LDA performs dimensionality reduction by considering class discriminatory information. It finds directions along which the classes are best separated whereas PCA finds axes of maximum variance. Instead of a covariance matrix, LDA uses a within-scatter matrix of all c classes and a between-scatter matrix [25].

Kernel PCA is the nonlinear variant of PCA for performing nonlinear dimensionality reduction. The performance of Kernel PCA depends on the selection of appropriate kernel function which requires prior knowledge about the data. MDS has two variations metric and non-metric MDS that performs linear and nonlinear dimensionality reduction respectively. It is simple and relatively easy to implement method. MDS is very useful for data visualization and are able to uncover hidden structures in the data, but it has a got numerous limitations. There are chances for MDS to generate suboptimal or degenerate solutions, also probable to produce meaningless output. Isomap, LLE, Fastmap etc. are variations of MDS that eliminates its limitations and result in more effective solutions. Fastmap [18] is developed as an alternative to MDS, produces fast and efficient mapping of high dimensional data to low dimensional spaces. It provides distance preserving projections. Here, different pairs of pivot objects can be randomly selected to produce different projections of the data. It is highly scalable and can effectively handle large data sets.

LLE generates highly nonlinear embedding where the local topography is preserved by linear neighborhood relations rather than by the pairwise distances. Isomap shares many of the properties of LLE which performs nonlinear dimensionality reduction by preserving the geodesic distances between pairs of data points. Laplacian Eigenmap produces a low dimensional embedding by taking into account the structure of the manifold on which the data may possibly reside [21]. Isomap is a global method that takes into account distances between all pairs of points in the nearest neighbor graph. LLE and Laplacian Eigenmap are local methods that consider only the local neighborhood structures for mapping. LTSA computes the local tangent planes around each data point and all such tangent planes are aligned to produce the low dimensional embedding. LLE, LTSA and Laplacian Eigenmap operate on sparse matrices that save the computational complexity.

REFERENCES

[1] C. J. C. Burges, "Dimension reduction: A guided tour," Foundations and Trends in Machine Learning, 2010.

[2] F. S. Tsai, "Dimensionality reduction techniques for blog visualization," Expert Systems with Applications, Elsevier, 2010.

[3] Y.-Z. W. Y. C. Hua-Wei Shen, Xue-Qi Cheng, "A dimensionality reduction framework for detection of multiscale structure in

heterogeneous networks," IEEE Journal of Computer Science and Technology, 2012.

[4] "Principle component analysis (pca)." Notes.

[5] A. Kumar, "Analysis of unsupervised dimensionality reduction techniques," Advance Data and Information Engineering Confernece, Springer, 2013.

[6] H. Z.-K. J. A.-A. Snasel, V, "Reducing social network dimensions using matrix factorization methods," IEEE Social Network Analysis and Mining, 2009.

[7] P. D. M. W.Mahoney, "Cur matrix decompositions for improved data analysis," The National Academy of Sciences of the USA, 2009.

[8] Y. X. H. Z. C. F. Jimen Sun, "Less is more: Compact matrix decomposition for large sparse graphs," Statistical Analysis and Data Mining, ACM, 2008.

[9] H. A. M. P. Vaclav Snasel, "Behavior of the concept lattice reduction to visualizing data after using matrix decompositions," Innovations in Information Technology, IEEE, 2007.

[10] R. Gutierrez-Osuna, "Dimensionality rereduction-lda." Introduction to Pattern Recognition,Wright State University.

[11] S. Raschka, "Linear discriminant analysis bit by bit." Blog, August 2014.

[12] M. V. John A. Lee, Non Linear Dimensionality Reduction. Springer Science & Business Media, 2007.

[13] B. F. Manly, Multivariate Statistical Methods: A Primer, Third Edition. CRC Press, 2004.

[14] J. J. K. P. A. C. Nagiza F. Samatova, William Hendrix, Practical Graph Mining with R.CRC Press, 2013.

[15] S. Raschka, "Kernel tricks and nonlinear dimensionality reduction via rbf kernel pca." Blog,September 2014.

[16] J. A. B. Mathieu Fauvel, Jocelyn Chanussot, "Kernel principal component analysis for feature reduction in hyperspectrale images analysis," Signal Processing Symposium, 2006.NORSIG 2006. Proceedings of the 7th Nordic, 2006.

[17] N. T. T. G. W. Imran Khan, Joshua Zhexue Huang, "Ensemble clustering of high dimensional data with fastmap projection," International Workshop on Algorithms for Large-Scale Information in Knowledge Discovery, PAKDD 2014, 2014.

[18] K.-I. L. Christos Faloutsos, "Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets," Proceedings of the 1995 ACM SIGMOD international conference on Management of data, 1995.

[19] J. C. L. Joshua B. Tenenbaum, Vin de Silva, "A global geometric framework for nonlinear dimensionality reduction," Sciencemag.org, 2000.

[20] L. K. S. Sam T. Roweis, "Nonlinear dimensionality reduction by locally linear embedding,"Science Magazine, 2000.

[21] P. N. Mikhail Belkin, "Laplacian eigenmaps for dimensionality reduction and data representation," Neural Computation, ACM, 2003.

[22] M. F. Luca Bergamaschi, Enrico Bozzo, "Computing the smallest eigenpairs of the graph laplacian," 2013.R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[23] H. Z. ZHENYUE ZHANG, "Principal manifolds and nonlinear dimension reduction via local tangent space alignment," SIAM Journal on Scientific Computing, 2005.

[24] R. Z. Harry Strange, Open Problems in Spectral Dimensionality Reduction. Springer Science & Business Media, 2014.

[25] R. M. S. S. S. Swamidoss Sathiakumar, Lalit Kumar Awasthi, Proceedings of International Conference on Internet Computing and Information Communications: ICICIC Global2012. Springer Science & Business Media, 2013.