Overview on Reduction of Dimensionality

Jata Shankar Yogesh
Department of Computer Science and Engineering
THE NORTHCAP UNIVERSITY, Gurgaon

Gaurav Aggarwal
Department of Computer Science and Engineering
THE NORTHCAP UNIVERSITY, Gurgaon

Abstract- Every minute or second data are increasing exponentially. So, we must need to find ways to automatically analyze, classify, summarize and characterize the data. There are many methods to reduce the dimensionality. So here explain different methods to reduce the dimensionality i.e. PCA, SVM, Isomap, Autoencoder, Boltzmann Machine, Deep Belief Network etc.

Keywords- Reduction of dimensionality, PCA, Boltzmann Machine, Deep Belief Network

I. INTRODUCTION

Today world are living in an informational age or data age. Every minute or second data are increasing exponentially. Real world data means digital photographs, speech signals, scientific observations, statistic calculations etc. We have been collected a mass of data, from simple mathematical measurements and text documents, to more complex information such as structural data, multimedia channels and hypertext documents [10]. But users have no time to look investigate data. They want only precious resource. So, we must need to find ways to automatically analyze, classify, summarize and characterize the data. Improvements in technology have also improved data quality and quantity. This improvement also helps in increasing the features or dimensions of observed data. As the dimensions increases the size of data increases. Second, introduction of interdisciplinary fields motivating students from diverse fields to work together and develop new road maps in different research areas this also increase the size of data as the accuracy increases with new mile stones. The challenge is to store data in minimum space and retrieve it adequately and accurately.

Data Mining is a truly interdisciplinary approach which can be defined as discovery of useful, possibly unexpected, interesting patterns in large data also termed as Knowledge discovery from data, or KDD [9] which follow an essential iterative sequence of steps in the process of KDD as follows:

1. Data Cleaning: Detection of bogus data, removing inconsistent and noise data.

2. Data Integration: It is process of combining data coming from heterogeneous sources like the web, databases, data warehouses, other information repositories or the data streamed into system dynamically.

3. Data Selection: It is process of retrieving data from database relevant to the analysis task.

4. Data Transformation: It is process of transforming and consolidating data into appropriate forms for mining by applying summary or aggregate operations.

5. Data Mining: It is process of extracting data patterns by using various good methods.

6. Pattern Evaluation: It is process of identifying interesting patterns that represent knowledge based on interestingness measures like Decision trees, Clusters, hidden-Markov etc…

7. Knowledge Presentation: It is process of representing mined knowledge to users by applying visualization and knowledge representation techniques.

First 4 steps are different forms of data pre-processing where data are prepared for mining. After mining interesting patterns are presented to the users and may be stored as new knowledge in the knowledge base.

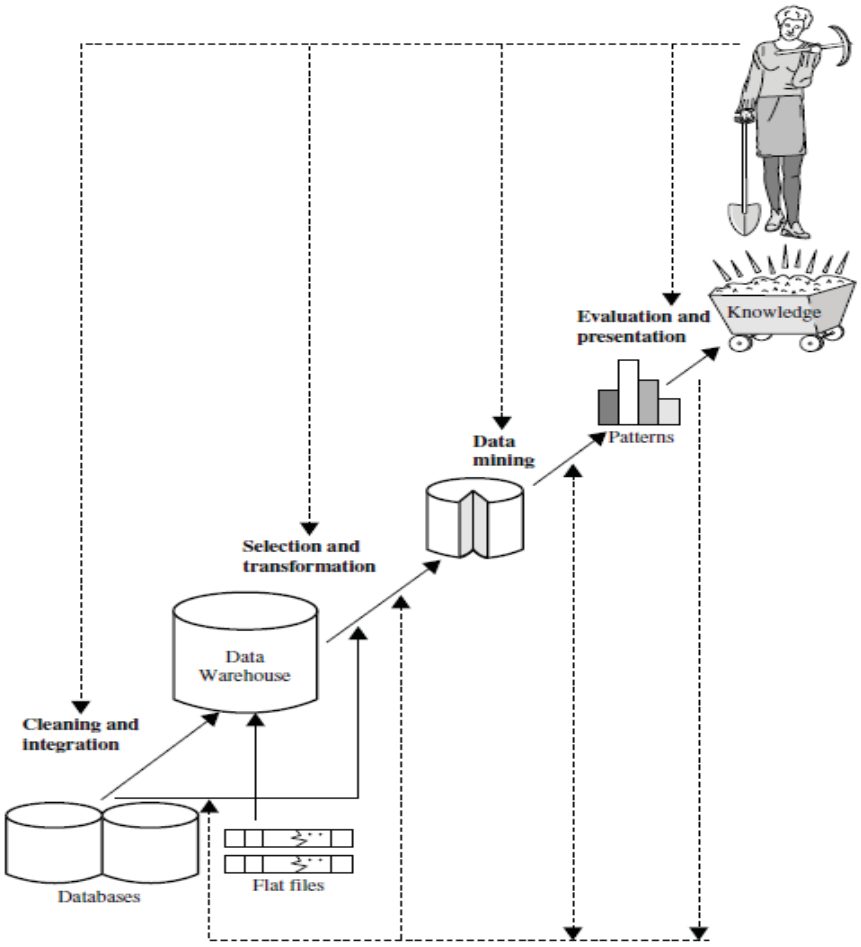
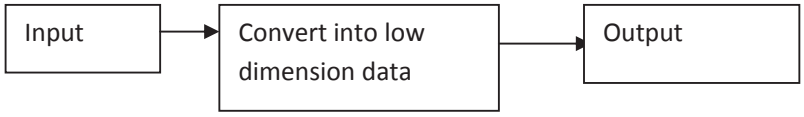


Figure 1. Knowledge discovery from data

Reduction of dimensionality is used to minimize space, fast information retrieval, image processing, easy visualization, good classification of dataset easily etc. Actually reduced dimensional representation should have a dimensionality that means the minimum number of dimensions used to account for the observed properties of the data. Reduction of dimensionality follows stages:



Dimensionality reduction method can also be thought of as a tool for visualizing high dimensional data. High dimensional data introduce much more mathematical challenges with many more opportunities, and are also derive new concepts. One of the biggest problem with the high dimensional data is that, in many cases, all the variables measured are not important or useful for the understanding the dataset, while only certain of very less

dimensions are sufficiently descriptive about the dataset. So feature selection during dimensionality reduction is a problem.

II. TECHNIQUES

2.1 Principal Component Analysis

Principal component analysis (PCA) [1] is a most common unsupervised technique. It is used for reducing the dimension of a given set of data. It is a linear dimension reducing technique. In Principal component analysis, we are finding a mapping from original d-dimensional space to a new k-dimensional subspace where d>k, with minimum loss of information. This new k-dimensional subspace is known as principal components and is orthogonal. The major applicability of PCA technique is to identify the principal direction in which data varies even when data is having large number of variables and redundant that is some of the variables are highly correlated. PCA is used to reduce the dimensionality of given data variables, extract and identify the important data variables [2].

Steps for PCA [3],

Suppose that dimension of data set X is 'p', where X=$\{x_1, x_2, \ldots, x_p\}$

1. Subtract the mean
   In this step, we have subtracted mean value from each data. This process produce a data set whose mean is 0(zero).
   Mean (m) of X= $\frac{1}{p}\sum_{i=1}^{p} x_i$
2. Calculate the covariance matrix
   In this covariance matrix, non-diagonal elements are positive.
   Covariance matrix(c) = $\frac{1}{p-1}\sum_{i=1}^{p}(x_i - m_i)(x_i - m_i)^T$

3. Measure the eigenvectors and eigenvalues of the covariance matrix

4. Reduce dimension and form feature vector

The eigenvector with the highest eigenvalue is the principal component of the data set. Now, ignore the components of less significance so we can loss some information but eigenvalues are small, we don't loss much.

Examples-

In given data, there are p dimensions

Find out n eigenvectors and eigenvalues

Select only first q eigenvectors

Final dataset has only p dimensions

5. Deriving new data

FinalData= RowFeatureVector X RowZeroMeanData

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.

RowZeroMeanData is the mean-adjusted data transposed, i.e. the data items are in each column, with each row holding a separate dimension.

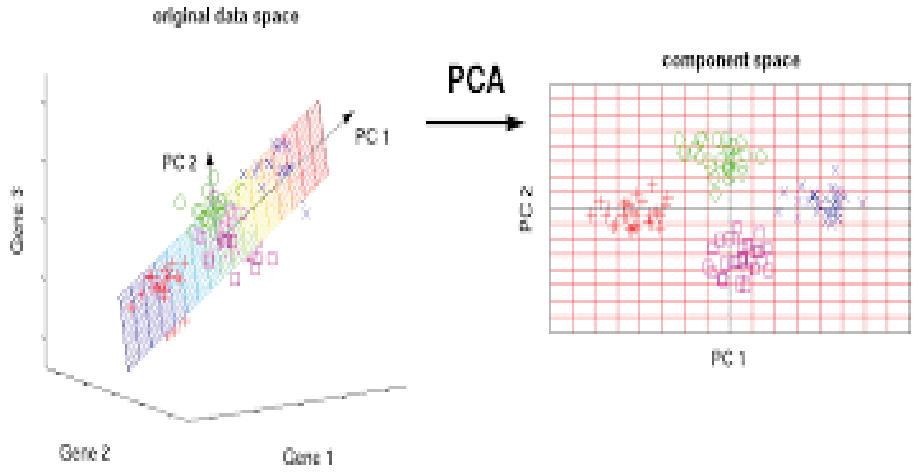


Figure 2. PCA

2.2 Support vector machine

Support vector machine (SVM) is a most common used supervised learning method. It is used for both classification and regression. It is belonging to a family of generalized linear classification [11]. Support vector machine has property that is SVM minimize the empirical classification error and maximise the geometric margin simultaneously. So Support vector machine also called Maximum Margin Classifiers. In Support vector machine, two parallel hyperplanes are constructed so that both hyperplanes separate the data. The separating hyperplane is maximizing the distance between two parallel hyperplanes. Presently, Support vector machine is widely used in object detection and recognition, speech recognition, text recognition, context based image retrieval, biometrics etc. Support vector machine issued for linear and non-linear data but need to select correct model.

Given a set of data points

$$\{(x_i, y_i)\}, i = 1, 2 \ldots, n$$

Where

$$For\ y_i = +1, \quad w^T x_i + b > 0$$

$$For\ y_i = -1, \quad w^T x_i + b < 0$$

With a scale transformation on the both w and b, the above is equivalent to

$$For\ y_i = +1, \quad w^T x_i + b \geq 1$$

$$For\ y_i = -1, \quad w^T x_i + b \leq -1$$

We know that

$$w^T x^+ + b = 1$$

$$w^T x^- + b = -1$$

The margin width is:

$$M = (x^+ - x^-) \cdot n$$

$$= (x^+ - x^-) \cdot \frac{w}{\|w\|} = \frac{2}{\|w\|}$$

Formulation:

$$Maximize \ \frac{2}{\|w\|}$$

Such that

$$For \ y_i = +1, \quad w^T x_i + b \geq 1$$

$$For \ y_i = -1, \quad w^T x_i + b \leq -1$$

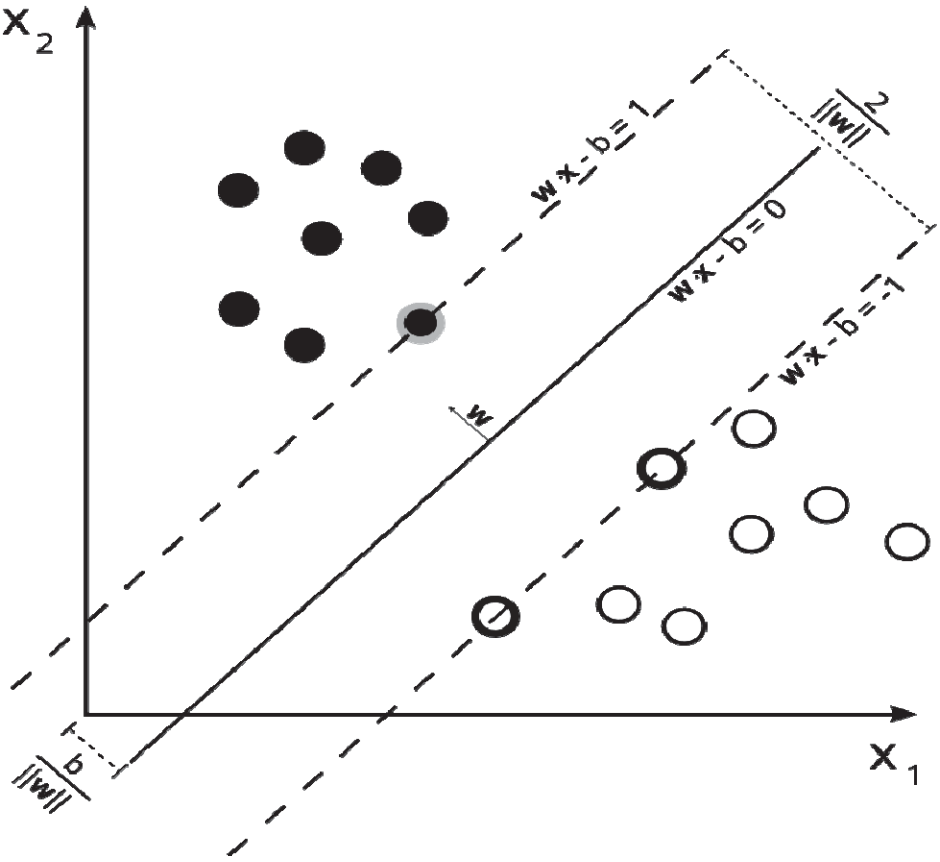


Figure 2. Support Vector Machine

2.3 Multidimensional Scaling

MDS is also a linear approach for dimensionality reduction [12]. It helps in visualizing the level of similarity of given data set by calculating the distance between each and every data and point. Let A = ($x_1, x_2, x_3, \ldots, x_N$) is a n by N data set matrix where every column represents one n dimensional data vector, second matrix B is square distance matrix with dimensions N by N, third matrix Q is inner product matrix of every two data vector[2]. Then:

$$Q = A^T A$$

$$q_{ij} = A_i^T A_j$$

$$B = (x_i - x_j)^T (x_i - x_j)$$

$$B = x_i^T x_i - 2x_i^T + x_j^T x_j$$

This gives: $$Q = -\frac{1}{2}B$$

After that calculating Eigen values of Q we get:

$$\xi = VUV^T$$

Where:

- U = diagonal matrix having Eigen values
- V = contains Eigen vectors as its columns

Chose first Eigen vectors 'm' (m<n) having largest Eigen values to construct low dimensional space V' and corresponding U'. The resultant low dimensional space is as follows

$$Y = V'U'^{\frac{1}{2}}$$

Multidimensional scaling works like as PCA if row and column of given data set A having zero mean. MDS and PCA both methods have the same restriction.

2.4 Isomap

The key feature of isomap is to map the high dimensional data into non linear low dimensional data [13]. Multi dimensional Scaling works on the basis of Euclidean distance, and does not take attention about the data point distributed in neighbour. Let the data is in curved manifold, MDS calculate Euclidean distance between two points which may be very small whereas the distance of manifold may be very large then the Euclidean distance. Isomap [2] attempts the solution of this problem by maintaining pair wise Geodesic distance (distance between two points measure over the manifold) by estimating shortest path in a neighbourhood graph driven from data. Isomap works as:

- Make neighbourhood graph.
- Estimate Geodesic distance of each pair using shortest path algo.
- Use pair wise distance in MDS for finding low dimensional space.

This method fails when parameters are non convex. Second problem with this method is it fails to recover correct dimension for space with high intrinsic curvature. Third it is very slow with large training data set.

2.5 Restricted Boltzmann Machines

Boltzmann Machines [8] are fully connected symmetric network of binary (0, 1) stochastic processing units. Boltzmann Machines (BM) are used for learning some important features of unknown probability distribution which are based on samples from this distribution. But Boltzmann Machines technique is very complicated and time consuming. For making simple and easy use of this model, some restrictions are done over the existing network Boltzmann Machines. So new simplified Boltzmann Machines is called Restricted Boltzmann Machines (RBMs). Restricted Boltzmann Machines (RBMs) are also known as Markov Random Field (MRF).So Restricted Boltzmann Machine [6] is two layer, undirected graphical model. In which one layer is observed data layer that is called visible layer and another layer is latent variables that is called hidden layer. This visible and hidden layer is fully symmetric connected with undirected weight and there are no intra layer connections between the hidden units and visible units. So visible units have edge connection to hidden units and hidden units are edge connections to visible units. In Boltzmann machines [4] everything is defines in terms of energies of joint configuration of visible and hidden units. So Restricted Boltzmann Machine [6] determine the energy of a joint of visible and hidden units E(v, h).
Probability of every possible visible-hidden vector pair via energy function

$$E(v,h;\theta) = -\sum_{i=1}^{V}\sum_{j=1}^{H} v_i h_j w_{ij} - \sum_{i=1}^{V} b_i v_i - \sum_{j=1}^{H} a_j h_i$$

then,

$$p(\mathbf{v},\mathbf{h}) = \frac{1}{Z}e^{-E(\mathbf{v},\mathbf{h})}$$
$$Z = \sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}$$

RBMs need two different phase for learning (a) Pre training and (b) Fine tuning. During pre training constructs RBM with an input layer 'v' and hidden layer 'h'. Now train the network. After training use backpropagation for fine tune for getting good weights. Fix the weights between layer 'v' and 'h'. Stack another layer 'h1' as hidden layer and assume 'h' as input layer, so the lastly done process again. This process can be repeated as the number of hidden layers introduced. Successful learning of RBMs gives a compact representation of input data.

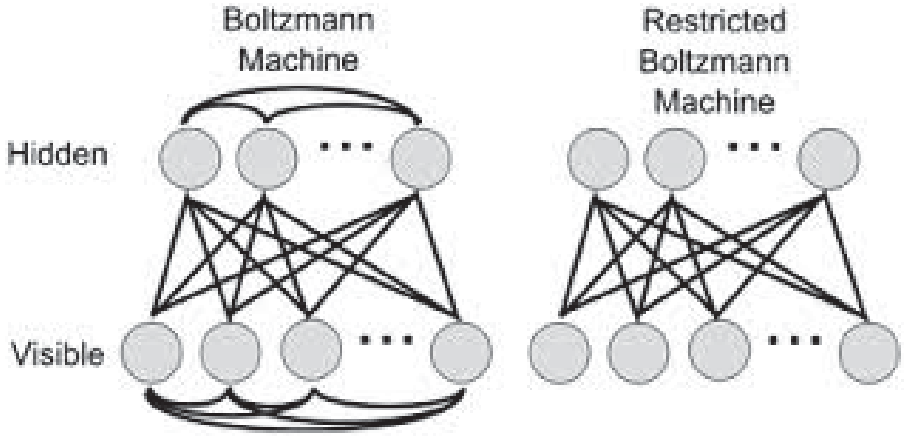


Figure 5. Boltzmann Machine and Restricted Boltzmann Machine

2.6 Autoencoder

Auto-encoders [14] are simple tool which transfer input into output with the least possible amount of distortion using unsupervised learning. An Auto-encoder neural network is unsupervised learning algorithm. Auto-encoders use three or more layers to do the task. Three layers means one input layer, one output layer with at least one hidden layer. Input layer and output layer should have same number of units. One of many back-propagation algorithms (conjugate gradient method, steepest descent, etc.) is used to train an auto-encoder for setting the target value to be equal to the input i.e. it uses y(i) = x(i) , where {x(1) , x(2) , x(3) , …} Є Rn and y(i) is output layer with hw,b(x) as identity function. The auto-encoder mean is trying to learn an approximation to the Identity function. Layer L1 is input layer, layer L2 is hidden layer and layer L3 is output layer. For auto-encoders generally number of units at hidden layer are less in number than input layer and output layer.

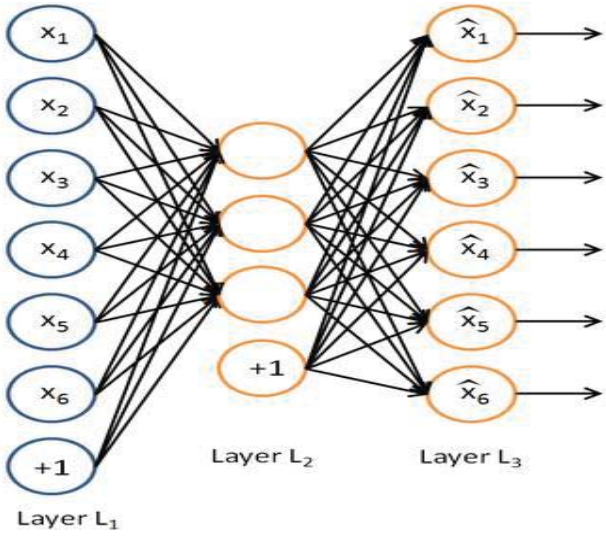


Figure 6. Autoencoder with three layers

By minimizing the number of unit at hidden layer we generate a structure that can compress the input data in low dimension and then reconstruct that in same as input. This is a good structure for discovering some correlation if the input vector has some correlation. It became some complicated if input data is independently and identically distributed (iid). When input data has some correlation the auto-encoder works similar to PCA. Some time we can use an auto-encoder with more number of units at hidden layer then input layer and output layer, it will also generate an interesting structure with some imposed constrains on network. The auto-encoder works well if the weight of network initializes accurately that means the big problem with auto-encoder using back-propagation is setting of initial weights when network contain many layers. Auto encoder with one hidden layer is a simple autoencoder. Simple autoencoder can be converted into multiple layered autoencoders by using many hidden layers.

2.7 Deep Belief Network

Deep belief nets are probabilistic generative models. Deep belief nets consists of multiple layers of stochastic binary (0, 1) values (latent variables).This multiple layers are called hidden layer [5].we can also called that Deep belief network is composition of simple learning model that is Restricted Boltzmann Machine.RBM that is Restricted Boltzmann Machine consists of only 2 layers i.e. visible layer and hidden layer but in Deep belief network consists of multiple hidden layers. Deep belief network is a special type of model in which top two layers forms undirected associative memory and remaining hidden layer form a directed acyclic graph [8].

Some important properties of Deep belief network

1. There is a fast, greedy learning algorithm that can find a good set of even in deep networks with number of parameter and many hidden layers

2. Deep belief algorithm is unsupervised but also applied to labelled (supervised) data.

3. There is a fine-tuning algorithm that learns an excellent generative model that outperforms discriminative methods on the MNIST database of hand-written digits.

4. It is fast and accurate.

5. Algorithm is local and depend only the states of the presynaptic and postsynaptic neuron.

6. The communication is simple. Neurons need only to communicate their stochastic binary (0,1) states.

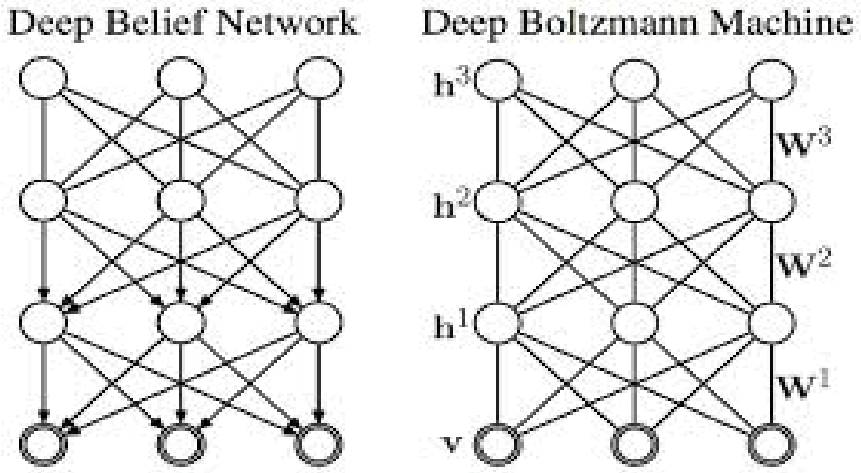


Figure 7. Deep Belief Network and Deep Boltzmann Machine

III. CONCLUSION

After study of different types of dimensionality reduction methods, I identified that different method used for different purpose. Principal component analysis (PCA) is a supervised linear dimension reducing technique. Support vector machine (SVM) is also a most common used supervised learning method. It is used for both classification and regression and it is issued for both linear and non-linear data but need to select correct model. MDS is also a linear approach for dimensionality reduction. It helps in visualizing the level of similarity of given data set by calculating the distance between each and every data and point. Isomap is to map the high dimensional data into non linear low dimensional data. Auto-encoders are simple tool which transfer input into output with the least possible amount of distortion using unsupervised learning. An Auto-encoder neural network is unsupervised learning algorithm. Boltzmann Machines (BM) are used for learning some important features of unknown probability distribution which are based on samples from this distribution. Deep belief network is composition of simple learning model that is Restricted Boltzmann Machine.

REFERENCES

[1] A tutorial of Principal component analysis, Lindsay I Smith, February 26, 2002.
[2] http://www.inf.ed.ac.uk/publications/thesis/online/IM110952.pdf
[3] .http://kybele.psych.cornell.edu/~edelman/Psych-465-Spring-2003/PCA- tutorial.pdf
[4] http://www.scholarpedia.org/article/Boltzmann_machine
[5] http://www.scholarpedia.org/article/Deep_belief_networks
[6] Sarikaya, R., Hinton, G. E., & Deoras, A. (2014). Application of deep belief networks for natural language understanding. Audio, Speech, and Language Processing, IEEE/ACM Transactions on, 22(4), 778-784.
[7] Mohamed, A. R., Dahl, G. E., & Hinton, G. (2012). Acoustic modeling using deep belief networks. Audio, Speech, and Language Processing, IEEE Transactions on, 20 (1), 14-22.
[8] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. Neural computation, 18(7), 1527-1554.
[9] Han, J., Kamber, M., & Pei, J. (2011). Data mining: concepts and techniques. Elsevier.
[10] .https://webdocs.cs.ualberta.ca/~zaiane/courses/cmput690/notes/Chapter1/ch1.pdf
[11] http://www.cs.toronto.edu/~hinton/MatlabForSciencePaper.html
[12] http://www.obgyn.cam.ac.uk/cam-only/statsbook/stmulsca.html
[13] http://en.wikipedia.org/wiki/Isomap
[14] http://en.wikipedia.org/wiki/Autoencoder
[15] Brief Introduction of Back Propagation (BP) neural Network Algorithm and Its Improvement, Jing Li, Ji-hand Cheng, Jing-yuan Shi, and Fei Huang, Springer 2012, pp.553-558.