Journal of Theoretical and Applied Information Technology
20th October 2015. Vol.80. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A REPORT ON REDUCING DIMENSIONS FOR BIG DATA USING KERNEL METHODS

CH. RAJA RAMESH, K RAGHAVA RAO, G.JENA, C V SASTRY

[1]Research Scholar, K. L. University, Associate Professor, Sri Vasavi Engineering College,
Tadepalligudem, Pedatadepalli,A.P 534101
[2]Professor, Dept. of CSE, K. L. University
[3]Professor & Principal Ronald Institute of Technology, Berhampur
[4]Director R&D, Dept. of CSE, Regency Institute of Tecnology , Yanam

ABSTRACIT

Big-Data is very popular word to perform huge data processing; it brings so many opportunities to the academia, industry and society. Big data hold great promise for discovery of patterns and heterogeneities which are not possible with small data. Big Data faces many challenges like unique computational and statistical challenges including scalability and storage. Among these challenges some maybe mentioned as noise accumulation, spurious correlation, incidental endogeneity and measurement errors. Most of the problems occur based on the size of the data associated with large number of attributes. Irrelevant attributes add noise to the data and increase the size of the model. Moreover datasets with many attributes may contain groups of data that are correlated. All these attributes may be measuring the same feature. One way of dealing with this problem is to eliminate some attributes (dimensions) which do not exhibit large variance and hence do not affect the clusters. Several techniques exist to ignore certain attributes or dimensions such as Principle component analysis (PCA), Singular Value Decomposition (SVD) etc. We review these techniques in this paper with respect to clustering. We plan to use principle component analysis and Kernel methods for Dimensionality reduction which is an essential preprocessing technique for large scale data sets. It can be used to improve both the efficiency and effectiveness of classifiers.

Keywords: *Big Data , Dimensionality Reduction, Feature Extraction, Fuzzy , Term Data.*

1. INTRODUCTION

Some new sources of data like Satellites automatically generating huge volumes of data calls for a wide variety of data processing and analysis tools. The traditional ideas of mathematical statistics leading to inferences such as hypothesis testing and confidence statements have relatively little to offer. There exists no formal definition of the phenomenon or artifact, in terms of a carefully stated mathematical model. There exists no formal derivation of the proposed processing strategy, suggesting that it is in some sense naturally associated with the phenomenon to be treated.

Over the last forty years, Data Analysis has developed at breakneck pace, responding to the rapid advances in information technology: massive data storage, rapid throughput, effective algorithms for basic problems. Over the last twenty years particularly, the largest contributions of statistics to data analysis have been in the formalization of information technology for data analysis, in the form of software packages like S and S-Plus, CART, Data Desk, GLIM, MacSpin, and in the identification of ways we can use information technology to substitute for analysis – bootstrap, Monte-Carlo Markov Chain.etc.

Traditional dimensionality reduction approaches fall into two categories. Feature selection and feature extraction. Feature selection procedures can be seen as a search technique for proposing new feature sub-sets along with an evaluation measure. The choice of the evaluation matrix heavily influences the algorithm. Feature extraction is an attribute reduction process. The transformed attributes may be a linear combination of the existing attributes thus achieving attribute reduction or dimensionality reduction, typically more effective than feature selection. This process (feature extraction) reduces the time complexity, reduces noise accumulation, and generates stabilized data processing. We wish to consider the analysis of big data and use the above techniques for clustering and classification.

Big Data hold great promises for discovering subtle population patterns and heterogeneities that are not possible with small-scale data. On the other hand, the massive sample size and high dimensionality of Big Data introduce unique computational and statistical challenges, including scalability and storage bottleneck, noise accumulation, spurious correlation, incidental endogeneity and measurement errors. These challenges are distinguished and require a new computational and statistical paradigm.

2. DATA TRENDS

2.1 Data: Last few decades, data management, and data processing have become very important everywhere and impacts our daily life and work [4].

2.2. Recent Data Trends: Huge investments have been made in various data gathering and data processing mechanisms. The information technology industry is the fastest growing and most lucrative segment of the world economy, and much of the growth occurs in the development, management, and warehousing of prodigious streams of data for scientific, medical, engineering, and commercial purposes. Some recent examples include:

2.3. Biotech Data: Virtually everyone is aware of the fantastic progress made in the last five years in gathering data about the human genome. A common sight in the press is pictures of vast warehouses filled with genome sequencing machines working night and day, or vast warehouses of computer servers working night and day, as part of this heroic effort.

This is actually just the opening round in a long series of developments. The genome is only indirectly related to protein function and protein functions are only indirectly related to overall cell function. Over time, the focus of attention will go from genomics to proteomics and beyond. At each round, more and more massive databases will be compiled.

2.4. Financial Data: Over the last decade, high-frequency financial data have become available. Now with the advent of new exchanges such as Island.com, one can obtain individual bids to buy and sell, and also the full distribution of such bids.

2.5. Satellite Imagery: Providers of satellite imagery have available a vast database of Satellite Images. Projects are in place to compile databases to resolve the entire surface of the earth to 1 meter accuracy. Applications of such imagery include natural resource discovery.

2.6. Hyper spectral Imagery: It is now becoming common, both in airborne photographic imagery and satellite imagery to use hyper-spectral cameras which record, instead of three color bands RGB, thousands of different spectral bands. Such imagery is, presumably, able to reveal subtle information about chemical composition and is potentially very helpful in determining crop identity, spread of diseases in crops, in understanding the effects of droughts and pests, and so on. In near future we can expect hyper-spectral cameras to be useful in food inspection, medical examination, and so on.

New optical sensors are able to obtain imagery not just with red-green-blue color sensitivity, (3 numbers per pixel) but instead a full spectrum with thousands of frequency bands being measured. Thus an image is not just, say, a 4096*4096 pixel array, but a 4096*4096*1024 pixel volume. Such data can be viewed as an N by D array. Suppose we have I images in our database, each of size n by n with S spectral bands. We can let $D = S$ and let $N = In^2$.

We will consider what statisticians consider the usual data matrix, a rectangular array with N rows and D columns, the rows giving different *observations* or *individuals* and the columns giving different *attributes* or *variables*. In a classical setting we might have a study like the Framingham Heart study, with data gathered very carefully over decades, and ultimately consisting of about N = 25,000 records about the individual residents of Framingham Massachusetts on D = 100 variables.

2.8. Web Term-Document Data: Document retrieval by web searching has seen an explosive growth over the last 5 years. One approach to document retrieval is the vector space model of information retrieval. In this model, one compiles *term-document matrices*, N by D arrays, where N, the number of documents, is in millions, while D, the number of terms (words), is in tens of thousands, and each entry in the array measures the frequency of occurrence of given term in the given document, with a suitable normalization.

2.9. Sensor Array Data: In many fields we see the use of sensor arrays generating vector-valued observations as functions of time. For example, consider a problem in study of evoked potential analysis in neuroscience. An array of *D* sensors is attached to the scalp, with each sensor recording *N* observations over a period of seconds, at a rate of X thousand samples, per second. One hopes to use such data to witness the response of human neural system to various external stimuli. This array allows one potentially to localize various effects within the head. [21].

2.10. Gene Expression Data: A very “hot” data analysis topic at the moment involves gene expression data. Here we obtain data on the relative abundance of D genes in each of N different cell lines. The goal is to learn which diseases are associated with which arrangement.

2.11. spectral Imagery. It is now becoming common, both in airborne photographic imagery and satellite imagery to use hyper spectral cameras which record, instead of three color bands RGB, thousands of different spectral bands. Such imagery is presumably able to reveal subtle information about chemical composition and is potentially very helpful in determining crop identity, spread of diseases in crops, in understanding the effects of droughts and pests, and so on. In the future we can expect hyper spectral cameras to be useful in food inspection, medical examination, and so on.

2.12. Consumer Preferences Data: Recently on the world-wide-web we have seen the rise of attempts to gather information about browsing and shopping behavior of consumers – along with demographics and the survey results are used to modify the presentation of information to users. Examples include recommendation systems used at Netflix and Amazon, and personalization systems like X-amplify. We mention briefly the Netflix scheme http://www.netflix.com. Each consumer is asked to rate about 100 films; based on that rating, the consumer is compared to other customers with similar preferences, and predictions are made of other movies which might be of interest to the consumer based on experiences of other customers who viewed and rated those movies. Here we have a rectangular array giving responses of N individuals on D movies, with N potentially in the millions and D in the hundreds (or eventually, thousands).

2.13. Consumer Financial Data: Every transaction we make on the web, whether a visit, a search, a purchase, is being recorded, correlated, compiled into databases, and sold and resold, as advertisers scramble to correlate consumer actions with pockets of demand for various goods and services

How Useful is all this?

One can easily make the case that we are gathering too much data already, and that fewer data would lead to better decisions and better lives [4]. But one also has to be very naïve to imagine that such wistful concerns amount to much against the onslaught of the forces that are mentioned. In science, engineering, and even government administration and business we see major efforts to gather data into databases. In some cases commercial enterprises have made huge collection of datasets concerning the surfing habits of internet users assuming that knowledge of consumer web surfing click streams can be sold, traded or otherwise leveraged into value.

Similarly, giant investments have been made to decode the human genome and make it available to biological researchers. It is claimed that this data will translate into an understanding of protein expression, and then to underlying biology and biochemistry.

We can’t say at the moment whether such assumptions will prove valid or not. What we can say is that our society has chosen to gather massive amounts of data, and this trend is accelerating yearly, and that major efforts are underway to exploit large volumes of data.

Data Stores Types: In **Key-Value stores**, the schema can differ from row to row. Key-Value stores are great for stock quoting, parts lists and other forms of high-volume data storing.

Column stores are, again, key-value, but “super columns” or “column families” are declared in the schema. For example, a super column could be Name, which is broken down into First Name and Last-name columns. This slight deviation to the schema makes column stores very useful for Big Data with its mix of known and unknown in every row. Column stores are very useful for time series data.

Graph Databases primarily store the aforementioned relationship information. They therefore handle a workload very different from other NoSQL data stores. “Edge” is the term for a relationship and the edges can connect any nodes (which are like rows in a table) that have a relationship.

3. REPRESENTATION OF DIMENSIONALITY REDUCTION

In this paper, we have given a sample mathematically represented by a matrix f X n, where n is the number of objects and f is the feature number. Each object is denoted by a Column vector x_i i = 1,2,3,…n and the kth entry of x_i is denoted by x_{ik} k = 1,2,3,…f. Assume that these feature vectors belong to different classes and the sample number of jth class is n_j. We use c_j to represent class j, j = 1,2,3,…c . The mean vector of the j^{th} class is $m_j = \frac{1}{n_j}\sum_{x_i \in c_j} x_i$ and the mean vector of all samples is $= \frac{1}{c}\sum_{j=1}^{c} m_j$. The dimensionality reduction problem can be stated as the problem of finding a function $f{:}R^d \rightarrow R^p$, where p is the dimension of data after dimensionality reduction (p <<f) so that object $x_i \in R^f$ is transformed into $y_i = f(x_i) \in R^p$.

4. DATA ANALYSIS

In studying an *N*-by-*D* data matrix, we often refer to it as *D*-dimensional – because we take the view of *N* points in a *D*-dimensional space. In this section we describe a number of fundamental tasks of data analysis. Good references on some of these issues include [9, 10, 11].

4.1 Classification: In classification, one of the *D* variables is an indicator of class membership. Examples include: in a consumer financial data base, most of the variables measure consumer payment history, one of the variables indicating whether the consumer has declared bankruptcy. The analyst would like to predict bankruptcy from credit history; in a hyper spectral image database all but one of the variables give spectral bands, an extra variable gives an indicator of ground truth chemical composition; the analyst would like to use the spectral band information to predict chemical composition.

Many approaches have been suggested for classification, ranging from identifying hyper planes which partition the sample space into non-overlapping groups, to *k*-nearest neighbor classification; see [10].

4.2 Regression In regression setting, one of the *D* variables is a quantitative response variable. The other variables are used to predict it. Examples include: in a financial data base, the variability of exchange rates today are predicted, given recent exchange rates. There is a well-known and widely used collection of tools for regression modeling; see [9, 20]. In linear regression modeling, we assume that the response depends on the predictors linearly,

$$X_{i,1} = a_0 + a_2\, X_{i,2} + Z_i \qquad (1)$$

The idea goes back to Gauss, if not earlier. In nonlinear regression modeling, we assume that the response depends on the predictors in a general non linear fashion,

$$X_{i,1} = f\left(X_{i,2} \dots X_{i,D}\right) + Z_i \qquad (2)$$

Linear regression modeling involves mostly linear algebra: the estimated coefficients by the least-squares method can be obtained by $\hat{}a = (X^T X) - X^T Y$, where *Y* is the column vector of response data. Nonlinear regression can involve: local linear fits, neural nets, radial basis functions, etc.

4.3 Latent Variables Analysis: In latent variables modeling we propose that

$$X = AS$$

where X is a vector-valued observable, *S* is a vector of unobserved latent variables, and *A* is a linear transformation converting one into the other. Often, the hope is that a few underlying latent variables are responsible for the structure we see in the array *X*, and by uncovering those variables, we have achieved important insights. Principal Component Analysis [18, 15, and 17] is an early example of this. One takes the covariance matrix *C* of the observable *X*, obtains the eigenvectors, which will be orthogonal, bundles them as columns in an orthogonal matrix *U* and defines

$$S = U'X.$$

This tool is widely used throughout data analysis in sciences, engineering, and commercial applications. Projection on the space spanned first *k* eigenvectors of *C* gives the best rank *k* approximation to the vector *X* in a mean square sense.

A now standard application comes in latent semantic indexing, where it is used to perform web searching [9, 20]. One extends the PCA method to a singular value decomposition factorization

$$X = UDV'$$

where *V* is the matrix of eigenvectors of *C* and D is the diagonal matrix with square roots of the

Eigen values of *C*. A query is a vector *α* indicating a list of terms to search for and responses are sorted based on values of $UD_kV'\alpha$ to find documents with large query values. Here D_k is a *k*-term approximation to the diagonal matrix *D* keeping only the *k* biggest terms. In effect the *k*-term approximation causes grouping of both terms and documents together, so that one can obtain 'hits' on documents that do not contain the precise term used, but that do contain a highly correlated term or terms.

PCA has been tried in image analysis, where it has been used to study images of faces. In that application, the eigenvectors can be viewed as images – "eigenfaces" – searching for matches of faces in a database of faces. It can then be processed in a fashion similar to the LSI model: if *α* gives the data vector for a new face, look for large entries in the output of the rank-*k* approximation for appropriate *k*, in $UD_kV'\alpha$.

In the last decade, an important alternative to PCA has been developed: ICA – independent components analysis [12, 13, and 14]. It is valuable when, for physical reasons, we really expect the model $X = AS$ to hold for an unknown *A* and a sparse or non Gaussian *S*. The matrix *A* need not be orthogonal.

An example where this occurs is with Array Data, where one assumes there are several sources, each one coupled with different strength to different sensors (for example, based on proximity of source to sensor). A typical example is the cocktail party problem: one has several microphones and several human speakers, the speakers are talking simultaneously and each microphone is picking up all the speakers at once.

4.4 Clustering: Cluster Analysis could be considered a field of its own, part art form, part scientific undertaking.

One seeks to arrange an unordered collection of objects into a collection of objects which are similar. There are many ways to do this, serving many distinct purposes, An obvious application area would be in latent semantic indexing, where we might seek an arrangement of documents so that nearby documents are similar and an arrangement of terms so that nearby terms are similar. See for example [19].

Recently, more quantitative approaches have been developed, of which we mention two here. The first, Gene Shaving, is described in [22]; it has been developed by a team of statisticians and pharmacists. The underlying model is

$$X_{i,j} = \mu_0 + \sum_{k=1}^{K} \alpha_k \beta_k^T$$

Where each β_k is a *D*-vector, each α_k is an *N* vector taking values 0 and 1, and in addition is sparse (relatively few 1's). An iterative, heuristic algorithm is used to fit layers $k = 1, 2, \ldots, K$ of the gene expression array.
The second, Plaid Modeling, [16] has been developed and seeks in addition to constrain each vector β_k to have entries either 0 or 1.

$$X_{i,j} = \mu_0 + \sum_{k=1}^{K} \mu_k \alpha_k \beta_k^T$$

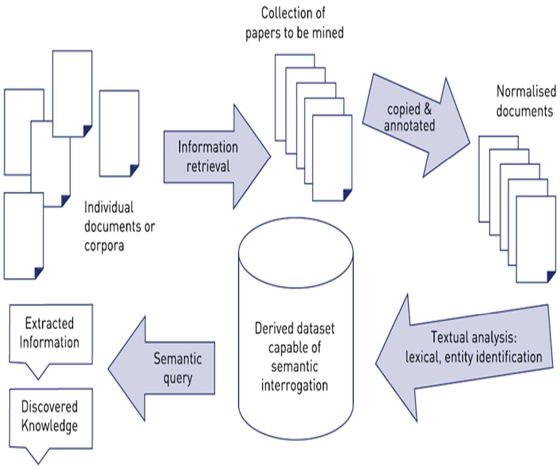
Again an iterative, heuristic algorithm is used to fit layers of the gene expression array, layers $k = 1, 2, \ldots, K$, one at a time. The two models differ in that plaid models have a complete 0-1 nature, giving a strict clustering form, while Gene Shaving clusters rows but does not constraint individual rows to have constant behavior.

5. FEATURE EXTRACTION TECHNIQUES [3]

Extraction of features involves reducing required amount of resources to describe a large set of data. While analyzing data one of the major problems stems from the number of variables involved. Analyzing with huge number of variables generally requires a large amount of memory and computation power or a classification algorithm which over-fits the training sample and generalizes poorly to new samples. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy.

We have many Feature extraction techniques for reducing the resources required to describe huge datasets. Whenever we have to use large datasets it requires good number of variables and large amount of memory and computation power and the classification algorithms which produce very poor results. Feature Extraction is a general term for constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy.

If no such expert knowledge is available to construct feature vectors from the attributes, general dimensionality reduction techniques may help. Some of the dimensionality reduction techniques are described below.


Collection of papers to be mined
Information retrieval
copied & annotated
Normalised documents
Individual documents or corpora
Derived dataset capable of semantic interrogation
Textual analysis: lexical, entity identification
Semantic query
Extracted Information
Discovered Knowledge

Feature 1 extraction Based on Knowledge Extraction

6. TYPES OF KERNEL METHODS

6.1 Principal component analysis

Principal component analysis (PCA) is a statistical process that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.

PCA uses an orthogonal transformation to convert a set of observations into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the original variables. The transformation takes care so that the first principal component has the highest variance possible and each succeeding component, in turn, has the highest variance. The principal components are orthogonal since they are the Eigen vectors of the covariance matrix which is symmetric.

PCA can be thought of as filling an n-dimensional ellipsoid to the data where each axis of the ellipsoid represents a principal component. If an axis of the ellipsoid is small, then the variance along that axis is also small and by ignoring that axis we lose very little information.

PCA is mathematically defined as an orthogonal linear transformation which transforms the data to a new co-ordinate system such that the largest variance comes in the first component and so on. The full principal components decomposition of X can therefore be given as

$$T = XW$$

where W is a p X p matrix whose columns are the Eigen vectors of $X^T X$.

Dimensionality reduction: The transformation T = XW maps a data vector x (I) from the original space of n variables into a new space of p variables which are uncorrelated over the data set. However not all the components need to be kept since the lower components do not have much variance? Keeping only L components, we can write the transformation as

$$T_L = XW_L$$

where the matrix T_L has now n rows and l columns. Such dimensionality reduction can be a very useful step for processing higher dimensional data sets. For example, selecting l=2 (i.e) keeping only the first two principal components finds the two-dimensional plane in which the data is most spread . So, if the data contains clusters these too may be spread out and therefore most visible in a two-dimensional diagram.

Singular value decomposition

The principal component transformation can also be associated with another matrix transformation, the Singular Value decomposition (SVD) of X:

$$X=U\Sigma W^T$$

Σ is n x p rectangular diagonal matrix of positive numbers σ (k) called the singular values of X. U is an n x n matrix, the columns of which are orthogonal unit vectors of length n called the left singular vectors of X and W is a p x p matrix, where columns are orthogonal unit vectors of length p called the right singular vectors of X. The matrix $X^T X = W\Sigma U\ ^T U\Sigma W^T$ and is equal to $W\Sigma^2 W^T$. It is obivous that the right singular vectors of X are equivalent to the Eigen vectors of $X^T X$ while the singular values σ(K) of X are equal to the square-roots of the Eigen values χ(k) of $X^T X$. Using the singular value decomposition, the transformation matrix can be written as:

$$T = XW$$

$= U\Sigma W^T W = U\Sigma.$

So each column of T is given by one of the left singular vectors multiplied by the corresponding singular value.

6.2 Kernel PCA: In multivariate statistics, kernel principal component analysis (kernel PCA) is an extension of principal component analysis (PCA) .

Semi definite embedding (SDE) or maximum variance unfolding (MVU) is an algorithm in computer science that uses semi definite programming to perform non-linear dimensionality reduction of high-dimensional input data.

6.3 Multifactor dimensionality reduction: Multifactor dimensionality reduction (MDR) is a data mining approach for detecting and characterizing combinations of attributes or independent variables that interact to influence a dependent class variable. MDR was designed specifically to identify interactions among discrete variables that influence a binary outcome and is considered a nonparametric alternative to traditional statistical methods such as logistic regression.

6.4 Multilinker subspace learning Multi linear subspace learning (MSL) aims to learn a specific small part of a large space of multidimensional objects having a particular desired property. It is a dimensionality reduction approach for finding a low-dimensional representation with certain preferred characteristics of high-dimensional tensor data through direct mapping, without going through vectorization. The term tensor in MSL refers to multidimensional arrays.

6.5 Isomap: is a nonlinear dimensionality reduction method, also one of several widely used low-dimensional embedding methods. Isomap is used for computing a quasi-isometric, low-dimensional embedding of a set of high-dimensional data points. This algorithm provides a simple method for estimating the intrinsic geometry of a data manifold based on a rough estimate of each data point's neighbors on the manifold. Isomap is highly efficient and generally applicable to a broad range of data sources and dimensionalities.

6.6 Multi linear PCA

It is a mathematical procedure that uses multiple orthogonal transformations to convert a set of multidimensional objects into another set of multidimensional objects of lower dimensions. There is one orthogonal (linear) transformation for each dimension (mode); hence *multilinear*. This transformation aims to capture as high a variance as possible, accounting for as much of the variability in the data as possible, subject to the constraint of mode-wise orthoganality.

6.7 Latent semantic analysis (**LSA**) is a technique in natural language processing, in particular in vectorial semantics, of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. LSA assumes that words that are close in meaning will occur in similar pieces of text. A matrix containing word counts per paragraph (rows represent unique words and columns represent each paragraph) is constructed from a large piece of text and a mathematical technique called singular value decomposition (SVD) is used to reduce the number of rows while preserving the similarity structure among columns. Words are then compared by taking the cosine of the angle between the two vectors (or the dot product between the normalizations of the two vectors) formed by any two rows. Values close to 1 represent very similar words while values close to 0 represent very dissimilar words.

6.8 Singular Value decomposition

Singular value decomposition is another way to arrange the data sets in order of decreasing variance. Let M be an mxn matrix. Singular value decomposition splits the matrix M as

$$M = U\Sigma V^T \qquad (1)$$

where U is m x m matrix, Σ is m x n diagonal matrix and V is n x n matrix. U and V are matrices with columns as the eigenvectors of MM^T and M^TM respectively. The columns of U and the columns of V form ortho-normal vectors. A non-negative real number σ is a singular value of M if $Mu = \sigma v$ and $M^Tv = \sigma u$. The vectors u and v are called left singular and right singular vectors respectively. With the singular value decomposition as shown in (1) we have

$M^T M = V\Sigma^T U^T U\Sigma V^T = V(\Sigma^T \Sigma)VT$

$MM^T = U\Sigma V^T V\Sigma^T U^T = U(\Sigma^T \Sigma)U^T$

The right-hand sides indicate Eigen value decomposition of $M^T M$ and MM^T respectively. The non-zero elements of Σ are the square roots of the Eigen values of $\Sigma^T \Sigma$ or $\Sigma\Sigma^T$

The elements of Σ are arranged in the decreasing order of variance.

Kernel Methods to reduce the dimensionality

The use of a kernel [5] function is an attractive computational short cut. If we wish to use this approach, there appears to be a need to first create a complicated feature space, and then work out what the inner product in that space would be, and finally find a direct method of computing that value in terms of the original inputs. In practice the approach taken is to define a kernel function directly, hence implicitly defining the feature space. There are different types of kernel methods.

7. CONCLUSION AND FUTURE WORK

Kernel methods are very popular dimensionality reduction techniques in normal data mining implementation and gives better results compared to other dimensionality reduction methods. I wish to implement similar kernel methods for big data.

REFERENCES

[1]. The Discriminate Analysis and Dimension Reduction Methods of High Dimension, Lan Fu,

[2] Big data solutions (http://mike2.openmethodology.org/wiki/Big_Data_Solution_Offering #Hadoop)

[3] http://en.wikipedia.org/wiki/Feature_extraction

[4] Donoho, L. (2000) High Dimensional Data Analysis: the Curses and Blessings of Dimensionality. Present Data AmericanMathematics Society Conference.

[5] “Dimensionality Reduction for Optimal Clustering In Data Mining” Ch. Raja Ramesh*, International Journal on Computer Science and Engineering (IJCSE), Vol. 3 No. 10 October 2011

[6] K. V. Mardia, J. T. Kent, J. M. Bibby. Multivariate analysis , London ; New York : Academic Press, 1979.

[7] Ripley, Brian D. (1996). Pattern recognition and neural networks. New York: Cambridge University Press, 1996.

[8] W.N. Venables, B.D. Ripley. Modern applied statistics with S-PLUS, 3rd ed. New York:Springer, 1999.

[9] Michael W. Berry and Susan T. Dumais, and Gavin W. O’Brien (1995) Using Linear Algebra for Intelligent Information Retrieval. SIAM Review 37:4, pp. 573-595.

[10] Declan Butler (1999) Computing 2010: from black holes to biology Nature, 402 pp C67-70. Dec 2, 1999.

[11] Cand`es, E. and Donoho, D. (1999). Ridgelets: the key to high-dimensional intermittency?. Phil. Trans. R. Soc. Lond. A. 357 2495-2509.

[12] Bell, A.J. and Sejnowski, T.J. (1995) An information-maximization approach to blind separation and blind deconvolution. Neural Computation 7 1129-1159.

[13] Cardoso, J.F. and Souloumiac, A. (1993) Blind Beamforming for non-Gaussian Signals. IEE Proceedings-F. 140 352-370.

[14] Comon, P. (1994) Independent Component Analysis, a new concept? Signal Processing 36 287-314

[15] Kari Karhunen (1947) Uber Lineare Methoden in Wahrscheinlichkeitsrechnung. Ann. Acad. Sci. Fenn. 37.

[16] Lazzeroni, Laura, and Owen, Arthur. Plaid Models for Gene Expression Data Technical Report, Department of Statistics, Stanford University, March, 2000.

[17] Michel Lo`eve. Fonctions aleatoires de second ordre. C. R. Acad. Sci. 220 (1945), 222(1946); Rev. Sci. 83 (1945), 84 (1946).

[18] Hotelling, H. (1933) Analysis of complex statistical variables into principal components. Journel of Educational Psychology 24, 417-441, 498-520.

[19] Murtaugh, F., Starck, J.-L., Berry, M.W. (2000) Overcoming the curse of dimensionality in clustering by means of the wavelet transform. *Computer Journal* 43, pp. 107-120.

[20] Jerome Friedman, Trevor Hastie and Robert Tibshirani (2001) *ELEMENTS OF STATISTICAL LEARNING: Prediction, Inference and Data Mining* Springer: New York

[21] Tzyy-Ping Jung, Scott Makeig, Colin Humphries, Te-Won Lee, Martin Mckeown, VicenteIragui, Terrence Sejnowksi (2000) Removing Electroencephalographic Artifacts by blind source separation. *Psychophysiology* **37** 163-178.

[22] Trevor Hastie, Robert Tibshirani, Michael Eisen, Pat Brown, Doug Ross, Uwe Scherf, John Weinstein, Ash Alizadeh, Louis Staudt, David Botstein, Gene Shaving: a New

Class of Clustering Methods for Expression Arrays, Technical Report, Department of Statistics, Stanford University, January, 2000.