Dimensionality Reduction and Nearest Neighbor Search in Large and High-Dimensional Data

A thesis submitted during 2020 to the University of Hyderabad in partial fulfillment of the award of Ph.D. degree in Computer Science

by

RAGHUNADH PASUNURI

SCHOOL OF COMPUTER AND INFORMATION SCIENCES

UNIVERSITY OF HYDERABAD

(P.O.) CENTRAL UNIVERSITY

HYDERABAD - 500 046, INDIA

2020

CERTIFICATE

This is to certify that the thesis entitled "**Dimensionality Reduction and Nearest Neighbor Search in Large and High-dimensional Data**" submitted by **Raghunadh Pasunuri** bearing Reg. No. **10MCPC08** in partial fulfillment of the requirements for the award of **Doctor of Philosophy** in **Computer Science** is a bonafide work carried out by him under my supervision and guidance.

This thesis is free from plagiarism and has not been submitted previously in part or in full to this or any other University or Institution for award of any degree or diploma.

Parts of this thesis have been published in the following Journals and Conferences:

1. *Second International Conference on Contemporary Computing and Informatics (IC3I) 2016, IEEE Xplore (SCOPUS)*
2. *International Journal of Applied Engineering Research (IJAER) Vol 10, No. 86, 2015. (SCOPUS)*
3. *Fourth International Conference on Harmony Search, Soft Computing and Applications (ICHSA) 2018, Springer (SCOPUS)*
4. *Second International Conference on Computing and Communication (IC3), 2018, Springer (SCOPUS)*
5. *International Conference on Communication and Intelligent Systems (ICCIS-2019), Springer, LNNS, (SCOPUS) (Presented)*

Further, the student has passed the following courses towards fulfillment of the coursework requirement for Ph.D.

	Course Code	Name	Credits	Result
1	CS801	Data Structures and Algorithms	3	Pass
2	CS802	Operating Systems and Programming	3	Pass
3	AI851	Trends in Soft Computing	3	Pass
4	AI810	Metaheuristic Techniques	3	Pass

Supervisor
Prof. V. China Venkaiah
School of Computer and Information Sciences
University of Hyderabad

Dean
School of Computer and Information Sciences
University of Hyderabad

DECLARATION

I, **Raghunadh Pasunuri** , hereby declare that this thesis entitled "**Dimensionality Reduction and Nearest Neighbor Search in Large and High-Dimensional Data** " submitted by me under the guidance and supervision of **Prof. V. China Venkaiah** is a bonafide research work. I also declare that it has not been submitted previously in part or in full to this University or any other University or Institution for the award of any degree or diploma. I here by agree that my thesis can be deposited in Shodganga / INFLIBNET.

A report on plagiarism statistics from the University Librarian is enclosed.

Date:

Name: **Raghunadh Pasunuri**

Signature of the Student:
Regd. No. **10MCPC08**

//Countersigned//

Signature of the Supervisor:

ACKNOWLEDGEMENTS

I thank University of Hyderabad for providing me this opportunity to carry out my Ph.D. work successfully. During my Ph.D. studies, there are many peopole whose guidance, support, encouragement and sacrifice has made me remember them and indebted for my whole life. I take the previlage to express my sincere thanks and gratitude to all these people. Firstly, I would like to thank my supervisor **Prof. Vadlamudi China Venkaiah** for his invaluable guidance and encouragement throghout my research work. His guidance and support from preliminary to concluding level enabled me to develop an understanding of the subject. I am thankful to him for his patience and constant encouragement in motivating me with the words of hope throughout this work. It is a great honour to thank my doctoral review comittee members Prof. Chakravarthy Bhagavati and Prof. S. Durga Bhavani for their valuable suggestions during my research tenure. My interactions with them have been of great help in defining my reasearch goals and in identifying ways to achieve them. My sincere thanks to Prof. Arun Agarwal Pro-VC-I (Former Dean, SCIS), Prof. V. Krishna (former Controller of Examinations, UoH) for helping me to complete my Ph.D. I thank Prof. Chakravarthy Bhagvati, Dean, School of Computer and Information Sciences for the academic support and facilities provided to carry out the research work.

I thank Prof. K. Narayana Murthy (Ex-Dean of SCIS) for supporting me during my Ph.D. tenure. I thank Dr. T. Sobha Rani, Dr. B. Wilson Naik, Prof. Vineet Padmanabhan Nair, Dr. Naveen Nekuri, Dr. Anupama P, Prof. Bapi Raju Surampudi and all the faculty members and staff of School of Computer and Information Sciences for thier helpful suggestions and encouragement. I would like to acknowledge the financial support of University Grants Commission (UGC), New Delhi for my doctoral studies.

I am vey much indebted to my friends/soulmates/brothers Dr. Uday Bhanu, Dr. Mendem Bapuji, Bugga Rajender and Dr. Harshavardhan, who are always with me to share my happiness and to encourage me in grief/sorrow and helped me a lot to finish my Ph.D., and I thank Dr. Nandi Chinni, his brotherhood, affection and help resulted in completion of my Ph.D.

I extend my thanks to co-researchers Bheekya, Govind, Pandu, Anil, Abhimanyu, Abdul Basit, Rajesh, Rakesh, Anil G.R., Dr. Ramesh Babu, Dr. Jagadeeswar Rao, Bobby Ramesh, Umesh, Narsimhulu, for their support during the Ph.D. tenure.

I thank my teachers Dr. Jayadev Gyani, Dr. C. Srinivasa Kumar, Mr. Agarwal, who are my main motivation to think of pursuing research career. I thank my friends Rajaram Jatothu, Satyanarayana Vollala, Satish, Sheik Indivar, Ranjith, Srinivas Rao, Sudhakar, Sameer, Kumar, Mamidi Srinivas, Bhaskar and Satyam for their comradery during my M.Tech. studies.

I am grateful to my collegues Prof. Venkat Rama Raju, Srinivas Sachdev, Ravinder Thammadi, Venkat, Venkanna, Srinivas Sura, Praveen, Sridevi, Rama Devi, Pandu Naik and Mahesh, Mamtha and Aparna for their cooperation in all the times.

I am very indebted to my B.Tech. friends Shiva, Ravi, Vivek, Naresh, Suman, Srikanth, Murali, Prashanth, Nishanth, Hathiram, Nagaraju, Pradeep Reddy, Vishwanath Reddy, Rahul(IT), who helped me a lot, without their help and support I couldn't complete my B.Tech. course and I thank my teachers Tarun, Rajanna.

I thank Srinath, Prasad, Chandra Mouli, Rajalingam, who taught me at Intermediate. I feel happy to express my gratitude to Srinu , Shiva, Ravi, Kiran, Bikshu, Kumar who are always with me. I thank the Scholars Tutorial teachers: Nagaiah, Chary, Karim and Venu, who taught me at High school level.

I would like to express my sincere thanks to my teachers K. Mohan Rao, Jayanth, Ram Chandar Rao, Chary, Shathrugna (Acharya), Muralidhar Rao, Kumara Swamy, Mrs. Kaalindi, Tirupathi Reddy, Venkatraman who taught me at school level and their motivation and inspiration is invaluable and I am indebted to them for my whole life.

I extend my special thanks to Dr. Sreedhar Bhukya, for his continuous support and encouragement during my Ph.D. tenure. I express my thanks to my close friends at UoH: Dr.Vijender Reddy, Dr. Ilaiah, Anil, Dr. Srinivas Naik, Praveen, Kedarinath, Hanumanthu, Dr. Sachindanand.

In this journey, my family is always with me, I am indebted to my parents (late) Veeraswamy , Varamma for their love and trust in me, my brothers Dr. Pasunuri Ravinder, Dr. Pasunuri Ramesh, Pasunuri Raja.

I thank my wife Swapna, for supporting me in all the times throughout my Ph.D. as well as in life, and I wont forget my son Siddharth, who missed his joyful time due to my research work, understanding me in crucial times, and his presence made me joyful.

- Raghunadh Pasunuri

ABSTRACT

Nearest neighbor search is an extensively used technique in pattern recognition, object recognition, Content Based Image Retrival (CBIR), text classification, Recommender Systems. The Nearest Neighbor (NN) search problem is defined as : Given a set of N points in space R^d, and a query q, then we need to find closest points of q in the set. The reason for its extensive use is in its simplicity. NN techniques can be categorized into : structure less and structure-based. Cover and Hart (1967) proposed k-nearest neighbor (kNN) rule in which the value of k plays important role in finding the nearest neighbor. The k value tells us how many nearest neighbors are to be considered and eventually to determine the class of a sample data point considered (Classification). The same problem is referred to as post office problem in Knuth (1998). In this thesis work, we mainly studied nearest neighbor search problem and dimensionality reduction problems for analyzing large scale data in view of high dimensionality.

The nearest neighbor search problem can be defined more formally in the following way: Assume that X is a data set with N points $X = X_i, i = 1, ..., N$. Then given a query point q, and a distance metric $dist(\cdot)$, find the q's nearest neighbor X_{NN} in X, i.e., $dist(X_{NN}, q) \leq dist(X_i, q), i = 1, ..., N$. Like in most statistics or machine learning research, here $X = X_i, i = 1, ..., N$ and q are assumed to i.i.d. samples. Note that in the following chapters, sometimes X also represents the data matrix consisting of all data points, X_i represents the i^{th} data point, which is X's i^{th} column.

We proposed two methods for partitioning large scale data, to reduce the cost of searching and retrieving k-nearest neighbors of a given query. This partitioning has been studied by many researchers in the past. However, we come up with a new and effective partitioning strategies: one, which is based on the farthest reference point (minmax as pivot), and the second one is based on the set of weighted reference points (pivots). The ultimate goal of these partitioning methods is to reduce the search space as much as possible, which is done by computing the distance between the data points (in the database) and reference point (pivot). This distance will guide

us to eliminate the inappppropriate search space. The effectiveness of the proposed methods is demonstrated by conducting sufficient experimental study.

Another contribution of this work is to study different dimensionality reduction techniques for processing high-dimensional data to find the hidden patterns in it. For improving this pattern recognition task, one might need to reduce the dimensionality of data down to a sufficient and comfortably small. In this study, this task is achieved by proposing a variant of IRP-K-means algorithm, and also proposed two hybrid methods, which combines Principal Component Analysis (PCA), Random Projection (RP) with K-means Clustering. By integrating these dimensionality reduction methods, the quality of clustering is improved in the reduced feature space. Various data sets both low-dimensional and high-dimensional, are used for experimental study.

Finally, we proposed a new method of dimensionality reduction, which projects the data to a lower-dimensional space with the help of the projection matrix that is constructed by taking a random sample from the data and subsequently the most significant eigen vectors of the resulting correlation matrix. A novel feature of the proposed method is that the projection matrix, unlike random projection matrix, is dependent on the data and hence it is expected to preserve the pair-wise distances more accurately in the reduced space. It is observed in our experiments that only 10% of the data is enough to give good results. We have tested our proposed method on high-dimensional as well as on low-dimensional real world data sets, and the experimental results better advocate the use of our proposed method. We also tested our method on the given data sets by varying the reduced dimension (D), and from the results we conclude that the RP-based dimension reduction method is producing worse results when the D is approaching the original dimension, whereas the proposed method is performing well and also improving when reduced dimension (D) reaches original dimensionality of the data.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

NNS	Nearest Neighbor Search
MSE	Mean Squared Error
DR	Dimensionality Reduction
RP	Random Projection
PCA	Principal Component Analysis
k-NN	k-Nearest Neighbor
k-Means	k-means clustering
LD	Low-dimensional
HD	High-dimensional
SVD	Singualr Value Decomposition
CoD	Curse of Dimensionality
ZINC	ZINC Is Not Commercial
CS	Compressive Sensing
LDA	Linear Discriminant Analysis
MDS	Multi Dimensional Scaling
LLE	Locally Linear Embedding

NOTATION

X	Input Data set
N	Number of points in the data set
d	Dimensionality of the input data
D	reduced Dimension/target dimension
R^d	Original Space
R^D	Reduced/Projected Feature Space
P	Projection Matrix
R	Set of Reference Points
l	No of Reference Points
r_i	A Reference point in R
W	Set of Weights
i, j, k	Indices
$dist(\cdot)$	distance fundtion
d_i	i^{th} distance value
$dist$	set of distances for all points in X
$D_{A,B}$	Distance between the points A,B
q	Query object/point
X_i	i^{th} point in data set X

CHAPTER 1

Introduction

Data Mining process acquires, processes and models the data with an aim to uncover or discover hidden knowledge that is present in the data itself. In todays voluminous data era, we mainly encounter large volumes of data to be analysed. This increase can be found mainly in two aspects:

One is database size i.e. number of observations acquired in the intial step of applications such as weather forecasting, customer market analysis, social networks, gene expression analysis, information retrieval. These applications generally comes under large-scale.

Second aspect is dimensionality (no. of features/attributes size) which defines the characteristics of samples. High dimensionality is very common in many of the applications today. Examples include text and image retrieval, recommender systems, gene expression analysis which are high-dimensional in nature.

This high-dimensional data to processed and analyzed to find patterns present in the data, which are useful in knowledge discovery and decision making process.

1.1 Nearest Neighbor Search

Nearest neighbor search (NNS) is an extensively used technique in pattern recognition, object recognition, Content Based Image Retrival (CBIR), text classification, Recommender Systems etc. NNS problem can be stated as : Given a N points set in R^d space and a query q , then we need to find closest points of q in the point set. The reason for its extensive use is its simplicity. NN techniques can be categorized into : structure less and structure-based. Cover and Hart in (Cover and Hart (1967)) proposed a kNN rule where k value is crucial in finding the nearest neighbors, which tells how many nearest neighbors are to be considered to classifiy the sample data point. The same problem is referred to as post office problem by Knuth in (Knuth

(1998)). In this thesis, we will study NNS problem and dimension reduction problems in view of large and high dimensionality of data.

The NNS problem can be defined more formally in the following way:
Assume that X is a given input data set with N points $X = X_i, i = 1,2,3,...,N$. Then given a query point q, and a distance metric $dist(\cdot)$, then we need to retrieve neighbor of q, X_{NN} in X, such that $dist(X_{NN}, q) \leq dist(X_i, q), i = 1,2,3,...,N$. Like in most statistics or machine learning research, here X and q are assumed to be i.i.d. samples. Note that in the following chapters, sometimes X also represents the data matrix which contains all data points, X_i represents the i-th data point, which is i^{th} column of X.

1.2 High-dimensional Applications

The normal behaviour of the distances shown in the low-dimensional setup is different from that in high-dimensional scenarios. The high-dimensionality of data exhibits some strange behaviour. As the dimensionality increases, the cost of finding distances between the points, storage cost and the cost of handling the data also increases. The high dimensionality introduces two things: *distance concentration*: where we cannot differentiate between the nearest and farthest point. The second one is concentration of cosine similarity in the information retrieval field. In Radovanovic et al., 2010 (Radovanovic et al. (2010)), they have shown that as the dimensionality increases then cosine similarity and standard deviation converges to constant and zero, respectively. Thus the realtive contrast converges to 0 and the cosine similarity is said to concentrate. We can trace these issues to the commonly used term *Curse of Dimensionality* (CoD). In literature, many methods proposed to deal with it, Dimensionality Reduction (DR) is the first and foremost method of them. The goal of any DR method is to represent or embed an HD data in a LD space, then doing any processing in that low dimensional subspace instead of original high dimensional data. This will save time, space, computations, and resources etc.

1.3 Motivations

Transfomation of high-dimension (HD) data into a low-dimensional (LD) one (subspace) is regarded as Dimensionality reduction (DR). Usually the reduced data dimension resembles its intrinsic dimension. According to (Fukunaga , 1990), intrinsic dimension is defined as the lowest set of parameters that are required to show the data properties. Dimensionality Reduction is inevitable in various domains, which is used as a pre-processing step in classification task, visualization of HD data and compression of HD data, by diminishing the curse of dimensionality and other unwanted properties of HD spaces (Jimenez et al. (1997)).

In literature, we can find various techniques that are intend to reduce the dimensionality. See Agrafiotis (2003), Baudat and Anouar (2000), Belkin and Niyogi (2002), Brand (2004), Donoho and Grimes (2005), He and Niyogi (2004), Hinton and Roweis (2002), Hinton and Salakhutdinov (2006), Lafon and Lee (2006), Roweis and Saul (2000), Sha and Saul (2005), Scholkopf et al. (1998), Tenenbaum et al. (2000), Teh and Roweis (2002), Verbeek (2006), Weinberger et al. (2005), Zhang and Zha (2004), Zhang et al. (2007) for some of these techniques. PCA, LDA, MDS and many other techniques present in the above list. Many of these techniques are suitable for non-linear data, which we often find in the real world, (Duda et al. (2001)).

Our main motivation for the work done in this thesis is two-fold: (a) search space reduction for nearest neighbor search in large and high-dimensional data by implemeting space partitioning methods, (b) reducing cost of computation for processing high-dimensional data. Because of the challenges that are posed by the high-dimensionality of the data, we have been searching for the better methods which can reduce the additional cost that is incurred in dealing with high-dimensional data. Especially we have studied a data-independent Random Projection (RP) dimension reduction method, along with its counterpart data-dependent dimension reduction methods (like PCA) and we conclude with some experimental observations about the suitability of these methods on different data sets. For the performance evaluation purpose, we first apply a dimensionality reduction method on the original (HD)data and apply clustering (like K-means) on the resultant feature (LD) space. This methodology has been followed in the chapters which deals with DR problem.

1.4 Contributions

1.4.1 Proximity-based Nearest Neighbor Search

In this chapter, we studied the nearest neighbor search problem i.e. extracting the identical objects to a query. In this we proposed two proximity-based data partitioning methods which improves the similarity search. The first proposed method works by dividing the data by using a reference (pivot) point, into different bins (partitions). This facilitates to return nearest neighbors of a given query by searching in one of these bins and avoiding the remaining bins from searching. The second method works by taking multiple reference points which are weighted according to the distance from the points in the dataset, and points are segregated into multiple bins/groups. These segregated groups are useful while finding the neighbors of a query. The major difference between these two methods is, the first method does data partitioning based on only one reference point, whereas the second method performs partitioning of data based on multiple weighted reference points. Both the methods aims at search space reduction and eventually saving computational time.

In fisrt part of this chapter, we have presented a metric-based data partitioning method by using a pivot point from the database which improves nearest neighbor search quality. The method works by taking a reference point (MinMax) from the given input data, and then finds the distances of all the points from this reference point. Then by taking the minimum and maximum distance values, the distance range is identified, this distance range is divided into equal bins. Now the distance of the points from the reference point are compared with that of the range and place those points in the corresponding bin that satisfies the range criterion. This way, the construction phase of the proposed method partitions the data into subgroups. Finding neighbors of q is the main goal of search phase. For this, first we calculate distance from query to reference point, and then we check this value in distance range (IndexSpace) to find the partition in whcih its neighbors are present. We have experimented on various data sets. Infact our method (MinMax) is performing well when compared to BO-Heuristic (Bozkaya and Ozsoyoglu (1999)), and a large portion of the search space is reduced. So, it is effective and efficient in retrieving the

nearest neighbors for a given query with less computational requirement. The effectiveness of the proposed pivot point is validated by plotting the distance distribution along with the distance distribution of BO-Heuristic. The proposed pivot distance distribution is flatter than the BO-Heuristic. This indicates that while querying a lot of points are going to be eliminated from the search space, whereas for BO-Heuristic, it is not possible.

In the second part of this chapter, we proposed a variant of the proximity-based data space partitioning method, which is called WSRP method. This speeds up the nearest neighbor search by reducing the search space. This method works by taking multiple weighted reference points for partitioning the data into groups and returns the nearest neighbors of a query. Reference points weighting is done as per the distance of a point from pivot (reference point) set. WSRP is effective and efficient in retrieving the nearest neighbors for a given query with less search space. The average precision of WSRP (proposed method) is compared with two other pivot-based methods (BO-Heuristic and MinMax), and our method is outperforming those methods.

1.4.2 Random Projection for Dimensionality Reduction and Clustering in High-dimensional Data

In this chapter, we have studied iterative random projections and combined the dimensionality reduction methods with clustering methods for improving the quality of the clustering solutions in reduced space. In this chapter, we propose a variant of IRP K-means algorithm, in which the dimension is gradually decreased regularly in iterations; thereby preserving the inter-point distances efficiently. This has been proved empirically by large number of experiments. The proposed method is compared with the Single Random Projection (RP), IRP K-means (IRP) methods. Compared to these two methods, our proposed method is giving best results for the given HD datasets. In the second part of this chapter, we proposed two hybrid algorithms by combining different dimensionality reduction methods (PCA, RP) with K-means clustering for better clustering in HD and LD data. We have observed that the K-means is giving good performance when combined with PCA than the normal K-means. The details of the proposed hybrid algorithms are as follows: one com-

bines PCA with K-means and the second one combines PCA and RP with K-means, third one combines RP with PCA then K-means is applied. PCA when combined with Random Projection and RP combined with PCA produces good quality clusters in the reduced dimensional space. Our proposed algorithm works by combining PCA with RP and also RP first then PCA (for DR), then performs clustering on reduced data.

A comparative analysis is done with simple K-means, PCA reduced K-means algorithms on 12 bench mark datasets, by taking k-means objective function as performance measure. The results of experiments reveal that the proposed PCA-Kmeans and PCA+RP-Kmeans and RP+PCA-K-means are outperforming the classic K-means on both low and HD datasets.

1.4.3 A Computationally Efficient Data-Dependent Projection for Dimensionality Reduction

A new projection was proposed, which maps HD data onto a LD space by using a projection matrix. This projection is constructed by taking a random sample from the data, construct the covariance of sample, then take the most significant eigen vectors of this covariance matrix. A novel feature of the proposed method is that the projection matrix, unlike random projection matrix, is dependent on the data and hence it is expected to preserve the pair-wise distances more accurately in the reduced space. It is observed in our experiments that only 10% of the data is enough to give good results. We have tested our method on high-dimensional as well as on low-dimensional datasets, and the superiority of proposed projection is clear from the empirical results. Proposed projection is achieving better pair-distance preservation than random projection. We also tested our projection on the given data sets by varying the reduced dimension (D), and from the results we conclude that the RP-based dimension reduction method is producing inferior results when the D is approaching the original dimension, where as the proposed method is performing well and also improving when D is reached original dimension.

1.5 Contents of the Thesis

The contents of thesis are: Chapter 1 gives an introduction to Nearest Neighbor Search and High-Dimensionality and the Motivations for this research. Basics of the subject matter related to the problems studied here and the Related Work is presented in Chapter 2. In Chapter 3, we discuss about the Proximity-based Nearest Neighbor Search Algorithms. Chapter 4 deals with the Randomized Algorithms for Dimensionality Reduction Problem in the context of High-dimensionality. In Chapter 5, we present a data-driven, deterministic construction of Projection matrix.This is an alternative approach we proposed to its counterpart Random Projection method. Chapter 6 concludes the thesis with future research guidelines.

CHAPTER 2

Preliminary Study and Related Work

2.1 Similarity/Distance Measures

Distance or similarity measures play a vital role in solving many pattern recognition problems such as clustering, classification and information retrieval. The basic Euclidean distance is not suitable for all types of data. Various distance/similarity measures have been proposed for a variety of applications in the scientific literature. These measures are designed for different subject areas such as biology, anthropology, chemistry, ecology, computer science, information theory, mathematics, geology, psychology, physics, statistics etc.

We find many studies in the literature. These are aimed at finding the appropriate measures among the vast list, due to the importance of similarity/distance measure in many tasks such as classification, clustering and information retrieval (Duda et al. (2001)).

Distance is a quantitative degree, which tells us about any two objects , how far apart to each other. Distance is also called dissimilarity. The distance measure satisfy the metric properties then it is called as a *metric*, and the other non-metric distance measures are called *divergences*. Similarity measures are also called similarity coefficients, and the similarity and dissimilarity both comes under one term that is proximity.

The data representation is important in selecting a suitable proximity measure.

Distance function between two vectors a and b is a function $dist(a,b)$ which defines the distacne between both vectors as a non-negative real number. This function can be called as a metric if it satisfies the following propoerties (Deza and Deza (2009)):

- **Non-negativity:** Distance between any two vectors is always positive.

$$dist(a,b) \geq 0 \tag{2.1}$$

- **Identity:** Distance becomes zero only when a = b.

$$dist(a,b) = 0 \iff a = b \tag{2.2}$$

- **Symmetry:** Distance from a to b is same as the distance from b to a.

$$dist(a,b) == dist(b,a) \tag{2.3}$$

- **Triangular inequality:** In addition to a, b, if a third point c exists, then distance from a to b is always less than or equal to the sum of the distance from a to c, from b to c.

$$dist(a,b) \leq dist(a,c) + dist(c,b) \tag{2.4}$$

If a distance coefficient which satisfies the first three properties, then it is called a pseudometric. A distance coefficient which doesnot satisfy the triangular inequality property, then it is called a non-metric (Kamichety et al. (2002)).

When the distance value is in the range [0,1], then the corresponding similarity measure $sim(a,b)$ is given by:

$$sim(a,b) = 1 - dist(a,b) \tag{2.5}$$

Cha S-H (Cha S-H (2007)) and Prasath et. al, (Prasath et al. (2017)) presented a vast list of distance measures which belongs to eight major families, these constitute a total of 54 distance measures.

2.2 Curse of Dimensionality (CoD)

For the first time, the term CoD was used by Bellman R (1961), in area of spaces by connecting it to the difficulty of optimization by exhaustive enumeration on product. According to Bellman R (1961) "*considering a Cartesian grid of spacing* 1/10 *on the unit cube in* 10 *dimensions, the number of points equals* 10^{10} *; for a* 20*-dimensional cube, the number of points further increases to* 10^{20}. *Here is the Bellman's interpretation: If we want to optimize a function over a continuous domain of a few dozen variables by exhaustively searching a discrete search space defined by a crude discretiza-*

tion, one could easily be faced with the problem of making tens of trillions of evaluations of the function. In other words, CoD refers to the fact that without any assumptions of simplification, the number of data samples required to estimate a function of multiple variables to a given accuracy (low-variance) on a given domain, grows exponentially with the number of dimensions. Because, data contains very less observations, HD spaces are naturally sparse". This reality or the fact, is responsible for the curse of dimensionality. It is often called the "empty space phenomenon ". CoD and empty space phenomenon causes surprising behaviour/properties to HD spaces, Scott and Thompson (1983), Lee and Verleysen (2007). Following are the list of problems that are result of the dimensionality curse:

- Hypervolume of cubes, spheres and a thin spherical shell
- Tail probability of isotropic Gaussian distributions
- Concentration of norms and distances
- Diagonal of a hypercube

Refer to Lee and Verleysen (2007) for an in detail description of the above problems.

2.3 Dimensionality Reduction

In real world, many objects such as images, speech signals, hypersprectal images, gene expression information, text documents, fingerprints and handwritten characters and numbers, etc. be represented with only high-dimensional data. We have to analyze these data and process them to get the required useful information from them. For example, identifying a person's fingerprint, finding relevant documents on the Internet with keywords, to search for obscured patterns in imagery, to outline the objects from videos and the relevant information. In order to achieve the relavant tasks, one has to develop a system that can process the entire data. But, owing to the HD of the data, these systems may not perform the job as desired and these systems may be complicated, unstable and infeasible. Generally, many systems work well with low-dimensional data. When the dimensionality of the data exceeds, one cannot process or handle the data correctly. Therefore, reducing the dimensionality

is a must to process HD data.

In most of the modern machine learning tasks, the number of available data features are generally very large - often larger than the number of available data patterns.These vast features may contain noise and redundancy, which is the reason for the difficulty of extracting meaningful information from the data (Carreira-Perpinan (2001)). These extra features slow the learning algorithms and curse of dimensionality can lead to overfitting or may increase the chance that certain optimization algorithms get stuck at local minima (Ding et al. (2002)). To address these problems, dimensionality reduction has become an important tool in machine learning. The main aim and goal of the tool is to lessen the number of dimensions or features of a dataset before running a learning algorithm. In principle, eliminating or reducing features only lead to a minor reduction in the algorithm's effectiveness(or, by avoiding overfitting and local minima), actually improves learning performance (Carreira-Perpinan (2001)).

Dimensionality reduction techniques are classified into two main categories (Liu and Motoda (1998)):

Feature Selection: Selecting a small set of features from the original database is called feature selection. The selection follows some measure to cull out the redundant features. These measures include, Laplacian score (Fisher score), Feature variance. Features may be randomly sampled or highest ranked features are selected, based on probability proportional to importance.

Feature Extraction: Instead of picking up a subset of features from original feature set (done in Feature selection), feature extraction generates a completely novel features set by transforming the original ones. Principal Component Analysis (PCA) is one such example, and nonlinear methods are also common as well.

These two methods of DR methods are popular and a more techniques have been studied theoritically and empirically. Theoritical analysis is often inspired by the observed effectiveness in practice and, in turn, often inspires new algorithms that work well experimentally.

Now, we give details on some of the dimensionality reduction techniques which are used in this thesis work.

2.4 Principal Component Analysis (PCA)

It is a multivariate data analysis tools, which was developed by Pearson in 1901 (Pearson (1901)) and H. Hotelling in 1933 (Hotelling (1933)), and Jolliffe (2002) is the latest reference. It is used to reduce the data from HD to LD one. The applications of PCA include, Data Compression, Data Visualization, Feature Extraction and so on.

PCA reduces the original high dimensionality to a much smaller, uncorrelated feature set with minimum information loss. The reduced or the transformed features are called principal components.

According to Shlens (2005), this transformation can be defined by:

$$Y = XP \tag{2.6}$$

Where $P_{d\times D}$ represents projection matrix containing D eigen vectors of corresponding highest eigen values, $X_{N\times d}$ is data matrix which was mean centered and Y is resulting projected data.

As per Hotelling (Hotelling (1933)), PCA is such that the D principal axes, for given set of vectors $X_i, i = 1,2,...N$, are those orthogonal axes onto which the variance retained under projection is maximum. The derivation of PCA following this definition is as follows (Ali Ghodsi (2006)): Let the centered observations $X_i, i = 1,2,...,N$ be stacked into columns of an $d \times N$ matrix X, where d is the dimensionality of the observations. Choose the first principal component U_1 as a linear combination of the vectors in X so that it captures the maximum variance. That is, choose

$$U_1 = W^T X \tag{2.7}$$

so that $var(U_1) = var(W^T X)$ is maximum, where $W^T = [w_1, w_2, ..., w_d]$. So choosing U_1 is equivalent to determining the weight vector W^T so that the variance is maximum. Since $var(U_1) = var(W^T X)$ can be made arbitrarily large by choosing larger values for the components of W, it follows that the determination of the optimum weight vector requires normalization of its magnitude. Summarizing, we have that

the choice of the first principal component with maximum variance is equivalent to determining the weight vector W as a solution to the following optimization problem:

$$Maximize \qquad W^T SW \tag{2.8}$$

$$Subject\ to \qquad W^T W = 1 \tag{2.9}$$

where S is the $d \times d$ sample covariance matrix of X. Introducing Lagrange multiplier, the problem becomes

$$Maximize \qquad L(W,\alpha) = W^T SW - \alpha(W^T W - 1) \tag{2.10}$$

Differentiating with respect to W and equating it to zero, we have

$$SW = \alpha W. \tag{2.11}$$

That is, the eigen vectors of the covariance matrix are the extreme points of the $L(W,\alpha)$. Premultiplying Eq.2.11 both sides by W^T, we have

$$W^T SW = W^T \alpha W = \alpha W^T W = \alpha \tag{2.12}$$

A characteristic of PCA is that the projection $\hat{x}_i$ of the observations $x_i, i = 1,2,...,N$ onto the subspace spanned by the principal components minimizes the squared reconstruction error, $\sum_{i=1}^{N} \|x_i - \hat{x}_i\|^2$.

Principal components can be obtained by finding the SVD of X. In fact the D principal components can be determined from the first D columns of the left singular matrix of X, i.e. from the first D columns of U of $X = U \sum V^T$.

Variations to PCA that cater to the special requirements of various application scenarios are proposed in the literature. These include Dual PCA, Kernel PCA, Metric Multidimensional Scaling (MDS), Semi definite Embedding (SDE), etc. Dual PCA is a variation to PCA that exploits the SVD structure and makes the technique feasible

or, the least, amortizes the amount of computation required, for the case where the dimensionality (d) is very large compared to N, the number of observations. Kernel PCA handles nonlinear DR problems. MDS maps the original HD space to a LD space while attempting to preserve pairwise distances. That is, MDS constructs a configuration of N points in Euclidean space by using information about the distances between the N observations. MDS is identical to dual PCA if the distances are measured using Euclidean metric. Semi Definite Embedding (SDE) is a variation of Kernel PCA in that the kernel is learned from the data as against choosing apriori a kernel function as in kernel PCA. Algorithmic descriptions of PCA are present in Ali Ghodsi (2006).

One can refer to Jolliffe (2002),Shlens (2005),Ali Ghodsi (2006) for more detailed information on PCA.

2.5 Random Projection (RP)

It is a linear dimensionality reduction method which is based on matrix multiplication. In this, the original HD data is represented in LD embedding, by using a matrix for projection, which satisfies certain properties from JL Lemma (Johnson and Lindenstrauss Lemma). RP can preserve the inter-point distances approximately (Fradkin and Madigan (2003), Bingham and Mannila (2001)).

RP maps the original d-dimensioanl data onto a subspace of dimensionality D, where D≪d. For this, it uses a random orthogonal matrix P of size $d \times D$. The orthogonal matrix P_{dXD} contains columns of length one (unit length). RP is defined as:

$$X^{RP}_{N\times D} = X_{N\times d} P_{d\times D} \tag{2.13}$$

Johnson-Lindenstrauss (JL) lemma is the basis for RP. Johnson-Lindenstrauss lemma (Johnson and Lindenstrauss (1984)) states that if vector space with N points and d dimensionality, is randomly projected to a D-dimensional subspace, then distance (Euclidean) retained approximately. JL-Lemma details can be found in Dasgupta and Anupam Gupta (2003). The statement of the JL lemma is as follows:

Theorem 1 (JL Lemma)**.** *For any* $0 < \epsilon < 1$ *and* N *is size of database, let* D *be the*

reduced dimension (JL bound), such that

$$D \geq 4\left(\frac{\epsilon^2}{2} - \frac{\epsilon^3}{3}\right)^{-1} \ln N.$$

Then, for any N sized set V, in R^d, we can define a map $f : R^d \to R^D$ such that $\forall$ $a, b \in V$,

$$(1-\epsilon)\|a-b\|_2 \leq \left\|f(a)-f(b)\right\|_2 \leq (1+\epsilon)\|a-b\|_2 .$$

Additionally, this mapping process from d-space to D-space requires only a randomized polynomial time.

This lemma got proved by Dasgupta and Anupam Gupta (2003) and Achlioptas (2001). Many researchers proposed different methods for constructing the random matrix, See Achlioptas (2001),Li et al. (2006). Usually, these random matrices are not orthogonal, and also making them orthogonal is an expensive task, which takes more computational time. But, according to Nielsen (1994), the vast orthogonal directions present in an high-dimensional space is abundantly orthogonal. Many researchers presented different methods for constructing random projection matrix entries which obeys JL Lemma. Achlioptas (Achlioptas (2001)) is one of these methods that has been used extensively. This method uses integers and sparseness while constructing P, which eventually minimizes the cost of computation.

Achlioptas method of random matrix P elements are defined as:

$$p_{ij} = \begin{cases} +1 & \text{with } P_r = \frac{1}{2}; \\ -1 & \text{with } P_r = \frac{1}{2}. \end{cases} \tag{2.14}$$

or

$$p_{ij} = \begin{cases} +\sqrt{3} & \text{with } P_r = \frac{1}{6}; \\ 0 & \text{with } P_r = \frac{2}{3}; \\ -\sqrt{3} & \text{with } P_r = \frac{1}{6}. \end{cases} \tag{2.15}$$

$O(dDN)$ is the computational cost of RP, where d represents the original features (dimension), D represents the embedded (reduced) features of reduced data and N represents the size of the data. If the given data matrix has sparseness of c non-zero

entries per column, then the cost will become $O(cDN)$ (Papadimitriou et al. (1998)).

2.6 K-Means Clustering

K-means algorithm is implemented or executed to perform cluster analysis on given data. It is a prototype based clustering method, in which a given input database is divided/grouped into K number of non-intersecting clusters. This method starts by initializing K centroids randomly, and checked every point in database to find its nearest centroid and the point is assigned to that nearest centroid. This will be repeated for all the points in the database. Now the clusters centroids are recalculated as the mean of each cluster. This is repeated until no change in assignment (i.e. the algorithm converged).

Let $X = \{X_i, i = 1, ..., N\}$ be the database of N points. We want to cluster these N points into K distinct clusters, $C = \{c_k, k = 1, 2, ..., K\}$, where $K \ll N$. K-means tries to decrease squared Euclidean distance between the points and cluster centroid, μ_k is the mean of cluster c_k, which is given by (A. K. Jain (2010); Alshamiri et al. (2014)):

$$\mu_k = \frac{1}{N_k} \sum_{X_i \in c_k} X_i \tag{2.16}$$

N_k is size of a cluster c_k.

Mean squared error between centroid/mean μ_k and points in a cluster c_k is defined as in A. K. Jain (2010):

$$J(c_k) = \sum_{X_i \in c_k} \left\| X_i - \mu_k \right\|^2 \tag{2.17}$$

K-means objective is minimizing sum of squared error (SSE) of K clusters:

$$J(C) = \sum_{k=1}^{K} \sum_{X_i \in c_k} \left\| X_i - \mu_k \right\|^2 \tag{2.18}$$

Gradient descent approach can be used to reduce the error, which is scalable for large datasets and it is a good heuristic for optimising the distance. K-means algorithm contains the following steps:

1. K cluster centroids are randomly initialized.
2. Compute Euclidean distance of a point from K centroids.
3. Assign a point to its closest centroid.
4. Compute new centroid of each cluster using Eq. (2.16)
5. Steps 2 and 3 are repeated until converged (no change in mean).

2.7 Related Work

The main aim of many indexing techniques or search algorithms is reducing search space, to speed up nearest neighbors retrieval for given query object.

Efficient retreival of nearest neighbors requires the database to be partitioned, which can be done by two methods: One is clustering based and the second one is Pivot based.

Now, here onwards we will focus on the techniques or methods those are Pivot-based.

In the Pivot-based partitioning, a set of pivot (anchor or reference points) points are selected according to a pre-defined criteria, and the database is divided into multiple partitions based on these pivots.

There are many similarity search algorithms present in literature, which are based on the use of pivots for partitioning the space. These include: Burkhard-Keller Tree (BKT) Burkhard and Keller (1973), Approximating Eliminating Search Algorithm (AESA) Vidal (1986), Vantage Point Tree (VPT) Yianilos (1993), Fixed-Queries Tree (FQT) Baeza-Yates et. al, (1994), Fixed-Height FQT (FHQT) Baeza-Yates et. al, (1994), Linear AESA Mico (1994), Multi VPT (MVPT) Bozkaya and Ozsoyoglu (1997), Excluded Middle Vantage Point Forest (VPF) Yianilos (1999), Spagettis Chavez et. al, (1999) and Fixed Queries Array (FQA) Chavez et. al, (2001).

AESA (Vidal (1986)) is one such index technique, which uses a random pivot in querying, to cull out the unwanted objects from the search space using triangular inequality property of a metric. Pivot-based metric indexing technique takes at least $O(log(n))$ random reference (pivots) points. This was proved by Chavez et al. in

(Chavez et al. (2005)).

Survey on Dimensionality Reduction:

Dimensionality reduction is a process in which high-dimensionality is reduced to low dimensionality by using various techniques viz. PCA and MDS, Kernel PCA, LDA, Isomap, LLE and Laplacian Eigenmaps etc. The study is first initiated by Bellman in 1961 while studying the high-dimensional data (Bellman R (1961)). While studying that, he coined a term called *curse of dimensionality*. According to him the term refers to the fact that "without simplifying the assumption, the sample size requires to estimate a function with several variables to a given degree of accuracy grows exponentially with the increasing number of variables". He also proved with an example that the estimate of local average grows exponentially. He has drawn the conclusion to estimate local average, based on the density smoothers, local average of the neighbouring points. Finally, he also states that to find enough neighbours, one has to make use of multi variant smoothers in HD spaces to reach out farther, by loosing the locality.

Later Scott and Thompson in 1983 (Scott and Thompson (1983)) found another reason called as "empty space phenomenon" for the curse of dimensionality . According to this phenomenon, the high-dimensional spaces are inherently scant and thin. In other words it is sparse. To explain this phenomenon, he gives an example of sparseness which explains the probability densities in a unit of sphere in R^{10}. According to him the study of distribution of points in a unit sphere is most important thing in which the densed points situated. Earlier scholars in the field considered only the mean centered portion of Bell curve or normal distribution which is known as $N(0,1)$. But Scott and Thomson reveal that the mass in the unit sphere consists only 0.02% of $N(0,1)$ and the major points spread across the tails of the distribution instead of mean centered area.

Perpinan produced his first report titled a review of dimension reduction techniques (Perpinan (1997)). The report is divided into six parts. The first part, which is an *Introduction*, introduces the dimension reduction problem. Second part deals with the principle component analysis by explaining the disadvantages of Principal Component Analysis. Third part describes the *Projection Pursuit*, which is a visual

representation of the data by unsupervised technique. Projection Pursuit is used for low dimentional linear orthogonal projection of a high dimentional space or data. This part also higilights, how human beings discovers the patterns in low dimentional projections i.e. (1 - 3-D); discusses about visual presentations of projected data density viz. histograms, smoothed density estimates, scatter plots and contour plots. The fourth part explains about the principal curves and principle surfaces. The fifth part discusses topological continuous maps which includes Kohonen's Self-Organizing Maps (SOM) and density networks. The sixth part of the study describes the neural networks implementation of the stasticlal models described in the above chapters. Over all, the study is a survey on different dimension reduction techniques which includes PCA, projection pursuit, projection pursuit regulations, principal curves and methods based on topologically continuous maps such as Kohonen maps or generalized topographic mapping. Along with the survey he also had experimented the above techniques using neural network implementations. While experimenting he had detected two major issues: to obtain the reasonable results for dimensionality reduction problem requires huge sample size and determining the intrinsic dimension of the given distribution of the data are open problems in the area.

Maaten et.al prepared a technical report titled *Dimensionality Reduction: a Comparative Review* (Maaten et. al. (2009)). The report addresses the limitations of traditional linear techniques such as PCA and Classical Scaling. The report systematically describes and compares a variety of non-linear dimension reduction techniques which are also implemented and experimented on natural and synthetic data sets. The main contribution of the report is how the problems encountered in linear techniques are being solved by the implementation of non-linear techniques. The weaknesses of the current nonlinear techniques is also explained in a vivid way after the experiential studies. The later part of the report explains how the performance of the non-linear techniques will be improved with the major observations and suggestions from the study. Another contribution of the study which was submitted in the form of a report on dimensionality reduction a comparative review is systematic comparison of dimensionality reduction techniques which is new to the field at that time. This report helped the researcher as preliminary foundation for the present study; which also follows similar comparisons as given in Figure 2.1.

Cardoso and Wichert (2012) paper on Random Projection for high dimensions, proposes an iterative version for the random projection K-means algorithm (Cardoso and Wichert (2012)). They have compared the proposed algorithm (IRP K-means) with related approaches viz. RP K-Means, K-Means on original HD spaces. For the study they had done experimentation on the image and text data. IRP-K-means is showing low Mean Squared Error (MSE) in the original space, than single RP.

Martins and Gurjao (2013) (Martins and Gurjao (2013)) paper on Random Projection, applies random projections on house hold electric meters to describe behavioural usage of the house hold energy consumptions of 443 homes. For this, they selected 443 households consumption usage of electric consumption from UMas-Trace repository. They applied dimensionality reduction via random projection to obtain reduced sketch of the smart meters original data. The method they employed in the study is that 443 house hold electric consumption is measured at a sample rate of one sample for minute. The 443 households consumption has come to 1440 samples. So the size of the data set is then 443 by 1440. When analysing on the dimensionality they found ample number of redundant dimensions. By using RP they have reduced the dimensionality by 50% of the original data with an achievement of 2% reduction in average relative error. So, this helped in reducing time, space and energy.

Aleshinloye et al. (2017) (Aleshinloye et al. (2017)) discusses the efficiency of DR tools for demand side management. This was a preliminary study from the perspective of management. It is also a critical analysis of DR on smart meters data for smart grid applications. In the study they compared the performances of two DR techniques viz. PCA and RP. As part of the method they used RP on high dimensional data. The technique reduced the dimension to low dimensional feature space. Later a cluster technique is applied on this low dimensional space which resulted in some of the square errors (SSE), distance between data points. Similar clustering also applied on the original HDD. Later the results of HDD and LDD were compared, which indicated that the PCA had better performed than RP in LDD. They also concluded that RP is better for smaller dimensions than PCA.

Tariq et.al (2018) (Tariq et al. (2018)) discusses an efficient approach, which helps in feature construction of high-dimensional micro array data by using Random Projections (RPs). The paper highlights the integration of Genetic Programming (GP) techniques with random projections. The study considered two ways of employing techniques viz. apply the Decision Tree (DT), Random Forest (RF), Naive Baiyes (NB), Support Vector Machines (SVM) and K-nearest neighbor method (KNM) on input data directly or apply RP to convert the data into low dimensions and then apply the former five techniques. Eight data sets were taken for the implementation of the techniques. The over all results project that when 50 features are constructed using GP the results were found to be best all the times whereas the accuracy was gradually decreasing as the number of constructed features increases. Finally, the results indicate significant increase in the over all accuracy with the use of RP based constructed features. It was also observed that there is a decrease in the standard deviation.

(Rana et al. (2013)) is a study on Deterministic Construction of projection matrix for adaptive trajectory compression. It is a thought-provoking paper in the area. The study proposes an adaptive compression algorithm. This algorithm defines a deterministic and data driven construction of the projection matrix. This projection matrix is obtained by applying a singular value decomposition to a sparsifying dictionary, learned from the data set. This study contributes to the field of compressive sensing and signal recovery as follows: 1. proposed an adaptive compression frame work which under pins compressive sensing theory and support vector regression. 2. proposed a data driven and deterministic construction of projection matrix which was combined with the trained data dictionary to offer better compression ratio compared to the predefined matrix and dictionary pairs. 3. validated the proposed compression frame work performance by using large data sets which includes pedestrian data of 91 different students and volunteers from five different sites, and an animal trajectory data from 36 cows of CSIRO's Belmont deployment. Finally, they conclude from the case studies including GPS trajectory data sets, pedestrian and animal data sets which contain more than 120 subjects. The adaptive compression is more useful to increase the performance of the trajectory compression. This is because the adaptive compression with proposed compressive matrix can be saved maximum 40% of transmission for pedestrian data sets and about 85% transmission

for animal data sets. The study also highlights the more suitability of the deterministic construction of projection matrix compared to the free defined random matrices to achieve improved trajectory performance.

Rana et. al. (2014) (Rana et al. (2014)) paper on Trajectory Compression, where they apply deterministic projection method of Matrix for Mobile Sensor Networks, is an informative and thought-provoking paper in the field of compressive sensing. The paper also proposes a method for predicting the size of projections needed for mobile nodes adoptively based on their speed. The result of the research is that it (i) proposed a different method from earlier ones which computes projections from a learned dictionary, (ii) a new and simple adaptive method is proposed, that uses support vector regression which enables the MSN nodes to choose the number of projections instantly, based on their speed and (iii) shows, based on the enormous experimental results of the 6 data sets that the average distance between the original and the reconstructed trajectories, (ADE) reduces by $10-60$ centimeters by the proposed one when equated with SQUISH.

Juvonen, A and T, Hamalainen (2014) (Juvonen and Hamalailen (2014)) is a land mark paper in the area. The paper highlights the Efficient Network Log Anomaly Detection System, which uses Random Projection Dimensionality Reduction. This study develop a system which facilitates a quick anomaly detection by visualizing the network traffic structure. For the study DR method is employed. The developed system works by taking webserver log data as an input and pre-processes it in order to extract the numerical features from it which directly or indirectly forms the future metrics. The study also highlights that while extracting the numerical features the dimensionality of this feature metrics is reduced using random projection methodology. The outliers are visualized and highlighted from the reduced dimension.

(Sachin and Kaban (2014)) is also another study in the field of dimensionality reduction. It is a comparative study in which two methods viz. Random Projection (RP) and Random Feature Selection (RF) are compared. The study mainly focuses on classifying HD data points. It is basically an empirical investigation using a comparative method on RP and RF by means of dimensionality reduction for classification. The results of the study indicate that RP better classifies data than RF when the dimensionality is larger than the number of points. RF is a bit competitive to RP in

some occasions.

Chris Ding and Xiaofeng He (2004) paper on K-means clustering via principal component analysis is a statistical method of analysis, which was used for the unsupervised dimension reduction (Ding and He (2004)). In this work, they proved that principal components are progressive solutions for the cluster membership. For the study two data sets are used viz. DNA gene expression and internet news groups. The results of the experiment revealed that newly derived L (low) bounds for k- means are 0.5-1.5% of the ideal (optimal) values.

Qi and Hughes (2012) studied PCA and RP for data analysis, experimental results reveals that the output of the data, which is the result of applying on low-dimensional random projections is equal to that applying PCA on original data set.

Extreme Learning Machine (ELM) is combined with K-means clustering in Alshamiri et al. (2014). The proposed method first projects the low-dimensinoal data to HD feature space by using ELM, then applies K-means clustering in that ELM space, it improves clustering quality.

Alshamiri et al. (2015) paper on "Combining ELM with RP for Low and High-dimensional data classification and clustering" is an experimental study in the field of RP wherein the proposed algorithms are tested on low and HD data for classification and clustering (Alshamiri et al. (2015)). There are two scenarios in the study. In the first scenario RP is used for reducing the dimensionality by cutting down the ELM hidden layer neurons. Later, they performed the classification and clustering in ELM (reduced) space. Second scenario: LD data is first projected to ELM space (as this increases data dimensionality, now data attains linear separability). Now, RP is applied to reduce the dimension along with preserving linear separability. For the experimentation, they used different types of data sets viz. 12 low-dimensional and 8 high-dimensional data sets. Finally, the study suggests that the integration of RP and ELM gives satisfying results, also provides a clear agrrement between performance and computational cost.

(Khandelwal et al. (2016)) is a review of applications of PCA in multi model biometrics system. In this review they proposed a system which is a rank level fusion of multiple domain experts information. For the review they compared two works: an

unimodal biometric system and a multimodal biometric system. They also checked the other methods like highest rank method, borda count method and logistic regression method. All the above three are methods to combine the ranks of different biometric systems. Finally, they conclude that multimodal biometric is more accurate compared to the unimodal biometric system.

Sanjay Dasgupta (2000) paper on Experiments with random projection is an informative paper in the field of RPs, See (Dasgupta (2000)). In this paper he has done an experiment which is already present in literature in the form of theory. He has conducted the experiment on synthetic as well as real data sets of Gaussian and OCR. These experiments were done in order to illustrate the benefits of the RP technique. PCA and RP, RP and EM methods were used for the experiments and compared them for the better performance as well as benefits. The paper concludes, based on the experimental results that, RP performance is better when compared to other methods like PCA, EM etc. The paper also highlights that RP has reduced the dimensionality from 256 to 40 and had better benefits.

Samuel Kaski (1998) paper Random Projection by Random Mapping (RM) for fast similarity computation for clustering is an experimental one in the area of dimensionality reduction (Kaski (1998)). He is the first person to introduce the RM method for the text documented data and classification. He explained the process of reduction where he mentions that the usage of RM method reduces the original document and preserves the original dimension even after the reduction. Through this paper he promotes the RM method by saying that it is a promising, computational and feasible alternative in the area of dimensionality reduction. Along with the above promotions, he also mentions that in the above situations like dimensionality reduction, the reduced dimensional data vectors will be used for the clustering as well as for other similar approaches. Later the method is extensively applied in the areas of WEBSOM document organization system. He also promoted that this method is as good as PCA or the original high dimensional data vectors. Finally the paper suggests that RM produces better separability of different topic areas for the news groups with 68% better than its counterpart methods.

Hegde et al (2007) in their paper titled *Random Projections for Manifold Learning* is an experimental as well as a theoretical paper in the area. In this work the

researchers proposed a novel method for linear dimensionality reduction of manifold modeled data. The main contributions of this work is: Let HD ambient space R^N can be learned in a manifold of dimension K, it can also be embedded in a lower RP space R^M, and $M = CKlog(N)$.

This can be described in three main points as follows: (i) a minmum bound is defined for intrinsic dimension (ID) estimation. This is achievd upto an accepted accuracy level of the Grassberger-Procaccia algorithm Grassberger and Procaccia (1983) and Camastra (2003), which is a widely accepted geometric approach for intrinsic dimensionality estimation.

(ii) To discover the non-linear structure of the manifold, they presented a bound (minimum number of measurements per sample point required) for manifold learning algorithm - Isomap Tenenbaum et al. (2000). In both (i) and (ii), M logarithmic in N, linear in K. (iii) proposed a linear algorithm (ML-RP, which is weakly adaptive) for DR and manifold learning, without any information about data, it finds lower bound of M, in practical settings.

(Ailon and Chazelle (2006)) is a theoretical paper on Approximate Nearest Neighbors and Fast Johnson-Lindenstrauss Transform . The paper is highly theoretical in its nature. The researchers have introduced a technique called Fast JL Transform (FJLT). It preconditions the sparse projection matrices with a randomized Fourier transform. This technique speeds up the search algorithms based on the Linear Dimension Embedding (LDE) in L_1 and L_2 spaces.

Dasgupta and Anupam Gupta (2003) paper on JL Lemma proof, lucidly explains the JL-Lemma and its proof. This is also a bench mark paper in the area of, specifically RP which provides a proof for JL Lemma by using some elementary probabilistic techniques. The theorems that are presented in this paper are analogous to Indyk and Motwani (1998) and Achlioptas (2001), which gives a lower bound for the JL-lemma.

Juvonen, A. et. al. (2015) paper on online anomaly detection by using dimensionality reduction techniques for HTTP log analysis is an experimental frame work in the area (Juvonen et al. (2015)). This frame work is used to find out the abnormal behaviour of the network logs. The proposed network has special character in the

sense that it has online capabilities of detecting an intruder. This is demonstrated by using the real world network log data. Three methodologies were used in the research viz. RP, PCA and DM. From the experimental results, the study suggests that the RP and DM should be combined for the efficient analysis of the network traffic and for better detection.

((Han et al., 2017)) discusses about Online multi linear principal component analysis (OMPCA). The paper is an extension of Multi-linear Principal Component Analysis (MPCA) learning method. The OMPCA proposed by them is tested for higher order tensor machine for classification. The results of the study show that OMPCA significantly reduces the time of DR with a little loss of recognition accuracy.

(Tasoulis et al., 2013) paper on Random direction divisive clustering is an experimental study in the area of RP and DR. The study is basically about the performance analysis of RP with various clustering algorithms i.e. RP with clustering algorithms for high-dimensional cases. The study proposes a new RP clustering algorithms viz. rp-de PDDP, RDDP and RL RDDP. Later the study explains about the achievement of high quality data partitions with orders of magnitude faster. Finally the study concludes with the experimental analysis on seven HD data sets (gene, face recognition). The results of the study suggest that the proposed clustering frame work has computational savings of RP with a minimal performance loss.

(Bettoumi et al., 2016) compares k-means variants for mono-view clustering is a comparative work in the area, which is broadly related to the area of our research. This reviews various clustering methods. These methods implement RP and DR on K-means, IRP K-means and Fuzzy K-means (FKM). The three methods were thoroughly experimented and tested on a set of images using five separate descriptors with different sizes. The performance measures were used for the comparison of different clustering characteristics viz. purity, accuracy and running time. The review suggests that IRP K-means has better accuracy than the other.

(Yu and Zhang, 2016) paper on a 3-way decision clustering approach for HD Data is an experimental one with a proposed quantitative technique. The study proposes a new 3-way decision clustering approach using RP. It works by applying 3-way K-medoids multiple times on given input, and increases the dimensionality of the data after each iteration of three-way K-medoids. The proposed method also experimen-

tally compared with IRP K-means, Fuzzy Subspace Clustering (FSC) and K-medoids. For the study three parameters were used. These are accuracy, Normalized Mutual Information (NMI), and CPU time. The study finally concludes that the novel 3-way decision approach is best suitable for Hd data with higher accuracy and no compromise on computation time.

2.8 Summary

In this chapter, we discussed the Curse of Dimensionality, which is the basis for the Dimension Reduction. Then explained the term Dimension Reduction or Dimensionality Reduction. After that, a preliminary discussion on some of the basic DR methods like PCA, RP is presented, and also K-means clustering is explained. These methods have been used in the works for comparing the performance of the proposed dimension reduction methods. Later on we have reviewed the literature on Nearest Neighbor Search and also on Dimensionality Reduction. These works have motivated us to work in the area of DR of HD data and Nearest Neighbor Search.

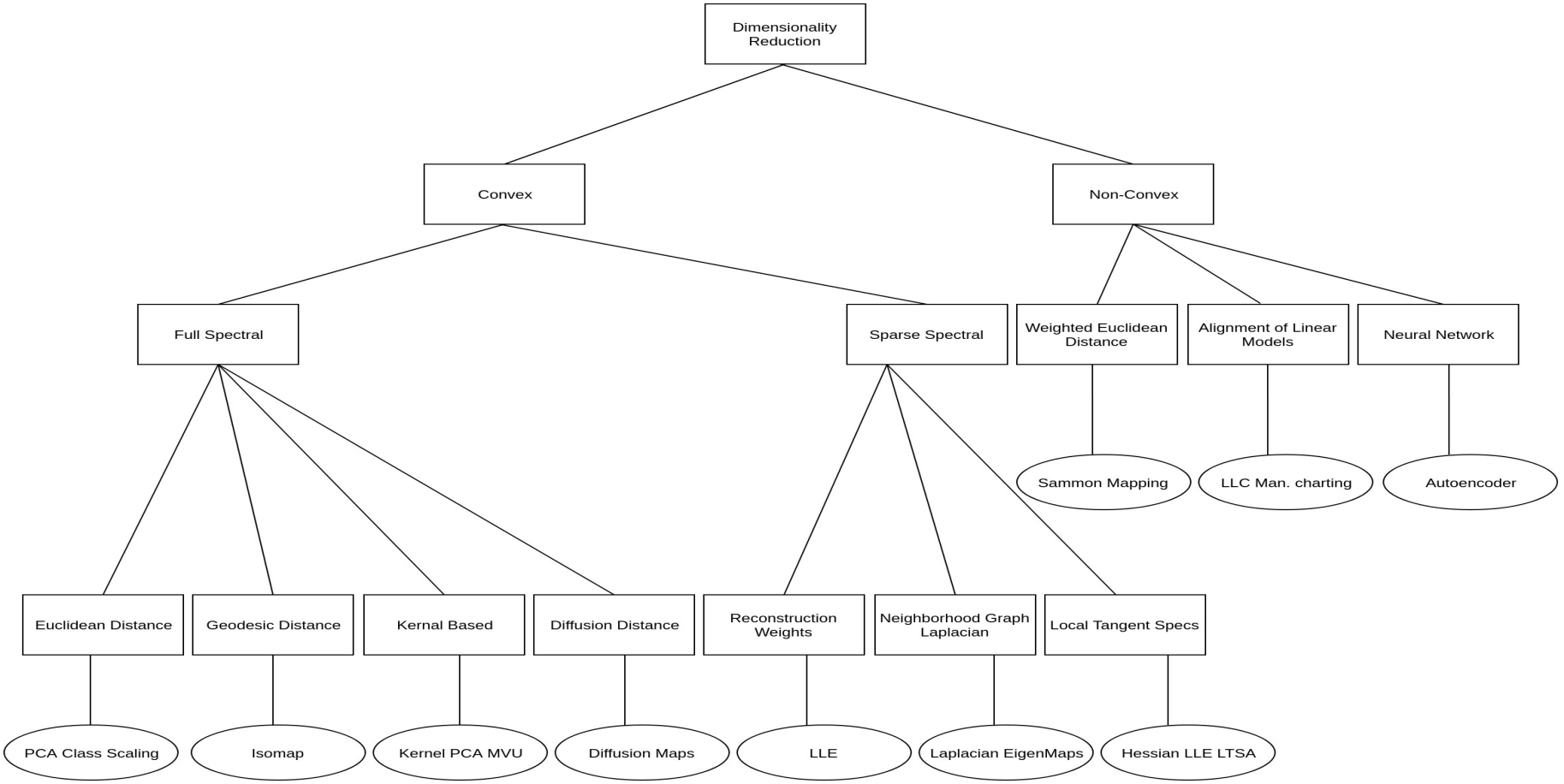

Figure 2.1: Taxonomy of dimensionality reduction techniques (Maaten et. al. (2009))

CHAPTER 3

Proximity-based Nearest Neighbor Search Algorithms

Proximity Searching refers to finding similar objects of a given query object by using a similarity/distance measure. There are many measures or metrics present to find out the similarity or dissimilarity between two objects.

Nearest Neighbor Search (NNS) is a very common form of similarity search. Similarity search requires pre-processing of the given database so that the similar objects of a given query can be found efficiently. Similarity search can be done on a database using two methods: **(i)** ***Cluster-based similarity search***- in which a clustering algorithm is applied on the database, and while querying we can omit the clusters which are not in the range of the given query. **(ii)** ***Pivot-based similarity search***- in which the given database is pre-processed (partitioned based on the distance from pivot point) so that while searching for similar points (kNN's) of a given query, we can find them in any one of the partitions, and the remaining partitions are simply omitted. We considered the later one in this chapter.

In this chapter, we propose two methods for partitioning space to speed up the nearest neighbor search. These two methods (proposed) are pivot-based partitioning methods. The first one uses a single pivot point (reference point/anchor point) for partitioning the database, and the second method uses group of pivots (reference points) to partition the database. Both the methods are compared with the existing methods and experimentally evaluated on various datasets.

3.1 Reference (Pivot) Points Selection for Space Partitioning

Many real world application require searching a large amount of data to find the objects similar to the search queries. Nearest Neighbour Search is the common operation in similarity search. The similarity search is useful in many domains viz.

Content-Based Image Retrieval (CBIR), Web search engines, micro array data analysis etc. Dimensionality forces us to look at data from a different perspective when dealing with such large data.

In the present work, we propose a distance-based partitioning method, which uses a pivot (reference) point to divide the data into disjoint groups for finding the nearest neighbors of a given query from large and HD data.

The proposed method divides the given database into separate groups based on the distance from a reference point. In the next level for each sub-partition a reference point is selected and again it is partitioned into further sub-sub-partitions if required. Main objective of partitioning is to reduce search space. To find the nearest neighbours we use Soergel Distance metric which is a dissimilarity-based distance metric which finds the association between any two database points.

By experimenting we proved that, the proposed method is capable of retrieving the top nearest neighbours (k-NN) by searching in only a single bin, and the remaining bins need not be searched (search space reduction).

We have validated our method by conducting experiments with the following data sets: ZINC data set, AT & T (ORL) Faces Database, Yale, Leukemia, Lung and GCM. Proposed method reduces or prunes the search space and saves lot of computation time.

3.1.1 Introduction

Nearest Neighbor Search (NNS) has been a research problem in World Wide Web era beacause voluminous data produced by the on-line sites, Content Based Image Retrieval (CBIR), and micro array data analysis in Bioinformatics and so on. There are large number of studies in literature which were focused on this problem.

Curse of dimensionality is the biggest challenge in high-dimensional data. The processing complexity becomes high as the data dimensionality grows. Exponential time is needed for nearest neighbor retrieval with increasing dimensionality, Indyk (2004).

Large dimensionality/feature is inevitable in domains like, Chemical similarity

search, CBIR, Gene Expression (Micro Array) Data Analysis, etc. These all applications require NNS operation to do some processing, and NNS is the most common operation in these data domains.

In this work, we try to improve the nearest neighbor search on large and high-dimensional data sets by pre-processing the input database into a number of partitions based on the distance to the points computed from a selected reference point. After an enormous experimentation, MinMax reference point was found to be suitable for data partitioning.

3.1.2 Related Works in the Area

Several researchers all over the world are working on finding nearest neighbors in HD space. Most of them are interested in constructing search/index structures for similarity search in HD such as genome databases, time-series, text documents and image databases. But unfortunately scalability of these structures with the dimensionality is poor; for example the k-d tree performance degrades to such an extent it is worse than even brute-force linear search if the dimensionality of the data exceeds 10.

Gionis et al. (1999) presented a novel hashing-based method called Locality-Sensitive Hashing (LSH) for approximate similarity search. LSH works by hashing the database points so that the most similar points will have more collisions so that they will fall in the same bucket, where as the dissimilar points will fall into different buckets. The LHS method works up to a dimensionality of 50 or more.

In (Indyk and Motwani (1998)), Indyk and Motwani proposed a variant for exact nearest neighbor search which is called approximate nearest neighbor search. To overcome the dimensionality curse, one can trade the search performance by opting for approximate NNS. Instead of finding exact similar point of a given query, in approximate NNS, we find closer points to the query. Indyk (2004) presents theoritical syudy on approximate NNS algorithms.

A tree based index structure called Multi-Vantage Point (MVP) tree was proposed by Bozkaya and Ozsoyoglu (1997). MVP-tree is a distance-based index structure.

Another metric-based data structure is proposed by Yu et al. (2001), which is called iDistance. In this the data is projected onto 1-dimension (1-D line) from high-dimension. They used pyramid technique for data partitioning. This index is more scalable and adaptive to high-dimensionality than the previous methods present in literature.

In Chandrasekhar and Rani (2012), a rank based feature selection type DR method is proposed along with a storage layout for storing the retrieved nearest neighbors. Correlation Fractal Dimension (CFD) as a descrmination measure was used to select fature subset from original data. By using CFD, dimension is reduced from 58 to 7. The nearest neighbor revrieval capacity of the proposed method is invariant of the dimension, i.e. the nearest neighbors for a given query, both from original data (with 58 features) and from reduced data (with 7 features) are same. This has been proved by the experimental results.

Samet (1989) presented a survey on various data structures used for NNS geometrical spaces.

PIVOT SELECTION STRATEGIES:

Pivot selection is very important for any pivot-based indexing technique, selecting the good pivots is the question of interest. A golden rule is to randomly select pivots, but the selection impacts the search performance.

Many pivot selection strategies were introduced by Bustos et, al. in (Bustos et al. (2003)). The main contribution of Bustos et al. is : comparison method for a collection of pivot sets, and find which set is best (performance-wise) when compared to the other one. The authors found that a best pivots are those that are having largest mean in the mapped space. They proposed an *efficiency criteria* for comparing two pivot sets. In this they proposed three methods to select best among pivot sets: i) selecting random groups ii) incremental selection iii) local optimum selection.

A dynamic selection of pivot selection algorithm is proposed in (Bustos et. al, (2008)). This method computes each pivot's contribution to decide whether to keep that point in pivot set or to replace it with a new pivot, the higher the contribution of a pivot the higher the chance for it, to become part of pivot set.

Another pivot selection technique is Sparse Spatial Selection (SSS) by Pedreira

and Brisaboa (2007). Here the selection of pivots is automatic, and the intrinsic dimensionality of the database has effect on the SSS pivots, irrespective of the database size. The main aim of this method is to select a set of pivots that are well distributed over the space. This method works by fixing the maximum distance d_{max} between any two points in the database. It starts with an empty set and incrementally selects the pivots by checking for *well coverage* of present set of pivots in each step. A new point will become a new pivot if its distance to all the points is greater than or equal to ϵd_{max}, and $0 < \epsilon \leq 1$, $\epsilon = 0.4$ is the experimental suggestion by the authors.

Extreme Pivoting (EP) method (EP Table) is presented in (Ruiz et. al, (2013)), which is a new index for proximity searching. This method selects a set of non-redundant pivots which covers the full database.

The Pivot placement problem was studied and addressed by Angiulli and Fassetti in (Angiulli and Fassetti (2013)), in which they proposed a strategy for pivot placement by using the data orientation that is present in the given database. This orientation will guide us to select the best set of pivots. The proposed method is called Principal directions-based Pivot Placement (PPP) algorithm. It uses clustering to determine the small clusters then inter-cluster directions are found. Now, the angles between these directions are computed, then prioritized fixed width clustering is peformed, eventually the best pivots are determined. The experimental results prove that the indexing performance can be improved by the best alignment of the pivots by using the proposed method.

Chavez et. al, (Chavez et. al, (2015)) presented a novel framework for approximate nearest neighbor (ANN) search algorithms, called *K* Nearest References (K-nr). This works by selecting a subset R_s of R reference points from database. Now the original proximity (similarity search) problem is mapped to a signature space, that is constructed from the K nearest references to that object. While querying, a candidate set is prepared by finding the similarity between signatures of the objects and query q, then the q is directly compared with the obtained candidates.

Sergey Brin in his paper on GNAT Tree (Brin (1995)), proposed a method for selecting split points (reference points). His idea of selecting m - number of split points starts with selecting randomly one point from candidate list ($3m$ points are there in candidate list, and this 3 is found empirically). Then select a farthest candidate point

from the earlier point. Now, select a candidate point which is farthest from the previous two points. Then pick a point which is fathest from these three points, and so on, and stop when the desired number of split points are selected. A dynamic programming solution requires $O(nm)$ time to do this, n represents size of candidate points and m is size of desired split points.

3.1.3 Reducing the Search Space for Efficient retrieval of Nearest neighbors

Partitioning Algorithm

To lessen the search space and hence the computational complexity we use Partitioning. It is a data space partitioning method, where in the data space is divided into bins based on the distance from a chosen reference point to the data objects in the sample. We use $3/2(min+max)$ point from X as the reference point to partition the data, and calling it as MinMax method. There are two phases in this method. One is construction phase and the other is search phase. In the construction phase, we project the distances onto a straight line and that the line is divided into equal parts in some interval range and this will define the bin boundaries. Once the bin boundary is defined, in the construction phase, we calculate the distance of a data object from its reference point and check the bin range in which the calculated distance falls, then that data object will be stored in that corresponding bin. The same procedure is followed for all the data objects and that completes the construction phase. This is rephrased in the following algorithm, Algorithm 1.

Algorithm 1 Construction

1: Read and pre-process the input data set X.

2: Select a reference point based on selection criteria.

3: Compute distance between reference point to all the points in X.
4: Equally partition the range of the Distance into intervals (bin boundaries).
5: Distribute all the data objects among the bins and sort them.
6: Repeat from step 2, for each sub-partition until stopping condition is met.

In the search phase, we are given a query object in HD space and expected to return its nearest neighbors. For this, we first compute the distance from reference

Algorithm 2 Search

1: Read query q, k (no. of neighbors to be retrieved).

2: Calculate the distance between reference point and q.

3: Go to the appropriate bin, and if it is further partitioned into bins then repeat from step2, else continue.

4: Return the top k-objects from the bin as nearest neighbors.

point to query point and check in which bin it is falling, so that we will directly eliminate the other bins from searching; concluding that no nearest neighbors will be found in those excluded bins. By searching in only one bin, we can retrieve the nearest neighbors according to the distance in the order. The same procedure will be followed in the next level also, up to a specified recursion depth. This greatly reduces the computational effort. The above described search process is presented in Algorithm 2.

3.1.4 How to calculate nearest neighbors

Nearest Neighbor Search problem is defined as retrieving the closest points for a given query from the database. The nearness is determined in terms of distance from a reference point. So the selection of reference point is crucial in nearest neighbor search. To calculate the similarity (dissimilarity) between any two data objects, we have measure called Tanimoto Coefficient (TC), which is basically a similarity coefficient whose value is in the range 0 to 1, See Rao et al. (2011) for more details on TC. The similarity value 1 indicates that the points are similar and the similarity value 0 indicates dissimilarity. A, B are any two points, then TC is computed as:

$$TC(A,B) = \frac{\sum_{i=1}^{n} a_i b_i}{\sum_{i=1}^{n} a_i^2 + \sum_{i=1}^{n} b_i^2 - \sum_{i=1}^{n} a_i b_i} \tag{3.1}$$

where a_i and b_i represent i^{th} feature value of A and B respectively.

We use a dissimilarity-based metric called Soergel Distance (SoD) in our experiments, See (Cha S-H (2007)) for more details. Soergel distance value ranges from 0 to 1, value 0 indicates that the two data points are similar and value 1 indicates that the two data points are dissimilar. SoD is complement to Tanimoto coefficient. For

any points A, B, SoD is computed as:

$$SoD(A,B) = \frac{\sum_{i=1}^{n} |a_i - b_i|}{\sum_{i=1}^{n} max(a_i, b_i)} \tag{3.2}$$

where a_i the i^{th} feature value of A and b_i is i^{th} feature value of B.
For more on metric properties like, non-negativity, identity, symmetry and triangular inequality, See Section 2.1.

3.1.5 Experimental Results

Data sets used for empirical study

We used ZINC database, which is a drug like chemical structures data, available online (John Irwin and Brian Shoichet (2005)). ZINC is a free virtual screening database that comprises of chemical compounds. Each structure contains ZINCID and SMILES notation of it.

Linear representation of a chemical compound is called SMILES. SMILES stands for Simplified Molecular Input Line Entry System, that represents a 2-D or 3-D molecule in string format.

The drug-like data set is having 8,783,230 chemical structures in it. It has 9 physical features, 10 Atom count features and 9 structural features, total of 28 features, See 3.1. See Rao et al. (2011) for more information on the features of this dataset.

For this study we have taken nearest neighbors of 100 selected molecues and from this we took the neighbors of first 5, and the resulting data set is called as ZINC5, its size is 596 by 7 by 5.

Besides ZINC5, we have used another data set i.e. AT&T Database of Faces (ORL Faces) composes of 400 image samples, belongs to 40 persons, 10 per each one. Each image size 92 by 112 pixels (so 10304 dimensions), and has 256 gray levels, and also another version of ORL used in experiments, i.e. with 1024 dimensionality, in which each image size is 32 by 32 pixels, see ORL Faces (2002) for more details on this data set.

GCM (Global Cancer Map) has 190 tumor, normal tissue samples are 90 in num-

ber.

Leukemia has 72 samples of two types: 25 acute lymphoblastic leukemia (ALL), 47 acute myeloid leukemia (AML).

Lung cancer contains 181 samples. This gene expression data samples are classified into malignant pleural mesothe-lioma (MPM) and adenocarcinoma (ADCA).

Yale comprises of 165 faces of 15 persons and 11 images per person, with a dimensionality of 1024.

Physical(9)	**Atom Count(10)**	**Structural(9)**
Molecular Weight	Br. Count	Cyclic
logP	C Count	Acyclic
De_ apolar	Cl Count	Mono Cyclic
De_ polar	F Count	Bi Cyclic
HBD	I Count	Tri Cyclic
HBA	N Count	Tet Cyclic
tPSA	Na Count	Hi Cyclic
Change	O Count	Hetero Cyclic
NRB	P Count	Chiral Centers
	S Count	

Table 3.1: The names of 28 features of ZINC data set, which are classified into three major types, from these, 9 are Physical features, 10 are Atom count features and 9 are Structural features

Experimental Results on various data sets

We have experimented by implementing both BO_ Heuristic (Bozkaya and Ozsoyoglu (1999)) and MinMax methods and tested these methods on various data sets. The average precision of these experimental results are reported.

On the ORL_ 1024 dataset, the average precision for BO_ Heuristic is 43% whereas it is 56% for MinMax method.

On the Yale dataset, the average precision is 23% for both BO_ Heuristic and MinMax method.

On the Leukemia dataset, the average precision for BO_ Heuristic is 55% whereas it is 54% for MinMax method.

On the ORL_ 10304 dataset, the average precision for BO_ Heuristic is 37% whereas

Data set	Average Precision (%)	
	BO_Heuristic	MinMax (proposed)
ORL_1024	43	52
Yale	23	23
Leukemia	55	54
ORL_10304	37	44
Lung	73	80
GCM	61	75
ZINC5	27	56

Table 3.2: Average precision (percentage) of 10 queries on various datasets, BO_ Heuristic (Bozkaya and Ozsoyoglu (1999)) versus MinMax method (proposed)

it is 44% for MinMax method.

On the Lung dataset, the average precision for BO_ Heuristic is 73% whereas it is 80% for MinMax method.

On the GCM dataset, the average precision for BO_ Heuristic is 61% whereas it is 75% for MinMax method.

On the ZINC5 dataset, the average precision for BO_ Heuristic is 27% whereas it is 56% for MinMax method.

In summary, both the methods are equally performing on Yale and Leukemia datasets, and for the remaining datasets the proposed MinMax method is outperforming the BO_ Heuristic method.

Distance Distribution Plots for BO_ Heuristic and MinMax methods

As a validation experiment, we have plotted the distance distribution in histogram plots for both the methods. As per the good pivot properties the intuition is that a good pivot will give a flatter distribution, which is useful for eliminating most of the points while finding the neighbors of a given query. This elimination is possible by using triangular inequality property of distance metric, here we used Soergel distance metric, which obeys all the three properties of a metric. From the Figure 3.1, we can see that the distance distribution of MinMax method (first column plots) is flatter than that of the BO_ Heuristic method (second column plots), this helps in eliminating large candidates while finding nearest neighbors of a given point.

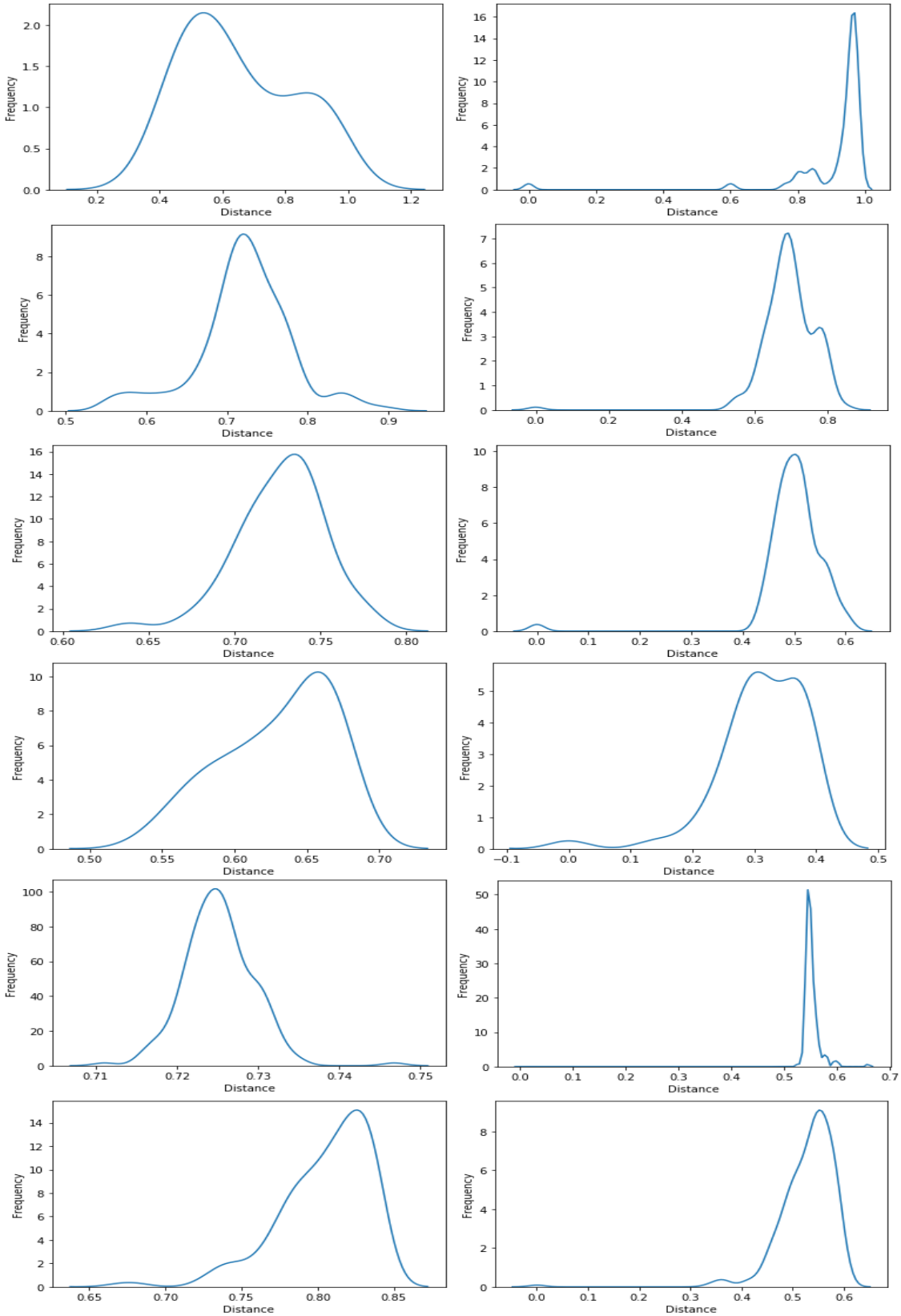

Figure 3.1: Distance distribution plots for various datasets: The First column is of proposed MinMax method and the second column is for BO_ Heuristic method. 1st row: ORL1024, 2nd row: Yale, 3rd row: Leukemia, 4th row: ORL10304, 5th row: Lung, 6th row: GCM.

Conclusion

Pivot-based partitioning is studied and proposed a pivot-based data partitioning method called MinMax method, which divides the data into different bins by taking MinMax reference point. MinMax is tested on various datasets and its performance is equated with existing method called BO_ Heuristic (Bozkaya and Ozsoyoglu (1999)). The empirical results project that the proposed and suggested MinMax technique is best on the given data sets when compared to BO_ Heuristic method.

The efficiency of MinMax (proposed) method is validated through plotting the distance distribution graphs for the considered data sets. These distance distribution graphs shows that the proposed MinMax method has flatter distance distribution than the BO_ Heuristic method, which defines a golden rule in eliminating more candidates in similarity search or finding nearest neighbours of a given query.

3.2 Weighted Set of Reference Points (WSRP) Method

In this, we have generalized the previous partitioning (MinMax Reference Point-based) method by taking a set of reference points and weighting them according to the distance from first reference point (R_1).

3.2.1 Proposed Weighted Set of Reference Points (WSRP) Method

The proposed method consists of two parts: Construction or Partitioning phase (off-line), in which the given input data is partitioned into smaller groups based on the proposed partitioning method and Search phase (on-line), in which, a query is given and we have to find the nearest neighbors the given query.

Partitioning Algorithm

To fasten the search operation and to reduce the search space and the computational complexity we use Space Partitioning Approach. It is a data space partitioning method, wherein the data is divided into bins based on the distance from a chosen set of reference points to the data points in the given data set.

There are two phases in this method. One is Construction phase and the other is Search phase. The proposed method is a variant of the proximity-based data partitioning method that speeds up the nearest neighbor search by reducing the search space.

This method works by taking multiple weighted reference points for partitioning the data into groups, finds the nearest neighbors of given query and returns them.

The core concept is defined by the following main points:

- Instead of taking single reference point, a reference points set is chosen for data partitioning.
- Reference point set is defined according to a pre-defined criteria
- Computes weighted distances of all points in X to Reference Points Set.
- Find the range of these distances, divide the range into equal interval bins.

- Distribute all the points of X, into these bins.

This is explained in detail, in the following text. Let $X_{N\times d}$ be the given input database of N points and d dimensions. The proposed method takes MinMax as the first reference point (R_1) for selecting l number of reference points (including R_1). First we compute $dist(R_1, X_i), i = 1,2,3,...N$ which gives N number of distance values. We find the range of these distances and divide the range into $l-1$ equal bins. Now, randomly pick a point from each bin. Let the random point picked from the i^{th} bin be $R_{i+1}, 1 \leq i \leq (l-1)$. These points will form the Reference Set $\Re$ (including R_1). The weights of these reference points are computed as: $d_{R_1} = 1$, $d_{R_2} = dist(R_1, R_2)$, $d_{R_3} = dist(R_1, R_3)$,...$d_{R_l} = dist(R_1, R_l)$, and $W_1 = 1/d_{R_1} = 1$, $W_2 = 1/d_{R_2}$,...,$W_l = 1/d_{R_l}$. Now, the weighted distance of each point in X is computed as: $d_{W_i} = \sum_{i=1,j=1}^{i=N,j=|\Re|} W_j * d_{x_{ij}}$, where $d_{x_{ij}} = dist(x_i, R_j)$. This gives N distance values. Divide the range of d_{W_i} into m equal bins. Assign each point to its corresponding bin according to the weighted distance, and sort the bins. The psuedocode of the this method is present in Algorithm 3, See the Figure 3.2.

Algorithm 3 Weighting the reference points

Input: Input data set X

Output: Reference Points Set R_l, Weights set W_l

1: Compute R_1= MinMax Reference Point (3/2*(min+max))
2: Compute distances of all the points from R_1
3: Determine the range of these distances.
4: Randomly select one point from X that is at a distance of d_{R_1}
5: Similarly, do the same for points those are at a distance of d_{R_2}, d_{R_3}, d_{R_4}, d_{R_5}, d_{R_6}.
6: Now, form the reference set from these randomly selected points: R_1 to R_6
7: Assign weight 1 to R_1, $(1*1/dr_1)$ weight to reference point that is at dr_1 distance and so on.
8: **return** R_l, W_l

In the search phase, for a given query q, first we compute distance between query and set of reference points $\Re$: $d_{q_j} = dist(q, R_j), j = 1,2,....l$. Now, weight these l distances by assigning weights specified above to get a scalar value: $d_{W_i} = \sum_{i=1}^{|\Re|} W_i * d_{q_i}$. Find the bin range in which d_{W_i} value is falling, retrieve the nearest neighbors of q from that bin.

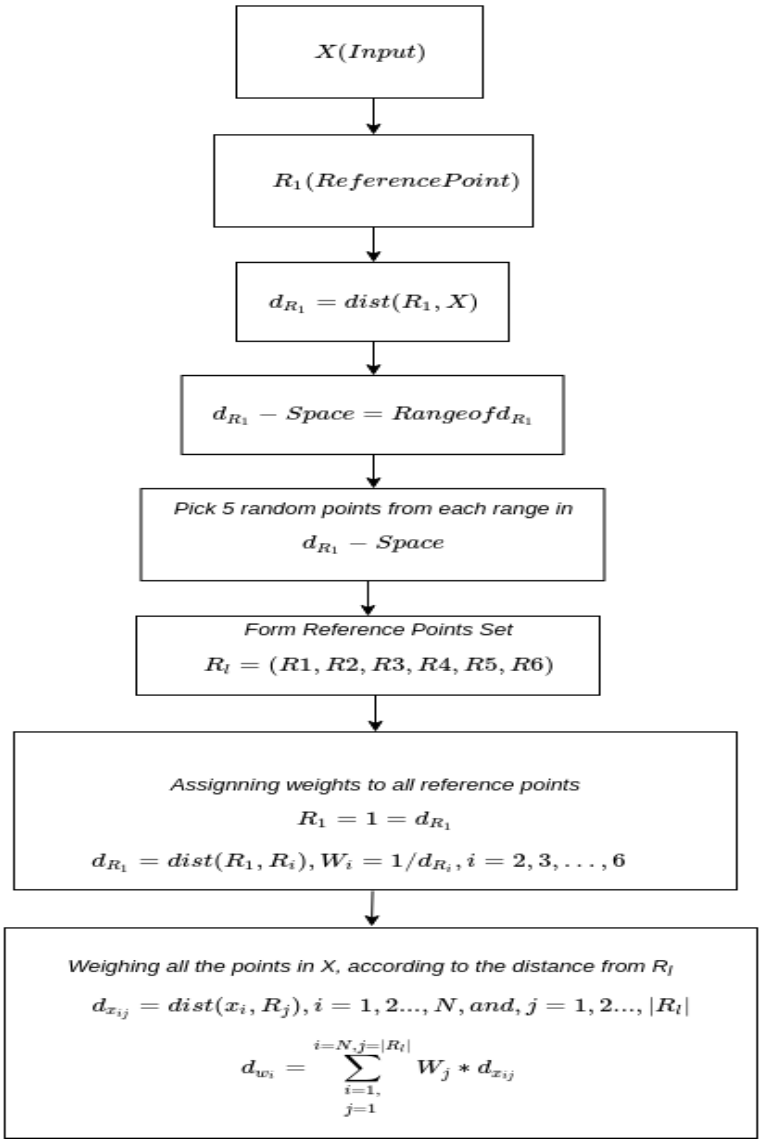

Figure 3.2: Weighting the Reference Points

The functioning of proposed technique (WSRPM) is equated with other partitioning methods like: Mean, MinMax, Set of Reference Points (SRPM). Precision (what percent of positive predictions were correct) is the performance measure we have taken for comparing these methods. From empirical study, one can say that the suggested (WSRP) method is showing good precision over the other methods.

This proposed partitioning method implemented by taking the mean as reference point, minmax as reference point (Pasunuri (2015)) and set of reference points without weighing for the purpose of performance comparison. When compared with the above said three methods, the proposed (WSRP) method is giving a good improvement in average precision. Algorithm for Construction (offline) phase is given in Algorithm 4.

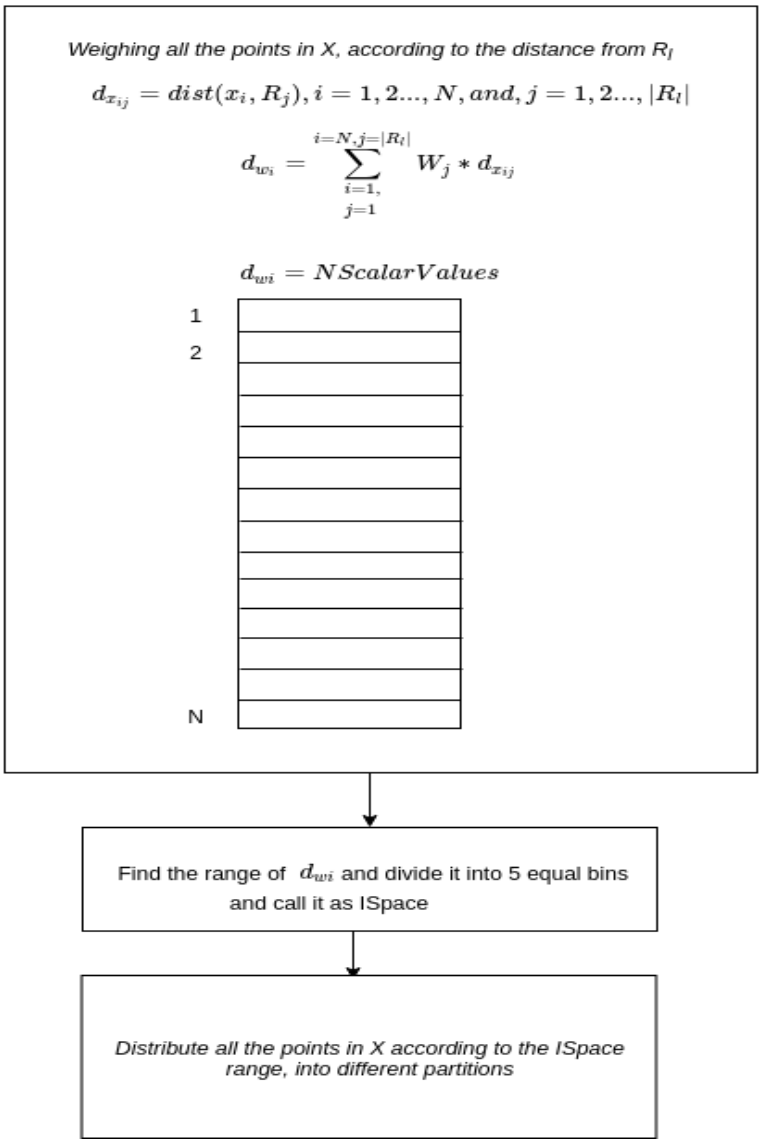

Figure 3.3: Construction Phase

Algorithm 4 depicts the proposed partitioning method with an initial assumption of five subgroups.

Search Algorithm

The process of searching the nearest neighbors for a given query is presented in Algorithm 5. In the search phase, which is also called online phase, we are given a query point q from the data space and we have to retrieve its nearest neighbor points from the data base. For this the Search algorithm calculates the distances to the query from the predefined set of reference points, and these distance values are weighted with W and summed to arrive at a scalar value d_q. This distance value is

Algorithm 4 Construction Phase

Assumptions: Let X be the data set of size N.
Input: Data Set X, Set of Reference points R,
l is the no.of ref.points,m is the no.of partitions,
W be the array of Weights.
Output: Partitions P, $dist$: array of distances.

for $i = 1$ to N **do**
  Compute $d(\mathbf{x_i}, R) = \sum_{k=1}^{l} W_k d_k$
end for
Find minimum(min) and maximum(max) of distance array d (min)
$IntervalLength = (max - min)/m$
for $j = 1$ to m **do**
  $range = min + IntervalLength$
end for
for $i = 1$ to N **do**
  for $j = range(1)$ to $range(m)$ **do**
    if $dist(i) \geq range(j)$ and $dist(i) \leq range(j+1)$ **then**
      Assign x_i to the j^{th} bin
    end if
  end for
end for
return P

compared with the range of bin boundaries so that we get location in the data space where its nearest neighbors are present.That location only we will search and report the nearest neighbors.

Algorithm 5 Search Phase

Input: query q, k (k-NNs), dataset X, weights W, Set of Reference Points R.
Output: k-NNs of a given query q

for $j = 1$ to l **do**
  Compute $d_j = d(q, R_j)$
end for
Compute $d_q = \sum_{i=1}^{l} d_i W_i$
for $i = 1$ to $size(range)$ **do**
  if $d_q \geq range(i)$ and $d_q \leq range(i+1)$ **then**
    Return k-nns of q from the partition P_i
  end if
end for
return k-NN's

3.2.2 Experimental Analysis

In this study, we used 5 high-dimensionsional data sets and one low-dimensional data set to assess the functioning of projected WSRPM technique for efficiently re-

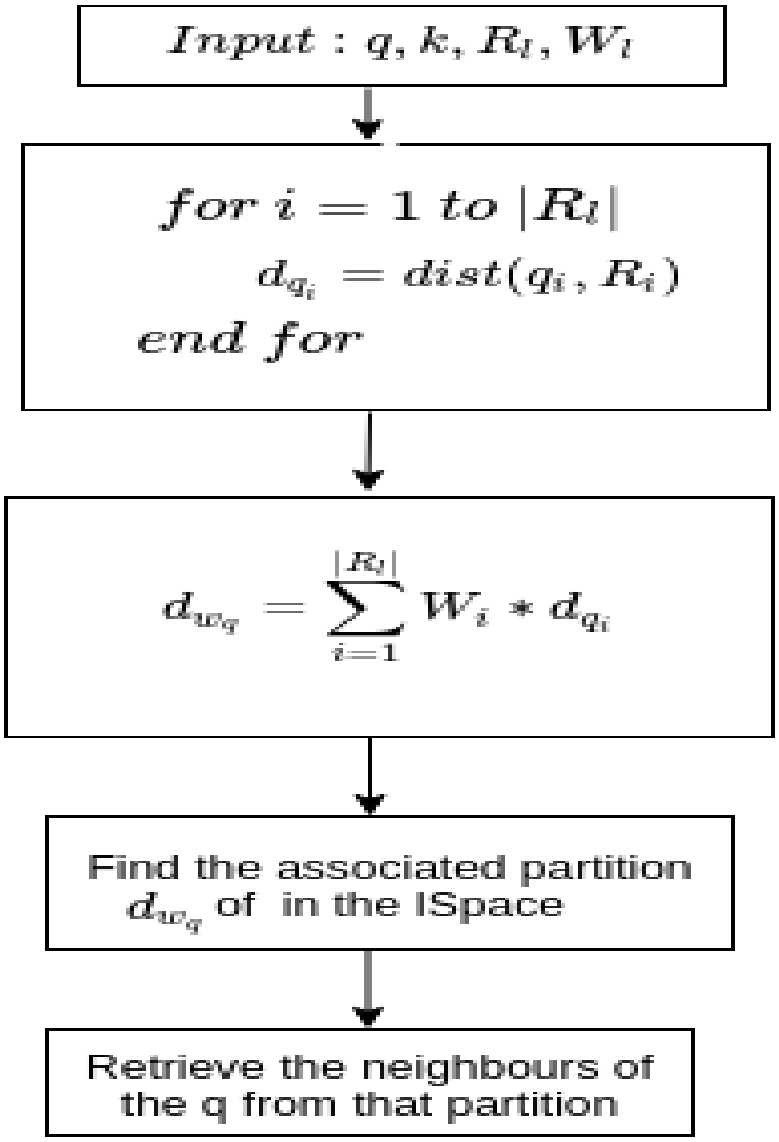

Figure 3.4: Search Phase

trieving nearest neighbors from the data. Table 3.3 gives the description of the datasets. A chemical (drug like structures) database from an online source ZINC John Irwin and Brian Shoichet (2005), is the only one low-dimensional data set and remaining all are high-dimensional data sets.

ZINC is a virtual screening database of chemical compounds. This provides a ZINCID and SMILES notation for each chemical structure in the data base.

ZINC provides 9 features by default which are physical properties, Rao et al. (2011) extracted 49 more feaures, summing to a toatl of 58 features, which are from different classes: physical, atom count, structural and functional. From these 58 features a subset of 28 features after excluding the functional groups are used by

(Chandrasekhar and Rani (2012)) for the experiments.

Yale, Leukemia, Lung and GCM are the other high-dimensional data sets used in experimentation, See Section 3.1.5 for details.

Dataset Name	# Samples	# Dimensions	# Classes
ORL	400	10304	40
Yale	165	1024	15
GCM	280	16063	2
Leukemia	72	7129	2
Lung	181	12533	2
ZINC5	596	7	5

Table 3.3: Specifications of Data sets.

We have applied the weighted set of reference points method to partition the data space into bins. Then performed online search with query samples of different size on various data sets.

The Precision is defined by the following formula:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{3.3}$$

Table 3.4 gives the average precision percentage for a group query of size 50. That is, these values are the average of 50 queries. From the results it is evident that the methods mean, minmax (Pasunuri (2015)), and Set of Reference Points Method are giving almost similar precision for the ZINC5 data set.

The Average Precision for the set of reference points and MinMax (Pasunuri (2015)) are also similar. The mean reference point method is 10% less than these two, and 30% less from that of Weighted Reference Points Method. The Weighted Reference Points method is having 30% more gain in performance.

For the ORL (AT & T Face images) database, the minmax and set of reference points methods are having same performance index, mean method is some what poor in performance and the proposed method is giving 70% precision.

For the Yale data set the three methods (mean, minmax, set of reference points) are giving same precision, and proposed method is giving almost 2-times better precision than the remaining three methods that are compared.

For the GCM data set the first three methods are giving an average precision of 71%, and proposed is 77%.

For the Luekemia data set mean method is giving 83.8% which is nearly 2% greater than the proposed.

For Lung data set, we got an average performance growth of 30% compared to the first three methods. From all these experimental results, WSRPM is giving good precision for almost all the given data sets.

Ref.Pt.Method	ZINC5	ORL	Yale	GCM	Luekemia	Lung
Mean	40	30	40	70	**84**	45
MinMax	42	40	39	71	80	48
Set of Ref.Pts.	41	38	42	72	81	46
Proposed (Weighted Set of Ref.Pts)	**68**	**70**	**80**	**77**	82	**76**

Table 3.4: Average Precision values of different Reference Point selection methods for partitioning the data, and these methods are tested on various data sets of low and high-dimensionality. Except for Luekemia, on all the remaining data sets, the proposed method (WSRP) is showing better average precision

From empirical results it is clear that the projected technique is functioning good on low-dimensional (ZINC5- which has only seven features) data set and also for all the five high-dimensional data sets. See Figure 3.5.

Pivot Method	ZINC5	ORL	Yale	Luekemia	Lung	GCM
BO- Heuristic	27	43	23	55	73	61
MinMax	56	52	23	54	80	75
WSRP(proposed)	**68**	**70**	**80**	**82**	**76**	**77**

Table 3.5: Average Precision of WSRP (proposed method, in bold face) method is compared with the two other existing methods namely, BO-Heuristic (Bozkaya and Ozsoyoglu (1999)) and MinMax (Pasunuri (2015)). Except for Lung, on all the remaining data sets, the proposed method (WSRP) is showing better average precision. The results are average of 10 runs

In another experiment, we equated the average precision of WSRP technique with two existing pivot-based methods namely, BO-Heuristic (Bozkaya and Ozsoyoglu (1999)) and MinMax (Pasunuri (2015)), and the results are presented in Table 3.5. The average precision percentage of WSRP method is higher than the compared two methods, for all data sets considered, except Lung data set.

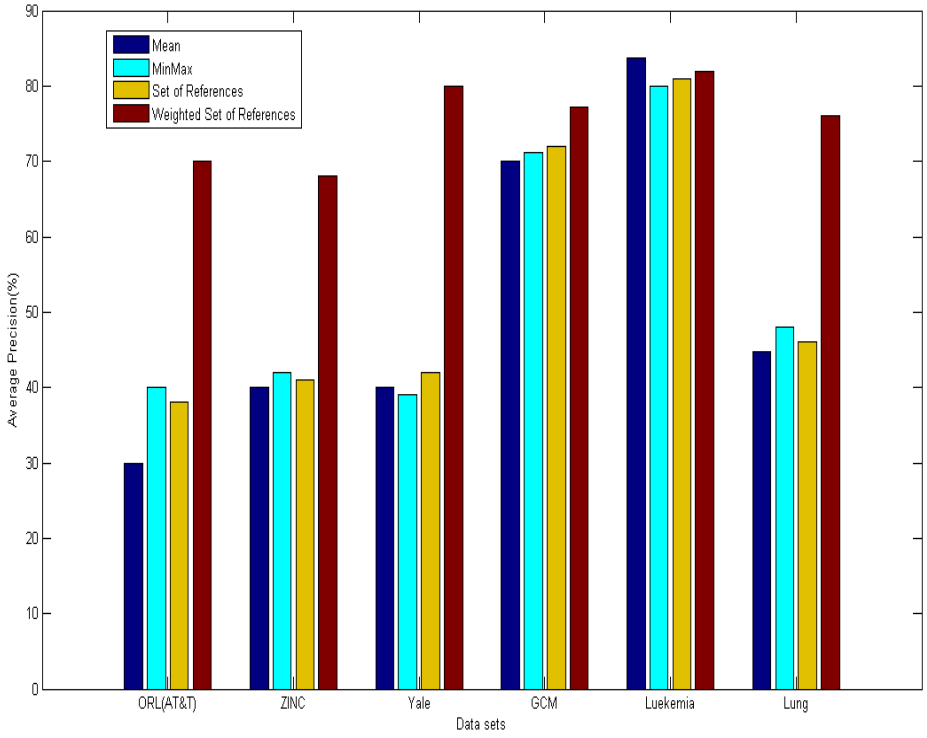

Figure 3.5: Performance Analysis for different Data sets

3.2.3 Summary

In this chapter we have presented two algorithms (which are pivot-based) which uses proximity for data space partitionin, and speeds up the nearest neighbor search.

The first method proposed is a pivot-based data partitioning method for Nearest Neighbor processing. It works by taking a farthest $(3/2*(min+max))$ point as reference point (pivot) for partitioning the data into subgroups. This helps in search phase, when we want to search for the nearest neighbors of a given query. Based on the distance between reference point to the query, we will move to a particular group in which we can find all the neighbors to the query, and the remaining groups or partitions need not to search. Thus, culling out the majority of the partititons and finding nearest neighbors of given query in a single group; thereby saving the computational cost and search time.

During the process of experiment, our study with empirical data showed that the projected technique was effective. When compared with the existing (BO-Heuristic), our method is able to reduce majority of the search space, and giving correct predic-

tions of a query.

In the second part of this chapter, we proposed a variant of the proximity-based data space partitioning method, that speeds up the nearest neighbor search by reducing the search space. This method works by taking multiple weighted reference points for partitioning the data into groups and returns the nearest neighbors of given query. The weighting of the reference points is done according to the distance from the data points to reference point set. The empirical results show that the WSRP technique is effective and expeditious in retrieving the nearest neighbors for a given query with less search space. The average precision percentage from the proposed method (WSRP) is compared with two other methods (BO-Heuristc and MinMax) and our method is outperforming the other methods.

CHAPTER 4

Random Projections for Dimensionality Reduction and Clustering in High-dimensional data

4.1 Ascending and Descending Order of Random Projections: Comparative Analysis of High-Dimensional Data Clustering

4.1.1 Introduction

A variation to the iterative random projections K-means (IRP-Kmeans) was proposed. IRP K-means efficiently performs clustering on HD data, which is based on RP, K-means algorithm. In this, clustering runs for many epochs, projection dimension increased in each epoch. In our method, we propose a variation in the projection of data, that is, instead of increasing the dimension, we gradually decrease the dimension in the successive iterations, which reduces the computational complexity. IRP K-means is a simulated annealing type of projection wherein we reduce the dimension in one step from HD to LD. Whereas, in our proposed variation we reduce the dimension to a target dimension from HD to LD in successive iterations. These iterations stop when a stopping criterion is met.

The proposed variant of IRP-K-means has been validated experimentally on various data sets. IRP K-means (Cardoso and Wichert (2012)) is a fusion of reduction (RP) and clustering (K-means) algorithm. It starts with a chosen low-dimension and gradually increases the dimensionality in each K-means iteration. K-means is used in each iteration on RP-reduced data. The proposed variant, in contrast to the IRP K-means, starts with the high dimension and gradually reduces the dimensionality while honouring the Johnson-Lindenstrauss lemma in each iteration.

VIRP-K-means (proposed variant) tested on five HD data sets. Of these, two are image and three are gene expression data sets. Comparative Analysis is carried out

for the cases of K-means clustering using RP-Kmeans and IRP-Kmeans. The analysis is based on K-means objective function, that is the mean squared error (MSE). It indicates that our variant of IRP K-means method is giving good clustering performance compared to the previous two (RP and IRP) methods. Specifically, for the AT & T Faces data set, our method achieved the best average result (9.2759×10^9), where as IRP-Kmeans average MSE is 1.9134×10^{10}. For the Yale Image data set, our method is giving MSE 1.6363×10^8, where as the MSE of IRP-Kmeans is 3.45×10^8. For the GCM and Lung data sets we have got a performance improvement, which is a multiple of 10 on the average MSE. For the Luekemia data set, the average MSE is 3.6702×10^{12} and 7.467×10^{12} for the proposed and IRP-Kmeans methods respectively. In summary, our proposed algorithm is performing better than the other two methods on the given five data sets.

4.1.2 Basic Concepts and Related Work

The K-means (Lloyd (1982)) is a clustering algorithm in which the data with N points given in R^d and an integer K is specified. The algorithm finds K cluster centers such that the average distance from a point to its cluster center is minimum. It starts by initializing the centers by randomly. That is, it selects K cluster centers randomly. Then, each point is compared with all these centers and the point is assigned to its closest cluster center. Now the new mean of each cluster is calculated and these mean points are now become the new cluster centers. Likewise the cluster centers are updated regularly after each iteration by taking the mean of each cluster. That is every iteration the means are recalculated and all the points are reassigned to its closest new center. The total mean squared error is reduced in each of the iteration. The algorithm converges when it reaches the minimum squared error. The disadvatage of K-means is that it can be caught in local minimum.

Random Projection (RP) (Johnson and Lindenstrauss (1984)) is a very famous and powerful technique for dimensionlaity reduction. It uses matrix multiplication to map data onto a smaller embedding space, by utilizing a random matrix for projection, by this mapping, distance between the data points is approximately preserved.

(Fradkin and Madigan (2003)) conducted a comparative analysis on the combination of PCA and RP with SVM, decision trees (DT) and proximity (k-NN) methods.

Bingham and Mannila (Bingham and Mannila (2001)) is another work in the literature, in which different dimensionality reduction methods have been compared for image and text data. The distortion rate and computational complexity are considered as performance parameters. In this work, they use RP to reduce dimensionality of image and text data. RP is compared with PCA, SVD, LSI and Descrete cosine transform (DCT), on datasets from different domains. Noisy and noiseless images, newsgroup text documents are used for testing and comparison. The distortion proportion and computational complexity are the measures for performance comparison. Their emperical results show that the random projection preserves the data similary very well even at a moderate projecting (embedding) dimensions, and projections computation is also fast.

Fern and Brodley (2001) used RP and ensemble methods for HD data clustering quality improvement. They use RP for unsupervised learning, by using RP for clustering HD data using multiple random projections with ensemble methods. They equated the proposed method with single random projection and PCA for EM clustering. Their method outperforms PCA with better clusters on all data sets.

Deegalla and Bostrom Deegalla and Bostrom (2006) applied PCA and RP for Nearest Neighbor Classifier to report the advantage of performance increse when dimensions grow fastly. In this study, feature reduction is achieved through PCA, RP, then on this feature reduced space kNN classifier is applied. They have taken image, micro array data sets for this study. The study reports that PCA performance is more dependent on the number of reduced dimensions. PCA accuracy decreases after reaching a maximum as dimension increasing, whereas accuracy increases with the growing dimension, in case of RP.

Cardoso and Wichert (2012) proposed IRP+K-means, which is an iterative version of RP+K-means algorithm for HD data clustering. K-means objective function (i.e. Mean squared error (MSE)) and running time (number of iterations clustering takes to converge) are the two parmaeters used in performance analysis.

We proposed a variant to IRP K-means (called VIRP K-means) method that per-

forms clustering of HD data using random projections in the iterative dimensions of IRP K-means algorithm Cardoso and Wichert (2012), in this work. The performance of VIRP Kmeans is compared with the related methods namely, IRP K-means, RP K-means. From the empirical results, we can say that the performance (mean squared error) of VIRP Kmeans is improved when compared to RP Kmeans and IRP Kmeans methods. Results of the conducted experiments reveal that gradual decrease in the reduced dimensionality and then clustering on that LD space achieves high clustering quality.

See Sections 2.5, 2.6 for more details on random projection technique, K-means clustering respectively.

4.1.3 RP K-means

Several researchers combined the K-means clustering algorithm with random projection (Boustsidis et al. (2010), Dasgupta (2000), Li et al. (2006)). The fundamental idea to project the original HD data into a LD space and then perform clustering on this low-dimensional subspace. This reduces the K-means iteration cost effectively. The solution we get in low-dimensional space is same as the one in the high-dimension.

The RP K-means, first initializes cluster membership G randomly. Then generate a random matrix P to map input data. Map input data $X_{N\times d}$ to D dimensions where $D < d$, using the projection matrix $P_{d\times D}$. The initial cluster centers C^{RP} defined by the mean of each cluster in X^{RP} with the help of projected data $X^{RP}_{N\times D}$ and G. We apply K-means clustering upto convergence or we will stop based on some stopping condition. The details of this method is described in Algorithm 6, See Cardoso and Wichert (2012).

4.1.4 Iterative version of RP K-means

It is an iterative algorithm proposed by Cardoso and Wichert (2012). This algorithm increases the dimension of the space in each iteration so that the local minimums are avoided in the original space. Each solution, constructed in one iteration, is used in the following iteration; thereby saving the computations. This is same

Algorithm 6 Random Projection K-means

Input: Reduced dimension D, $X_{N\times d}$, K
Output: G (cluster membership).
begin
  Partition X randomly into K groups.
  Set a random matrix $P_{d\times D}$
  Set $X^{RP}_{N\times D} = X_{N\times d} P_{d\times D}$
  Set $C^{RP}_{k\times D}$ by finding the mean of each cluster in $X^{RP}_{N\times D}$ according to G.
  Find G with K-means on $X^{RP}_{N\times D}$ with $C^{RP}_{k\times D}$ as initialization.
  return G
end

as simulated annealing clustering, Selim and Alsultan (1991). The wrong cluster assignments are reduced as dimensionality is increased i.e. the chance of falling a point into a wrong cluster is reduced as the dimensionality is increased. A wrong cluster is identified by the euclidean distance from center to the point in the original space. The algorithm is same as RP K-means, but here the projection and clustering is applied in many iterations. The projection dimension is increased in each iteration. The clusters in the previous iterative dimension are the base for initializing the clusters in the present dimension.

The algorithm randomly partitions the input data set X and initializes as cluster membership G. The algorithm starts in dimension D_1, Initial centroids are the means of the initial clusters. The input data X is projected onto a $D_1(D_1 < d)$ dimension space by random projection P_1, obtaining X^{RP_1}. K-means clustering is performed in X^{RP_1} to get the new cluster membership G, and this G will become the basis for next dimension (D_2) for initilizaing K-means. Again compute centroids now in dimension $D_2(D_1 \leq D_2 < d)$ by using the cluster membership G obtained from K-means in dimension D_1 and X^{RP_2} to obtain the new initial centroids C^{RP_2}, in a new D_2 dimensional space. Now in D_2, we perform K-means clustering again using C^{RP_2} as initialization. This process is repeated until the last $D_l(D_1 \leq D_2 \leq ... \leq D_l < d)$ is reached, returning the cluster membership from D_l. This algorithm is based on a heuristic relation $D_1 \leq D_2 \leq ... \leq D_l < d$ which is analogous to simulated annealing cooling. The procedure is presented in Algorithm 7.

Algorithm 7 Iterative RP K-means

Input: Dimensions list $D_a = 1,2,3,...,l$, Dataset $X_{N\times d}$, K (clusters)
Output: G (cluster membership).
begin
  Partition X into K random clusters and assign as G.
  for $D_a = 1$ to l **do**
    Specify $P_a(d \times D_a)$ (random matrix)
    Set $X^{RP_a}(N \times D_a) = XP_a$
    Set $C^{RP_a}(K \times D_a)$ by finding the mean of each cluster in X^{RP_a} according to G.
    Apply K-means on X^{RP_a} with C^{RP_a} as initialization to get G.
  end for
  return G
end

4.1.5 Proposed variant of IRP-Kmeans

The proposed variation is based on Iterative dimension reduction using random projections and K-means Algorithm. But instead of gradually increasing the dimension, we decrease the dimension from the high-dimension to low-dimension in the random projection part of the algorithm. Similar to IRP-Kmeans, we try to capture the solution constructed in one iteration and use it in subsequent iteration. In this way, it transfers the characteristics of previous generation to following generation. In our experiment, we ran our method for the reduced dimensions from the list $(d, d/2, d/4, d/8)$. See Algorithm 8.

Algorithm 8 Proposed Variant of IRP-K-means

Input: list of dimensions $D = (d/2, d/4, d/8)$,
Data Set $X_{N\times d}$,
No. of clusters K
Output: G which is cluster membership.
begin
1: Partition X randomly and assign cluster centroids as G.
2: Set $D_a = d/2$
3: Specify random matrix $P_a(d \times D_a)$
4: Set $X^{RP_a}(N \times D_a) = XP_a$
5: Set $C^{RP_a}(K \times D_a)$ by finding the mean of each cluster in X^{RP_a} according to G.
6: If $D_a < d/8$
7: $D_a = D_a/2$
8: and Goto STEP 3
9: Apply K-means on X^{RP_a} with C^{RP_a} as initialization to get G.
10: **return** G
end

4.1.6 Experimental Study

Performance of the proposed variant of IRP-K-means is analyzed is done on five high-dimensional data sets, two image (AT & T, Yale), three micro array (also called gene expression data) datasets, which are: GCM, Leukemia and Lung.

The mean squared error (MSE) which is the objective function of K-means clustering is taken as a measure to report the performance of the proposed method.

4.1.7 Data Sets

In this study, we considered five high-dimensional data sets to evaluate the performance of the proposed variation of IRP-K-means algorithm. See Section 3.1.5 for details. A detailed specifications of the data sets are presented in Table 4.1.

Dataset Name	No. of Samples	No. of Features	No. of Classes
AT&T Faces (ORL)	400	10304	40
Yale	165	1024	15
GCM	280	16063	2
Leukemia	72	7129	2
Lung	181	12533	2

Table 4.1: Specifications of data sets

4.1.8 Results and Discussion

Using Theorem 1, we have calculated the bound for the data sets that are considered for experimentation. The ϵ value is fixed at 0.99 in all the experiments. The MSE for several data sets with the implementation of Cardoso and Wichert (2012) and by using Achlioptas Random matrix (our own implementation), we got almost similar results, except for the Lung data set with a difference of 10^1 times in the MSE for AT & T Faces, Lung and GCM data sets. These results are presented in Table 4.2.

The average MSE over 20 runs for the proposed variant along with two other methods is shown in Table 4.3. From this, it is evident that the proposed variant outperforms the IRP-K-means method on the given five high-dimensional data sets. When compared with RP-K-means, the performance of the proposed one is almost

same for all the data sets considered except GCM, wherein the The performance of VIRP is doubled for GCM data set when compared with RP-Kmeans Algorithm. VIRP is showing 6 times improvement than IRP, when GCM data set is considered. It is double of IRP on first four data sets, and it is 10 times improved for Lung data set.

Data set	D	IRP-Kmeans (Classical Normal matrix)	IRP-Kmeans (Achlioptas random matrix)
AT&T Faces (ORL)	221	7.8850×10^8	8.1216×10^8
Yale	166	1.2311×10^8	1.459×10^8
GCM	234	4.5467×10^{11}	4.9832×10^{11}
Leukemia	212	4.1263×10^{11}	4.1620×10^{11}
Lung	226	10.88×10^{10}	4.43×10^{10}

Table 4.2: MSE for several datasets. When the dimensionality of the data is reduced from original dimension to JL Limit (D), The results reported are sample average over 20 runs.

S.No.	Data sets	RP	IRP	Proposed (VIRP)
1	AT&T Faces(ORL)	8.53×10^9	19.134×10^9	9.2759×10^9
2	Yale	1.61×10^8	3.45×10^8	1.6363×10^8
3	GCM	1.20×10^{13}	1.551×10^{13}	0.7444×10^{13}
4	Leukemia	4.17×10^{12}	7.467×10^{12}	3.6702×10^{12}
5	Lung	1.19×10^{12}	13.3×10^{12}	1.309×10^{12}

Table 4.3: MSE of the proposed method (IRP-K-means variant) is compared with two other methods namely: RP-K-means, IRP-K-means for several datasets. The values reported are sample average 20 runs.

4.1.9 Summary

We proposed a variant for IRP K-means algorithm by gradually decreasing the dimension in each iteration thereby preserving the inter-point distances efficiently. This can be confirmed by the empirical results presented above. Our method is compared with the Single Random Projection (RP), IRP K-means (IRP) methods. Compared to these two methods, our proposed method is giving best results for the given HD datasets. Using other methods to generate random matrix and verify if the method preserves the inter-point distances will be the future research work. And also to carry out comparative analysis of the proposed method with some standard clustering algorithms.

4.2 Clustering High-Dimensional Data: A Reduction - level Fusion of PCA and Random Projection

Principal Component Analysis (PCA) is a very famous and commonly used statistical technique, which is also an unsupervised DR method. K-means centroid based clustering technique used in learning tasks which are unsupervised. Random Projection (RP) is a widely used dimension reduction technique. The basic idea of RP is to change the high-dimensional data representation into a much lower-dimensional subspace, and also to ensure preservation of proximity among the points. Here we prove the effectiveness of these methods by combining them for efficiently clustering the low as well as high-dimensional data. Our proposed algorithm works by combining PCA with RP to lessen the dimensionality of the data set, then performs K-means clustering in the reduced space. We compare the proposed algorithm performance with simple K-means, PCA reduced K-means algorithms on 12 bench mark datasets. Of these, 4 are LD and 8 are HD datasets. Our proposed algorithms outperforms the other methods.

PCA is a data analysis technique, and also a dimension reduction method which was introduced by Pearson in 1901 (Pearson (1901)).

It is used to reduce the data from HD to LD with maximum variance preservation. The applications of PCA include, Data Compression, Data Visualization, Feature Extraction and so on. K-means (Lloyd (1982)) is a clustering algorithm in which the data with N points given in R^d and an integer K is specified. The algorithm finds K cluster centers such that the mean squared error is minimized. It starts by initializing the centers by randomly selected K points. In each iteration, every point is checked with all the centers to find its relevant cluster, this creates new clusters, new centers are computed after completion of an iteration. Mean of points defines cluster center. Objective of an iteration is to reduce the total squared error. This error becomes minimum when the algorithm reaches convergence state, but it need not be a global minimum every time.

4.2.1 Related Work

Many researchers worked on to combine two or more dimensionality reduction methods before applying a clustering or classification task on the data. Here the dimensionality reduction process acts as a pre-processing step to ease the eventual clustering or classification task, See Section 4.1.2.

Ding and He (2004), has proved that the principal components are the continuous solutions to the for K-means clustering. They related unsupervised dimension reduction with unsupervised learning, and they claim that these both are complement to each other.

Qi and Hughes (2012) has shown that, PCA on random projections (of low-dimensions) gives the same results as PCA on the original data set with certain conditions. The empirical results from both synthetic and real-world data sets show that, PCA when applied on RP reduced data, then it successfully recovers center of data, and also princiapl components.

In this work, we have combined PCA and RP with K-means clustering to enhance clustering performance. Here we have done a two-step preprocessing of the input data that is applying PCA first to get reduced directions then apply Random Projection on these reduced principal components, and also vice-versa. Then performing clustering on the reduced data.

We have implemented PCA and RP in the preprocessing step of fusion method. See Section 2.4, 2.5 for more details on PCA and RP respectively.

4.2.2 K-Means Clustering

K-means is prototype-based clustering algorithm. It finds k groups/clusters in the given database. This iterates through N observations and divides them into K independent clusters. It first initializes K random points as cluster centroids. Then each point is tested for its closest centroid, and the point is assighned to that cluster. This is repeated for all the points. Now K clusters are formed. Now the new cluster centers are defined as the mean of each cluster. This process repeated until there is no change in the cluster centers. For more details on K-means clustering, refer to

Section 2.6.

4.2.3 Fusion of dimensionality reduction methods for Clustering High-Dimensional Data

This section describes the proposed fusion (of DR methods) algorithms for performing clustering on the high-dimensional data efficiently. Most clustering algorithms (like K-means) present in the literature are basically distance based. These algorithms which are meant for low-dimensioanl data clustering, cannot produce significant clusters in HD data because many features are redundant, useful patterns can be found in the subspaces with small dimensionality, Bouveyron et al. (2007), Assent (2012). To make the conventional clustering algorithms suitable for HD data clustering, first convert data into a LD space then apply clustering task on this projected (dimension reduced) points. We have proposed fusion of algorithms in this work to perfrom clustering in HD data efficiently and effectively. In Algorithm 9, PCA is first applied on the original HD data, then K-means is applied on the reduced data. We set $D < d$ in Algorithm 9.

In Algorithm 10, PCA is applied then RP is applied on the PCA resulted reduced space, then K-means is applied to this reduced space (PCA$^{\text{rp}}$ K-means).

PCA mapping is analogous to kernel-based clustering.

Given two data points a and b and the PCA defines a map ϕ from R^d input space to R^D reduced/feature space

$$\phi : R^d \rightarrow R^D. \tag{4.1}$$

Euclidean distance between x, y is defined in input space as:

$$d(a,b) = \sqrt{\|a-b\|^2} \tag{4.2}$$

After the mapping (to reduced space), the euclidean distance is:(Zhang and Cao (2011), Alshamiri et al. (2015))

$$d_{R^D}(a,b) = \sqrt{\left\|\phi(a)-\phi(b)\right\|^2} \tag{4.3}$$

Equation 4.2 can be replaced by Equation 4.3 as of projected space. Similarly, proposed PCA K-means can be potrayed as a mapping from HD to LD features and then applying clustering in that LD space.

PCA K-means consists of 3 steps and is described in Algorithm 9. Let X be the input data, G be the cluster membership vector. Here, we first perform PCA on primary input X, to get reduction space. Then we execute K-means on reduced space to get the clustering results (G). The basic idea is, instead of applying K-means clustering on the original high-dimensional data, first we reduce the dimensionality of the data into low-dimensional one, and then we perform K-means clustering in reduced space.

Algorithm 9 PCA K-means

Input: Data set $X_{N\times d}$, K (clusters), Reduced Dimension D
Output: G (cluster membership)
begin
1: Apply PCA on the original input X
2: Perform K-means in PCA reduced space
3: **return** G
end

The pseudo code for PCA+RP-Kmeans Algorithm is given in Algorithm 10:

Algorithm 10 PCA+RP-K-means

Input: Data Set $X_{N\times d}$, K (clusters), Reduced Dimension D

Output: G (cluster membership).

begin

1: Apply PCA on $X_{N\times d}$ to get $X^{pc}_{N\times d_1}$
2: Set random projection matrix $P_{d_1\times D}$
3: Set $X^{RP}_{N\times D} = X^{pc}_{N\times d_1} P_{d_1\times D}$
4: Perform K-means clustering on the resultant $X^{RP}_{N\times D}$
5: **return** G

end

The RP+PCA-Kmeans Algorithm described in Algorithm 11:

Algorithm 11 RP+PCA-K-means

Input: Data Set $X_{N\times d}$, K (clusters), Reduced Dimension D

Output: G (cluster membership).

begin

1: Set a random projection matrix $P_{d\times d_1}$

2: Set $X^{RP}_{N\times d_1} = X_{N\times d} P_{d\times d_1}$

3: Apply PCA on $X^{RP}_{N\times d_1}$ to get $X^{pc}_{N\times D}$

4: Apply K-means on resultant $X^{pc}_{N\times D}$

5: **return** G

end

In the Algorithm 11, N is size of database, d gives original size of features present in the input data set. d_1 is the size of the reduced dimension to which the input data set is projected by random projection and $d_1 < d$. D is the final reduced dimension after applying PCA on the RP reduced space.

This algorithm starts by reducing the dimensionality of input data from d to d_1 by applying random projection. By this the $X_{N\times d}$ becomes ${X^{\mathrm{RP}}}_{N\times d_1}$. Then we apply PCA on this RP reduced space to get $X^{pc}_{N\times D}$. Now, we perform K-means clustering on this reduced matrix to get the clusters. Then we report clustering pefromance in the form of Mean Squared Error (MSE) as performance measure for comparing the methods.

4.2.4 Empirical Study

We equate the functioning of projected technique on eight high-dimensional and four low-dimensional data sets. Performance measure used is MSE, which is objective function of K-means clustering. The low the MSE, the higher the clustering accuracy.

Data Sets

In this study, we have considered eight HD data sets and four LD data sets to assess the functioning of presented hybrid (fusion) algorithms. A detailed specifica-

tions are in Table 4.4. Their description is as follows:
HD data sets: COIL20 (Columbia Object Image Library) comprises of 72 gray-scale images belongs to 20 classes. Colon has 62 samples, each with 200 genes. This data samples can be classified into 22 normal and 40 tumor tissue samples. Prostate cancer data set contains 136 tissue samples: 72 tumor, 59 normal.

Along with the above, we have used ORL, Yale, GCM, Leukemia and Lung data sets, for details of these, see Section 4.1.7

LD data sets: The dataset Iris compose of 150 iris flower samples. For each flower, we have four measurements: sepal length, sepal width, petal length and petal width, giving 150 points $x_1, x_2, ...x_{150} \in R^4$. The data points are in 4 dimensions. Wine dataset contains 178 samples and 13 dimensions. ZINC is a data repository of chemical structures. This database has $8,783,230$ chemical compounds. From this we have taken 50000 samples randomly with 7 features (ZINC7) as one data set and with 28 features (ZINC28) as another data set for our experiments.

S.No.	Data set	Patterns	Features	Classes
1	AT&T Faces (ORL)	400	10304	40
2	Yale	165	1024	15
3	GCM	280	16063	2
4	Leukemia	72	7129	2
5	Lung	181	12533	2
6	COIL20	1440	1024	20
7	Colon	62	2000	2
8	Prostate	136	12600	2
9	Iris	150	4	3
10	Wine	178	13	3
11	ZINC7	50000	7	-
12	ZINC28	50000	28	-

Table 4.4: Specifications of data sets

Results and Discussion

All the results we have reported here are the average of 20 independent runs. These results are shown in Table 4.5. The Performance of the K-means is compared with the PCA-Kmeans and PCA-RP-Kmeans on low-dimensional data is shown in Figure 4.1. From the Figure 4.1, it is evident that, PCA-Kmeans is better than K-

means for the low-dimensional data sets considered. The clustering performance of PCA-RP-Kmeans is far better than that of K-means on all LD data sets studied except ZINC28, for which both the methods are same in the performance.

From the Figure 4.2, we can say that, the PCA-RP-Kmeans is giving better clusters compared to K-means. PCA-K-means is showing good results for GCM, Luekemia, Yale and Lung datasets when compared with K-means. For Prostate, COIL20 and ORL data sets, both the methods are giving almost similar performance. For the Colon data set, K-means(3.16×10^7) is performing better compared to PCA-Kmeans (4.6×10^7). When comapred with PCA-K-means, the PCA-RP-K-means is giving better performance with a 5 times improvement for Leukemia dataset, 4 times improvement for ORL and COIL20 datasets, 3 times improvement for GCM and Yale data sets, 2 times improvement for Colon, Lung and Prostate data sets. The overall performance of PCA-RP-K-means is much better than PCA-K-means method.

Data set	Classic K-means	PCA-K-means	PCA-RP-Kmeans
Iris	0.69	0.077	0.441
Wine	1231	310	1185
ZINC7	309	58	136
ZINC28	99	75	99
AT and T Faces (ORL)	6.51×10^6	6.668×10^6	1.61×10^6
Yale	2.39×10^6	9.162×10^5	2.66×10^5
GCM	6.92×10^9	2.6575×10^9	0.823×10^9
Leukemia	5.02×10^9	1.4101×10^9	2.96×10^8
Lung	9.55×10^8	3.183×10^8	1.21×10^8
COIL20	16.424	17.381	4.51947
Colon	3.16×10^7	4.6×10^7	1.82×10^7
Prostate	15×10^7	14.8×10^7	7.24×10^7

Table 4.5: MSE for several datasets. Sample average over 10 runs.

On ORL dataset, the average mean squared error (MSE) of RP-K-means is increased when the reduced dimension (D) is gradually increasing, and the proposed fusion method (RP+PCA-K-means) MSE is gradually decreasing with increase in D,

Reduced dimension (D)	**Avg. MSE of RP-K-means**	**Avg. MSE of RP+PCA-K-means**
10	1.43×10^3	5.08×10^{-1}
20	3.43×10^3	3.17×10^{-1}
50	9.40×10^3	2.29×10^{-1}
100	19.4×10^3	1.71×10^{-1}

Table 4.6: Average MSE for ORL dataset.

Reduced dimension (D)	Avg. MSE of RP-K-means	Avg. MSE of RP+PCA-K-means
10	214	1.68×10^{-1}
20	447	1.29×10^{-1}
50	1263	5.83×10^{-2}
100	2565	4.34×10^{-2}

Table 4.7: Average MSE for Yale dataset.

Reduced dimension (D)	Avg. MSE of RP-K-means	Avg. MSE of RP+PCA-K-means
10	187	1.18×10^{-1}
20	436	7.91×10^{-2}
50	1195	5.43×10^{-2}
100	2487	4.07×10^{-2}

Table 4.8: Average MSE for COIL20 dataset.

Reduced dimension (D)	Avg. MSE of RP-K-means	Avg. MSE of RP+PCA-K-means
10	563	4.29
20	1164	2.87
50	2908	2.47
100	6091	3.83

Table 4.9: Average MSE for Colon dataset.

Reduced dimension (D)	Avg. MSE of RP-K-means	Avg. MSE of RP+PCA-K-means
10	2144	27
20	4326	19
50	11316	10
100	22878	13

Table 4.10: Average MSE for Leukemia dataset.

Reduced dimension (D)	Avg. MSE of RP-K-means	Avg. MSE of RP+PCA-K-means
10	2129	37
20	4165	28
50	10852	15
100	21619	9

Table 4.11: Average MSE for Lung dataset.

Reduced dimension (D)	Avg. MSE of RP-K-means	Avg. MSE of RP+PCA-K-means
10	1696	30
20	3462	22
50	9007	16
100	17972	21

Table 4.12: Average MSE for Prostate dataset.

Reduced dimension (D)	Avg. MSE of RP-K-means	Avg. MSE of RP+PCA-K-means
10	2011	34
20	4021	24
50	10298	14
100	21499	9

Table 4.13: Average MSE for GCM dataset.

Reduced dimension (D)	Avg. MSE of RP-K-means	Avg. MSE of RP+PCA-K-means
1	2.64×10^{-2}	3.28×10^{-2}
2	1.01×10^{-1}	3.11×10^{-2}
3	1.39×10^{-1}	1.83×10^{-2}

Table 4.14: Average MSE for Iris dataset.

Reduced dimension (D)	Avg. MSE of RP-K-means	Avg. MSE of RP+PCA-K-means
2	0.33	5.24×10^{-2}
4	0.94	2.26×10^{-2}
6	1.34	2.00×10^{-2}
8	1.89	1.18×10^{-2}
10	2.51	1.12×10^{-2}
12	3.03	1.14×10^{-2}

Table 4.15: Average MSE for Wine dataset.

Reduced dimension (D)	Avg. MSE of RP-K-means	Avg. MSE of RP+PCA-K-means
1	2.24×10^{-3}	2.99×10^{-3}
3	5.30×10^{-2}	7.14×10^{-4}
5	1.28×10^{-1}	6.11×10^{-4}
7	2.17×10^{-1}	5.29×10^{-4}

Table 4.16: Average MSE for ZINC7 dataset.

Reduced dimension (D)	Avg. MSE of RP-K-means	Avg. MSE of RP+PCA-K-means
1	0.03	2.22×10^{-2}
7	2.21	5.43×10^{-3}
14	5.41	3.79×10^{-3}
21	7.72	3.80×10^{-3}
28	11.27	4.72×10^{-3}

Table 4.17: Average MSE for ZINC28 dataset.

See Table 4.6.

From Table 4.7, it is clear that on Yale dataset, the MSE of the RP-K-means is increasing from 214 to 2565 when increasing the reduced dimensionality ($D = 10$, 20, 50, 100), whereas the MSE of the proposed method is gradually decreasing from 0.17 to 0.04 for the increasing D value.

From Table 4.8, it is clear that on COIL20 dataset, the MSE of the RP-K-means is increasing from 187 to 2487 when increasing the reduced dimensionality ($D = 10$, 20, 50, 100), whereas the MSE of the proposed method is gradually decreasing from 0.12 to 0.04 for the increasing D value.

From Table 4.9, it is clear that on Colon dataset, the MSE of the RP-K-means is increasing from 563 to 6091 when increasing the reduced dimensionality ($D = 10$, 20, 50, 100), whereas the MSE of the proposed method is gradually decreasing from 4.29 to 2.47 for the increasing D value.

From Table 4.10, it is clear that on Leukemia dataset, the MSE of the RP-K-means is increasing from 2144 to 22878 when increasing the reduced dimensionality ($D =$ 10, 20, 50, 100), whereas the MSE of the proposed method is gradually decreasing from 27 to 13 for the increasing D value.

From Table 4.11, it is clear that on Lung dataset, the MSE of the RP-K-means is increasing from 2129 to 21619 when increasing the reduced dimensionality ($D = 10$, 20, 50, 100), whereas the MSE of the proposed method is gradually decreasing from 37 to 9 for the increasing D value.

From Table 4.13, it is clear that on GCM dataset, the MSE of the RP-K-means is increasing from 2011 to 21499 when increasing the reduced dimensionality ($D = 10$, 20, 50, 100), whereas the MSE of the proposed method is gradually decreasing from 34 to 9 for the increasing D value.

From Table 4.14, it is clear that on Iris dataset, the MSE of the RP-K-means is increasing from 0.03 to 0.14 when increasing the reduced dimensionality ($D = 10$, 20, 50, 100), whereas the MSE of the proposed method is gradually decreasing from 0.03 to 0.02 for the increasing D value.

From Table 4.15, it is clear that on Wine dataset, the MSE of the RP-K-means is

increasing from 0.33 to 3.03 when increasing the reduced dimensionality ($D = 10$, 20, 50, 100), whereas the MSE of the proposed method is gradually decreasing from 0.05 to 0.01 for the increasing D value.

From Table 4.16, it is clear that on ZINC7 dataset, the MSE of the RP-K-means is increasing from 0.002 to 0.22 when increasing the reduced dimensionality ($D = 10$, 20, 50, 100), whereas the MSE of the proposed method is gradually decreasing from 0.003 to 0.0005 for the increasing D value.

From Table 4.17, it is clear that on ZINC28 dataset, the MSE of the RP-K-means is increasing from 0.03 to 11.27 when increasing the reduced dimensionality ($D = 10$, 20, 50, 100), whereas the MSE of the proposed method is gradually decreasing from 0.02 to 0.04 for the increasing D value.

4.2.5 Summary

In the first part of this chapter, we have proposed a variant for IRP K-means algorithm by gradually decreasing the dimension in each iteration thereby preserving the inter-point distances efficiently. This has proved by the empirical results. Our propopsed method is compared with the Single Random Projection (RP), IRP K-means (IRP) methods. Compared to these two methods, our proposed method is giving best results for the given high-dimensional data sets.

In the second part, we have incorporated different dimensionality reduction methods (like PCA and RP) with K-means clustering method for better clustering in high and low-dimensional data. The K-means is giving good performance when combined with PCA than the normal K-means. We have proposed two hybrid algorithms: one combines PCA with K-means and the second one combines PCA and RP with K-means. PCA when combined with Random Projection gives us a good quality clusters in the reduced dimensional space. This method works by combining PCA with RP as preprocessing, then performs clustering (K-means) in PCA/RP reduced space.

Proposed fusion techniques are compared with K-means, PCA-K-means on 12 datasets. The proposed PCA-K-means and PCA-RP-Kmeans algorithms are outperforming the classic K-means, RP+PCA-K-means is outperforming RP-K-means on the given low and HD data sets. Our experimental results strongly advocates the us-

age of the proposed fusion algorithms.

Data set	Avg. MSE of RP-K-means				Avg. MSE of RP+PCA-K-means			
	Reduced dimension (D)				Reduced dimension (D)			
	10	20	50	100	10	20	50	100
ORL	1.43×10^3	3.43×10^3	9.40×10^3	1.94×10^4	5.08×10^{-1}	3.17×10^{-1}	2.29×10^{-1}	1.71×10^{-1}
Yale	214	447	1263	2565	1.68×10^{-1}	1.29×10^{-1}	5.83×10^{-2}	4.34×10^{-2}
COIL20	187	436	1195	2487	1.18×10^{-1}	7.91×10^{-2}	5.43×10^{-2}	4.07×10^{-2}
Colon	563	1164	2908	6091	4.29	2.87	2.47	3.83
Luekemia	2144	4326	11316	22878	27	19	10	13
Lung	2129	4165	10852	21619	37	28	15	9
Prostate	1696	3462	9007	17972	30	22	16	21
GCM	2011	4021	10298	21499	34	24	14	9

Table 4.18: Average MSE for the high-dimensional datasets.

Dataset	Reduced dimension (D)	Avg. MSE of RP+K-means	Avg. MSE of RP+PCA-K-means
Iris	1	2.64×10^{-2}	3.28×10^{-2}
	2	1.01×10^{-1}	3.11×10^{-2}
	3	1.39×10^{-1}	1.83×10^{-2}
Wine	2	0.33	5.24×10^{-2}
	4	0.94	2.26×10^{-2}
	6	1.34	2.00×10^{-2}
	8	1.89	1.18×10^{-2}
	10	2.51	1.12×10^{-2}
	12	3.03	1.14×10^{-2}
ZINC7	1	2.24×10^{-3}	2.99×10^{-3}
	3	5.30×10^{-2}	7.14×10^{-4}
	5	1.28×10^{-1}	6.11×10^{-4}
	7	2.17×10^{-1}	5.29×10^{-4}
ZINC28	1	0.03	2.22×10^{-2}
	7	2.21	5.43×10^{-3}
	14	5.41	3.79×10^{-3}
	21	7.72	3.80×10^{-3}
	28	11.27	4.72×10^{-3}

Table 4.19: Average MSE for the low-dimensional datasets.

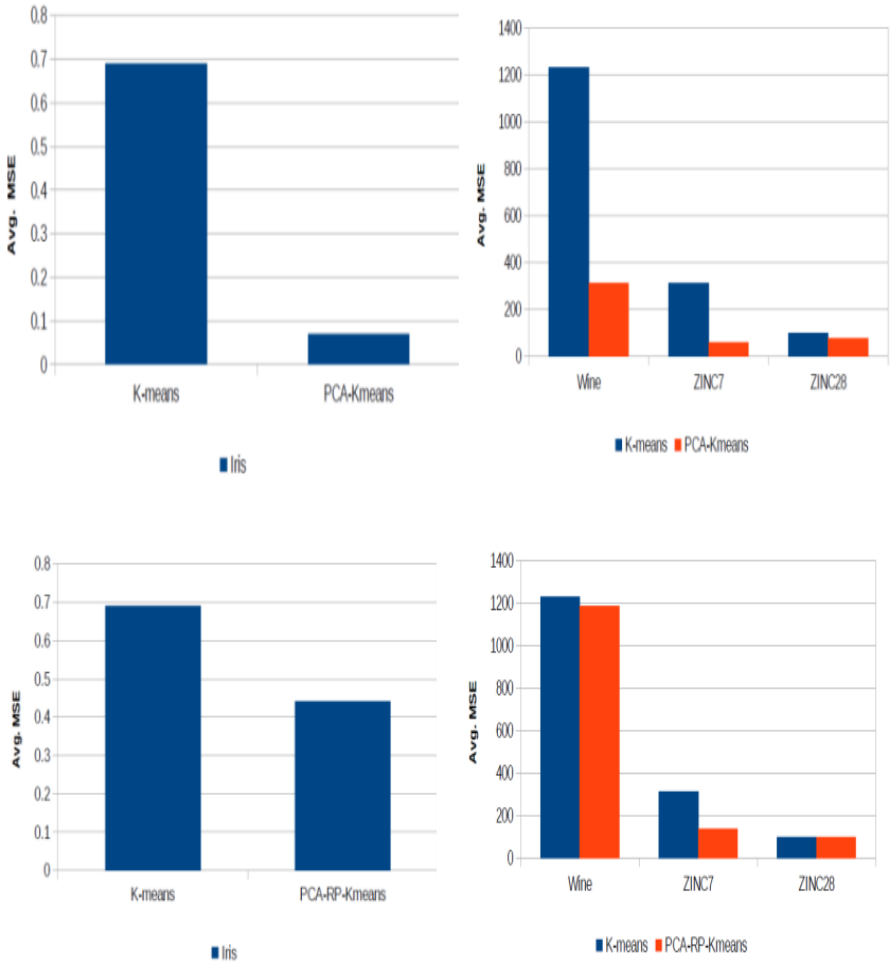

Figure 4.1: Average Mean Squared Error for Low-dimensional Data

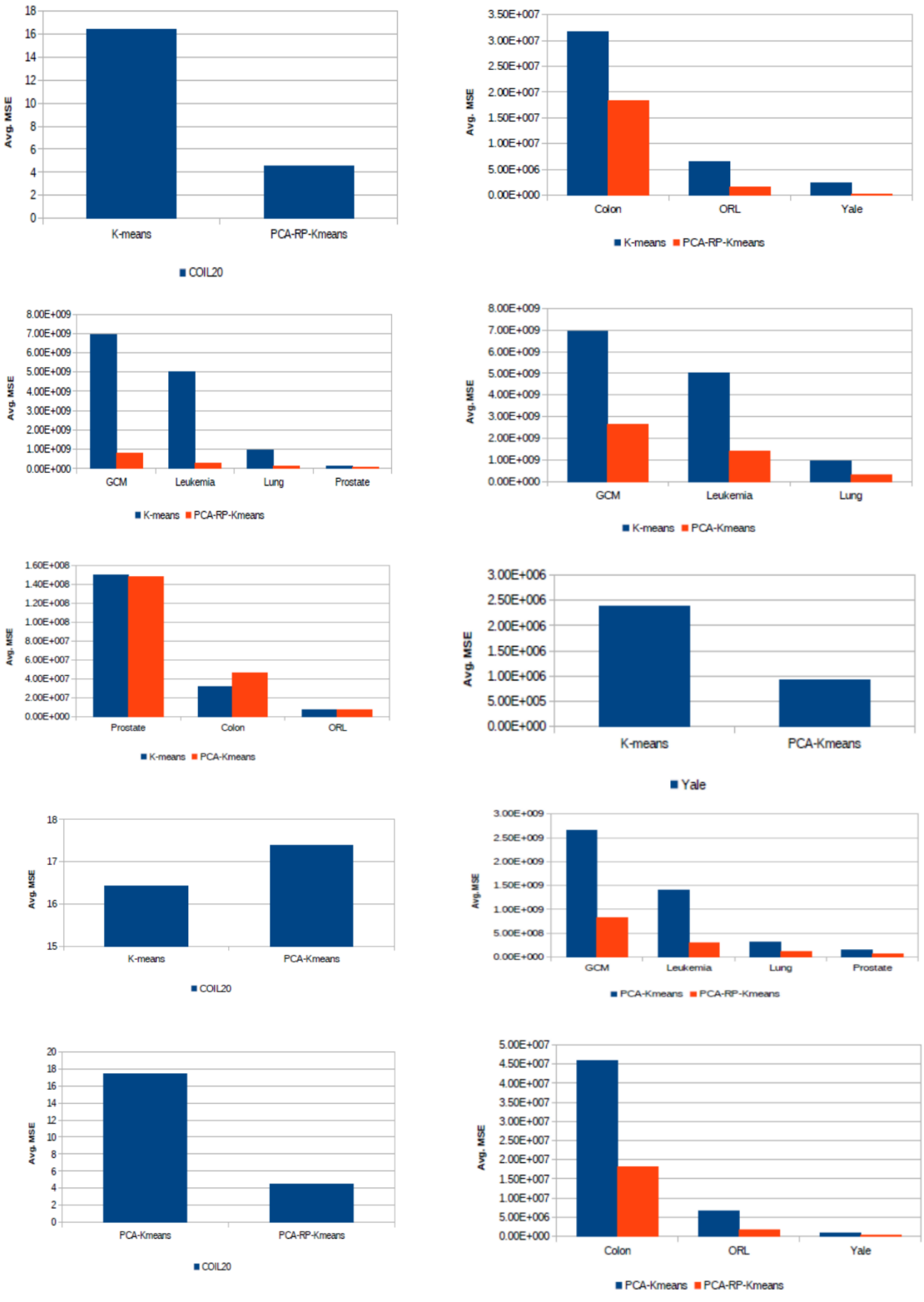

Figure 4.2: Average Mean Squared Error for High-dimensional Data

CHAPTER 5

A Computationally Efficient Data-Dependent Projection for Dimensionality Reduction

5.1 Introduction

Many objects in the world such as images, videos, hand-written letters and numbers, speech signals, text documents, fingerprints, gene-expression micro arrays and hyperspectral images, etc have large number of attributes and hence their representation requires large number of dimensions. Analysis and processing of large scale data is a common requirement in various applications. For example, identifying the fingerprint of a person, searching the documents on Internet by giving keywords, hidden information finding in images, object tracing in videos etc. The system that is processing with this high-dimensional data may be unstable and infeasible. This motivates the need for dimensionality reduction method to bring down the dimensionality of the data and process it.

PCA is a basic linear type of technique which has been used by many researchers in the literature for dimension reduction purpose. PCA is basically a linear projection, that represents a High-dimensional data in lower dimensions. In PCA, the projection matrix is derived from the data it projects. So, the performance of PCA is good among all other DR methods. The wealth of PCA is its simplicity and gives good performance when compared to all other methods. But, computation-wise PCA is very costly, than others. The performance guarantee is more and stable for PCA method, while other methods are unstable in some of the applications like clustering. Eventhough PCA's performance is well and good, the high-computational cost requirement initiates the need to look for other D R methods.

Random Projection is another technique for dimension reduction, which is independent of data, as it uses random matrices which are generated from various distributions which do not correspond to the data that is going to be reduced. Random

projection has been showing good peformance for high-dimensional data clustering, For more details see (Diaconis and Freedman 1984) and (Dasgupta 2000). In these, the authors showed how the shape of the clusters can be changed with the use of RPs (from eccentric to spherical).

Random matrices have been widely used for dimensionality reduction and in Compressive Sensing (CS) due to their provable good performance and they can be easily constructed. The inconsistent performance of random projections can be solved by constructing the projections matrices deterministically Shen et al. (2013). This need of determinisstic way of developing or constructing projection matrices, leads to defining optimally various projection matrices, and these can be found in the literature like: Elad (2007), Julio and Sapiro (2009), Zhou et al. (2012).

To learn from high-dimensional data, we need some some simple and efficient methods like, Random Projection, Random Subspace method, which reduces the dimensionality. Random Projection (RP) is a computationally inexpensive (to its counterpart PCA) technique of dimensionality reduction. PCA has been giving better results for the low-dimensions, where as RP is giving better results at the large reduced dimension. Many researchers in the literature have been documented this. The main feature of RP is its data independent nature, i.e. the RP works by projecting the HD input to LD embedding, such that the pair-wise distance among the points in original data is approximately preserved. Because of its data-independent projection, we cannot expect the local structural properties of the input data are to be preserved. Here comes the elegant data-dependency feature of PCA, eventhogh it is computationally expensive compared to RP, the quality of the solution is very good.

This motivated us to think of a projection which is not based on the point set in the data, instead we will take a representative random sample (10% for example) from the data and do the projection based on eigenvectors with most significant eigenvectors(those are with largest variance) that preserve the structural properties of the given input efficiently. This sample-based projection matrix reduces the computational cost.

5.2 Related Work

In (Papadimitriou et al. (1998)), Papadimitriou et al., shown that embedding the points of a given data X in a low-dimensional space can greatly speed up low-rank approximation to X, without any distortion in its quality.

In (Indyk and Motwani (1998)), Indyk and Motwani proved the usefulness of JL-embeddings in solving the ϵ-approximate neighbor problem. The input data set X is preprocessed to answer the queries like, given an arbitrary point x, find a point $y \in X$: for every point $z \in X$, $\|x-z\| \geq (1-\epsilon)\left\|x-y\right\|$.

In (Schulman (2000)), Schulaman used the JL-embeddings for clustering and the JL-emddings are part of the approximation algorithm in this clustering, where the sum of the squares of intra-cluster distances are minimized.

In (Indyk (2000)), Indyk used the JL-embeddings for the data stream computation, where the memory is limited and we can perform only a single pass over the data stream.

In (Achlioptas (2001)), Achlioptas used random ± 1-matrices for JL, in place of Gaussian matrices. His main motivation in this work is to make the random projections easier to use in practical applications. He obtained a constant speed up by using null values in random projection matrix entries roughly 2/3rd, which means it is a sparse matrix.

In (Frankl and Maehara (1987)), Frankl and Maehara presented a simple and short proof of JL-Lemma by projections onto random orthogonal vectors. Some more simplified proofs are presented in (Indyk and Motwani (1998)), (Dasgupta and Anupam Gupta (2003)).

Kleinberg (Kleinberg (1997)) proposed two algorithms which give approximate nearest neighbors (ANN) in l_2^d. Projecting data onto random lines is the basis for these two algorithms. This idea has been used by many researchers in ANN problem. The first algorithm time complexity is $O((d\log^2 d)(d+\log N))$, and space complexity is in exponential of d. Second algorithm time complexity is $O(N+d\log^3 N)$, which improves on an algorithm that has time complexity $O(Nd)$, require a storage space of only $O(dN\ polylog\ N)$, where N represents database size.

In (Bingham and Mannila (2001)), the authors studied DR methods based on sparse projection heuristics, for noisy and noise-less image data, text data. They showed that the representing HD data on a random subspace (of less dimension) gives same result as the dimension reduction done by a PCA. They also justify with their experiments that, using random projections for dimension reduction is siginficalty improving the computational expense than PCA, and less computations are required for random projection, if we use sparse random matrices.

In (Deegalla and Bostrom (2006)), Deegalla and Bostrom compared two dimension reduction methods namely, PCA and RP. They conducted experiments on high-dimensional image and micro array data sets by integrating dimension reduction method with nearest neighbor classifier. PCA is giving better performance than RP on all the datasets used in the experiments. They have drawn another inference from their experimental study that, PCA accuracy gradually reaches a peak and then decreases with the increasing dimension, RP accuracy increases as the dimensionality grows.

Fern and Brodley in (Fern and Brodley (2001)) studied HD data clustering. They examined RP, identified its instability problem, proposed a solution based on cluster ensemble framework, that works well for clustering high-dimensional data . The main motivation for this work is unstable clustering performance of random projection, but they want robust clustering performance. Lack of stability of clustering results, as each projections is giving a different clustering result, is the main motivation for this RP-based cluster ensemble framework.

In (Xie et al. (2016)), Xie et al. showed the classification performance of RP can be increased when it is combined with other DR methods like PCA, Linear Discriminant Analysis (LDA) and Feature Selection (FS), and this has been proved experimentally by using three micro array data sets and a synthetic data set. The reason for integrating the above methods with RP is stated in their paper as: direct applying PCA, LDA and FS methods require practically infeasible computaional power and not upto the mark. So, they combined these methods with RP and the data dimensionality is reduced so as to maintain a proportionate computational complexity to that of classification accuracy. Eventhough the computational cost of RP is much lesser than PCA, LDA and FS. RP has it's own limitations, that it cannot model the intrinsic

properties of input data, because the projected space is from a random matrix and not from the input matrix. This leads to low accuracy for the classifiers which are based on RP. So, Xie et al. decided to go for combining PCA, LDA and FS with RP. Combination of PCA, LDA, and FS with RP allows one to find the features with best decriminative power than the features obtained by applying RP alone. Finally, this paper concludes that the classification accuracy is improved by fusing the RP with FS or LDA, and this has been proved experimentally. RP combined with FS scores top performance (high classification accuracy and low computation time) on majority of the data sets considered, while RP combined with PCA is not giving better classification accuracy for the given data sets.

Now we will move on to the discussion on some of the related works that have been studied in the field of Compressive Sensing (CS) about the problem of deterministic construction of projection matrix. This is some what divergent from the discussion we are following in the present work, and we will not go into the full details, as this is not in the scope of this work. We just give some of the works on CS, we are not going deep into CS, and we will stick to our main goal of representing the HD data in a LD embedding which preserves the pair-wise distance (local structural properties of the data) approximately.

Elad in (Elad (2007)) and Julio et al. in (Julio and Sapiro (2009)) have optimized projection matrix to achieve better compression ratio. Elad has defined a new concept called mutual coherence, which describes the correlation between the dictionary and projection matrix. The smaller the mutual coherence, the better the compression performance. Elad has minimized the mutual coherence with respect to the projection matrix - keeping the dictionary fixed. In addition to just optimizing the projection matrix, Julio et al. has also optimized the dictionary simultaneously. In particular, Julio uses recently proposed K-SVD algorithm given in (Elad et al. (2005)) to learn dictionary and then jointly optimize the dictionary and projection matrix by maximizing the number of orthogonal columns in their product. Rana et al. in (Rana et al. (2013)) uses SPAMS to learn the dictionary, which is different from that of K-SVD. For the optimization of projection matrix, in (Rana et al. (2013)) they used a special SVD of the dictionary, which produces low coherence projection matrix and dictionary pair.

After thoroughly investigating all the above works, we found that the main drawbacks of random projection are: RP is lagging in capturing the intrinsic structure of original data due to its randomness, whcih may cause a low classification accuracy. It gives unstable clustering performance because of its randomness. To address this problem, we have come up with a moderate and via-medium solution which uses PCA kind of projection but with less computation cost (compared to PCA) and also captures the intrinsic structure of the original data very well. This is because we are generating the projection matrix from the original data, which implies that the underlying structural properties of data are preserved and eventually, quality of the solution is improved.

5.3 Random Projections for Dimensionality Reduction

Input data is mapped onto an independant random feature space so that the pair-wise similarity between the points is preserved. Here we describe the basic independant projection technique, Random Projection. See Section 2.5 for more details on random projection technique.

5.4 Proposed Deterministic Construction of Projection Matrix

We present a deterministic construction of projection matrix from the given input data. Projection matrix, so constructed, offers better performance compared to predefined matrix constructed independently from the given input data. And also validate the proposed one using large datasets, including MNIST sample of 2500 points of handwritten digits; ORL images of 40 people, 10 images per each; Yale contains a total of 165 samples, belongs to 15 subjects, 11 per each; COIL20 of 1440 sample size, 1024 dimensionality and Colon data set which 62 samples of 2000 dimensionality, with 2 classes.

5.4.1 Proposed Approach

Take a random sample S (say 10% of the input data set) from the input data X, perform mean centering for this sample. Compute the covariance of S, and determine eigen values, eigen vectors for this covariance matrix of the sample. Now, the projection matrix P is constructed by taking the eigen vectors corresponds to most significant eigenvalues as columns of P. Now, projection is done by multiplying X with P which gives the projected data matrix Y. After this, we compare the pair-wise distances in original High-Dimensional data with the pair-wise distances of reduced data. Algorithm 12 gives a pseudo code for this.

Algorithm 12 Deterministic Construction of Projection Matrix (DCPM) Algorithm

Input: Data set $X_{N\times d}$, Reduced Dimension D
Output: pdist (pair-wise distance between the points), L_2 -norm
begin

1: Read the input data X.
2: Extract a sample (say 10%) randomly from X and normalize it (mean centering) and call it as S
3: Find the covariance of S and call it as $CovS$.
4: Compute the eigen values and eigen vectors for $CovS$
5: Take the eigen vectors corresponding to the most significant eigenvalues. The most significant eigenvalues are according to a user defined threshold variance and copy these eigenvectors as the columns of Projection Matrix P.
6: $Y_{N\times D} = X_{N\times d} \cdot P_{d\times D}$ //projection step.
7: Compare the pair-wise distances in X and Y.

end

5.5 Experimental Evaluation

5.5.1 Data Sets

In this empirical study, we have taken five data sets of high-dimensionality and two data sets of low-dimensionality, for experimentation. A detailed specifications of the data sets are given in Table 5.1.

In HD (high-dimension) category, we took MNIST data base, that contains handwritten digits, which comprises of 60000 training samples, 10000 testing samples. It is a subset of a larger set available from NIST. The digits are size-normalized and

centered. We have taken a subset of size 2500 samples with a dimensionality of 784 for our experiments. The original MNIST database can be found in (LeCun et al. (1998)).

The ORL Database, comprises of 400 images belongs to 40 people (classes), 10 per each class. Each image is expressed by 1024 sized vector with 256 gray levels. Details about Yale, COIL20, Colon, Iris and Wine datasets are available in Section 4.2.4. Empirical results reported are averages of 10 trials.

S.No.	Data set	# Patterns	# Features	# Classes
1	MNIST	2500	784	10
2	ORL Faces	400	1024	40
3	Yale	165	1024	15
4	COIL20	1440	1024	20
5	Colon	62	2000	2
6	Iris	150	4	3
7	Wine	178	13	3

Table 5.1: Specifications of data sets

5.5.2 Results and Discussion

The methodology we followed in this work for comparing the two methods is as follows: Let X is our input data, Y is the reduced data we got after applying the proposed DR method on X. The pair-wise distance in the original HD data X is denoted as D_1. The pair-wise distance in the reduced data Y is D_2. D_3 is the difference of D_1 and D_2. Y_{rp} is the reduced data after we apply RP-based DR on original HD data X, and the pair-wise distance in Y_{rp} is denoted by D_4. Now D_6 is the difference between D_1 and D_4. Then we calculate the L_2-norm of D_3 (Original v/s Proposed) and L_2-norm of D_6 (Original v/s RP-based DR method). From the intuition, it is clear that the smallest the L_2-norm is the best, that is the pair-wise distance is preserved efficiently.

From the experimental results, we report the error as the L_2-norm of pair-wise distance D_3 (Original v/s Proposed) and L_2-norm of pair-wise distance D_6 (Original v/s RP-based DR method).

In the first experiment, we have compared the RP reduction method and the proposed method by taking the reduced dimension (D) is equal to 50 for MNIST, ORL,

Yale, COIL20 and Colon; $D = 2$ for Iris and Wine datasets, the sample size is fixed at 10% and the values reported are average of 10 runs, See Table 5.2.

On the Iris dataset the error is 240 and it is 224 for RP method. For Wine dataset the error values are 4.93×10^4 and 4.94×10^4 respectively, which means both are almost same.

On MNIST dataset, proposed method got a 5× improvement with 2.65×10^4 over RP method (13.4×10^4).

On ORL dataset, the proposed method (5.24×105) is showing a 5× improvement when compared to RP method (26×10^5).

On Yale dataset, the RP method is giving an error value 16.1×10^5 and the proposed method is 5× better than RP, with an error 3.12×10^5.

On COIL20 dataset, proposed method (1.40×10^4) is showing 5× improvement over the RP method (6.96×10^4).

On Colon dataset, proposed method (1.38×10^6) is showing 5× improvement over RP method (7.30×10^6).

In summaary, the proposed method is giving a better pair-wise distance preserving performance with a 5× improvement over the RP method for all the high-dimensional datasets, and performance of both the methods is almost similar on low-dimensional datasets (Iris, Wine). From Table 5.2, it is evident that, our proposed reduction method is best in pair-wise distance preserving performance (by means of L_2-norm, denoted as error) with that of RP-reduction on all the data bases considered in this study.

By varying sample size in another experiment, we compared the performance of the two methods. We have varied the sample size starting from 10% to 100%. On low-dimensional datasets (Iris, Wine), the sample size has no effect on the performance, and both the methods are showing almost same results.

On MNIST dataset, when varying the sample size, the proposed method is having error values between 2.66×10^4 and 3.11×10^4 with an average of 4× improvement over the RP method (min=13.2×10^4 and max=13.5×10^4).

On ORL dataset, the error is almost constant (on average 26.3×10^5) for RP, for

the proposed :it is in between 5.26×10^5 and 5.91×10^5 when varying sample size (5× improvement over RP).

On Yale dataset, the error ranges from 15.9×10^5 to 16.6×10^5 for RP, for the proposed the range is from 3.10×10^5 to 3.60×10^5 when varying sample size (5× improvement over RP).

On COIL20 dataset, the error ranges from 6.82×10^4 to 7.16×10^4 for RP, for the proposed the range is from 1.49×10^4 to 1.58×10^4 when varying sample size (4× improvement over RP).

On Colon dataset, the error ranges from 6.96×10^6 to 7.37×10^6 for RP, for the proposed the range is from 1.39×10^6 to 1.45×10^6 when varying sample size (5× improvement over RP). These results are present in Table 5.3.

In the third experiment, we have tested our proposed method by varying reduced dimension (D)and reported the average error of 10 runs.

On Iris dataset, the error is not changing much with the increase in D for RP method, but the error is gradually decreasing with the increase in D for the proposed method.

On Wine dataset, when we increase the reduced dimension gradually from 1 to 11 the error is also inreasing for RP method, whereas it is decreasing for the proposed method.

On MNIST sample, the proposed method is showing a good improvement (1.6 to 8×) over the RP method with the increasing reduced dimension (D).

On ORL dataset, the proposed method is showing a good improvement (2.6 to 8×) over the RP method when we increase the reduced dimension.

On Yale dataset, the proposed method is showing a good improvement (2 to 8×) over the RP method when we increase the reduced dimension (D).

On COIL20 dataset, the proposed method is showing a good improvement (1.6 to 7×) over the RP method when we increase the reduced dimension (D).

On Colon dataset, the proposed method is showing a good improvement (1.6 to 8×) over the RP method when we increase the reduced dimension (D).

In summary, average error is decreasing with increase in reduced dimension (D), whereas for RP-reduction average error has been decreasing with the increasing reduced dimension. This experiment results are presented in Table 5.4.

S.No.	Dataset	Average Error	
		RP Reduction	Proposed Reduction
1	Iris (2)	240	224
2	Wine (2)	4.93×10^4	4.94×10^4
3	MNIST (50)	13.4×10^4	2.65×10^4
4	ORL (50)	26×10^5	5.24×10^5
5	Yale (50)	16.1×10^5	3.12×10^5
6	COIL20 (50)	6.96×10^4	1.40×10^4
7	Colon (50)	7.30×10^6	1.38×10^6

Table 5.2: L_2-norm (error) values of the Proposed DR method v/s RP method on various data sets, The reduced dimension (D) is given in brackets with each data set, $D = 2$ for both Iris, Wine datasets and $D = 50$ for other five datasets. Sample average of 10 runs.

5.6 Summary

We proposed a new method of dimension reduction, which maps HD data to a space with small dimension, with the help of a projection matrix that is constructed by taking a random sample from the data and the most significant eigen vectors of this sample forms the projection matrix. The supremacy of the proposed method is, it works effieciently to preserve the pair-wise distance in the reduced space, while making use of only a 10% sample from the original data.

We have tested our proposed method on five high-dimensional and two low-dimensional real world data sets, by fixing the reduced dimension (D) at 50 for high-dimensional datasets and $D = 2$ for low-dimensional datasets. This experiment results shows the improvement in performance of our proposed method is much higher than RP-reduction, showing a 5× improvement on all HD datasets and for LD datasets, both the methods are showing almost same performance.

We also tested the two methods by varying sample size. Our method is showing 5× improvement over RP-reduction on all HD datasets, and both the methods are showing nearly similar performance on low-dimensional datasets.

When we vary the reduced dimension (D), the RP-based dimension reduction

method is producing inferior results when the D is increasing, where as the proposed method is performing well and also improving when D is gradually increased.

The proposed method is giving a good performance improvement over the RP-based method, when we vary sample size and reduced dimension, on all the high-dimensional datasets, the performance of both methods is same on low-dimensional datasets.

In this study, we have used Achlioptas method for generating the entries in random matrix for the projection in RP-based DR method, and as a future work one can use the other random matrices available in the literature, to check the performance varying with the change of random matrix for the projection.

		Sample Size (%)									
Data set	Method	10	20	30	40	50	60	70	80	90	100
Iris	RP	2.20×10^2	2.22×10^2	2.09×10^2	2.39×10^2	2.81×10^2	1.81×10^2	1.95×10^2	1.41×10^2	2.52×10^2	1.42×10^2
	Proposed	2.28×10^2	2.29×10^2	2.27×10^2	2.26×10^2	2.26×10^2	2.29×10^2	2.27×10^2	2.28×10^2	2.28×10^2	2.28×10^2
Wine	RP	4.98×10^4	6.21×10^4	4.80×10^4	5.44×10^4	4.97×10^4	5.25×10^4	5.04×10^4	5.23×10^4	4.73×10^4	5.72×10^4
	Proposed	4.55×10^4	5.15×10^4	5.38×10^4	5.25×10^4	5.44×10^4	5.78×10^4	5.82×10^4	6.09×10^4	6.14×10^4	6.25×10^4
MNIST	RP	1.34×10^5	1.36×10^5	1.34×10^5	1.35×10^5	1.34×10^5	1.32×10^5	1.33×10^5	1.35×10^5	1.35×10^5	1.34×10^5
	Proposed	2.66×10^4	2.88×10^4	3.09×10^4	3.10×10^4	3.10×10^4	3.10×10^4	3.10×10^4	3.11×10^4	3.11×10^4	3.11×10^4
ORL	RP	2.58×10^6	2.62×10^6	2.63×10^6	2.65×10^6	2.60×10^6	2.63×10^6	2.68×10^6	2.62×10^6	2.68×10^6	2.57×10^6
	Proposed	5.26×10^5	5.43×10^5	5.54×10^5	5.63×10^5	5.69×10^5	5.74×10^5	5.78×10^5	5.83×10^5	5.87×10^5	5.91×10^5
Yale	RP	1.63×10^6	1.59×10^6	1.62×10^6	1.59×10^6	1.64×10^6	1.60×10^6	1.60×10^6	1.64×10^6	1.60×10^6	1.61×10^6
	Proposed	3.10×10^5	3.24×10^5	3.30×10^5	3.37×10^5	3.41×10^5	3.47×10^5	3.50×10^5	3.54×10^5	3.57×10^5	3.60×10^5
COIL20	RP	6.82×10^4	7.00×10^4	7.16×10^4	7.00×10^4	6.88×10^4	6.93×10^4	6.98×10^4	6.89×10^4	6.86×10^4	6.94×10^4
	Proposed	1.49×10^4	1.52×10^4	1.54×10^4	1.55×10^4	1.55×10^4	1.56×10^4	1.57×10^4	1.57×10^4	1.57×10^4	1.58×10^4
Colon	RP	7.13×10^6	7.24×10^6	7.06×10^6	7.37×10^6	7.33×10^6	7.19×10^6	7.24×10^6	6.96×10^6	7.23×10^6	7.16×10^6
	Proposed	1.39×10^6	1.41×10^6	1.41×10^6	1.42×10^6	1.43×10^6	1.43×10^6	1.44×10^6	1.44×10^6	1.45×10^6	1.45×10^6

Table 5.3: L_2 -norm (error) values when varying the sample size for several data sets

Dataset	Reduced dimension (D)	Average Error (L_2-norm)	
		RP reduction	Proposed reduction
Iris	1	238	248
	2	263	223
	3	252	185
Wine	1	5.11×10^4	4.84×10^4
	3	4.51×10^4	3.75×10^4
	5	7.01×10^4	2.58×10^4
	7	9.47×10^4	1.56×10^4
	9	11.2×10^4	1.08×10^4
	11	13.1×10^4	0.61×10^4
MNIST	10	4.93×10^4	2.92×10^4
	20	7.77×10^4	2.84×10^4
	50	13.3×10^4	2.66×10^4
	100	20.1×10^4	2.46×10^4
ORL	10	9.56×10^5	5.67×10^5
	20	14.6×10^5	5.51×10^5
	50	26.5×10^5	5.25×10^5
	100	38.6×10^5	4.97×10^5
Yale	10	6.26×10^5	3.38×10^5
	20	9.37×10^5	3.28×10^5
	50	16.2×10^5	3.11×10^5
	100	23.9×10^5	2.91×10^5
COIL20	10	2.49×10^4	1.55×10^4
	20	3.93×10^4	1.53×10^4
	50	7.04×10^4	1.50×10^4
	100	10.5×10^4	1.44×10^4
Colon	10	2.57×10^6	1.55×10^6
	20	4.21×10^6	1.46×10^6
	50	7.12×10^6	1.39×10^6
	100	10.6×10^6	1.31×10^6

Table 5.4: The effect of varying reduced dimension (D) on L_2-norm (error) for proposed DR method v/s RP method for various data sets.

CHAPTER 6

Concluding Remarks With Directions to Future Research

This work aims at exploring nearest neighbor search (NNS) techniques that can reduce the search space of the database. And also to explore efficient dimensionality reduction techniques (maps HD data to LD data) which preserves structural propoerties of original data, reduces the computational complexity while processing the data and finds the hidden patterns in the data. The outcomes of our research are discussed from Chapters 3 to 5 of this thesis. In this Chapter, first we summarize the salient features of the research contributions discussed made in this thesis. This chapter ends with a list of future research paths in our research area, that can be explored.

6.1 Details of Contributions

Chapter 1 presents an introduction. A brief literature survey on the studied problem is presented in Chapter 2. In Chapter 3, we studied the nearest neighbor search problem and proposed two data partitioning methods (MinMax, WSRP) which are based on the proximity from a chosen reference point from the database. The proposed methods finds the nearest neighbors of a given query point by searching in the reduced search space. The proposed pivot-based partitioning method works by dividing the data into different bins based on the distance of the database points from the MinMax reference point, and then querying returns the nearest neighbors for a given query by searching in a small search space. A comparative analysis was conducted between MinMax (proposed) and BO-Heuristic on various datasets. We have validated our proposed method by plotting the distance distribution of both the methods.

In addition, we have proposed another method for partitioning the space: Weighted Set of Reference Points Method (WSRPM). In which we choose a set of reference

points and we weight them according to the distance of a point in database to reference point set, so that these set of reference point are useful in partitioning the space. Now, in the search phase, we find the distance between the given query point q to all the reference points. And we will move forward by choosing nearest reference point to the q, and then weighting, then searching in that reduced search space, we retrieve the neighbors of query q. WSRPM is compared with other methods like BO-Heuristic, MinMax and it outperforms them.

In Chapter 4, we proposed a variant for Iterative Random Projections K-means algorithm which is primarily based on Random Projections (which is a Dimension Reduction Technique) and works by gradually decreasing dimension in the iteration subsequently preserves the inter-point distances efficiently. By conducting an enormous experiments, we compared our method with the Single Random Projection (RP), IRP K-means (IRP K-means) methods. Compared to these two methods, our variant is giving best results on HD data.

In the second part of this chapter, we have incorporated different dimensionality reduction methods (like PCA and RP) with K-means clustering for improving the efficiency of clustering high and low-dimensional data. The K-means is giving good performance when combined with PCA than the normal K-means. We have proposed two hybrid algorithms: one combines PCA with K-means and the second one combines PCA and RP with K-means. PCA when combined with Random Projection gives us a good quality clusters in the reduced dimensional space. Our proposed algorithm works by combining PCA with RP for feature (dimensions) reduction, then performs clustering (K-means) in reduced space. Clustering results on reduced data are compared with simple K-means and PCA reduced K-means algorithms on 12 bench mark datasets. Clustering results of fusion algorithms (PCA-K-means and PCA-RP-Kmeans and RP+PCA-K-means) are outperforming the classic K-means on given low and HD data sets.

In Chapter 5, we proposed a new mapping, which projects the data to a low-feature space by using a projection matrix that is constructed by taking a random sample from the data and the most significant eigen vectors of this sample forms the projection matrix. The main advantage of our proposed method is, it works effieciently to preserve the pair-wise distance in the reduced space, while making

use of only a 10% sample from the original data. We have tested our method on five high-dimensional and two low-dimensional bench mark real world data sets. Proposed projection is achieving better pair-distance preservation than random projection. We also tested our projection on the given data sets by varying the reduced dimension (D), and from the results we conclude that the RP-based dimension reduction method is producing worse results when the D is approaching the original dimension, where as the proposed method is performing well and also improving when D is reached original dimension.

6.2 Directions for Future Work

In this section, we discuss some future directions to explore in our problem area. Intrinsic Dimension (ID) Estimation is a future research problem, more specifically building robust esimators w.r.t. curse of dimensionality can be studied. Applying DR techniques for processing of Hyperspectral Images is another future research direction. Studying the feasibility of applying DR techniques in Compressive Sensing and Trajectory Compression. Study the effect of sparseness (present in data) on the Dimensionality Reduction Method is another open research problem. Studying the possibility of fusing different DR techniques for improved similarity search in large scale data. Studying the best way to combine these techniques (both data-dependant and data-independant) with nearest neighbour search techniques for improving the quality of search outcome.

REFERENCES

1. **T. M. Cover** and **P. E. Hart** , *Nearest Neighbor Pattern Classification.. In IEEE Trans. Inform. Theory, Vol, IT-13*, pp. 21-27, 1967.

2. **Radovanovic, Milos, Alexandros Nanopoulos, and Mirjana Ivanovic.** *Hubs in Space: Popular Nearest Neighbors in High-Dimensional Data* Journal of Machine Learning Research 11.Sep (2010): 2487-2531.

3. **Knuth, Donald E.** *The Art of Computer Programming, Volume 3: (2Nd Ed.) Sorting and Searching*. Addison Wesley Longman Publishing Co., Inc. Redwood City, CA, USA, 1998.

4. **Fukunaga K.** , *Introduction to Statistical Pattern Recognition.* Academic Press Professional, Inc., San Diego, CA, USA, 1990.

5. **L.O. Jimenez and D.A. Landgrebe.** *Supervised classification in high-dimensional space: geometrical,statistical, and asymptotical properties of multivariate data..* IEEE Transactions on Systems, Man and Cybernetics, 28(1): 39-54, 1997.

6. **Agrafiotis D.K.** *Stochastic proximity embedding.* Journal of Computational Chemistry, 24(10): 1215-221, 2003.

7. **G. Baudat and F. Anouar.** *Generalized discriminant analysis using a kernel approach.* Neural Computation, 12(10): 2385-2404, 2000.

8. **M. Belkin and P. Niyogi.** *Laplacian Eigenmaps and spectral techniques for embedding and clustering.* In Advances in Neural Information Processing Systems, volume 14, pages 585-591, Cambridge, MA, USA, 2002. The MIT Press.

9. **Brand M.** *From subspaces to submanifolds.* In Proceedings of the 15th British Machine Vision Conference, London, UK, 2004. British Machine Vision Association.2004.

10. **D.L. Donoho, C. Grimes.** *Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data.* Proceedings of the National Academy of Sciences, 102(21): 7426-7431, 2005.

11. **X. He, P. Niyogi.** *Locality preserving projections.* In Advances in Neural Information Processing Systems, volume 16, page 37, Cambridge, MA, USA, 2004. The MIT Press.

12. **G.E. Hinton, S.T. Roweis.** *Stochastic Neighbor Embedding.* In Advances in Neural Information Processing Systems, volume 15, pages 833-840, Cambridge, MA, USA, 2002. The MIT Press.

13. **G.E. Hinton, R.R. Salakhutdinov.** *Reducing the dimensionality of data with neural networks.* Science, 313(5786):504-507, 2006.

14. **S. Lafon, A.B. Lee.** *Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(9):1393-1403, 2006.

15. **S.T. Roweis, L.K. Saul.** *Nonlinear dimensionality reduction by Locally Linear Embedding.* Science. 290(5500):2323-2326, 2000.

16. **F. Sha, L.K. Saul.** *Analysis and extension of spectral methods for nonlinear dimensionality reduction.* In Proceedings of the 22nd International Conference on Machine Learning, pages 785-792, 2005.

17. **B. Scholkopf, A.J. Smola, and K.R. Muller.** *Nonlinear component analysis as a kernel eigenvalue problem.* Neural Computation, 10(5):1299-1319, 1998.

18. **J.B. Tenenbaum, V. de Silva, and J.C. Langford.** *A global geometric framework for nonlinear dimensionality reduction.* Science, 290(5500):2319-2323, 2000.

19. **Y.W. Teh and S.T. Roweis.** *Automatic alignment of hidden representations.* In Advances in Neural Information Processing Systems, volume 15, pages 841-848, Cambridge, MA, USA, 2002. The MIT Press.

20. **J. Verbeek.** *Learning nonlinear image manifolds by global alignment of local linear models.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(8):1236-1250, 2006.

21. **K.Q. Weinberger, B.D. Packer, and L.K. Saul.** *Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization.* In Proceedings of the 10th International Workshop on AI and Statistics, Barbados, WI, 2005. Society for Artificial Intelligence and Statistics.

22. **Z. Zhang and H. Zha.** *Principal manifolds and nonlinear dimensionality reduction via local tangent space alignment.* SIAM Journal of Scientific Computing, 26(1):313-338, 2004.

23. **T. Zhang, J. Yang, D. Zhao, and X. Ge.** *Linear local tangent space alignment and application to face recognition.* Neurocomputing, 70:1547-1533, 2007.

24. **R.O. Duda, P.E. Hart, and D.G. Stork.** *Pattern Classification.* Wiley Interscience Inc., 2001.

25. **R. Bellman** *Adaptative Control Processes: A Guided Tour.* Princeton University Press, Princeton, NJ, 1961.

26. **D.W. Scott and J.R. Thompson.** *Probability density estimation in higher dimensions.* In J.R. Gentle, editor, Proceedings of the Fifteenth Symposium on the Interface, pages 173-179. Elsevier Science Publishers, B.V., North-Holland, 1983.

27. **Lee J.A., Verleysen M.** *High-Dimensional Data.* In: Lee J.A., Verleysen M. (eds) Nonlinear Dimensionality Reduction. Information Science and Statistics. Springer, New York, NY

28. **M.A. Carreira-Perpinan.** *A review of dimension reduction techniques.* Technical report, University of Sheffield, Sheffield, January 1997.

29. **L.J.P. van der Maaten, E.O. Postma, and J.J. van den Herik** *Dimensionality Reduction: A comparative review.* Technical Report TiCC TR 2009-005.

30. **Deegalla S, Bostrom H** *Reducing high-dimensional data by principal component analysis vs. random projection for nearest neighbor classification.* In: ICMLA 2006: Proceedings of the 5th international conference on machine learning and applications. IEEE Computer Society, Washington, DC, pp. 245-250. doi:10.1109/ICMLA.2006.43

31. **A. Cardoso, A. Wichert** *Iterative random projections for high-dimensional data clustering* Pattern Recognit. Lett., 33 (13) (2012), pp. 1749-1755

32. **H. Xie, J Li, Q. Zhang and Y. Wang** *Comparison among dimensionality reduction techniques based on Random Projection for cancer classification.* Comput. Biol. Chem., 65 (2016), pp. 165-172, 10.1016/j.compbiolchem.2016.09.010

33. **Bingham, E., Mannila, H.** *Random projection in dimensionality reduction: application to image and text data.* In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 245-250.

34. **Fern, X.Z., Brodley, C.E.,** *Random projection for high dimensional data clustering: A cluster ensemble approach.* In: ICML. Vol. 3. pp. 186-193.

35. **Martins, A. D., Gurjao, E. C.** *Processing of smart meters data based on random projections.* Proc. IEEE PES Conference on Innovative Smart Grid Technologies Latin America, ISGT lA, 2013 (2013), pp. 1-4.

36. **Aleshinloye, A., Bais, A., Irfan Al Anbagi.** *Performance analysis of Dimension Reduction techniques for demand side management.* In Electrical Power & Energy Conference (EPEC) 2017 IEEE, 2017.

37. **Tariq, H., Eldridge, E. and Welch Iyan.** *An efficient approach for feature construction of high-dimensional microarray data by random projections.* PLoS One. 2018 Apr 27;13(4):e0196385. doi:10.1371/journal.pone.0196385.

38. **Rana R., Yang M., Wark T., Chou C. T., Hu W.** *A Deterministic Construction of Projection Matrix for Adaptive Trajectory Compression.* IEEE Transactions on Parallel and Distributed Systems 2013.

39. **Rana R., Yang M., Wark T., Chou C. T., Hu W.** *SimpleTrack: Adaptive Trajectory Compression with Deterministic Projection Matrix for Mobile Sensor Networks.* IEEE Sensors Journal 2014.

40. **Juvonen, A and T Hamalainen** *An Efficient Network Log Anomaly Detection System using Random Projection Dimensionality Reduction.* Proceedings of the 6th International Conference on New Technologies, Mobility and Security (NTMS), IEEE, Dubai, United Arab Emirates (2014), pp. 1-5.

41. **Sachin, Mylavarapu and Ata Kaban** *Random projections versus random selection of features for classification of high dimensional data.* Proc. 13th UK Workshop Comput. Intell. (UKCI), pp. 305-312, 2013.

42. **Ding C., He, X.** *K-means clustering and principal component analysis.* ICML, 2004.

43. **H. Qi and S. M. Hughes.** *Invariance of principal components under low-dimensional random projection of the data.* IEEE International Conference on Image Processing, October 2012.

44. **Alshamiri,A.K.,Singh, A., Surampudi, B.R.** *A novel ELM K-means algorithm for clustering.* In: Proceedings of 5th International Conference on Swarm, Evolutionary and Memetic Computing (SEMCO), Bhubaneswar, India, pp. 212-222 (2014)

45. **Alshamiri,A.K.,Singh, A., Surampudi, B.R.** *Combining ELM with Random Projections for Low and High Dimensional Data Classification and Clustering.* In Proceedings of the Fifth International Conference on Fuzzy and Neuro Computing (FANCCO) IDRBT Hyderabad, India, pp. 89-106 (2015)

46. **Khandelwal, C.S., Maheshewari, R., Shinde, U.B.** *Review paper on applications of principal component analysis in multimodal biometrics system* Procedia Comput. Sci. 92, 481-486 (2016)

47. **Dasgupta S.** *Experiments with random projection.* Uncertainity in Artificial Intelligence: Proceedings of the Sixteenth Conference (UAI-2000), pp. 143-151 (2000).

48. **Samuel Kaski.** *Dimensionality Reduction by Random Mapping* Proc. IEEE Int'l Joint Conf. Neural Networks, vol. 1, pp. 413-418, 1998.

49. **Hegde, Chinmay and B. Wakin, Michael and G. Baraniuk, Richard** *Random Projections for Manifold Learning.* Proceedings of the Advances in Neural Information Processing Systems, 2007

50. **Ailon, N., Chazelle, B.** *Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform.* Proc. 38th Annual ACM Symposium on Theory of Computing ,pp. 557-563. (2006).

51. **Dasgupta S., Gupta A.:** *An elementary proof of a theorem of Johnson and Lindenstrauss.* Random Structures & Algorithms 22, 60-65 (2003).

52. **A. Juvonen, T. Sipola, and T. Hamalainen** *Online Anomaly Detection Using Dimensionality Reduction Techniques for HTTP Log Analysis* Comput. Networks, vol. 91, pp. 46-56, 2015.

53. **Han L, Wu Z, Zeng K** *Online Multilinear Principal Component Analysis* J . Neurocomputing. (2017).

54. **S.K. Tasoulis, D.K. Tasoulis, V.P. Plagianakos** *Random direction divisive clustering* Pattern Recognition Letters, Volume 34, Issue 2, 2013,Pages 131-139, ISSN 0167-8655, https://doi.org/10.1016/j.patrec.2012.09.008.

55. **S. Bettoumi, C. Jlassi and N. Arous** *Comparative study of k-means variants for mono-view clustering* 2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Monastir, 2016, pp. 183-188. doi:10.1109/ATSIP.2016.7523092

56. **Hong Yu and Haibo Zhang** *A Three-Way Decision Clustering Approach for High Dimensional Data* In Proceedings of International Joint Conference, IJCRS 2016.

57. **E. Vidal Ruiz** *An algorithm for finding nearest neighbours in (approximately) constant average time* Pattern Recognition Letters, 4 (July) (1986), 145-157.

58. **E. Chavez, G. Navarro** *A compact space decomposition for effective metric indexing.* Pattern Recognition Letters, 26 (July) (2005), 1363-1376.

59. **B. Bustos, G. Navarro, E Chavez.** *Pivot selection techniques for proximity searching in metric spaces.* Pattern Recognition Letters, 24 (14) (2003), 2357-2366.

60. **O. Pedreira, N.R. Brisaboa** *Spatial selection of sparse pivots for similarity search in metric spaces.* in: J. van Leeuwen, G. Italiano, W. van der Hoek, C. Meinel, H. Sack, F. Plasil (Eds.), SOFSEM 2007: Theory and Practice of Computer Science, Lecture Notes in Computer Science, vol. 4362, Springer, Berlin, Heidelberg, 2007, pp. 434-445.

61. **W. Burkhard, R. Keller** *Some approaches to best-match file searching* Comm. ACM, 16 (4) (1973), pp. 230-236.

62. **Baeza-Yates, R., Cunto, W., Manber, U., Wu, S.,** *Proximity matching using fixed-queries trees.* In: Proc. 5th Combinatorial Pattern Matching (CPMâĂŹ94). In: LNCS, vol. 807. pp. 198-212

63. **E. Chavez, J. Marroquin, G. Navarro** *Fixed queries array: A fast and economical data structure for proximity searching* Multimedia Tools Appl. (MTAP), 14 (2) (2001), pp. 113-135

64. **Yianilos, P.,** *Data structures and algorithms for nearest neighbor search in general metric spaces.* In: Proc. 4th ACM-SIAM Symposium on Discrete Algorithms (SODAâĂŹ93). pp. 311-321

65. **Yianilos, P.,** *Excluded middle vantage point forests for nearest neighbor search.* In: DIMACS Implementation Challenge, ALENEXâĂŹ99, Baltimore, MD

66. **L. Mico, J. Oncina, E. Vidal** *A new version of the nearest-neighbor approximating and eliminating search (AESA) with linear preprocessing-time and memory requirements* Pattern Recognition Letters, 15 (1994), pp. 9-17

67. **E. Chavez, J. Marroquin, R. Baeza-Yates** *Spaghettis: An array based algorithm for similarity queries in metric spaces* Proc. String Processing and Information Retrieval (SPIREâĂŹ99), IEEE CS Press (1999), pp. 38-46

68. **Bustos B, Pedreira O, Brisaboa NR** *A dynamic pivot selection technique for similarity search in metric spaces.* In Proceedings of 1st international workshop on similarity search and applications (SISAPâĂŹ08). IEEE Press, pp. 105âĂŞ112

69. **Ruiz G., Santoyo F., ChÃąvez E., Figueroa K., Tellez E.S.** *Extreme Pivots for Faster Metric Indexes.* In: Brisaboa N., Pedreira O., Zezula P. (eds) Similarity Search and Applications. SISAP 2013. Lecture Notes in Computer Science, vol 8199. Springer, Berlin, Heidelberg.

70. **Angiulli F., Fassetti F.** *Principal Directions-Based Pivot Placement.* In: Brisaboa N., Pedreira O., Zezula P. (eds) Similarity Search and Applications. SISAP 2013. Lecture Notes in Computer Science, vol 8199. Springer, Berlin, Heidelberg

71. **E. Chavez, M. Graff, G. Navarro, E.S. Tellez** *Near neighbor searching with K nearest references* Information Systems, Volume 51, 2015, Pages 43-61.

72. **Sergey Brin** *Near Neighbor Search in Large Metric Spaces* In: Proc. 21st Conference on Very Large Databases (VLDBâĂŹ95). pp. 574-584.

73. **P. Indyk,** *Nearest Neighbors in High-Dimensional Spaces* Eds. J.E. Goodman and J. O Rourke, In Handbook of Discrete and Computational Geometr,, chapter 39. CRC Press, 2nd edition, 2004.

74. **Poorna Chandrasekhar A., Rani T. S.,** *Storage and Retrieval of Large Data Sets: Dimensionality Reduction and Nearest Neighbour Search* In: Parashar M., Kaushik D., Rana O.F., Samtaney R., Yang Y., Zomaya A. (eds) Contemporary Computing. IC3 2012. Communications in Computer and Information Science, vol 306. Springer, Berlin, Heidelberg 2012.

75. **Irwin, John J and Brian K Shoichet** *ZINC–a free database of commercially available compounds for virtual screening* Journal of chemical information and modeling vol. 45,1 (2005): pp. 177-182.

76. **Sankara Rao A., Durga Bhavani S., Sobha Rani T., Bapi Raju S., Sastry G.N.** *Study of Diversity and Similarity of Large Chemical Databases using Tanimoto Measure.* In: Venugopal K.R., Patnaik L.M. (eds) Computer Networks and Intelligent Computing. ICIP 2011. Communications in Computer and Information Science, vol 157. Springer, Berlin, Heidelberg (2011).

77. **Cha Sung-Hyuk.** *Comprehensive survey on distance/similarity measures between probability density functions.* Int. J. Math Models Meth Appl Sci. 2007;1:300-307.

78. **A. Gionis, P. Indyk, and R. Motwani** *Similarity Search in High Dimensions via Hashing* Proceedings of the 25th VLDB Conference, Edindurgh, Scotland, 1999.

79. **H. Samet** *The Design and Analysis of Spatial Data Structures* Addison-Wesley, Reading, MA, 1989.

80. **P. Indyk and R. Motwani** *Approximate Nearest Neighbor Towards Removing the Curse of Dimensionality* In Proceedings of the 30th Symposium on Theory of Computing, 1998, pp. 604-613.

81. **Cui Yu, Beng Chin Ooi, Kian-Lee Tan, H. V. Jagadish** *Indexing the Distance: An Efficient Method to KNN Processing* In Proceedings of the 27th VLDB Conference, Rome, Italy, 2001.

82. **T. Bozkaya, M. Ozsoyoglu** *Distance-based Indexing for High-dimensional metric spaces* In Proc. SIGMOD International Conference on Management of Data, 1997, pp. 357-368.

83. **Kamichety H. M., Pradeep Natarajan, Subrata Rakshit** *An Empirical Framework to Evaluate Performance of Dissimilarity Metrics in Content Based Image Retrieval Systems* Technical Report, Center of Artificial Intelligence and Robotics, Bangalore, 2002.

84. **T. Bozkaya, Z. M. Ozsoyoglu** *Indexing large metric spaces for similarity search queries* In ACM Transactions on Database Systems (TODS 1999), 1999, 24(3), pp. 361-404. ACM Press.

85. **R., Pasunuri.** *A Novel Algorithm for Nearest Neighbor Search in High-dimensional Spaces* International Journal of Applied Engineering Research, ISSN 0973-4562 Vol. 10 No.86 (2015) pp. 247-253.

86. **N., Bhatia and Vandana** *Survey of nearest neighbor techniques.* arXIv preprint arXiv:1007.0085, 2010.

87. **P.M. Riegger.** *Literature survey on nearest neighbor search and search in graphs.* 2010.

88. **ORL Faces** *The Database of Faces* AT & T Laboratories Cambridge, (2002). [Online]. Available:http://www.cl.cam.ac.uk/research/dtg/attarchive /facedatabase.html

89. **Y. Shen, W. Hu, M. Yang, B. Wei, and C. T. Chou,** *Projection matrix optimisation for compressive sensing based applications in embedded systems* In Proc. 11th ACM Conf. Embedded Netw. Sensor Syst. (SenSys), Nov. 2013, Art. ID 22.

90. **M. Elad** *Optimized projections for compressed sensing* IEEE Trans. Signal Process., vol. 55, no. 12, pp. 5695-5702, Dec. 2007.

91. **J. M. Duarte-Carvajalino and G. Sapiro** *Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization* IEEE Trans. Image Process., vol. 18, no. 7, pp. 1395-1408, Jul. 2009.

92. **M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, Z. Xing, D. Dunson and G. Sapiro** *Non-parametric Bayesian dictionary learning for analysis of noisy and incomplete images* IEEE Trans. Image Process., vol. 21, no. 1, pp. 130-144, Jan. 2012.

93. **Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner.** *Gradient based learning applied to document recognition.* Proceedings of the IEEE, 86(11):2278-2324, November 1998.

94. **C.H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala.** *Latent semantic indexing: A probabilistic analysis.* In 17th Annual Symposium on Principles of Database Systems (Seattle, WA, 1998), pages 159-168, 1998.

95. **L. J. Schulman.** *Clustering for edge-cost minimization.* In 32nd Annual ACM Symposium on Theory of Computing (Portland, OR, 2000), pages 547-555. ACM, New York, 2000.

96. **P. Indyk.** *Stable distributions, pseudorandom generators, embeddings and data stream computation.* In 41st Annual Symposium on Foundations on Computer Science (Redondo Beach, CA, 2000), pages 189-197. IEEE Comput. Soc. Press, Los Alamitos, CA, 2000.

97. **Achlioptas D.** *Database-friendly random projections: Johnson-lindenstrauss with binary coins.* Journal of Computer and System Sciences 66, pp. 671-687. Special Issue on PODS 2001.

98. **Frankl, P., Maehara, H.** *The Johnson-Lindenstrauss Lemma and the sphericity of some graphs.* Journal of Combinatorial Theory, Series A, 44 (1987), pages 355-362.

99. **Kleinberg 1997** *Two algorithms for nearest neighbor search in high dimensions.* Proc. 29th STOC (1997), 599-608.

100. **Michal, Aharon, Michael Elad, and Alfred Bruckstein.** *K-svd: Design of dictionaries for sparse representation.* Proceedings of SPARS, 5:9-12, 2005.

101. **Johnson W.,Lindenstrauss J.** *Extensions of lipschitz mappings into a hilbert space.* Contemp. Math. 26, pp. 189-206 (1984)

102. **Li P., Hastie T.J., Church K.W.** *Vary sparse random projections.* Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, New York, NY, USA. pp. 287-296. (2006).

103. **Hecht-Nielsen R.** *Context vectors: general purpose approximate meaning representations self-organized from raw data.* Computational Intelligence: Imitating Life, pp. 43-56, (1994).

104. **Ali Ghodsi** *Dimensionality Reduction: A Short Tutorial* Department of Statistics and Actuarial Science, 2006.

105. **J Shlens** *A tutorial on principal component analysis* [Online] Available:http://www.snl.salk.edu/âĹijshlens/pub/ notes/pca.pdf (2005).

106. **M. Carreira-Perpinan** *Continuous Latent Variable Models for Dimensionality Reduction and Sequential Data Reconstruction.* University of Sheffield, 2001.

107. **C. Ding, X. He, H. Zha, and H. Simon** *Adaptive dimension reduction for clustering high dimensional data.* In Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM), pages 147-154, 2002.

108. **H. Liu and H. Motoda** *Feature Extraction, Construction and Selection: A Data Mining Perspective.* Kluwer Academic Publishers, Norwell, MA, USA, 1998.

109. **P. Grassberger and I. Procaccia** *Measuring the strangeness of strange attractors* Physica D Nonlinear Phenomena, 9: 189-208, 1983.

110. **F. Camastra** *Data dimensionality estimation methods: a survey* Pattern Recognition , 36: 2945-2954, 2003.

111. **J. B. Tenenbaum, V.de Silva, and J. C. Lanford** *A global geometric framework for nonlinear dimensionality reduction* Science, 290: 2319-2323, 2000.

112. **Duda, R.O., Hart, P.E., and Stork, D.G.** *Pattern Classification* Second Edition. Wiley, 2001.

113. **Deza, E., and Deza, M. M.** *Encyclopedia of distances* Springer 2009

114. **Prasath, V., Alfeilat, H.A.A., Lasassmeh, O., Hassanat, A.:** *Distance and similarity measures effect on the performance of k-nearest neighbor classifier-a review* arXiv preprint arXiv:1708.04321 (2017)

115. **Lloyd S.** *Least squares quantization in pcm.* Information Theory, IEEE Transactions on 28, pp. 129-137 (1982)

116. **Fradkin D., Madigan D.** *Experiments with random projections for machine learning.* In KDD '03: Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data mining (2003).

117. **Boustsidis C.,Zouzias A., Drineas P.** *Random Projections for k-means Clustering.* Advances in Neural Information Processing Systems 23, pp. 298 - 306 (2010).

118. **Jain, A. K.** *Data Clustering: 50 years beyond K-means.* Pattern Recognition. 31, 651-666 (2010)

119. **Pearson, Karl.,** *On lines and planes of closest fit to systems of points in space* Philosophical Magazine, Series 6, vol. 2, no. 11, pp. 559-572. (1901)

120. **Hotelling, H.,** *Analysis of a Complex of Statistical Variables Into Principal Components* Journal of Educational Psychology, volume 24, pages 417-441 and 498-520 (1933)

121. **Jolliffe, I. T.,** *Principal Component Analysis.* Second ed. Springer Series in Statistics. New York: Springer-Verlag New York. (2002)

122. **Selim, S.Z., Alsultan, K.** *A simulated annealing algorithm for the clsutering problem.* Pattern Recognition 24, pp. 1003-1008, (1991).

123. **Bouveyron, C., Girard, S., Schmid, C.** *High dimensional data clustering.* Comput. Stat. Data Anal. 52, 502-519 (2007).

124. **Assent, Ira.** *Clustering high dimensional data* Wiley Interdisc. Rev. Data Min. Knowl. Discov. 2(4), 340-350 (2012).

125. **Zhang, L., Cao, Q.** *A novel ant-based clustering algorithm using the kernel method* Inf. Sci. 181, pp.4658-4672 (2011).

LIST OF PAPERS BASED ON THESIS

1. RAGHUNADH PASUNURI, VADLAMUDI CHINA VENKAIAH, AMIT SRIVASTAVA. **Clustering High-Dimensional Data: A Reduction Level Fusion of PCA and Random Projection** *In: Kalita J., Balas V., Borah S., Pradhan R. (eds) Recent Developments in Machine Learning and Data Analytics. Advances in Intelligent Systems and Computing,* Vol 740. Springer, Singapore 2018 (SCOPUS)

2. RAGHUNADH PASUNURI, VADLAMUDI CHINA VENKAIAH, BHASKAR DHARIYAL. **Ascending and Descending Order of Random Projections: Comparative Analysis of High-Dimensional Data Clustering** *In: Yadav N., Yadav A., Bansal J., Deep K., Kim J. (eds) Harmony Search and Nature Inspired Optimization Algorithms. Advances in Intelligent Systems and Computing,* Vol 741. Springer, Singapore 2018 (SCOPUS)

3. RAGHUNADH PASUNURI, VADLAMUDI CHINA VENKAIAH **An Optimal Proximity Method for Nearest Neighbor Search in High-Dimensional Data** *In Proceedings of 2nd International Conference on Contemporary Computing and Informatics (IC3I),* pages. 479-483. IEEE Xplore, 2016. (SCOPUS)

4. RAGHUNADH PASUNURI **A Novel Algorithm for Nearest Neighbor Search in High Dimensional Spaces** *International Journal of Applied Engineering Research (IJAER),* Vol 10, No. 86, pages 247-253, 2015. (SCOPUS)

5. RAGHUNADH PASUNURI, VADLAMUDI CHINA VENKAIAH **A Computationally Efficient Data-Dependent Projection for Dimensionality Reduction** *Presented at International Conference on Communication and Intelligent Systems (ICCIS-2019)* Springer, LNNS, (SCOPUS)

Dimensionality Reduction and Nearest Neighbor Search in Large and High-Dimensional Data

by Raghunadh Pasunuri

Dimensionality Reduction and Nearest Neighbor Search in Large and High-Dimensional Data

ORIGINALITY REPORT

9%	1%	9%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Raghunadh Pasunuri, Vadlamudi China Venkaiah. "An optimal proximity method for nearest neighbor search in high dimensional data", 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), 2016 Publication	2%
2	"Harmony Search and Nature Inspired Optimization Algorithms", Springer Science and Business Media LLC, 2019 Publication	1%
3	Raghunadh Pasunuri, Vadlamudi China Venkaiah, Amit Srivastava. "Chapter 44 Clustering High-Dimensional Data: A Reduction-Level Fusion of PCA and Random Projection", Springer Science and Business Media LLC, 2019 Publication	1%
4	Advances in Intelligent Systems and Computing, 2015.	<1%

Student Paper

<1%

19 Raghunadh Pasunuri, Vadlamudi China Venkaiah, Bhaskar Dhariyal. "Chapter 14 Ascending and Descending Order of Random Projections: Comparative Analysis of High-Dimensional Data Clustering", Springer Science and Business Media LLC, 2019

Publication

<1%

20 "Encyclopedia of Database Systems", Springer Nature, 2009

Publication

<1%

Exclude quotes	On	Exclude matches	< 14 words
Exclude bibliography	On		