“PATTERN ANALYSIS OF MACHINE OLFACTORY SYSTEM”

A thesis submitted towards the partial fulfillment of requirement

for the award of degree of

Master of Engineering

In

Electronics and Communication Engineering

Submitted by:

Shivam Choudhary

Roll No: 801361024

Under the guidance of:

Dr. Ravi Kumar

Assistant Professor, ECED

Thapar University, Patiala

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

(Established under the section 3 of UGC Act, 1956)

PATIALA – 147004 (PUNJAB)

Certificate

I hereby declare that the work which is being presented in the thesis entitled, "**PATTERN ANALYSIS OF MACHINE OLFACTORY SYSTEM**" partial fulfilment of the requirement for the award of degree of Master of Engineering (E.C.E) at the Electronics and Communication Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Ravi Kumar, Assistant Professor, ECED. The matter presented in this thesis has not been submitted in any other University/Institute for the award of any other degree.

Date: 10/7/2015

Shivam Choudhary
Roll. No: 801361024

It is certified that the above statement made by the student is correct to the best of my knowledge and belief.

Dr. Ravi Kumar
Assistant Professor
ECED,
Thapar University

Counter signed by:

Dr. Sanjay Sharma
Professor & Head
Thapar University
Patiala-147001

Dr. S. S. Bhatia
Dean of Academic Affairs
Thapar University
Patiala-147001

Acknowledgement

I take this opportunity to express my profound sense of gratitude and respect to all those who helped me through the duration of this thesis. I acknowledge with gratitude and humility my indebtedness to **Dr. Ravi Kumar, Assistant Professor,** Electronics and Communication Engineering Department, Thapar University, Patiala, under whose guidance I had the privilege to complete this thesis. I wish to express my deep gratitude towards him for providing individual guidance and support throughout the thesis work.

I convey my sincere thanks to **Head of the Department, Dr. Sanjay Sharma** as well as **PG Coordinator, Dr. Amit Kumar Kohli, Associate Professor, ECED**, entire faculty and staff of Electronics and Communication Engineering Department for their encouragement and cooperation.

Finally and above everyone else, my heartfelt thanks and gratitude goes to my parents and sister for their constant support and encouragement. I am also thankful to God who bestowed upon ability and strength in me to complete this work.

Shivam Choudhary

Abstract

Intelligent sensors require a robust recognition paradigm to discriminate and analyse the target species. Artificial olfactory systems (popularly known as E-Nose) typically suffers from poor selectivity of individual sensor elements due to which the final classification often turns out to be poor. The use of chemical sensors in the form of array helps to improve the selectivity. But the problem gets aggressive when real time classification of odors/gases is required. When different classification techniques are implies in real time applications, it is expected that the incoming test/data patterns gets classified in accordance of the distribution of the training data. But in all cases we do not have the initial training samples to train our network and hence we have to use unsupervised classification techniques. These unsupervised techniques use the distance parameter as a key to the classification problem. But the performance limitation of existing unsupervised algorithm like K-means clustering as it is highly sensitive to euclidean distance also limits the performance of E-nose to great extent. Thus it becomes imperative to find a distance parameter that corresponds to the similarity of incoming data patterns and then classify them. In this thesis we have proposed a novel normalized cosine K means clustering technique and Quaternion based approach for classification of data obtained from an E-Nose sensor.

Table of Contents

List of Figures

List of Tables

Page No.

CHAPTER 1

INTRODUCTION

1.1. Need of Artificial Olfaction

The olfactory system refers to the sensory system of an individual that distinguishes one smell from other. The olfactory machine, also referred to as E-nose is a sensory array instrument that is used to detect and distinguish complex odors/gases. It is a highly useful device that has been projected as an artificial alternative for human olfactory system [1]. This technology has a wide range of applications in quality control and assessment of beverages and food, detection of drugs and explosives, environmental monitoring etc. The E-nose constitutes of the gas sensor array and the pattern classifications methods. Pattern analysis techniques can be used to analyze the multivariate output of gas sensors, to differentiate between the odors/gases.

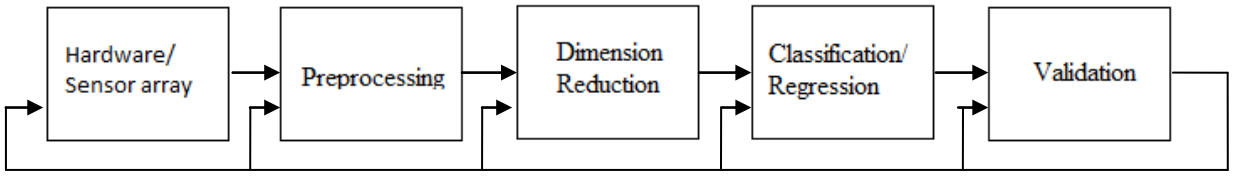

Figure 1.1 Sequential steps for the pattern analysis system of an electronic nose [1]

The data analysis start after the response of the sensor array have been received and stored into a memory space on a computer. The process of pattern classification after the sensor array hardware constitute of four sequential steps: signal preprocessing, reduction of dimensionality, clustering and validation as shown in the Figure 1.1. The first step of preprocessing extracts the descriptive parameters from the response of the sensor array and provide the feature vector that can be used for processing at the next stages. The dimensionality reduction step projects the feature vector from a higher dimension to a vector space of low dimension, to avoid the difficulties associated with the large dataset. The low dimensional vector is then processed to solve a clustering problem. Classification refers to the problem of classifying the unknown odor/gas as one of the learning set already learnt.

1.2. Poor selectivity of Sensing devices [10-11]

In the preprocessing stage, we are required to select a finite number of parameters that define the response of the sensor array, as these parameters are the inputs to the subsequent stages for the classification problem using pattern analysis. The array of sensors that has been used in the device suffers from a severe disadvantage as the inherit drift of gas sensor, that causes a large variation of the sensor output when the array is exposed to the same odors in the same environmental conditions and the cross sensitivity of sensors to different odors. Due to the poor selectivity of the sensor array, the sensor array response is not considered as the final classification but taken as an input to pattern analysis system.

1.3. Statistical Properties of the Sensor data

The sensor array response is the input to the pattern analysis system. The data enters the pattern analysis system in highly jumbled when expressed in pattern space. Hence we require some techniques so that we can differentiate the different odors after processing the data. The algorithms and the techniques take into account the hidden properties of the data and using them to cluster the data so that they can be easily differentiated in pattern space. Some of the statistical properties have been discussed below.

1.3.1 Mean

Mean of the data means the average value of the data. It corresponds to the central point of the distribution of data. It is also referred as the central tendency of the scattered data. If we have 'N' number of observations as $X = [x_1, x_2, x_3, .., x_N]$ then the mean is defined as

$$\bar{X} = \frac{\sum_{i=1}^{N} x_i}{N} \qquad (1.1)$$

Mean is also referred to as the first moment of X. The mean serves an important role in pattern classification as it tells the final cluster centers computed as the mean of the class evolved after the completion of all the iterations of the algorithm.

1.3.2 Variance and Standard Deviation

Variance and Standard deviation are termed as the second moment of X around the mean and provides the information about the spread of the data about the mean. It calculated the distance of the individual element of data from the mean, and then squares the distance to make all the values positive, and then find the average the resulting values. Hence it is also referred to as the average of the squared distances from the mean. Mathematically it is expressed as follows

$$Variance = \sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{X})^2 \qquad (1.2)$$

Standard Deviation (S.D) is the square root value of the variance. It also refers the deviation of the data points from the mean. A large value of standard deviation tells that the data is highly spread and less concentrated. On the contrary, small value of standard deviation tells that the data is highly concentrated around the mean.

$$S.D = \sqrt{Variance} = \sqrt{\sigma^2} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{X})^2} \qquad (1.3)$$

1.3.3 Correlation

Correlation refers to the relationship of two variables; in our study we take the relationship of two classes or clusters. Correlation is a statistical parameter which gives information of the fluctuation of two classes with respect to each other. The higher the correlation, the higher is the relationship of the properties of the data. Let the two classes X and Y have N data vectors each. Then the correlation of the two classes can be found as

$$r_i = \frac{\sum_{i=1}^{N}(x_i - \bar{X})(y_i - \bar{Y})/N}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{X})^2}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{Y})^2}} \qquad (1.4)$$

$\bar{X}$ and $\bar{Y}$ correspond to the mean of the sequence X and Y respectively. The value of the correlation should be high within a class and the different classes should be uncorrelated, such that the data vector in different classes should have minimum correlation coefficients.

1.4. Novel aspects of this work

The novel aspect of this study includes in depth analysis of the classification analysis using K-means clustering algorithm and its use in machine olfactory system to distinguish between different odors efficiently. Primary contributions are as follows:

- New distance measures to improve the efficiency of clustering techniques using Normalized cosine Distance parameter and comparing the results with the Euclidean distance parameter.
- Use of Quaternions to transform the data into different dimensions using rotation of axis, and subsequent clustering using different distance parameters to provide better results in the form of high Percentage Classification.

1.5. Organization of the thesis

This thesis is organized in the following six chapters.

Chapter 2 Gives an overview of the previous work and serves as the basis for identifying the research gaps.

Chapter 3 Deals with the data extraction from the sensor array and the techniques like Principal Component analysis (PCA) and a supervised clustering technique, Transformed Cluster Analysis (TCA). It also introduces validity measures for evaluating the performance of the clustering algorithms.

Chapter 4 Discusses the novel K-means Clustering algorithm and the formulation of a new distance measure as Normalized cosine distance used for the K-means clustering technique.

Chapter 5 Introduce the quaternion algebra and the use of quaternions to do the rotation of axis. The rotation of axis transforms the data to a new vector space and the formulations for the same have been presented.

Chapter 6 Contains the results and discussions for all the techniques used in the thesis work.

Finally, conclusion and future scope of this work has been presented in **Chapter 7.**

CHAPTER 2

Literature Survey

In this chapter, a brief account has been put forth of the available literature that has been studied extensively.

A. Papaioannou and S. Zafeiriou (2014) [4] presented that the nonlinear complex representations, through the utilization of complex kernels, could be connected to model and catch the nonlinearities of complex information. Despite the fact that the theoretical instruments of complex reproducing kernel Hilbert spaces (CRKHS) have been as of late effectively connected to the outline of computerized channels also, relapse and arrangement systems, there is a restricted research on part examination and dimensionality lessening in CRKHS. The point of this brief is to legitimately define the most prominent segment examination technique, i.e., Principal Component Analysis (PCA), in CRKHS. Specifically, authors characterized a general widely linear complex kernel PCA structure. Moreover, we demonstrate to proficiently perform broadly straight PCA in little example measured issues.

X. Huang *et al.* (2014) [19] presented an extension in K-means clustering for apportioning an information set into groups such that in a cluster data vectors are compact, the data vectors in distinctive clusters are well isolated. In this paper, progressions of new clustering algorithms by extending the existing K-means algorithm is proposed by incorporating both intra cluster minimization and inter cluster partition. The properties and exhibitions of these algorithms are explored on genuine and real time data sets. Trial studies illustrate that the proposed algorithms beat the cutting edge K-means Clustering calculations.

R. Kumar (2013) [2] This paper proposed a novel methodology toward the discrimination of odors/gases utilizing game theory for feature extraction. A formerly reported tin-oxide-based sensor array capable of operating at room temperature was chosen for extracting the raw data. The sensor array was exposed to four different smells/gases and the response curves uncover its poor selectivity. The assignment of classifying the sampled information into four classes was displayed as a coalitional game in which every sensor of the exhibit acted like a player framing coalitions with different players. A pay-off function is connected with every

conceivable coalition of players with higher pay-offs being given to coalitions that amplify class distinctness of the information. Shapley quality is utilized to evaluate the commitment of every player yielding a standard example for every scent class. A weighting plan for relative scaling of the test specimens is additionally proposed. It was watched that more than 89% of the examples were distinguished accurately utilizing the proposed procedure subsequently demonstrating its adequacy.

X. Chen *et al.* (2013) [20] proposed TW-k-means, a robotized two-level variable weighting classification algorithm for multi-view information, which can at the same time figure weights for perspectives and individual variables. In this calculation, a perspective weight is allotted to every perspective to distinguish the minimization of the perspective and a variable weight is additionally appointed to every variable in the perspective to recognize the significance of the variable. Both perspective weights and variable weights are utilized as a part of the separation capacity to focus the groups of items. In the new calculation, two extra steps are added to the iterative k-means bunching procedure to naturally register the perspective weights and the variable weights. They utilized two genuine information sets to examine the properties of two sorts of weights in TW-k-means and researched the contrast between the weights of TW-k-means and the weights of the individual variable weighting system. The examinations have uncovered the convergence property of the perspective weights in TW-k-means.

B K. Bao *et al.* (2012) [5] In this paper, authors addressed the lapse redress issue, that is, to reveal the low-dimensional subspace structure from high-dimensional perceptions, which are conceivably ruined by errors. At the point when the lapses were of Gaussian dispersion, principal component analysis(PCA) could locate the ideal (as far as least square errors) low-rank rough guess the high dimensional information. In any case, the accepted PCA system is known to be not significantly delicate to the vicinity of gross corruptions. Robust Principal Component Analysis (RPCA) is a transductive system and did not handle well the new specimens, which were not included in the training method. Given another datum, RPCA basically needed to recalculate over all the information, bringing about the high computational expense. Thus, RPCA was unseemly for the applications that oblige quick online reckoning. To defeat this impediment, in this paper, authors proposed an inductive

powerful key segment investigation (IRPCA) strategy. Given an arrangement of preparing information, dissimilar to RPCA that objectives on recuperating the first information framework, IRPCA went for taking in the fundamental projection grid, which could be utilized for effectively evacuate the conceivable debasements in any datum. The learning was finished by unraveling an atomic standard regularized minimization issue, which was curved and could be tackled in polynomial time.

Y. Xiong (2012) [16] presented that there were two procedures to form hierarchical structure of a document class. One was the hierarchical classification other was confusion matrix. In view of the confusion matrix of a plane classifier, the paper made utilization of hierarchical arrangement to carry forward tests, and it was demonstrated that the confusion matrix methodology was better than the hierarchical classification system. Also, contrasted with plane classifier, confusion matrix methodology enhanced review and exactness of clustering.

R. He *et al.* (2011) [6] presented a correntropy criterion based rotational invariant principal component analysis. Correntropy objective is calculated using half quadratic optimization algorithm that could correctly update the data mean. It provided robustness from the outliers and did not required the data to be zero mean. The optimum results consisted of the eigen vectors of covariance matrix showing the maximum of the eigen values.

R. Kumar *et al.* (2011) [10] presented a soft computational methodology for discrimination of odors/gases. The proposed method was used on the raw information acquired from the reactions of oxygen plasma treated thick film tin oxide sensor system presented to four diverse smells/gasses. The raw data from the sensor exhibit reaction was subjected to wavelet change and proper coefficients were chosen utilizing multi-scale principal component analysis (MSPCA). The preparation and test exhibitions of back-propagation neural system (BPNN) and radial basis function neural system (RBFNN) had been compared. Both the systems have been found to distinguish the odors/gases with a high achievement rate.

R. Kumar *et al.* (2011) [11] In this paper, a new smell/gas identifier-cum-quantifier is displayed. Dynamic reaction curves of an oxygen-plasma treated thick-film tin oxide sensor arrar exposed to four distinctive gasses were subjected to continuous wavelet transform (CWT). Suitable wavelet coefficients were chosen utilizing multiscale principal component

analysis (MSPCA).The quantitative data was encoded in the fuzzy subsethood estimations of the specific concentration bands in the output feature space, though the fuzzy entropy qualities were utilized to normalize the preparation information set comprising of MSPCA chose wavelet coefficients. A feedforward neural system was prepared with a backpropagation calculation with the preparation information containing the wavelet coefficients standardized with fuzzy entropies of individual scents/gasses. The objective information set was comprised of the fuzzy subsethood estimations of the specific concentration band. The proposed system accomplished recognizable proof furthermore, measurement of scents/gasses with a 100% achievement rate.

R. Kumar *et al.* (2010) [12] In this paper a novel neuro-fluffy classifier was introduced. The proposed classifier recovers both qualitative and quantitative data at the same time from the steady state reactions of thick-film tin oxide gas sensor cluster when it was presented to seven various types of alcohols and alcohol mixed drinks. The individual groups were represented in the output space by fuzzy subset hood measure. The qualitative as well as, quantitative arrangements were finished using an artificial neural system (ANN) with back-propagation algorithm. The proposed system gave acceptable execution and synchronous qualitative and quantitative classification of the alcohols and alcohol mixed drinks.

L. Wei *et al.* (2010) [21] Manifold learning algorithms can uncover the low-dimensional geometry structure of the information sets. In this paper, the authors consolidate K-means clustering calculation with manifold learning algorithms into a reasonable structure. They demonstrated the algorithm KCM (K-means bunching with complex) methodologies can get great clustering results on UCI information sets. They additionally outlined that the KCM clustering algorithm can be normally stretched out to semi-supervised clustering.

Haiping Lu *et al.* (2009) [7] proposed an algorithm for the unsupervised data space learning as Unrelated Multilinear Principal component analysis (UMPCA). It can also be thought as a multi linear extension to the novel principal component analysis (PCA). It produced uncorrelated features while capturing the variation in the input dataset. It also told the number of uncorrelated feature vectors that could be produced from the data. The

experimental results verified that the UMPCA effectively found the low dimension projection space required.

R. Kumar *et al.* (2009) [13] presented a novel way to deal with odor segregation of alcohols and hard drinks utilizing distributed information obtained from the response of thick film tin oxide sensor array created at author's lab. The technique used for classification is a mixture of TCA and radial basis neural network (RBFNN). The execution of the new classifier was analyzed with others taking into account back-propagation (BP) calculation. The new model has better classification power with a much lower error. Likewise, it was discovered to be less sensitive to the changes in learning parameters separated from being essentially speedier than the traditional models in view of BP calculation. Both crude information and information preprocessed by transformed cluster analysis (TCA) were utilized to train radial basis function neural network (RBFNN) and back-propagation system (BPN).

H. Xiong *et al.* (2009) [22] In this paper, the authors provided a comprehensive study of the impact of skewed information distributions on K-means clustering. Along this line, the authors first formally delineate that K-means has a tendency to produce clusters of moderately uniform size, regardless of the possibility that information have changed "genuine" cluster sizes. Moreover, the authors demonstrate that some clustering validation measures, for example, the entropy measure, may not catch this uniform impact and give deluding data on the clustering results. Seen in this light, the authors give the coefficient of variation (CV) as an important rule to validate the classification results. Results uncovered that K-means tends to deliver clusters in which the variation of cluster sizes, as measured by CV, are in a scope of around 0.3–1.0. Interestingly, for information sets with little variation in "genuine" cluster sizes (e.g., CV < 0.3), K-means expands a variety in resultant group sizes to more noteworthy than 0.3.

S. G. Dastidar *et al.* (2008) [8] presented an enhanced principal component analysis (PCA) cosine radial basis function classifier. A nine dimension feature space found in previous research is used in proper representation of electroencephalogram (EEG) is used as an input to a classifier. The two stages comprised using PCA in the first stage to improve the classification efficiency and in the next stage a radial basis neural network (RBFNN) is used.

The new method provided accuracy of the classification to 96.6%. For epilepsy diagnosis, the accuracy wet to 99.3% when normal EEGs were considered.

M. J. Li *et al.* (2008) [23] In this paper, authors presented an agglomerative fuzzy K-Means clustering technique for numerical information, an expansion to the standard fuzzy K-Means calculation by acquainting a penalty term with the objective to make the clustering process not sensitive to the starting cluster points. The new algorithm could create more steady clustering results from diverse arrangements of starting cluster centers. Consolidated with cluster validation methods, the new algorithm could focus the number of clusters in an information set, which is a big issue in K-Means clustering. Trial results on engineered information sets (2 to 5 measurements, 500 to 5,000 items and 3 to 7 clusters), the BIRCH two-dimensional information set of 20,000 items and 100 cluster and the WINE information set of 178 articles, 17 measurements, and 3 groups from UCI have exhibited the adequacy of the new calculation in delivering reliable clustering results and deciding the right number of clusters in diverse information sets.

L. Jing *et al.* (2007) [24] introduced another k-means calculation for classifying high-dimensional data in subspaces. In high dimensional information, clusters of articles frequently exist in subspaces instead of in the whole space. In data clustering, groups of data of distinctive themes are sorted by diverse subsets of terms or decisive words. The keywords for one class may not happen in the records of different groups. This is an information sparsity issue confronted in bunching high- dimensional information. In the new calculation, they developed the k-means clustering procedure to figure a weight for every measurement in every cluster and utilized the weight qualities to distinguish the subsets of imperative measurements that arrange diverse clusters. This is accomplished by incorporating the weight entropy in the goal capacity that is minimized in the k-means clustering procedure. An extra step is added to the k-means clustering procedure to find the weights of all measurements in every group. The investigations on both engineered and genuine information have demonstrated that the new calculation can create preferable clustering results over other subspace clustering calculations.

S. Bandyopadhyay and U. Maulik (2001) [14] presented a genetic algorithm (GA) of variable string length for making a novel nonparametric clustering procedure when the

number of clusters is not specified from the earlier. Chromosomes in the same populace might presently had distinctive lengths since they encode diverse number of classes. Cluster validity measure is utilized as a measure of the wellness of a chromosome. The execution of a few cluster validity measures, specifically, Davies–Bouldin (DB) record, Dunn's list, two of its novel versions and a newly developed index were compared.

C. Chang and Q. Du (1999) [9] presented that the Principal Component Analysis (PCA) suffered from two different problems. First, the algorithm found principal components considering the maximum variance of the information set but this did not provide the required image quality. To improve this a modified PCA based on SNR maximization was proposed. The major problem with this approach was that the noise cariance matrix had to be calculated before-hand which was not possible in real time applications. In this paper, these two issues were tended to by considering the interference as a different, obscure signal source, from which an interference and noise-adjusted principal components analysis (INAPCA) could be created in a way like the one from which the NAPC was determined. Two methodologies were proposed for the INAPCA, alluded to as signal to interference plus noise ratio-based principal components analysis (SINR-PCA) and interference-annihilated noise-whitened principal components analysis (IANW-PCA). It is demonstrated that if obstruction is dealt with appropriately, SINRPCA furthermore, IANW-PCA altogether enhance NAPC.

D. L. Davies and D. W. Bouldin (1979) [15] presented a measure which showed the likeness of clusters which are supposed to have an information density which is a declining function of distance from a vector for the cluster. The measure could be utilized to derive the fittingness of information segments furthermore, could in this way be utilized to analyze relative suitability of different divisions of the information. The measure does not rely on upon either the number of classes investigated nor the strategy for dividing of the information, can be utilized to guide a clustering algorithm.

Chapter 3

Data Extraction and Problem Formulation

3.1 Response recovery curve for a five sensor array [2]

An array of five sensors was prepared using screen printing technique and exposed to four different gases/odors. The sensors were fabricated at Centre for Research in Microelectronics (CRME). A gas sensitive tin oxide paste was prepared and printed on to an alumina substrate. The paste was doped with 1% Pd, Pt, CuO, and ZnO. Thus, four out of five sensors were doped with the above mentioned materials, while one sensor was left undoped. Gold electrodes (ESL 8080) were also printed along with gas sensitive layer of tin oxide and were subsequently fired onto alumina substrate. SAMCO plasma deposition system (model no. 10) was employed to generate oxygen plasma at low pressure by applying R.F. power at 13.56 MHz. The test gases (LPG, CCl_4, CO and C_3H_7OH) were injected into a test chamber of volume 2894.7 ml which contained the array. The resistance variations of all the sensors of array were measured simultaneously with time through a counter decoder circuit. Different concentrations (viz. 25ppm, 50ppm, 75ppm, 100ppm) of the test gases were injected into the chamber. The response recovery curves have been reported earlier [2] and the data obtained by sampling have been used in the present dimensionality reduction task. The data consist of four classes and the response curve of the data is shown in figure 3.1-3.4.These figures show the response of the sensor array to the gases whereas Figure 3.5 shows the scatter plot of the raw data to represent the low selectivity of the sensor array to distinguish the odors.

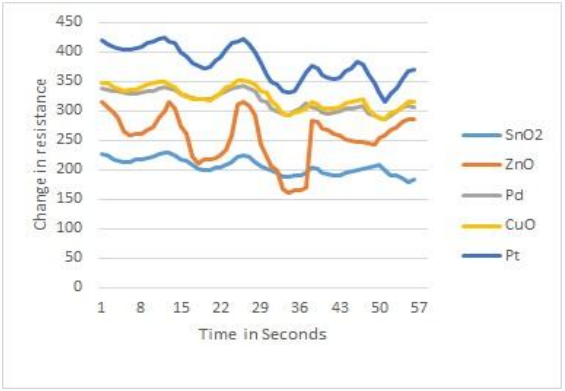

Figure 3.1. Data graph for the odor of LPG [2]

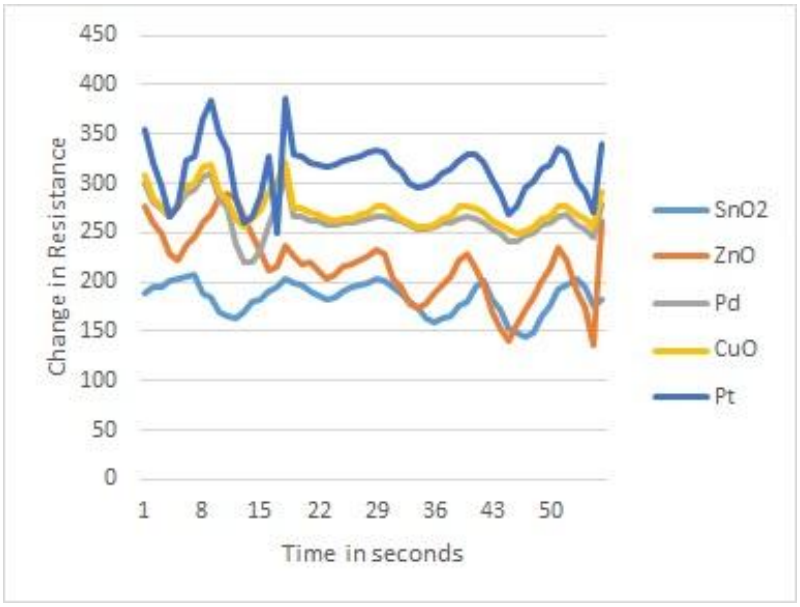

Figure 3.2. Data graph for the odor of CCl_4 [2]

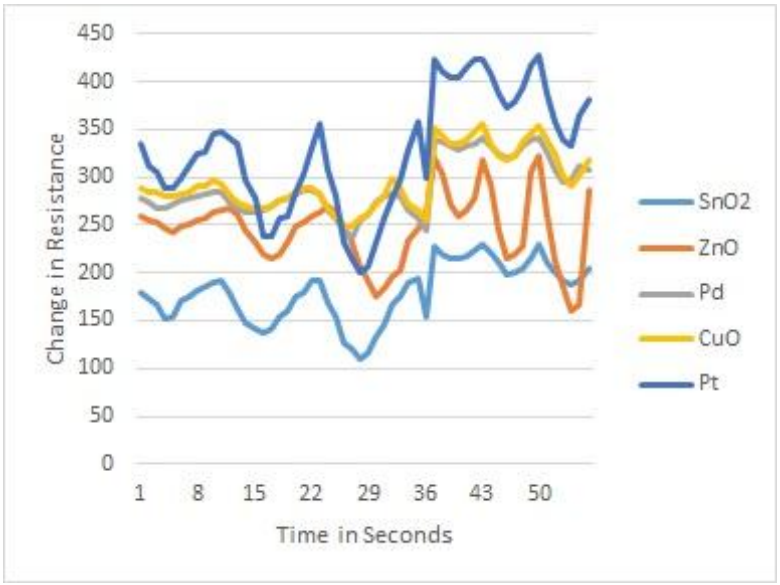

Figure 3.3. Data graph for the odor of CO [2]

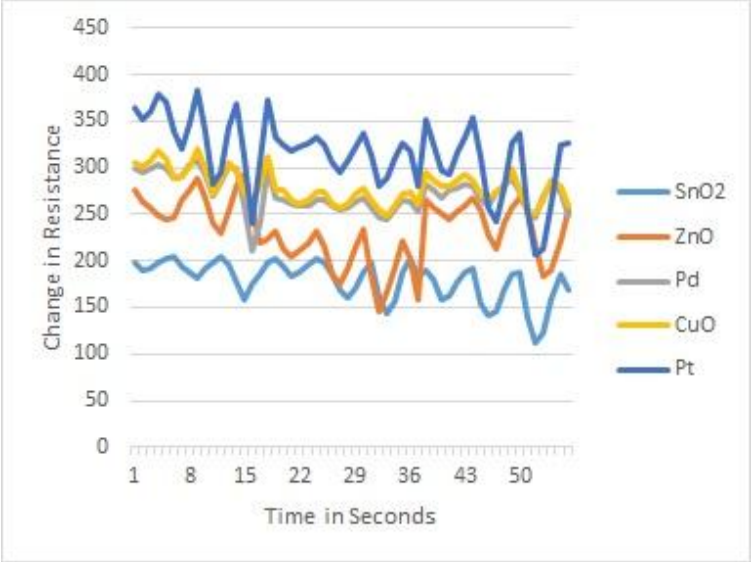

Figure 3.4. Data graph for the odor of $C_3H_7O_4$ [2]

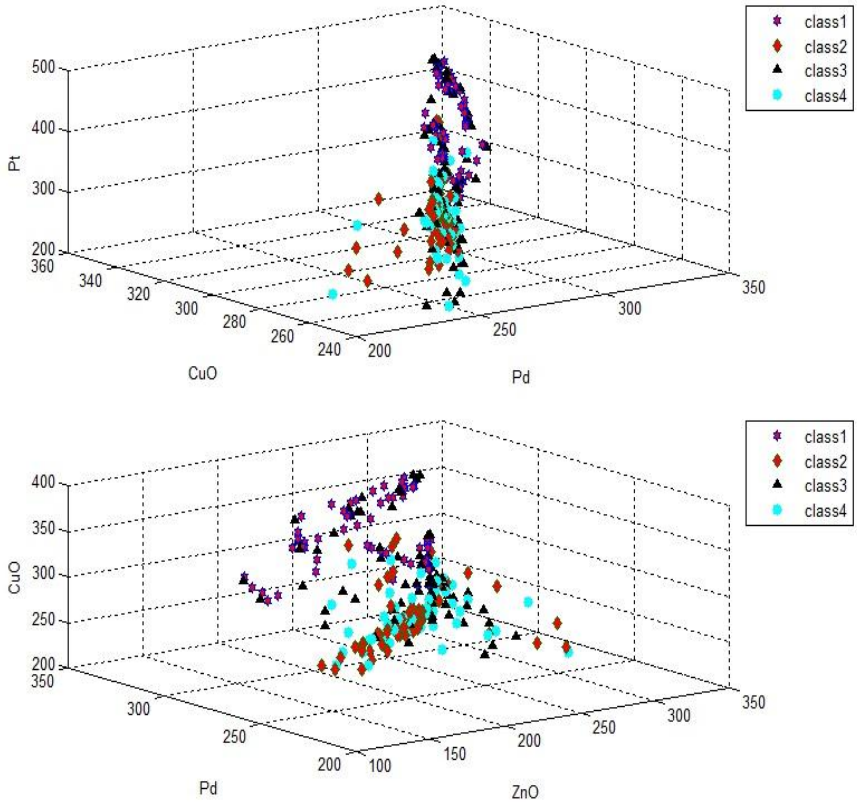

Figure 3.5. Scatter plot for the raw data. [2]

3.2. Principal Component Analysis [3]

Feature extraction or feature selection is a necessity requirement in statistical pattern classification. Feature selection often requires that the data space is transformed into a feature space in such a way that the transformed data has the same dimension as the input data but the data space can be represented by a fewer number of selected features instead of the complete feature space, yet containing the most important information of the input data. That is why it is also referred to as dimensionality reduction technique.[3]

Assume, we are having a problem where the input data that is nonlinearly separable has a very high dimension. According to Cover theorem for separability of patterns [3] the mapping the data nonlinearly to a higher dimension classifies a nonlinear data linearly, but if we map a data which already is of such a high dimension, we will be requiring a very large neural network with very large number of hidden layers and neurons. Hence it is required to decrease the dimensionality of the input data. We look for the redundancy in the 'm' dimensional data, so that we can select 'm_o' dimensions such that (m_o<m), taking into care that the dimensions that we are discarding should have unimportant data [4]. Truncating the data causes error. The mean square error hence obtained will be equal to the sum of the variance of the elements which are eliminated. We require the mean square error (MSE) to be as small as possible. Hence the data with the maximum information is restored [5].

In the pattern space of raw data, that is $\vec{x}$ domain we cannot decide what data points hold the required information hence it is decide what data points should be truncated. Therefore, we map the $\vec{x}$ vector into another space which we obtain by transforming it using a transformation matrix **T**. The transformation matrix **T** is m by m dimensional matrix and we pre multiply it with the input vector $\vec{x}$, to form a new vector $\overrightarrow{x_{new}}$ such that it will be easier for us to decide what data points should we discard so as to maximize the rate of decrease of variance.

$$\vec{x} = \left(x_1, x_2, x_{3\ldots\ldots} x_{m_0} x_{m_{0+1}\ldots\ldots\ldots} x_m\right) \tag{3.1}$$

$$\overrightarrow{x_{new}} = T\vec{x} \tag{3.2}$$

It is assumed that the vector $\vec{x}$ is a zero mean random vector and if it is not a zero mean, we can subtract the mean from all the elements of the vector to get a zero mean vector. We also consider an m dimensional unit vector $\vec{q}$, and we want to project the $\vec{x}$ onto the unit vector $\vec{q}$ to have a scalar value of projection as A.

$$E(\vec{x}) = 0 \tag{3.3}$$

$$\|\vec{q}\| = 1 \tag{3.4}$$

$$A = \overrightarrow{x^T}\vec{q} \tag{3.5}$$

We want to consider the expectation and the variance of the vector in the projected space. The unit vector $\vec{q}$ is not a random vector; hence it comes out of the expectation operator making the expectation of the projection zero.

$$E(A) = E\left(\overrightarrow{q^T}\vec{x}\right) = \vec{q}E(\vec{x}) = 0 \tag{3.6}$$

Now we require, finding the variance of the projection *A*, denoted as σ^2

$$\sigma^2 = E(A^2) = E\left[\left(\overrightarrow{q^T}\vec{x}\right)\left(\overrightarrow{x^T}\vec{q}\right)\right]$$

$$= \overrightarrow{q^T}E\left(\overrightarrow{x^T}\vec{x}\right)\vec{q} = \overrightarrow{q^T}\vec{R}\vec{q} \tag{3.7}$$

Where $\vec{R}$ is a correlation matrix of dimension m by m, and as it is a matrix of two same vectors multiplied after a transpose, they will be symmetric, which means that

$$\overrightarrow{R^T} = \vec{R} \tag{3.8}$$

And it can also be shown that if a and b are any m by one vectors then,

$$\overrightarrow{a^T}\vec{R}\vec{b} = \overrightarrow{b^T}\vec{R}\vec{a} \tag{3.9}$$

Our motive is to minimize the variance σ^2, but as we cannot control the $\vec{R}$ as it depends upon the input vector $\vec{x}$ and that is a random variable, hence we have to use $\vec{q}$ as a probe for finding the minimum of the variance. That is the variance of the projection A is a function of

the vector $\vec{q}$ and hence $\varphi(\vec{q})$ may e called as a variance probe.

$$\varphi(\vec{q}) = \sigma^2 = \overrightarrow{q^T}\vec{R}\vec{q} \tag{3.10}$$

Next, we have to find those unit vectors $\vec{q}$ such that we can find the extremal values or we can the local minima, having the constrained on the vectors $\vec{q}$ of having a Euclidean norm constrained. We can solve this difficulty by considering the eigen values of the correlation matrix $\overrightarrow{R}$. If we have a unit vector $\vec{q}$ such that we are having minima at the variance probe then a small change or perturbation in the vector $\vec{q}$ will cause no change in the value of the variance probe, such as

$$\varphi\left(\vec{q} + \overrightarrow{\delta q}\right) = \varphi(\vec{q}) \tag{3.11}$$

$$\varphi\left(\vec{q} + \overrightarrow{\delta q}\right) = \left(\vec{q} + \overrightarrow{\delta q}\right)^T \overrightarrow{R}\left(\vec{q} + \overrightarrow{\delta q}\right) \tag{3.12}$$

$$= \overrightarrow{q^T}\vec{R}\vec{q} + 2\overrightarrow{\delta q^T}\vec{R}\vec{q} + \overrightarrow{\delta q^T}\vec{R}\overrightarrow{\delta q}$$

Using the property in equation 3.9 in equation 3.11 we have got the equation 3.12. It can be seen that the last term $\overrightarrow{\delta q^T}\vec{R}\overrightarrow{\delta q}$ is negligible and hence can be neglected and using equation 3.10 in equation 3.12 we can get following results

$$\varphi\left(\vec{q} + \overrightarrow{\delta q}\right) = \varphi(\vec{q}) + 2\overrightarrow{\delta q^T}\vec{R}\vec{q} \tag{3.13}$$

But $\varphi\left(\vec{q} + \overrightarrow{\delta q}\right)$ is equal to $\varphi(\vec{q})$ to the first order of approximation; hence we can say that,

$$\overrightarrow{\delta q^T}\vec{R}\vec{q} = 0 \tag{3.14}$$

We have restricted ourselves to only those perturbations in which the euclidean norm remains to unity such that,

$$\left\|\vec{q} + \overrightarrow{\delta q}\right\| = 1 \tag{3.15}$$

$$\left(\vec{q} + \overrightarrow{\delta q}\right)^T\left(\vec{q} + \overrightarrow{\delta q}\right) = 1$$

$$\overrightarrow{q^T}\vec{q} + 2\overrightarrow{\delta q^T}\vec{q} + \overrightarrow{\delta q^T}\overrightarrow{\delta q} = 1$$

Using the result of equation 3.4 and taking the last term, $\overrightarrow{\delta q^T}\overrightarrow{\delta q}$ as negligible, we can say that

$$\overrightarrow{\delta q^T}\vec{q} = 0 \tag{3.16}$$

This shows that the vector $\vec{q}$ and the perturbation $\overrightarrow{\delta q^T}$ are orthogonal and only a change in the direction of the vector $\vec{q}$ is allowed. The elements of the vector $\vec{q}$ are dimension less, hence if we have to combine equation 3.14 and equation 3.15 we require a scalar of size m by m such that,

$$\overrightarrow{\delta q^T}\vec{R}\vec{q} - \lambda\overrightarrow{\delta q^T}\vec{q} = 0$$

Or, equivalently

$$\overrightarrow{\delta q^T}\left(\vec{R}\vec{q} - \lambda\vec{q}\right) = 0$$

$$\vec{R}\vec{q} - \lambda\vec{q} = 0 \tag{3.17}$$

The equation 3.17 can be seen as an eigen value formulation. The problem has non- trivial solution for some particular values of λ and those values are called the eigen values of the correlation matrix $\vec{R}$. Particular to those values of λ we have particular values of vector $\vec{q}$ and these values are called the eigen vectors. As the correlation matrix is symmetric the eigen values are non-negative and real, and assuming the eigen values are unique the eigen vectors are also different. If the eigen values and the corresponding eigen vectors are arranged in matrix form it can be expressed in following form

$$\vec{R}\overrightarrow{q_j} = \lambda_j\overrightarrow{q_j} \quad for\ j = 1,2,\ldots.m \tag{3.18}$$

The corresponding eigen values are arranged in the decreasing order as

$$\lambda_1 > \lambda_2 > \lambda_3 > \cdots > \lambda_j > \cdots > \lambda_m \tag{3.19}$$

Such that λ_1 is λ_{max} . Let the resulting eigen vectors be constructed by a m by m matrix.

$$Q = [q_1, q_2, \ldots q_j, \ldots q_m] \tag{3.20}$$

Hence we can combine the equation 3.18 with 3.19 and 3.20 as

$$\vec{R}\vec{Q} = \vec{Q}\Lambda \tag{3.21}$$

Where Λ is a diagonal matrix with diagonal elements the eigen values of the correlation matrix R.

$$\Lambda = diag[\lambda_1 > \lambda_2 > \lambda_3 > \cdots > \lambda_j > \cdots > \lambda_m] \quad (3.22)$$

The matrix Q is a unitary matrix, such that the column vectors of Q satisfy the orthonormality condition.

$$q_i^T q_j = \begin{cases} 1, & j=i \\ 0, & j \neq i \end{cases} \quad (3.23)$$

The above equation gives us different eigen values. Hence, we may write

$$\overrightarrow{Q^T}\vec{Q} = I \quad (3.24)$$

From which we can deduce that the inverse of the Q matrix is the transpose of the matrix itself.

$$\overrightarrow{Q^T} = \overrightarrow{Q^{-1}} \quad (3.25)$$

Hence the equation 3.21 can be written in orthogonal similarity transformation form as,

$$\overrightarrow{Q^T}\vec{R}\vec{Q} = \Lambda \quad (3.26)$$

It can also be written in the form,

$$q_j^T \vec{R} q_k = \begin{cases} \lambda_k, & k=j \\ 0, & k \neq j \end{cases} \quad (3.27)$$

PCA and the decomposition of matrix R is one and the same thing for solving the probe function. That shows the variance probe values and the eigen values are one and the same thing.

$$\varphi(\vec{q}) = \lambda_j \quad j = 1,2,\ldots,m \quad (3.28)$$

The sensor array response was sampled at regular intervals of time to yield a raw data set. 3D scatter plot of the data obtained from sampling the response recovery curves have been depicted in Figure 3.5. The figure reveals the poor selectivity of the array [10-11]. Then the

Principal Component Analysis (PCA) was performed on the raw data set PCA plot depicted in Figure 3.6, confirm the apprehension about high redundancy in the data since the three dimensions contribute only 60% of the total variance thereby making subsequent classification a challenging task.

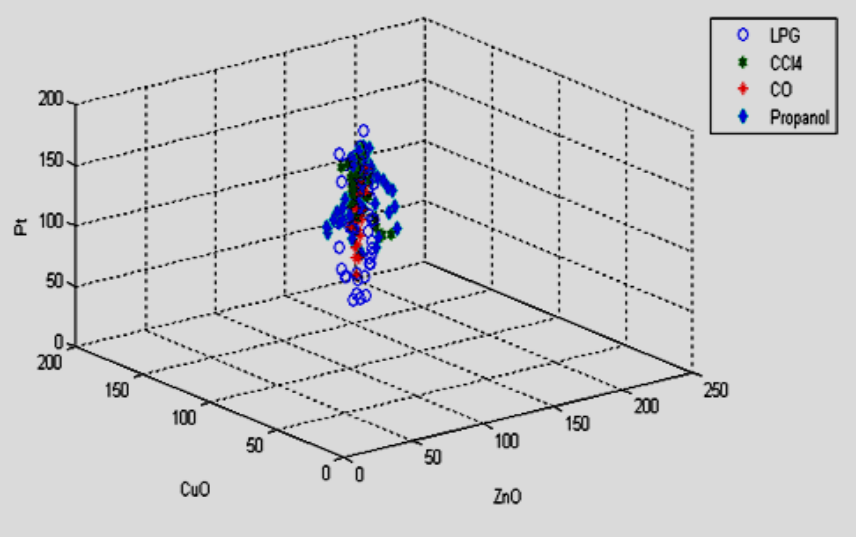

Figure 3.6. Scatter Plot for Responses of Sensors ZnO, CuO and Pt

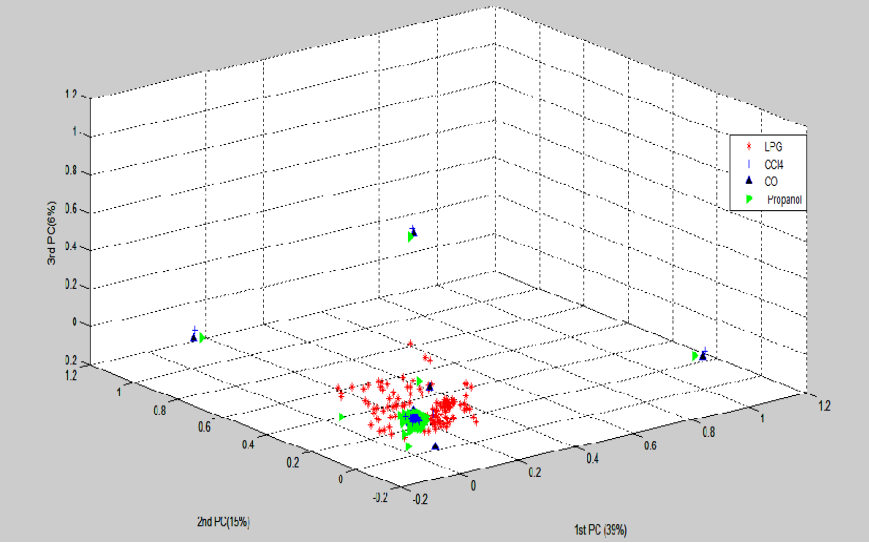

Figure 3.7. PCA Plot (60% variance contribution by first 3 PCs)

3.3. Transformed Cluster Analysis (TCA) [13]

Transformed Cluster Analysis is a supervised clustering technique that is used as a preprocessing module for the data that is coming from the sensor array. As shown in the figure 3.5 the sensor response is not spatially separated due to the overlapping clusters. Transformed Cluster Analysis arranges the data vectors around the mean of the particular class hence reducing the spatial spread of the raw data leading to a better classification [13]. For a particular gas/odor the mean of a sensor response given as

$$\overline{X_{ij}} = \frac{1}{r_{ij}} \sum_{k=1}^{r_{ij}} X_{ijk} \tag{3.29}$$

Where X_{ijk} is the response of the i[th] sensor for the j[th] odor at the concentration k. $\overline{X_{ij}}$ is the mean of the response of the i[th] sensor to the j[th] odor and r_{ij} refers to the number of observations taken from the i[th] sensor for the j[th] odor. Similarly the variance is calculated as

$$\overline{V_{ij}} = \frac{1}{r_{ij}} \sum_{k=1}^{r_{ij}} (X_{ijk} - \overline{X_{ij}})^2 \tag{3.30}$$

Where, $\overline{V_{ij}}$ is the variance of the i[th] sensor for the j[th] odor. The transformation of the data is obtained by the using the following equations.

$$T_{ijk} = \frac{(X_{ijk} - \overline{X_{ij}})}{\overline{V_{ij}}} + \overline{X_{ij}} \qquad (\text{for } \overline{V_{ij}} > 1) \tag{3.31}$$

$$T_{ijk} = (X_{ijk} - \overline{X_{ij}})\overline{V_{ij}} + \overline{X_{ij}} \qquad (\text{for } \overline{V_{ij}} < 1) \tag{3.32}$$

Here, T_{ijk} is the transformed result corresponding to the i[th] sensor for the j[th] odor at the concentration k. The results for the TCA have been discussed in the Chapter 6.

3.4. Cluster Validity Measures

Clustering is more of an unsupervised technique hence the evaluation of the clustering algorithms is of great importance. In the clustering process there are no predefined classes therefore it is difficult to find an appropriate metric for measuring if the evolving cluster configuration is acceptable or not. The result of a clustering algorithm can be very different from each other on the same data set as the other input parameters of an algorithm can

extremely modify the behavior and execution of the algorithm. The aim of the cluster validity is to find the partitioning that best fits the underlying data. Usually 2D data sets are used for evaluating clustering algorithms as the reader easily can verify the result. But in case of high dimensional data the visualization and visual validation is not a trivial tasks therefore some formal methods are needed. The process of evaluating the results of a clustering algorithm is called cluster validity assessment.

Distance Parameters: In clustering algorithms distance measure refers to the measure of similarity between different data patterns. Distance parameters can be classified as Inter cluster distance and Intra cluster distance.

- *Intra Cluster distance:* Intra cluster distance can be defined as the distance between the members of a cluster and it should be minimum possible.
- *Inter Cluster distance:* The clusters themselves should be widely separated. Inter cluster distance can be defines as the distance between the centroids of all possible pairs of clusters and it should be maximum possible.

3.4.1. Davies Bouldin Index [14-15]

This performance index is a function of the ratio of the sum of the with-in cluster scatter corresponding to intra cluster distance to the inter cluster distance [14]. The scatter within the ith cluster is formulated as

$$S_{i,q} = \left(\frac{1}{|C_i|}\sum_{x\in C_i}\{\|x - z_i\|^q\}\right)^{1/q} \quad (3.33)$$

And the inter cluster distance between C_i and C_j is formulated as

$$d_{ij,t} = \left\|z_i - z_j\right\|_t \quad (3.34)$$

$S_{i,q}$ is the qth root of the qth moment of the C_{ij} points in cluster C_i with respect to their mean z_i, and is a measure of the dispersion of the points in the cluster. Specifically $S_{i,1}$used, is the average Euclidean distance of the vectors in class *i* to the centroid of class *i*. $d_{ij,t}$is the Minkowski distance of order t between the centroids z_i and z_j that characterize clusters C_i and C_j [14]. Subsequently

$$R_{i,qt} = \max_{j,j\neq i}\left\{\frac{S_{i,q}+S_{j,q}}{d_{ij,t}}\right\} \quad (3.35)$$

The Davies–Bouldin index is then formulated as

$$DB = \frac{1}{K}\sum_{i=1}^{K} R_{i,qt} \quad (3.36)$$

The objective is to minimize the DB index for achieving proper clustering. Therefore, the fitness of chromosome j is defined as $(1/DB)$, where DB is the Davies-Bouldin index computed for this chromosome. Note that maximization of the fitness function will ensure minimization of the *DB* index.

It has been shown in the Chapter 6 that DB index is a valid performance measure for the clustering problems. It has been shown that due to TCA the spread of the data has been decreased in the pattern space and it has been proven that the inverse of the DB index increases due to TCA operation on the data [15]. The algorithm for the formulation of the DB index with the TCA if explained below

1. Input the set of data from the sensor response

   $Data = [data_j]\ for\ j = 1, \ldots.,224$

2. Divide the data into classes

   $C_i \quad for\ i = 1,2,3,4$

   $C_i \in < Data(56 * (i-1) + 1 : 56 * i, :) >$

3. Find the mean and variance of the respective classes as such

   $U_i = mean(C_i) \quad V_i = var(C_i)$

4. TCA: perform the TCA operation on the classes formed as

   $T_{ij} = \frac{(X_{ij} - U_i)}{V_i} + U_i \quad$ (for $V_i > 1$)

   $T_{ij} = (X_{ij} - U_i)V_i + U_i \quad$ (for $V_i < 1$)

5. Now we have the transformed classes as T_{ij} as the j elements in the ith class with the transformed values according to the TCA. Calculate the mean of the transformed class as

   $Z_i = mean(T_{ij})\ for\ i = 1,2,3,4$

6. DB index formulation: Calculate D_{ij} as the euclidean distance of the jth element in transformed class i from the corresponding mean Z_i of the class i.

   $D_{ij} = \|T_{ij} - z_j\|$

7. Calculate the inter cluster distance between C_i and C_j is formulated as

$d_{ik} = \|z_i - z_k\|$

This will be a four by four matrix calculating the distance of each class from the corresponding other transformed classes.

8. Find the average distance of the Euclidean distance of transformed class elements from the new mean as

$S_{i,q} = \left(\frac{1}{|C_i|}\sum_{x \in C_i}\{\|x - z_i\|^q\}\right)^{1/q} taking\ q = 1, i = 1,2,3,4$

9. Calculate the max of the inter cluster distance taking into account the intra cluster spread as

$R_i = \max_{j, j \neq i}\left\{\frac{S_i + S_j}{d_{ij}}\right\}$

10. Calculate DB index as

$DB = \frac{1}{K}\sum_{i=1}^{K} R_i$

It is important to consider the proper appropriate cluster in the supervised clustering algorithms like K-nearest neighbors where it is required to have a small seeding initial cluster for the further clusters to evolve from it. If we calculate all the DB index taking into consideration all the possible seeding clusters available, it is possible to find the proper initial seeding clusters for the clustering to proceed in a better spatial spread. Taking the same data response from the sensors, seeding clusters of size fourteen observations are made with four such clusters from each class hence there are sixteen such clusters each with fourteen samples. Now we calculate all the possible combinations of clusters that can be considered as seeding clusters and calculated DB index for them. There are two hundred and sixty four such combinations possible and the one with the minimum DB index or maximum inverse DB index is chosen. The corresponding algorithm is discussed below:

1. Divide the data set into a cell of size four-by-four having each column of cell corresponding to a particular class and each entity of cell contain fourteen observations from the class hence fifty six samples from each class is divided in four sub classes from each class.

$Data \rightarrow Dataset_{ijk}$ for $i \in 1,2,3,4 \quad j = 1,2,3,4 \quad k = 1,2 \ldots,14$

$Dataset_{ijk}$, refer to the i[th] class that has been divided into j sub-clusters, with each sub-cluster having k elements. Calculate DB index with each combination of $Dataset_{ij}$ and form the four dimensional DB index matrix

$$DB\ index\ martix_{1111} = \begin{bmatrix} DB_{1111} & \cdots & DB_{1141} \\ \vdots & \ddots & \vdots \\ DB_{1114} & \cdots & DB_{1144} \end{bmatrix}_{4X4}$$

$$\vdots$$

$$DB\ index\ martix_{4444} = \begin{bmatrix} DB_{4411} & \cdots & DB_{44441} \\ \vdots & \ddots & \vdots \\ DB_{4414} & \cdots & DB_{4444} \end{bmatrix}_{4X4}$$

There are a total of sixteen such matrices each having sixteen DB indices value. Find the minimum of the DB indices of all.

2. Find the cluster set for which the minimum DB index is obtained by finding the index values of each cluster.
3. Use the cluster set as a seeding cluster for the new clusters to evolve it.

The results of the formation of the DB index matrix and the Cluster with minimum DB index have been discussed in the Chapter 6

3.4.2. Percentage Classification

This index takes into consideration the accuracy of the clusters formed and not the spatial spread of the clusters evolved. Percentage Classification refers to the percentage of total number of samples that have been classified correctly to their particular class. We require a reference class for our evolved clusters to be compared with and hence calculate the percentage classification as

$$Percentage\ Classfication\ (P.C) = \frac{Number\ of\ samples\ correctly\ classified}{Total\ number\ of\ samples\ to\ be\ classified} * 100$$

3.4.3. Confusion Matrix [16-17]

To analyze the results of the classification performed we need to know how many samples have been classified to different classes. A confusion matrix conveys a thorough information on the performance of a classification algorithm and presents the relationship between the true class and predicted class of the sample [17]. It is a matrix where the row refers to the actual class and the sum of the elements of the row tells the number of elements in the actual class. Whereas the columns depicts the predicted class and hence the sum of the column refers to the number of samples that have been classified by the algorithm. The sum of the column tells the number of elements in the predicated class by the algorithm. This shows that the diagonal values of the confusion matrix correspond to the accurate classification and all the other elements show the misclassification. Let's assume that there are *N* classes with *Ti* samples each (i =1,…, *N*) in pattern space *D* . A classifier *C* is a black box or a function and its confusion matrix is:

$$CM(c,d) = \begin{bmatrix} cm_{11} & \cdots & cm_{1N} \\ \vdots & \ddots & \vdots \\ cm_{N1} & \cdots & cm_{NN} \end{bmatrix}$$

Where cm_{ij} means the samples of class *i* that have been assigned to *j* class, where *i* is the real class label, and *j* is the predicted class label. Confusion matrix presents the space distribution of class *C*, and shows the performance of the classification algorithm [refer2_confusion]. The diagonal elements of the matrix tell the number of correctly identified samples in the respective classes; the non-diagonal element tells the number of wrongly classified samples. Ideally, if the classifier's accuracy is 100%, then only the diagonal elements of matrix are non-zero value.

CHAPTER 4

K-means Clustering

Classification is the union of things in such way, that things in the same set which can be called as class are comparable to one another than to those in other. Classification can be viewed as the most indispensable unsupervised learning issue; it manages perceiving a pattern in a gathering of unlabeled information. A straightforward meaning of classification can be "the procedure of arranging things into classes whose individuals are undifferentiated from with a few properties". A class is a gathering of things which are reasonable inside in a class, however unlike the things fitting in with different classes [18-19].

Clustering algorithms have been utilized as a part of various applications viz. - image processing, information mining, and information investigation. Clustering aides in business field by helping advertisers to know hobbies of their particular clients in view of their buying examples and recognize gatherings of the customers. In geography, experts can use classification to perceive regions of similar atmospheric pressures, equivalent houses in a city and so on.

Clustering algorithms may be classified into the following

- Flat clustering: Creates an arrangement of classes with no express structure that would relate classes to one another; It is likewise called selective grouping.
- Hierarchical clustering: Creates a cluster tree by grouping data over a variety of scales. The cluster tree is a multilevel hierarchy rather than a group of clusters.
- Hard clustering: In hard clustering every data vector belongs to a particular cluster only. It cannot be a part of any other cluster.
- Soft clustering: It deals with the level of association of the data vector to different clusters, where data vector can belong to one or more clusters. The membership value is the degree to which the data vector is a representative of one cluster to another. Membership values lie between zero and one.

Data clustering is an intense field of exploration in which applications make their own imperative necessities. Data mining applications set the accompanying indispensable prerequisites on classification algorithms [20]:

1. Scalability: Classification applications can have a vast database. The quantities of items may range to million. Consequently, classification algorithms are obliged to be exceptionally adaptable for effectively classes to develop.
2. Ability to manage diverse sorts of properties: data vectors may have distinctive sorts, for example, numerical, ordinal, straight out, and twofold.
3. Arbitrary shape locating: Some classification methods focus classes taking into account separation measures, for example, Euclidean distance and Manhattan distance. These classification procedures structure spherical clusters while other classification algorithms are obliged to discover bunches of discretionary shapes and size, for example, taking into consideration the density of the class.
4. Insensitivity to noise: Classification methods are obliged to be obtuse to noise and outlier data vectors to evade the impact of terrible clustering.
5. High dimensionality: Many classification algorithms can capably discover classes of low dimensional data vectors. However, classification in high dimensional space is a troublesome errand in light of the fact that the separations between the classes turn out to be high and normal density of class in a certain space is prone to be low.

4.1. Introduction to K-means clustering [20-23]

K-means clustering is a data mining algorithm which performs clustering. The dataset is divided in clusters called classes such that the items of same properties lie in the same class. It is an unsupervised technique such that the clusters evolved are not known before the execution of the algorithm. Few of the clustering algorithms takes the number of desired clusters as input while some others opt the number of result clusters themselves.

In K-means clustering algorithm the data base is clustered using an iterative operation. The algorithm tales the input as the number of input clusters and the initial means of the clusters and the clustering algorithm creates the final mean of the evolved clusters as the output [20]. If the clustering algorithm takes K number of input clusters the number of initial and the final mean of the clusters evolved will also be K.

As the algorithm proceeds, a data vector is put in a particular class and the data vector becomes an entity of that cluster. The data vectors are subjected to cluster after finding the

distance from the initial cluster means and will be allotted to a class with which its distance comes out to be the minimum. The means of the new class after adding the new data vector in the class is calculated and is taken as the cluster center for the next iteration [21-22].

The K-means clustering algorithm classifies the data vectors in the dataset into the classes desired. The process of classification is an iterative task and the iterations will keep going till it reaches the convergence. After iteration, the means of the new class are calculated and updated is such a way that the new means are closer to the final required means. At the end the algorithm converges and the iteration of the process stops [23].

The concept of the convergence can be explained with the help of the figure given below. In the example the K-means clustering algorithm converges in three iterations. The initial means are represented by the blue points and can also be considered random within the data set of the approximation of the final means. The intermediate means are represented by the purple points in the data. The final means are represented by the red points in the pattern space and hence the final results of the K-means Clustering algorithm. As it has been presented by the example, the mean of a particular cluster move toward the centroid of the cluster through iteration of the K-means algorithm. The algorithm converges when the intermediate means reach the centroid of the cluster and iteration stops.

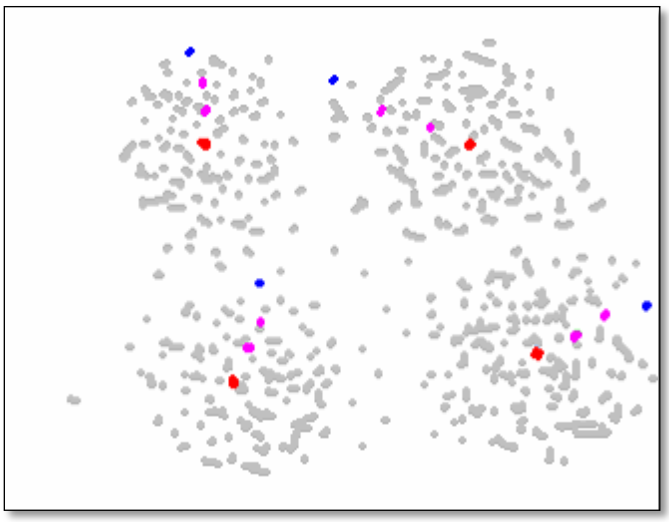

Figure 4.1. Illustration for the convergence of K-means Algorithm.

4.1.1. Measurement of Distance between Objects and Means

To compute the distance between the data vector and the cluster means different distance metrics can be used in the K-means clustering algorithm. The most common used distance metrics are Manhattan Distance and Euclidean Distance.

Manhattan distance is the simpler one of the two metrics. It is the absolute value of the difference between the data vector and the mean of the cluster. Euclidean Distance can be formulated as the square root of the addition of the squared of the differences the same dimension of the data vector and the mean of the cluster. Since, the Euclidean distance is the most commonly used distance metric when we are working with multi-dimensional data, hence we are using the Euclidean Distance parameter as the reference parameter for K-means clustering[25-27]. We have also incorporated two more different distance parameters and they will also be discussed.

4.1.2. Selection of Initial Means [28]

Selection of starting cluster means is up to the designer of clustering framework. This choice is autonomous of K-means clustering, on the grounds that these methods are inputs of K-means algorithm. A few designers want to choose initial cluster centers randomly from dataset while a few others like to create initial cluster centers.

It is realized that choice of initial cluster centers influences the execution time and accomplishment of K-means calculation. A few techniques are produced to accumulate better results considering the introductory cluster centers. The least complex of these methodologies is to execute K-means calculation with diverse arrangements of beginning means and after that select the best results. Yet, this system is not really achievable particularly for serial K-means when dataset is vast.

Another method to collect better clustering results is to refine initial cluster centers. In case it is possible to begin K-means algorithm with basic cluster centers which are closer to final cluster means, it is unequivocally possible that number of cycles that the computation needs to meet will lessen which in like manner declines the obliged time for change and extend the accuracy of final cluster means.

4.1.3. Steps of K-means clustering

The basic K- Means algorithm [26-27] description in steps in given below

a) Select ‘c’ cluster centroids arbitrarily.
b) Calculate the distance of first data point from all the ‘c’ cluster centroids.
c) Find the minimum distance and the cluster centroid corresponding to it.
d) Assign the data point to the cluster with the minimum distance.
e) Update the cluster centroid as follows

$$c_i = {}^{1}/_{N_i} \sum_{j=1}^{N_i} D_i \qquad (4.1)$$

Where c_i correspond to the updated i^{th} cluster centroid and N_i correspond to the total number of data points in the i^{th} cluster. D_i refers to the data point in the i^{th} cluster.

f) Calculate the distance of the next data point from the updated cluster centroids and repeat the steps from c to f.

The steps of K-means clustering can be easily understood in the form of a schematic representation as shown in figure 3.2. The steps show gives the information about the working of the algorithm. The iterative steps keep on proceeding till the final convergent means of the evolving clusters have been reached.

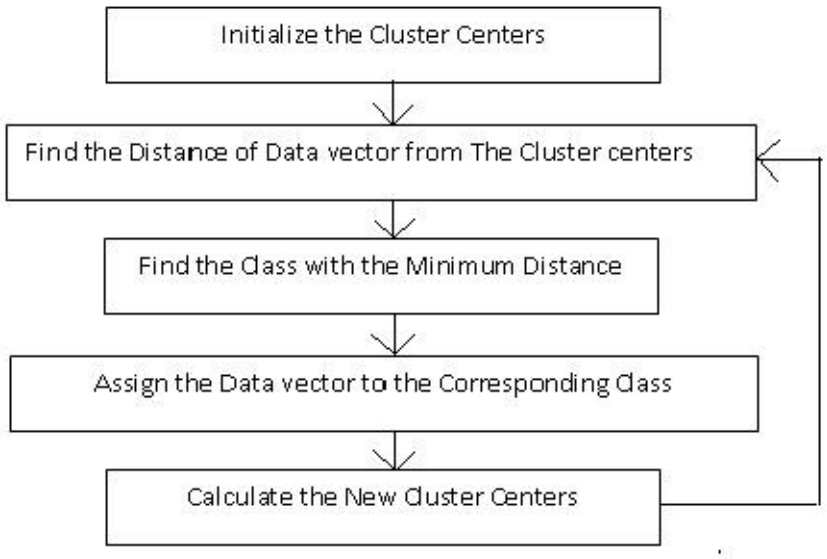

Figure 4.2.Steps of K-means Algorithm in Schematic form.

The iteration after which we want the cluster means to be updated is also dependent on the programmer. In this thesis we have used the batch processing as well as single data vector processing to update the cluster means. In the batch processing we have taken two different schematics. One with sixteen data vectors is processed and other with twenty eight data vectors being processed at a time. Hence, the number of iterations is sixteen and eight respectively in the two formats. The results are discussed in Chapter six with corresponding percentage classification and inverse DB indexes also calculated.

4.2. Normalized Cosine distance based K-means Clustering

As we have illustrated the distance parameters plays a vital role in the pattern classification, since the distance between the vectors defines the class to which the data vector belong. The euclidean distance gives the distance between two data vectors, but sometimes the spread of the class is really high in the pattern space and the spread of the cluster is more than the inter cluster distance that may lead to misclassification. Here we have considered another distance metric as normalized cosine distance where the data vectors are first divided by the magnitude of the data vector to form the unit vectors and then normalized by dividing each of the dimensions by the maximum of that dimension. Then the distance is taken as a measure of the cosine of the angle between the two vectors. Here we have taken into account the fact that the direction of the data vector will point to the cluster it corresponds to and hence the minimum angle will be formed with the class with which the data vector will a part of.

The key idea for using the cosine distance is to improve the percentage classification while increasing the Inter Cluster distance and decreasing the Intra Cluster distance. Cosine Distance measure, deals with the likeliness of two data vectors, by calculating the angle as a similarity measure between them. The angle between the two vectors refers to the direction of the vectors and hence the cosine of the angle is bounded by the value of [-1, 1]. Where the -1 shows that the direction of the vectors is opposite and the angle between them is 180 degree. Whereas the value 1 shows that the vectors are in the same direction, hence the angle between them is 0 degree [18]. Given two vectors, $\cos\theta$ function is represented by a dot product of the vectors with the magnitude as

$$\cos\theta = \frac{a*b}{\|a\|\|b\|} = \frac{\sum_{i=1}^{m} a_i * b_i}{\sqrt{\sum_{i=1}^{m}(a_i)^2} * \sqrt{\sum_{i=1}^{m}(b_i)^2}} \quad (4.2)$$

The modified cosine distance measure steps are as follows

1) Convert all the data vectors into unit data vectors as

$$\widehat{D_i} = \frac{D_i}{\sqrt{\sum_{j=1}^{n} D_i}} \quad (4.3)$$

'i' varies from [1 m] as there are ''m data vectors and each data vector is 'n' dimensional.

2) Normalize the complete data set by dividing the each dimension with the maximum value of the corresponding dimension.
3) Then calculate the normalized cosine distance, ND between the two vectors as

$$ND = 2 - a.b \quad (4.4)$$

The algorithm for the Normalized cosine distance parameter is discussed below

1. Input the set of data from the sensor response

   $Data = [data_j]\ for\ j = 1, \ldots\ldots, 224$.

   Where, the data vectors are 'i' dimensional.
2. Normalize the data

   $Normalized\_data = [normalized\_data_j] for\ j = 1, .., 224$

   $normalized\_data_j = \frac{data_{ij}}{\max_i(data_i)}\ for\ i = 1,2,3,4\ and\ j = 1, \ldots 224$
3. Convert the normalized data vectors into unit vectors

   $\widehat{Unit\_vector} = [unitvector_j] for\ j = 1, .. 224$

   $unitvector_j = \frac{normalized\_data_j}{|normalized\_data_j|}$
4. Initialize the cluster centers
5. Find the distance between the data vector and the cluster center using the cosine distance formula discussed above

   $d_{ij} = 2 - dot(unitvector_i, unitvector_j)$
6. Place the data vector in the cluster with the minimum distance.
7. Update the cluster centers as the mean of the classes updated.

8. Move to step 4
9. End

The data vectors after normalization have the values between zeros and one and after being converted to unit vector correspond to a particular direction and the cosine of the angle between the two directions proved to be a better distance parameter. The results of the normalized cosine distance parameter are discussed in the Chapter 6.

CHAPTER 5

Quaternion domain K-means Clustering

The advancement of quaternions is credited to W. R. Hamilton in 1843. Legend has it that Hamilton was strolling with his wife Helen at the Royal Irish Academy when he was abruptly struck by the thought of including a fourth dimension to multiply triples. Energized by this achievement, as the couple passed the Broome Bridge of the Royal Canal, he cut the recently discovered quaternion mathematical equation

$$i^2 = j^2 = k^2 = ijk = -1 \qquad (5.1)$$

into the stone of the extension bridge. This occasion is stamped by a plaque at the accurate area today. Hamilton spent whatever remains of his life chipping away at quaternions, which turned into the first non-commutative polynomial math to be concentrated on.

5.1. Introduction to Quaternions [29]

Up to this point we have discovered that a rotation in R^3 around an axis through the origin can be expressed by a 3×3 orthogonal framework with determinant 1. On the other hand, the matrix representation appears to be excess on the grounds that just four of its nine components are independent, likewise the geometric translation of such a framework is not get until we do a few stages of estimation to concentrate the rotation axis and angle. Besides, to form two rotations, we have to figure the result of the two respective matrices, which obliges twenty-seven multiplications and eighteen additions [29]. Quaternions are exceptionally effective for breaking down difficulties where axis in R^3 are included. A quaternion is a 4-tuple, which is a more compact representation than a rotation matrix. Its geometric importance is additionally more evident as the turn pivot and edge can be recuperated. The quaternion algebra permits us with effectively form rotations on axis. This is on the grounds that quaternion structure takes just sixteen multiplications and twelve additions.

5.2. Quaternion Algebra [29]

The arrangement of quaternions, together with the two operations of addition and multiplication, frame a non-commutative ring. Three unit vectors that represent the standard orthonormal basis of R^3 is defined as **i**=(1,0,0), **j**=(0,1,0), k=(0,0,1) . A quaternion is the sum of a scalar quantity q_0 and vector $\boldsymbol{q}=(q_1,q_2,q_3)$; as

$$q = q_0 + \boldsymbol{q} = q_0 + q_1\boldsymbol{i} + q_2\boldsymbol{j} + q_3\boldsymbol{k} \tag{5.2}$$

5.2.1. Addition and Multiplication [29]

The operation of addition on two quaternions acts component wise. The addition of two quaternions *q* above and quaternion *p*

$$p = p_0 + p_1\boldsymbol{i} + p_2\boldsymbol{j} + p_3\boldsymbol{k}$$

Then we have the result as

$$p + q = (p_0 + q_0) + (p_1 + q_1)\boldsymbol{i} + (p_2 + q_2)\boldsymbol{j} + (p_3 + q_3)\boldsymbol{k} \tag{5.3}$$

Every quaternion *q* has a negative $-q$ with components $-q_i$, i = 0, 1, 2, 3.

The product of two quaternions follows the following fundamental rule given by Hamilton:

$$\begin{gathered}\boldsymbol{i^2} = \boldsymbol{j^2} = \boldsymbol{k^2} = \boldsymbol{ijk} = -1,\\ \boldsymbol{ij} = \boldsymbol{k} = -\boldsymbol{ji},\\ \boldsymbol{jk} = \boldsymbol{i} = -\boldsymbol{ji},\\ \boldsymbol{ki} = \boldsymbol{j} = -\boldsymbol{ik}\end{gathered} \tag{5.4}$$

Now the product of two quaternions can be explained as

$$pq = (p_0 + p_1\boldsymbol{i} + p_2\boldsymbol{j} + p_3\boldsymbol{k})\,(q_0 + q_1\boldsymbol{i} + q_2\boldsymbol{j} + q_3\boldsymbol{k})$$

$$= p_0q_0 - (p_1q_1 + p_2q_2 + p_3q_3) + p_0(q_1\boldsymbol{i} + q_2\boldsymbol{j} + q_3\boldsymbol{k}) + q_0(p_1\boldsymbol{i} + p_2\boldsymbol{j} + p_3\boldsymbol{k})$$

$$+(p_2q_3 - p_3q_2)\boldsymbol{i} + (p_3q_1 - p_1q_3)\boldsymbol{j} + (p_1q_2 - p_2q_1)\boldsymbol{k} \tag{5.5}$$

We can use the dot product and cross product of two vectors on R^3, to write the above result in more compact easily understandable form.

$$pq = p_0q_0 - \boldsymbol{p}.\boldsymbol{q} + p_0\boldsymbol{q} + q_0\boldsymbol{p} + \boldsymbol{p} \times \boldsymbol{q} \tag{5.6}$$

In the above discussion $\boldsymbol{p}=(p_1,p_2,p_3)$ and $\boldsymbol{q}=(q_1,q_2,q_3)$ are the vector parts of p and q, respectively

5.2.2. Complex Conjugate, Norm and Inverse [29]

The complex conjugate of a quaternion *q*, denoted as q^*, is defined as

$$q^* = q_0 - \boldsymbol{q} = q_0 - q_1\boldsymbol{i} - q_2\boldsymbol{j} - q_3\boldsymbol{k} \tag{5.7}$$

From the above definition we can have the following results,

$$(q^*)^* = q_0 - (-\boldsymbol{q}) = q \tag{5.8}$$

$$q + q^* = 2q_0 \tag{5.9}$$

$$\begin{aligned} q^*q &= (q_0 - \boldsymbol{q})(q_0 + \boldsymbol{q}) \\ &= q_0q_0 - \boldsymbol{q}.\boldsymbol{q} + q_0\boldsymbol{q} + (-\boldsymbol{q})q_0 + (-\boldsymbol{q}) \times \boldsymbol{q} \\ &= {q_0}^2 + \boldsymbol{q}.\boldsymbol{q} \\ &= {q_0}^2 + {q_1}^2 + {q_2}^2 + {q_3}^2 \\ &= qq^* \end{aligned} \tag{5.10}$$

The norm of a quaternion *q*, denoted by |*q*|, is the scalar $|q| = \sqrt{q^*q}$. A quaternion is called a unit quaternion if its norm is 1. The norm of the product of two quaternions p and q is the product of the individual norms, for we have

$$\begin{aligned} |pq|^2 &= (pq)(pq)^* \\ &= pqq^*p^* \\ &= p|q|^2p^* \\ &= pp^*|q|^2 \\ &= |p|^2|q|^2 \end{aligned} \tag{5.11}$$

The inverse of a quaternion is defined as

$$q^{-1} = \frac{q^*}{|q|^2} \tag{5.12}$$

If *q* is a unit quaternion then the inverse of it will be its conjugate q^*.

5.3. Quaternion Rotation [29]

A vector in R^3 is expressed as a pure quaternion with real part zero. Assume a unit quaternion $q = q_0 + \boldsymbol{q}$ such that ${q_o}^2 + \|q\|^2 = 1$. Hence there exist an angle θ so that

$$cos^2\theta = {q_o}^2$$
$$sin^2\theta = \|q\|^2 \qquad (5.13)$$

Indeed, there exists an one of a kind $\theta \in [0, \pi]$ such that $cos\,\theta = q_o$ and $\sin\theta = \|\boldsymbol{q}\|$. The unit quaternion can now be composed regarding the angle θ and the unit vector $\boldsymbol{u} = \boldsymbol{q}/\|\boldsymbol{q}\|$.

$$q = cos\,\theta + \boldsymbol{u}\,sin\,\theta \qquad (5.14)$$

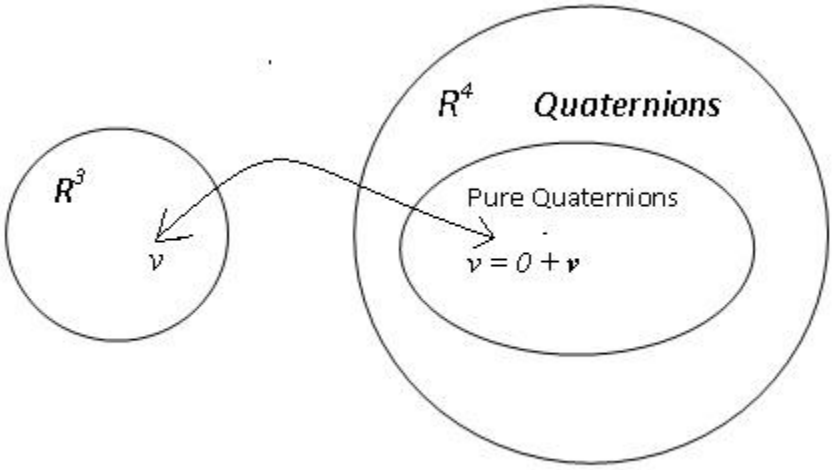

Figure 5.1. R^3 is viewed as the space of pure quaternion.

Taking the unit quaternion q in consideration we define an operator on vector $v \in R^3$:

$$L_q(\boldsymbol{v}) = q\boldsymbol{v}q^*$$
$$= (q_0^2 - \|\boldsymbol{q}\|^2)\boldsymbol{v} + 2(\boldsymbol{q}.\boldsymbol{v})\boldsymbol{q} + 2q_o(\boldsymbol{q} \times \boldsymbol{v}) \qquad (5.15)$$

Two observations are made from the above equations. First, the quaternion operator given by above equation does not affect the length of the vector ***v.***

$$\|L_q(\boldsymbol{v})\| = \|q\boldsymbol{v}q^*\|$$
$$= |q|\|\boldsymbol{v}\||q^*|$$
$$= \|\boldsymbol{v}\| \qquad (5.16)$$

Second, if the direction of $\boldsymbol{v}$ is along $\boldsymbol{q}$, the operator L_q causes no change in the direction of vector $\boldsymbol{v}$. Let us have $\boldsymbol{v} = k\boldsymbol{q}$ and

$$\begin{aligned} q\boldsymbol{v}q^* &= q(k\boldsymbol{q})q^* \\ &= \left(q_0^2 - \|\boldsymbol{q}\|^2\right)(k\boldsymbol{q}) + 2(\boldsymbol{q}.k\boldsymbol{q})\boldsymbol{q} + 2q_o(\boldsymbol{q} \times k\boldsymbol{q}) \\ &= k\left(q_0^2 + \|\boldsymbol{q}\|^2\right)\boldsymbol{q} \\ &= k\boldsymbol{q} \end{aligned} \tag{5.17}$$

Basically, any vector along $\boldsymbol{q}$ is subsequently not changed under $\boldsymbol{L_q}$. This makes us figure that the operator $\boldsymbol{L_q}$ acts like a rotation about $\boldsymbol{q}$, which will be made exact by the following theorem.

Theorem 1: For any unit quaternion

$$q = q_0 + \boldsymbol{q} = \cos\frac{\theta}{2} + \boldsymbol{u}\sin\frac{\theta}{2} \tag{5.18}$$

And for any vector $v \in R^3$ the operator works as

$$L_q(\boldsymbol{v}) = q\boldsymbol{v}q^*$$

On $\boldsymbol{v}$ is same as the rotation of the vector through and angle θ about $\boldsymbol{u}$ as the rotation axis.

Theorem 2: For any unit quaternion

$$q = q_0 + \boldsymbol{q} = \cos\frac{\theta}{2} + \boldsymbol{u}\sin\frac{\theta}{2} \tag{5.19}$$

And for any vector $v \in R^3$ the operator works as

$$L_{q^*}(\boldsymbol{v}) = q^*\boldsymbol{v}(q^*)^* = q^*\boldsymbol{v}q \tag{5.20}$$

Represents the rotation of the coordinate frame around the axis $\boldsymbol{u}$ through an angle θ while $\boldsymbol{v}$ is not rotated.

The rotation of the axis causes no effect on the length of the vector but due to the rotation of the axis a change in the distance between the two vectors is observed and it can also be explained using the Figures 5.2-5.4.

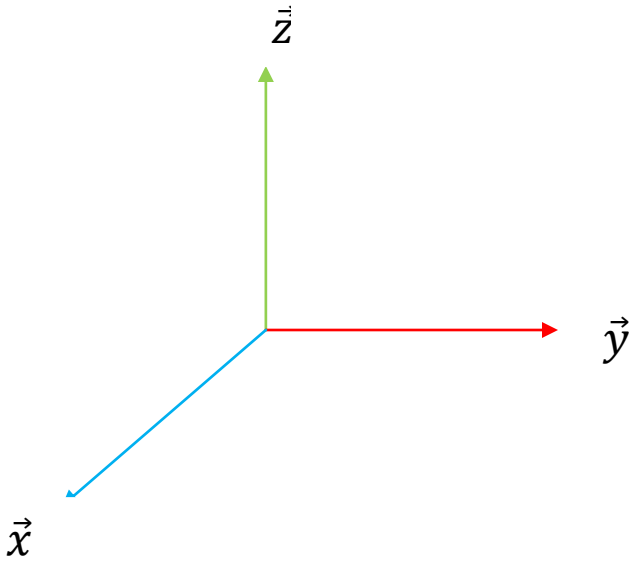

Figure 5.2 Coordinate system

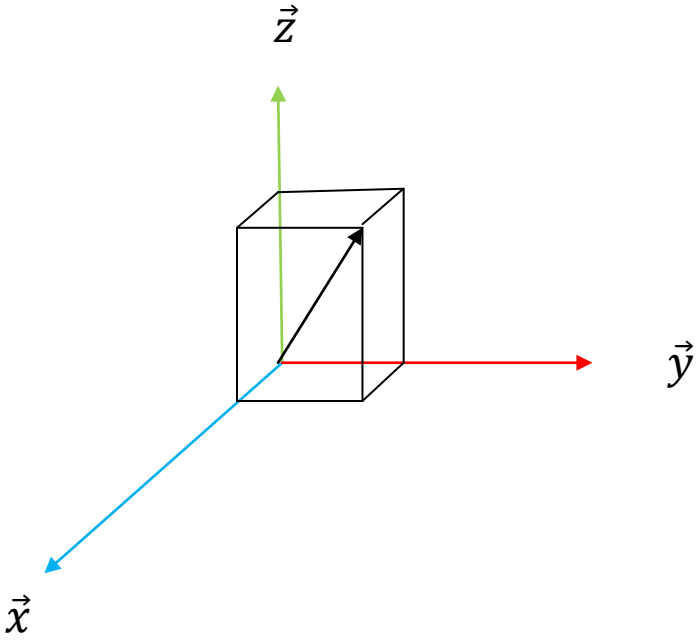

Figure 5.3 Vector in co-ordinate system

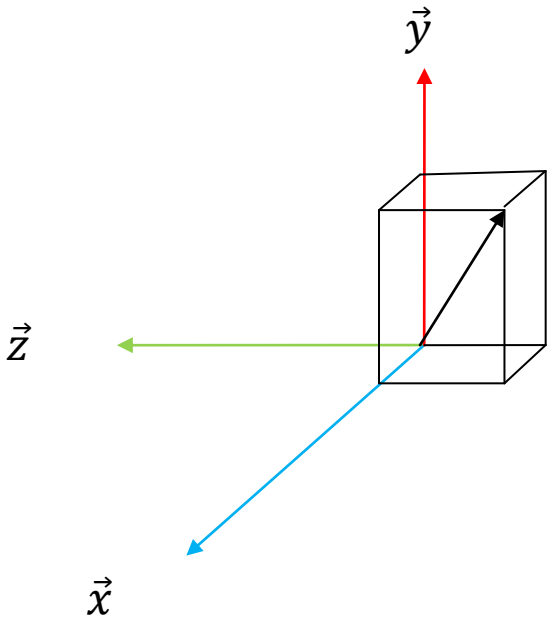

Figure 5.4 Vector in the coordinate system after the rotation of the axis

The Figure 5.4 shows the rotation of the axis by a 90 degree, which does not alter the length of the data vector. The rotation of the axis causes the data vectors to have a different orientation in the data space with every change in the degree of rotation. We have considered all the shifts caused by a rotation of 90 degree hence for every data vector we have four possible arrangements. Let us suppose, we have a data vector $\vec{v} = a_1 + a_2\hat{\imath} + a_3\hat{\jmath} + a_4\hat{k}$, then the quaternion rotation of the axis will transform the coordinate axis and the same data vector will be represented by another transformed vector representation as

$$\overrightarrow{v^1} = a_2 + a_3\hat{\imath} + a_4\hat{\jmath} + a_1\hat{k} \tag{5.21}$$

Similarly the data vector can be represented by having two more rotations of the coordinate axis each with 90 degree shift of the prior coordinate system. The new data vectors can be represented as

$$\overrightarrow{v^2} = a_3 + a_4\hat{\imath} + a_1\hat{\jmath} + a_2\hat{k} \tag{5.22}$$

$$\overrightarrow{v^3} = a_4 + a_1\hat{\imath} + a_2\hat{\jmath} + a_3\hat{k} \tag{5.23}$$

Such that

$$\vec{v} = [a_1\ a_2\ a_3\ a_4]^T$$

$$\overrightarrow{v^1} = [a_2\ a_3\ a_4\ a_1]^T \tag{5.24}$$

$$\overrightarrow{v^2} = [a_3\ a_4\ a_1\ a_2]^T$$

$$\overrightarrow{v^3} = [a_4\ a_1\ a_2\ a_3]^T$$

The rotation of the coordinate system can be explained pictorially using a three dimensional vector space and then rotating the coordinate axis by 90 degree.

For example we have a vector $\vec{v} = [1\ 2\ 3\ ]^T$ in coordinate system A.

When we rotate the coordinate axis 90 degree with reference to origin we get a new coordinate system B and for the same vector $\vec{v}$ we get a different representation $\overrightarrow{v^1}$. We can also rotate the axis by another 90 degree to get another coordinate axis C and similarly the vector $\vec{v}$ gets a new representation $\overrightarrow{v^2}$ in the new coordinate system.

$\vec{v} = [1\ 2\ 3]^T$ in Coordinate system A

$\overrightarrow{v^1} = [2\ 3\ 1]^T$ in Coordinate system B

$\overrightarrow{v^2} = [3\ 1\ 2]^T$ in Coordinate system B

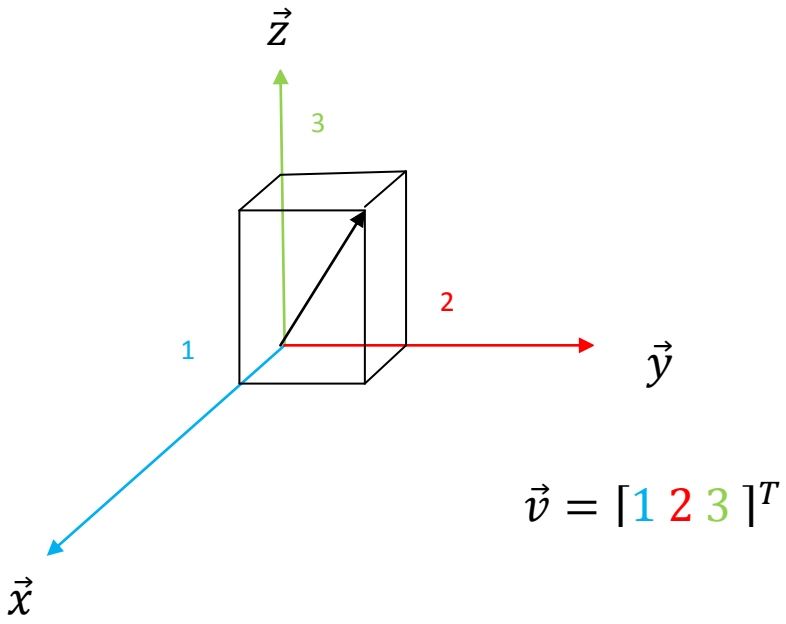

Figure 5.5. Representation of vector $\vec{v}$ in Coordinate system A.

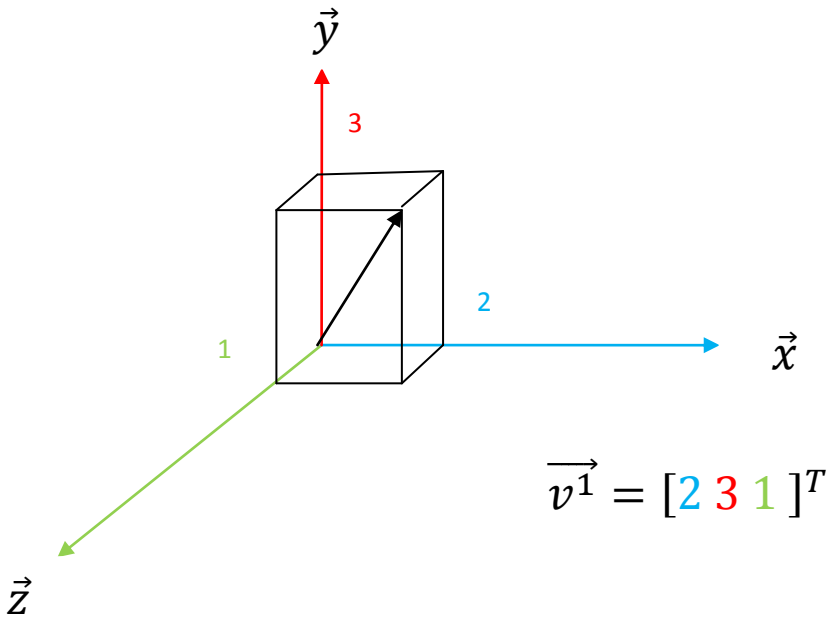

Figure 5.6. Representation of vector $\vec{v}$ in Coordinate system B as $\overrightarrow{v^1}$.

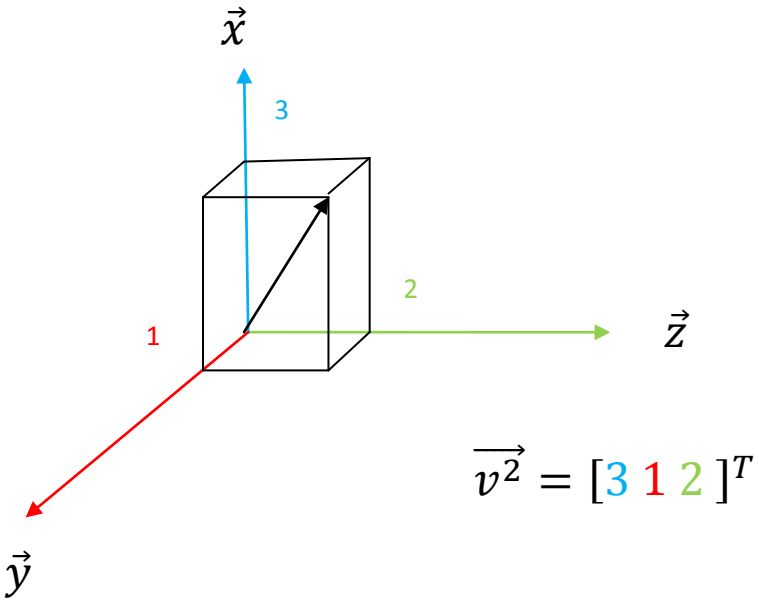

Figure 5.7. Representation of vector $\vec{v}$ in Coordinate system C as $\overrightarrow{v^2}$.

Figure 5.5-5.7 shows the transformation of the coordinate axis cause a change in the representation of the data vector. Hence for each data vector in quaternion, we have four different representations depending upon the coordinate system chosen and the data in

different coordinate system have distance between them. It must be noted that due to the rotation of the axis, the length of the data vector has not been altered.

Let us consider a data vector $\vec{x}$ =[1 2 3 4], and perform the transformation of rotating the axis to form four new data vectors,

$$\begin{aligned} \vec{x} &= [1\ 2\ 3\ 4] \\ \overrightarrow{x^1} &= [4\ 1\ 2\ 3] \\ \overrightarrow{x^2} &= [3\ 4\ 1\ 2] \\ \overrightarrow{x^3} &= [2\ 3\ 4\ 1] \end{aligned} \tag{5.25}$$

And let us have a test vector $\vec{t} = [5\ 7\ 9\ 8]$, we now calculate the euclidean distance of each quaternion from the test sample.

$$\begin{aligned} D1 &= \sqrt{(5-1)^2 + (7-2)^2 + (9-3)^2 + (8-4)^2} = 13.8390 \\ D2 &= \sqrt{(5-4)^2 + (7-1)^2 + (9-2)^2 + (8-3)^2} = 13.9572 \\ D3 &= \sqrt{(5-3)^2 + (7-4)^2 + (9-1)^2 + (8-2)^2} = 13.9703 \\ D4 &= \sqrt{(5-2)^2 + (7-3)^2 + (9-4)^2 + (8-1)^2} = 13.8785 \end{aligned} \tag{5.26}$$

From the above equations, it is evident that after transforming the data vector in the quaternion space, we have got different distance values as that of the conventional four dimensional space. This property is can be explored in the unsupervised K-means Clustering algorithm.

Hence we can use quaternions in the K-means clustering algorithm that provides us the initial coordinate axis as well as the transformed coordinate axis by rotating the coordinate axis around the origin. For every data vector and the cluster centroid we have four arrangements of the representation. With each arrangement the distance of the data vector from the cluster center (it is also considered in all the four arrangements due to quaternions) are considered and then data vector becomes a part of the class from which the minimum euclidean distance is obtained. This caused us to have a better knowledge of the data set and helps in understanding the cluster sparsity in the data space.

The results of the K-means using Quaternions have been discussed in chapter 6, and it has been shown that the quaternions help in getting better results than a normal K-means in terms of Percentage Classification (PC) as well as Inverse DB index.

CHAPTER 6

Results and Discussions

In this Chapter the results of the different techniques used in the K-means Clustering Algorithm and the results for the validation of the Inverse DB index have been put forward. The results of Transformed Cluster analysis (TCA) on DB index verifies that it can be used as a performance measure for the clustering algorithms and hence used in the performance check of all the algorithms used in the thesis.

6.1 Effect of TCA on Clustering and Inverse DB index

Figure 6.1 show the scatter plot for the raw data and the Figure 6.2 shows the transformed data using Transformed Cluster Analysis (TCA). Table 6.1 shows the comparison of Inverse DB Index for the raw data and the transformed data.

Table 6.1. Comparison of Inverse DB index for Raw Data and TCA data

	Inverse DB Index
Raw Data	0.0573
TCA data	2.3744

It is evident from the comparison of Figure 6.1 and Figure 6.2 that Transformed Cluster Analysis (TCA) decreases the Intra cluster Distance and increases the Inter cluster Distances. In Figure 6.2 the data is clustered near the mean of the clusters which cause the spread of the cluster to decrease and hence the Intra cluster distances decrease. The result has also been verified by finding the Inverse DB index, whose value increases by a factor of 41.43 times of the raw data. This shows that, Inverse DB index is a valid performance measure for the clustering techniques.

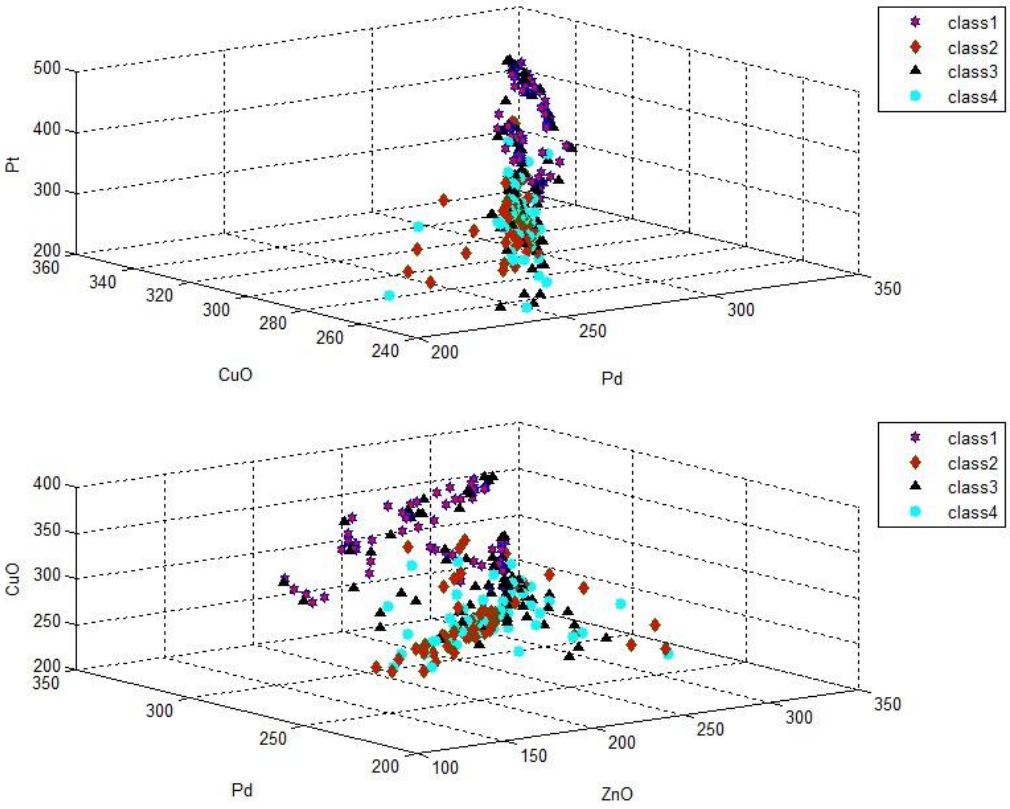

Figure 6.1. Scatter Plot of Raw data

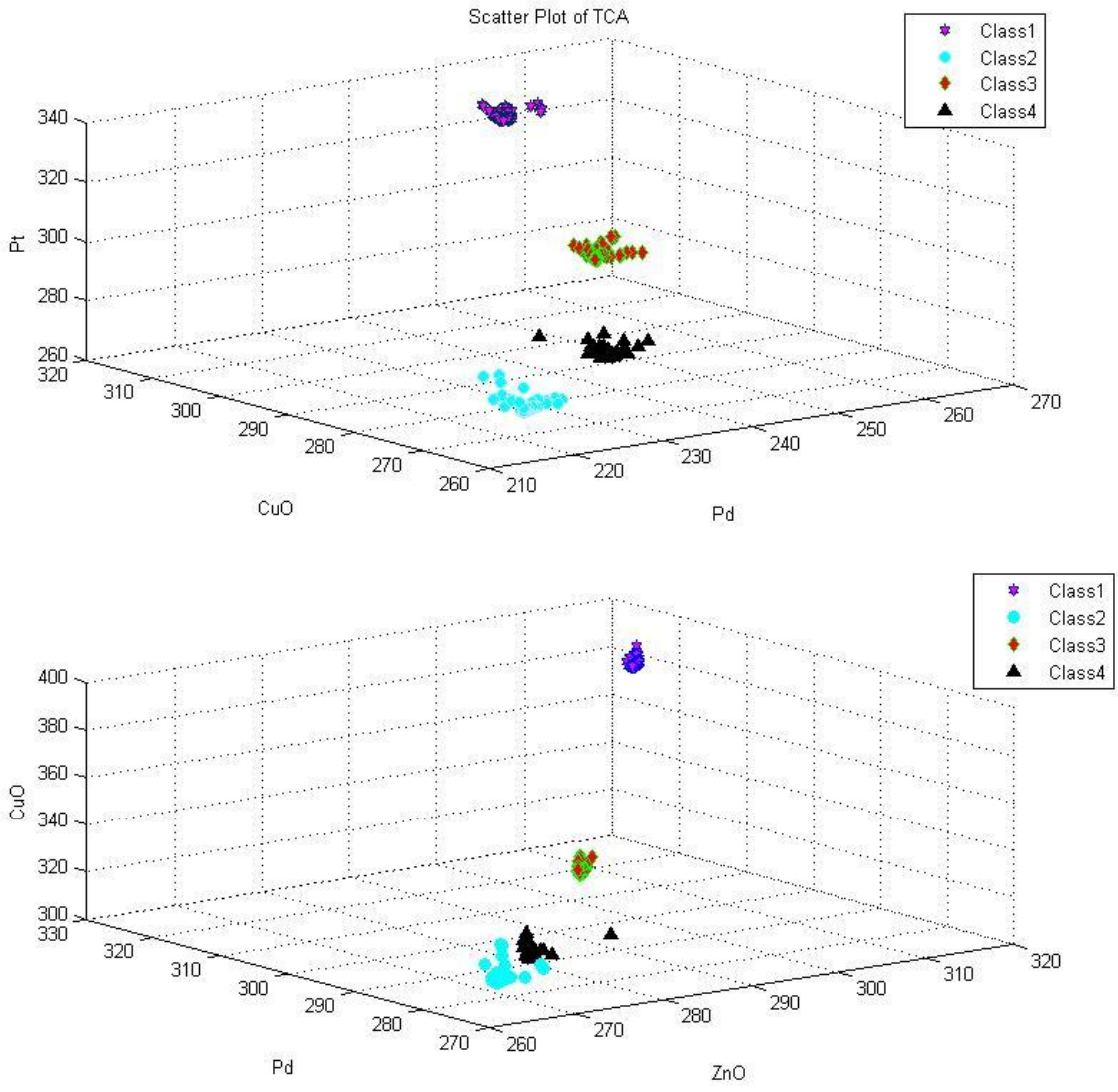

Figure 6.2. Scatter Plot from TCA

6.2. Formulation for the best group of Classes based on the Inverse DB index

For semi-supervised and supervised algorithms we need an initial seeding cluster from which the clusters are evolved. To have the best clustering results we require the seeding cluster that are very compact in the terms of Intra cluster distance and far away in terms of Inter cluster distance. The clusters should be such that the cluster mean should be near to the final means of the cluster. The data has been divided in four groups from each class comprising of fourteen data elements each, hence there are total sixteen sub classes hence there could be total two hundred and fifty six combinations of sub classes as shown in Table 6.2.

Table 6.2. Division of classes into sub-classes to form the initial seeding clusters.

Class 1	Class 2	Class 3	Class 4
C11=Data(1:14,:)	C21=Data(57:70,:)	C31=Data(113:126,:)	C41=Data(169:182,:)
C12=Data(15:28,:)	C22=Data(71:84,:)	C32=Data(127:140,:)	C42=Data(183:196,:)
C13=Data(29:42,:)	C23=Data(85:98,:)	C33=Data(141:154,:)	C43=Data(197:210,:)
C14=Data(43:56,:)	C24=Data(99:112,:)	C34=Data(155:168,:)	C44=Data(211:224,:)

Table 6.3. Minimum and Maximum Inverse DB index with the corresponding cluster sets.

	Inverse DB index	Index Of Classes	Set of Sub-Classes
Minimum	0.0127	2 4 2 4	C12,C24,C32,C44
Maximum	2.2686	1 3 2 3	C11,C23,C32,C43

While calculating the Inverse DB indexes we get a four dimensional matrix from which the minimum and maximum Inverse DB indexes have been sort. The corresponding sub-classes have also been calculated and shown in Table 6.3. The sub-classes for the maximum inverse DB index can be used for the supervised clustering algorithms. The K-means algorithm is implemented using the mean of the sub-class selected. The percentage classification and the confusion matrix is shown in Table 6.4 and Table 6.5.

Table 6.4. P.C and Inverse DB Index values for the K means Implementation.

Percentage Classification	39.2857
Inverse DB index	0.9233

Table 6.5. Confusion Matrix for the K-means implementation using the seeding cluster with maximum Inverse DB index.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	47	5	19	25
	Class2	1	22	9	68
	Class3	6	19	7	57
	Class4	2	10	21	26

6.3. Comparison of K-mean Clustering using Euclidean distance and Normalized Cosine distance.

The selection of initial cluster centers plays a vital role in the convergence of K-means clustering algorithm. The k-means algorithm is a un-supervised clustering algorithm hence if we have no knowledge about the cluster centers, we may take them randomly. Here we have done the tenfold cross validation using ten different cluster centers chosen arbitrarily from the data sets. Table 6.6 shows the selection of different cluster centers. Table 6.7 represents the Percentage Classification (P.C) and the Inverse DB index calculated for each cluster center taking two distance parameters Euclidean distance and Normalized Cosine distance parameter. Table 6.8 - Table 6.27 refer to the Confusion matrix calculations for the Euclidean distance and Normalized Cosine distance parameters.

Table 6.6. Percentage Classification and DB index values for Euclidean and Normalized Cosine Distance compared over ten folds of cross validation

Euclidean Distance		**Normalized Cosine Distance**	
Percentage Classification	Inverse DB index	Percentage Classification	Inverse DB index
37.9464	1.2836	29.0179	0.3092
24.5536	0.2751	24.5536	1.3398
31.2500	1.5355	31.2500	0.3230
31.2500	1.5355	40.6250	0.0553
42.4107	1.2590	31.2500	0.3550
31.2500	1.5355	40.6250	0.0553
41.5179	2.5344	31.6964	0.5068
32.1429	1.0943	20.0893	0.4151
34.3750	1.0906	29.9107	0.5073
43.3036	2.2323	20.5357	0.6951

Table 6.7. Confusion Matrix of the K-means Clustering for Center 1 using Euclidean Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	43	3	17	7
	Class2	6	28	18	29
	Class3	7	18	5	11
	Class4	0	7	16	9

Table 6.8. Confusion Matrix of the K-means Clustering for Center 2 using Euclidean Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	19	0	11	0
	Class2	18	6	9	17
	Class3	4	50	29	38
	Class4	15	0	7	1

Table 6.9. . Confusion Matrix of the K-means Clustering for Center 3 using Euclidean Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	43	3	17	4
	Class2	0	7	18	12
	Class3	8	36	5	18
	Class4	5	10	16	22

Table 6.10 Confusion Matrix of the K-means Clustering for Center 4 using Euclidean Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	43	3	17	4
	Class2	0	7	18	12
	Class3	8	36	5	18
	Class4	5	10	16	22

Table 6.11. Confusion Matrix of the K-means Clustering for Center 5 using Euclidean Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	44	5	16	8
	Class2	12	38	20	28
	Class3	0	6	3	10
	Class4	0	7	17	10

Table 6.12. Confusion Matrix of the K-means Clustering for Center 6 using Euclidean Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	43	3	17	4
	Class2	0	7	18	12
	Class3	8	36	5	18
	Class4	5	10	16	22

Table 6.13. Confusion Matrix of the K-means Clustering for Center 7 using Euclidean Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	28	12	14	19
	Class2	5	37	6	22
	Class3	23	0	13	1
	Class4	0	7	23	14

Table 6.14. Confusion Matrix of the K-means Clustering for Center 8 using Euclidean Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	11	0	7	0
	Class2	6	17	12	16
	Class3	25	3	11	7
	Class4	14	36	26	33

Table 6.15. Confusion Matrix of the K-means Clustering for Center 9 using Euclidean Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	27	10	14	20
	Class2	5	1	3	0
	Class3	24	0	13	1
	Class4	0	45	26	35

Table 6.16. Confusion Matrix of the K-means Clustering for Center 10 using Euclidean Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	24	0	14	1
	Class2	5	37	6	22
	Class3	27	12	13	19
	Class4	0	7	23	14

Table 6.17. Confusion Matrix of the K-means Clustering for Center 1 using Normalized Cosine Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	17	4	29	20
	Class2	10	33	8	0
	Class3	5	9	9	0
	Class4	24	10	10	6

Table 6.18. Confusion Matrix of the K-means Clustering for Center2 using Normalized Cosine Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	23	10	10	6
	Class2	20	4	29	0
	Class3	4	9	9	0
	Class4	9	33	8	23

Table 6.19. Confusion Matrix of the K-means Clustering for Center 3 using Normalized Cosine Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	26	17	11	10
	Class2	0	4	5	0
	Class3	24	27	21	0
	Class4	6	8	19	11

Table 6.20. Confusion Matrix of the K-means Clustering for Center 4 using Normalized Cosine Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	33	14	11	7
	Class2	16	29	16	0
	Class3	7	9	24	0
	Class4	0	4	5	3

Table 6.21. Confusion Matrix of the K-means Clustering for Center 5 using Normalized Cosine Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	23	13	27	23
	Class2	18	32	10	0
	Class3	0	1	10	0
	Class4	15	10	9	5

Table 6.22. Confusion Matrix of the K-means Clustering for Center 6 using Normalized Cosine Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	33	14	11	7
	Class2	16	29	16	0
	Class3	7	9	24	0
	Class4	0	4	5	3

Table 6.23. Confusion Matrix of the K-means Clustering for Center 7 using Normalized Cosine Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	10	14	8	9
	Class2	17	26	15	0
	Class3	24	13	30	0
	Class4	5	3	3	2

Table 6.24. Confusion Matrix of the K-means Clustering for Center 7 using Normalized Cosine Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	16	4	28	20
	Class2	24	10	11	0
	Class3	11	33	8	0
	Class4	5	9	9	7

Table 6.25. Confusion Matrix of the K-means Clustering for Center 9 using Normalized Cosine Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	39	20	22	27
	Class2	8	5	4	0
	Class3	9	26	12	0
	Class4	0	5	18	9

Table 6.26. Confusion Matrix of the K-means Clustering for Center 10 using Normalized Cosine Distance as distance parameter.

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	12	25	12	17
	Class2	39	22	23	0
	Class3	5	4	3	0
	Class4	0	5	18	9

6.4. Comparison of K-mean Clustering using Euclidean distance and Quaternion distance.

Quaternions provide a four dimensional rotation of a data vector. Rotation of axis in the four dimensions does not change the length of the vector but distance between two vectors gets altered. Hence the K-means Algorithm is implemented with the quaternion rotation to find new distance possibilities and compared with the standard results from the Euclidean distance parameter for K-means Clustering. Table 6.27 shows the computational result for percentage classification (PC) and the Inverse DB index.

Table 6.27. Percentage Classification and DB index values for Quaternion and Euclidean Distance compared over ten folds of cross validation.

Concept	**Quaternion**		**Euclidean Distance**	
Centre No.	Percentage Classification	Inverse DB. index	Percentage Classification	Inverse DB. index
Centre 1	43.7500	0.5345	39.2857	1.4633
Centre 2	30.8036	.2170	24.5536	0.8549
Centre 3	43.7500	.6856	31.2500	1.2431
Centre 4	43.3036	.4986	31.2500	1.3082
Centre 5	37.5000	.1582	41.9643	2.1396
Centre 6	43.3036	.5475	45.9821	1.6891
Centre 7	37.5000	.2579	41.9643	1.7136

Centre 8	38.3929	.1314	33.9286	1.3174
Centre 9	36.6071	.8244	31.2500	1.6053
Centre 10	44.6429	.1559	43.7500	2.1108

Table 6.28. Confusion Matrix of the K-means Clustering for Center 1 using Quaternion

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	22	0	0	0
	Class2	0	30	0	0
	Class3	13	7	0	0
	Class4	21	19	56	56

Table 6.29. Confusion Matrix of the K-means Clustering for Center 2 using Quaternion

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	22	0	0	0
	Class2	15	1	0	0
	Class3	10	21	0	0
	Class4	9	34	56	56

Table 6.30. Confusion Matrix of the K-means Clustering for Center 3 using Quaternion

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	23	0	0	0
	Class2	0	29	0	0
	Class3	13	10	0	0
	Class4	20	17	56	56

Table 6.31. Confusion Matrix of the K-means Clustering for Center 4 using Quaternion

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	22	0	0	0
	Class2	0	29	0	0
	Class3	10	7	0	0
	Class4	24	20	56	56

Table 6.32. Confusion Matrix of the K-means Clustering for Center 5 using Quaternion

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	26	0	0	0
	Class2	9	9	0	0
	Class3	0	24	3	0
	Class4	21	23	53	56

Table 6.33. Confusion Matrix of the K-means Clustering for Center 6 using Quaternion

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	23	0	0	0
	Class2	0	28	0	0
	Class3	13	8	0	0
	Class4	20	20	56	56

Table 6.34. Confusion Matrix of the K-means Clustering for Center 7 using Quaternion

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	16	1	0	0
	Class2	10	22	0	0
	Class3	22	0	0	0
	Class4	8	33	56	56

Table 6.35. Confusion Matrix of the K-means Clustering for Center 8 using Quaternion

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	18	0	0	0
	Class2	10	22	0	0
	Class3	24	0	0	0
	Class4	4	34	56	56

Table 6.36. Confusion Matrix of the K-means Clustering for Center 9 using Quaternion

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	13	3	0	0
	Class2	0	23	3	0
	Class3	23	0	0	0
	Class4	20	30	53	56

Table 6.37. Confusion Matrix of the K-means Clustering for Center 10 using Quaternion

Number of samples out of 224		Actual Classes			
		Class1	Class2	Class3	Class4
Predicted Classes	Class1	28	0	0	0
	Class2	0	26	0	0
	Class3	9	4	0	0
	Class4	19	26	56	56

The results verify that the quaternion transformation of the data helps in getting more information about the dataset and hence we are able to have more number of arrangements of data in the data space by rotation of the axis. Results show that in many cases of the tenfold cross validation we are able to have better results

CHAPTER 7

Conclusion and Future scope

The sensor arrays in the Machine Olfactory system when subjected to different odors gives a data that is highly correlated and hence very difficult to be distinguished in the pattern space. Due to the poor selectivity of the sensors to the different gases we have applied different processing techniques on the data for a better classification. The use of a popular unsupervised clustering technique viz. K-means has been applied to the data. Unsupervised clustering algorithms take into consideration that we have no information of the data prior and hence they are more suitable for real-time problems, but due to no knowledge the unsupervised clustering algorithms suffer for high misclassification and low accuracy. We have used two different techniques viz. Normalized cosine distance parameter and Quaternion based approach for the convergence and better classification results of the algorithm. Following conclusions can be drawn based upon the results obtained.

Normalized Cosine Distance provides an alternate distance measure for the pattern classification problem. In cases where the data is too large, Normalized Cosine distance could be a viable option. In the present work, it has been demonstrated that the proposed distance measure provides similar or some times better results over the standard Euclidean distance measure. It has also been observed that the quality of the cluster formed as demonstrated by the inverse DB index in Table 6.6, may increase even if the percentage s classification is same. This explains that the normalized cosine distance parameter evolves better clusters than the euclidean distance parameter.

Quaternions help us to find new arrangements of the data vector in data space by rotating the coordinate axis. In the proposed work we have assume four different rotations od the axis and hence four different set of arrangements for each data vector and it has been seen that due to the rotation of the axis we are able to classify the data accurately as the distance between the two vectors changes as the axis rotate. In the present work, it has been demonstrated that the Quaternions help in better classification in most of the cases as compared to the simple euclidean distance measure. But the percentage classification is

highly prone to the selection of initial cluster centers. Selecting the initial cluster centers, near the final centers, results in better classification results and better clusters.

FUTURE SCOPE

It is evident from the results presented above that there is a lot of scope to make the choice of an appropriate pattern recognition technique less dependent and more general purpose. Furthermore, future work can be extended to make the statistical properties of the data vary predictably from one set to another. This requires selection of an appropriate visualization and pre-processing technique, for which quaternion algebra can play a crucial role. Performance of the classifier with novel distance metrics should also be evaluated which in itself is a herculean task.

List of Publications

S. Choudhary, R. Kumar, “An improved K-means Clustering algorithm for classification of odor/gas sensor data,” in *STM Journal of Image Processing & Pattern Recognition Progress*, accepted 2015.

REFERENCES

[1] R. G. Osuna, "Pattern Analysis for Machine Olfaction: A Review," in *IEEE Sensors Journal,* vol. 2, no. 3, pp. 189-202, 2002.

[2] R. Kumar, "Game Theoretic Pattern Analysis for Identification of Odors/Gases Using Response of a Poorly Selective Sensor Array," in *IEEE Sensors Journal*, vol. 13, no. 3, pp. 1110-1116, 2013.

[3] S. Haykins, "Principal Component Analysis," in *Neural Networks and Learning Machines,* 3rd ed. Ontario: Pearson-Prentice Hall, 2009, pp. 373-378.

[4] A. Papaioannou and S. Zafeiriou, "Principal Component Analysis with Complex Kernel: The Widely Linear Model," in *IEEE Transactions on Neural networks and Learning systems*, vol. 25, no. 9, pp. 1719-1726, 2014.

[5] B. K. Bao, C. Xu, S. Yan, "Inductive Robust Principal Component Analysis," in *IEEE Transactions on Image Processing,* vol. 21, no. 8, pp. 3794-3800, 2012

[6] R. He, B. G. Hu, W. Zheng, and X. W. Kong, "Robust Principal Component Analysis Based on Maximum Correntropy Criterion," in *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1485-1494, 2011.

[7] H. Lu, K. Plataniotis, A. N. Venetsanopoulos, "Uncorrelated Multilinear Principal Component Analysis for Unsupervised Multilinear Subspace Learning" in *IEEE Transactions on Neural Networks,* vol. 20, no. 11, pp. 1820-1836, 2009.

[8] S. G. Dastidar, H. Adeli, and N. Dadmehr, "Principal Component Analysis-Enhanced Cosine Radial Basis Function Neural Network for Robust Epilepsy and Seizure Detection" in *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 2, pp. 512-518, 2008.

[9] C. I. Chang and Q. Du, "Interference and Noise-Adjusted Principal Components Analysis," in *IEEE Transactions on Geoscience and Remote Sensing,* vol. 37, no. 5, pp. 2387-2396, 1999.

[10] R. Kumar, R. R. Das, V. N. Mishra, and R. Dwivedi, "Wavelet Coefficient Trained Neural Network Classifier for Improvement in Qualitative Classification Performance of Oxygen-Plasma Treated Thick Film Tin Oxide Sensor Array Exposed to Different Odors/Gases," in *IEEE Sensors Journal*, vol. 11, no. 4,pp. 1013-1018, 2011.

[11] R. Kumar, R. R. Das, V. N. Mishra, and R. Dwivedi, "Fuzzy Entropy Based Neuro-Wavelet Identifier-Cum-Quantifier for Discrimination of Gases/Odors," in *IEEE Sensors Journal*, vol. 11, no. 7, pp. 1548-1555, 2011.

[12] R. Kumar, R. R. Das, V. N. Mishra, and R. Dwivedi, "A Neuro-Fuzzy Classifier-Cum-Quantifier for Analysis of Alcohols and Alcoholic Beverages Using Responses of Thick-Film Tin Oxide Gas Sensor Array," in *IEEE Sensors Journal*, vol. 10, no. 9, pp. 1461-1468, 2010.

[13] R. Kumar, R. R. Das, V. N. Mishra, and R. Dwivedi, "A Radial Basis Function Neural Network Classifier for the Discrimination of Individual Odor Using Responses of Thick-Film Tin-Oxide," in *IEEE Sensors Journal*, vol. 9, no. 10,pp. 1254-1261, 2009

[14] S. Bandyopadhyay and U. Maulik, "Nonparametric Genetic Clustering: Comparison of Validity Indices," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 31, no. 1, pp. 120-125, 2001.

[15] D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224-227, 1979.

[16] Y. Xiong, "Building Text Hierarchical Structure By Using Confusion Matrix," in *International Conference on Bio-Medical Engineering and Informatics*, pp. 1250-1254, 2012.

[17] X. Wang and X. Yao, "An Approach of Constructing ECOC Adaptively based on Confusion Matrix," in *International Conference on Computer Science and Network Technology*, pp. 545-549, 2012.

[18] L.Sahu, and B.R. Mohan, "An improved K-means algorithm using modified cosine distance measure for document clustering using Mahout and Hadoop," *International Conference on Industrial and information systems*, pp. 1-5, 2014.

[19] X. Huang, Y. Ye, and H. Zhang, "Extensions of Kmeans-Type Algorithms: A New Clustering Framework by Integrating Intracluster Compactness and Intercluster Separation," in *IEEE Transaction on Neural Networks and Learning Systems*, vol. 25, no. 25, pp. 1433-1446, 2014.

[20] X. Chen, X. Xu, J. Z. Huang, and Y. Ye, "TW-k-Means: Automated Two-Level Variable Weighting Clustering Algorithm for Multi-view Data" in *IEEE Transaction on Knowledge and Data Engineering*, vol. 25, no.4, pp. 932-944, 2013.

[21] L. Wei, W. Zeng and H. Wang, "K-means Clustering with Manifold," in *Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 5, pp. 2095-2099, 2010.

[22] H. Xiong, , J. Wu, and J. Chen," K-Means Clustering Versus Validation Measures: A Data-Distribution Perspective," in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 2, pp. 318-331, 2009.

[23] M. J. Li, M. K. Ng, Y. Cheung and J. Z. Huang, "Agglomerative Fuzzy K-Means Clustering Algorithm with Selection of Number of Clusters," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 11, pp. 1519-1534, 2008.

[24] L. Jing, M. K. Ng, and J. Z. Huang," An Entropy Weighting k-Means Algorithm for Subspace Clustering of High Dimensional Sparse Data, " in *IEEE Transaction on Knowledge and Data Engineering*, vol. 19, no. 8, pp. 1026-1041, 2007.

[25] J. Xie, S. Jiang, "A simple and fast algorithm for global K-means clustering," in *Second International workshop on Education technology and Computer Science*, vol. 2, pp. 36-40, 2010.

[26] K. Ishikawa, S. Morishita, "A simple but powerful heuristic method for accelerating K-means clustering of large scale data in life science," in *IEEE transaction on Computational Biology and Bioinformatics*, vol. 11, no. 4, pp. 681-692, 2014.

[27] S. Na, G. Yong, and L. Xumin, "Research on K-means clustering algorithm: An improved K- means clustering," in *International Symposium on Intelligent Information Technology and Security Information*, pp. 63-67, 2010.

[28] M. Xu, P. Franti, "A heuristic K-means Clustering algorithm by kernel PCA," in *International conference on Image Processing*, vol. 5, pp. 3503-3506, 2014.

[29] C. C. Took and D. P. Mandic,"The Quaternion LMS Algorithm for Adaptive Filtering of Hypercomplex Processes", *IEEE Transactions on Signal Processing*, vol. 57, no. 4, pp. 1316-1327, 2009.