Advances in Clustering based on

Inter-Cluster Mapping

A thesis submitted in fulfilment of the requirements for
The degree Doctor of Philosophy

by

Arshad Muhammad Mehar

Student ID: 16684993

September 2016

Master of Information Technology (Edith Cowan University, Australia) 2001

Graduate Diploma of Science (Computer Studies) (Edith Cowan University, Australia) 1999

Bachelor of Science (Punjab University Lahore, Pakistan) 1997

WESTERN SYDNEY
UNIVERSITY

School of Computing, Engineering and Mathematics

College of Health and Sciences, Western Sydney University

To the memory of my parents

Ghulam Rasool and Khursheed Bibi

ACKNOWLEDGMENT

First of all, I am deeply thankful to my supervisors, Professor Anthony Maeder and Dr. Kenan Matawie, for providing me an opportunity to study under their supervision. Their assistance, motivation and encouragement throughout the course of this work was amazing. I was very fortunate to have them and their guidance, feedback, scholarly and moral support. I will always treasure their comprehensive, broad knowledge and enthusiasm. I believe that without their care, supervision and friendship, I would not have been able to complete this work. I am most indebted to my parents Ghulam Rasool and Khursheed Bibi for their love, continuing affection, patience and constant encouragement during. May God protect them and bless them in paradise. I deeply thank my wife Sarwat Arhsad for her love and understanding, especially for her comforting me when I encountered sorrow and loneliness. I would like to express my appreciation to my children, Samia Arshad Mehar, Ismail Arshad Mehar, Yahiya Arshad Mehar and Laiba Arshad Mehar for their support during my study for this degree. Finally, my thanks also go to other faculty and staff members of the School of Computing, Engineering & Mathematics and to my fellow graduate students, for providing a friendly and enjoyable environment during my time here.

Declaration

The work presented in this thesis is, to the best of my knowledge and belief, original except as acknowledged in the text. I hereby declare that I have not submitted this material, either in full or in part, for a degree at this or any other institution.

Signed: _[redacted]

Date : 29-09-2016

Abstract

Data mining involves searching for certain patterns and facts about the structure of data within large complex datasets. Data mining can reveal valuable and interesting relationships which can improve the operations of business, health and many other disciplines. Extraction of hidden patterns and strategic knowledge from large datasets which are stored electronically, is therefore a challenge faced by many organizations. One commonly used technique in data mining for producing useful results is cluster analysis. A basic issue in cluster analysis is deciding the optimal number of clusters for a dataset. A solution to this issue is not straightforward as this form of clustering is unsupervised learning and no clear definition of cluster quality exists. In addition, this issue will be more challenging and complicated for multi-dimensional datasets. Finding the estimated number of clusters and their quality is generally based on so-called validation indexes. A limitation with typical existing validation indexes is that they only work well with specific types of datasets compatible with their design assumptions. Also their results may be inconsistent and an algorithm may need to be run multiple times to find a best estimate of the number of clusters. Furthermore, these existing approaches may not be effective for complex problems in large datasets with varied structure. To help overcome these deficiencies, an efficient and effective approach for stable estimation of the number of clusters is essential.

Many clustering techniques including partitioning, hierarchal, grid-base and model-based clustering are available. Here we consider only the partitioning method e.g. the *k-means* clustering algorithm for analysing data. This thesis will describe a new approach for stable estimation of the number of clusters, based on use of the *k-means* clustering algorithm. First results obtained from the *k-means* clustering algorithm will be used to gain a forward and backward mapping of common elements for adjacent and non-adjacent clusters. These will be represented in the form of proportion matrices which will be used to compute combined mapped information using a matrix inner product similarity measure. This will provide indicators for the similarity of mapped elements and overlap (dissimilarity), average similarity and average overlap (average dissimilarity) between clusters. Finally, the estimated number of clusters will be decided using the maximum average similarity, minimum average overlap and coefficient of variation measure.

The new approach provides more information than an application of typical existing validation indexes. For example, the new approach offers not only the estimated number of clusters but also gives an indication of fully or partially separated clusters and defines a set of stable clusters for the estimated number of clusters. The advantage of the new approach over several existing validation indexes for evaluating clustering results is demonstrated empirically by applying it on a variety of simulated and real datasets.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the last decade an explosive growth in information technology has increased our capabilities to generate and collect data electronically. Huge resulting datasets contain a wealth of information that could be used to improve the operations and quality of many discipline studies including business and health. Therefore, knowledge discovery from databases (KDD) has become of much interest as an option for analysing these huge datasets. Originally pattern analysis and KDD were not integrated into data management systems. Consequently data mining methods have been developed to extract structure and relationships from these types of datasets independently.

A variety of data mining methods are available for KDD today, but for complex data (as in health and economics) many challenges remain to be solved for achieving greater effectiveness and better outcomes. Often these datasets are not rich in all the important fields and this makes interpretation difficult. As an example, population datasets for large complex health problems are common yet analysis of disease clusters and multidimensional patterns of socio-economic differentials in health can be difficult. Also differences in access and use of hospitals may result in adverse health outcomes and major public health issues. Gaining insights for typical clinical or population health problems such as these, with associated large complex health datasets, increasingly relies on the use of adaptive or learning methods for the analysis, rather than simple statistical processing. Some general purposes driving data mining activities in health are diagnostics, prognostics, treatment optimizations and understanding of disease mechanisms.

Although numerous generic data mining methods and algorithms have been developed, currently available methods are not designed to handle the types of complex patterns that occur in some data. In one of the most widely used data mining activities, data clustering, typically standard clustering methods (such as *k-means*) are used despite complications such as sparseness in the datasets. Furthermore, in these situations no satisfactory techniques have been acknowledged to find the optimal number of clusters in the datasets. The research presented here is aimed at developing and applying a new approach to address these issues.

1.1 Healthcare Datasets

Today, many healthcare organizations are engaged in the generation and accumulation of different kinds of health datasets relating to clinical practice, patient information, clinical trials, resource administration, health expenditure, policies and research. State and national health agencies in both the government and private sectors maintain extensive electronic health record systems for patients and transactional record systems for episodes of their care. Health researchers and strategists continually conduct investigations leading to the derivation of new health datasets with information which complements and extends the data associated with the above record systems. Analysing healthcare problems related to subtle and interrelated datasets such as these is difficult due to the complex structure of the datasets, and consequently developing healthcare solutions for associated problems is both challenging and demanding.

This study contributes by addressing an issue at the intersection of analysing healthcare problems and developing a new approach for evaluating the clustering results to estimate the best number of clusters. Traditionally, statistical methods are used to obtain operational information from the data while data mining methods offer the opportunity to derive knowledge in an exploratory manner in terms of correlations, predictions, classifications, clustering and association rules. Such inductively derived healthcare knowledge can not only provide strategic insights into the practical delivery of healthcare, but also significantly impact other areas of health care systems. For example, adverse reactions to some medical pharmaceuticals are one of the leading causes of hospitalization and death [1-2]. Data mining techniques can complement existing systems for reporting spontaneous adverse drug reactions, by determining dependencies on variables such as underlying patient characteristics that are not captured in the normal drug reporting process.

1.2 Conceptual and Empirical Aspects

Many businesses and industries collect large volumes of data electronically, which are useful in determining trends in behaviour and broad pattern relationships, such as correlations and associations among different fields. Using a simple clustering approach may not reveal useful knowledge implicit within these kinds of large complex datasets. In this research, a concept of forward and backward mapping of common elements in a sequence of clustering results, with adjacent and non-adjacent clusters, is defined. There has been no such previous research found which is based on combined (forward and backward) consideration of different *k-means* clustering results. The word "combined" here means to map the resultant k number of clusters with k to $k+r$ (forward) and $k+r$ to k (backward) clusters $(r \geq 1)$ together to define a combined set of clusters, with more information, using inner product similarity measures.

This new approach will allow us to attack complex problems in large datasets with greater confidence of achieving useful results. Efficient exploitation of the new approach in a variety of simulated and real datasets will demonstrate how it can solve data mining problems faced today by researchers. This study will examine the use of clustering on the datasets by using the new approach and the empirical results will be compared with the performance of different existing cluster validation indexes.

1.3 Contributions to Knowledge

The main contribution of this thesis is the construction of an approach by using the standard form of *k-means* clustering algorithm to solve problems where the dataset is large and complex, and typically sparse in a number of relevant fields for the problem. This is achieved by defining a new forward and backward combined approach to determine the best choice of *k* in application of the *k-means* algorithm, and demonstrated across a range of simulated and real world health data problems in the domain of population health.

The research has resulted in the following specific significant contributions in analysing sequences of clusters for successive k values, making use of the approach*:*

- To determine the best value of k clusters
- To determine the stability between clusters at the best *k*
- To quantify properties of separation between clusters
- To determine the amount of overlapping of data element membership between clusters.

These contributions will help to answer the following major research questions:

- How can different choices of parameters and variables for a *k-means* clustering algorithm be combined to obtain more knowledge and explanation?
- How can different clustering results at adjacent and non-adjacent choice of k, number of clusters, be combined to form and obtain better clustering results and determining the best number of *k* ?

1.4 Research Outcomes

The research has produced the following outcomes:

1. A development of a mathematical formulation and computational implementation for forward and backward mapping of adjacent and non-adjacent clusters to realise the approach
2. Evidence of the effectiveness of the approach by applying it to different simulated and real datasets of different types
3. Analysis of some population health datasets by applying the approach, undertaken using R
4. Publications in conference proceedings and a book chapter related to the project.

The following three publications are the results of the contents of this research.

- Matawie, K., Mehar Muhammad, A. and Maeder, A. (2015). *An approach to determine clusters overlap for k-means clustering.* International Workshop on Statistical Modelling, Linz, Austria, vol 2, pp. 163-166.
- Mehar Muhammad, A., Matawie, K. and Maeder, A. (2013). *Determining an Optimal Value of K in K-means Clustering*. IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Shanghai, China, pp. 51-55.
- Mehar, A., Maeder, A., Matawie, K. and Ginige, A. (2010). *Blended Clustering for Health Data Mining*. Takeda, H.(ed.) E-Health, Springer Berlin Heidelberg, 978-3-642-15515-5, pp. 130-137.

1.5 Thesis Overview

The thesis is organized into six main areas as follows.

Chapter 2 (Data Mining) will include literature review on different kinds of data mining techniques especially for clustering, describing the fundamental steps and giving examples of its application in the real world in different industries.

Chapter 3 (Clustering Evaluation) will discuss different commonly used validation indexes for evaluating clustering to determine the best number of clusters.

Chapter 4 (New Approach) will define and develop the new approach, using forward and backward mapping of common elements to the corresponding clusters and combining the mapped results.

Chapter 5 (Application to Simulated Data) will discuss the generating of extensive numbers of simulated results with different levels of variation in cluster structure, as well as discussion of a collection of datasets from the literature with varying structures, using *k-means* clustering and comparing the new approach with the performance of eight existing validation indexes.

Chapter 6 (Application to Real World Datasets) will examine the effectiveness of the approach and compare this with other indexes, when applied on some real datasets from UCI and elsewhere for which results are expected based on prior clustering structure. The new approach will also use datasets from medical domain where no prior clustering information are available.

Chapter 7 (Conclusion) will summarise the work and discuss scope for improvements and better outcomes with the new approach.

Chapter 2

Data Mining

This chapter will provide a literature review and brief explanation of various types of data mining techniques including supervised and unsupervised learning approaches. It will also explain the necessary steps such as data preparation, cleansing, data types, visualization, variables selection and similarity/dissimilarity measures, to understand the description and characteristics of datasets. Also, application of different data mining methods especially the clustering approach for health datasets will be discussed.

2.1 Introduction

The systematic and progressive uptake of information and communication technologies (ICT) in a variety of fields (e.g. science, economics, engineering, business and health) has led to a rapid increase in the volume of data routinely being stored in electronic form. This makes it possible to carry out large scale studies to determine the underlying structure in the large datasets from these different fields. Different types of studies for investigating potential structure are commonly based on data mining. Thus, before describing these techniques, an explanation will first be given for the overall concept of data mining. The term data mining originated from statistics, computer science and related areas and is typically used in the context of large datasets [3]. It is a newer generation approach to data analysis by data scientists, which has grown rapidly out of the need to derive useful knowledge from massive amounts of high dimensional and large volume datasets. It is based on a paradigm of exploration and confirmation - “exploratory not analytic” [4] - also known as knowledge discovery [5], by analysing data from different perspectives and summarizing it into useful information. Technically, it is the process of finding correlations or patterns among dozens of fields [6-8] in large databases, using methods for searching through the data for patterns. Data mining can lead to the extraction of hidden predictive information from large databases which can help companies and organizations to focus on the most important information in their data warehouses [9, 10]. It is heavily used in numerous fields like banking, insurance and marketing etc. The basic steps involved in the conventional data mining process according to Fayyad [11] are shown in the figure below:

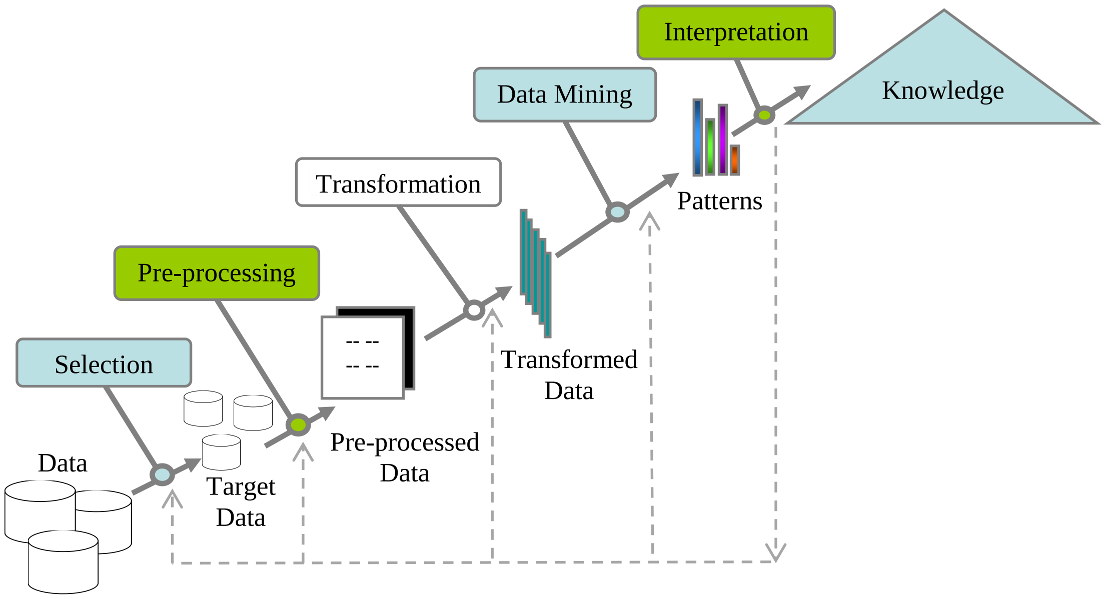


Figure 2.1: Data Mining Process.

It is important to first mention the related knowledge and understand the data mining methods and the datasets domains before we apply and progress with clustering. The basic steps for the data mining process to find and interpret patterns are summarized as follows:

- Create target data: for analysis select the appropriate data from the databases.
- Data cleaning and pre-processing: the process of cleansing data such as correcting data entry errors and deciding if outliers need removing.

- Data reduction and projection: the process of finding useful features to represent the data, using dimension reduction or transformation techniques to reduce number of variables considered.
- Choosing data mining methods: the process of selecting an appropriate data mining method in order to discover patterns of interest.
- Exploratory analysis, model and hypothesis selection: the process of deciding an appropriate model, algorithm and parameters.
- Interpretation: the process of providing information or knowledge discovered about the pattern.
- Using discovered knowledge: reporting and documenting the knowledge to the interested people and also checking and comparing with previously obtained knowledge.

Data mining techniques have been categorized into two different approaches: supervised or directed learning, and unsupervised or undirected learning. The supervised approach is used for hypothesis testing or verification while unsupervised data mining is used for knowledge discovery [12]. Association rules, clustering and feature extraction are examples of unsupervised learning. Classification, estimation, and prediction are examples of supervised learning. These two approaches are described below. Use of related techniques for health datasets are discussed later.

2.2 Supervised Learning

The supervised learning approach has already pre-defined labels (classes) for information and some prior knowledge involving predictor and response variables. The predictor is also known as the descriptor or independent variable, which is used to build the model, while the response is referred to as the dependent or outcome variable, which is predicted using a predictive model [13, 14]. The supervised learning is based on the use of a training dataset from a data source and associated response variables with already correct labels assigned [13, 15]. The several types of supervised learning techniques are regression analysis (linear, multiple, logistic) decision tree, classifier (rule base and naive base), artificial neural network and factor analysis, which will be described in more detail in the following sections.

2.2.1 Regression Analysis

Regression modelling in data mining is a method to find the relationship between dependent and independent (predictors) variables to build a model which can be used to make predictions. There are different types of well-known regression analysis techniques available which are commonly used including: linear regression, multiple regression and logistic regression which are described below.

2.2.1.1 Linear Regression

Primarily linear regression is used to predict the relationship between a single continuous predictor variable and a single continuous response variable [16]. It is a technique to produce a straight line function between independent (x) variable and dependent (y) variable. The mathematical form of linear regression and least squares [13, 16] are as follows:

$$y = a + bx + e \tag{2.1}$$

This equation shows the expected value of (y) is given by intercept (a) plus (x) multiplied by slope (b) and includes a (e) residual error.

Using least squares theory

$$b = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{2.2}$$

2.2.1.2 Multiple Regression

Datasets in many applications have hundreds or thousands of variables, many of which may have linear relationships with the response (target) variable. The main purpose of multiple regression is to provide a relationship between several independent or predictor variables and a dependent or criterion variable [16]. It is an extension of linear regression: if there are n independent (x) variables then the mathematical representation of multiple regressions [16] is

$$y = b_0 + b_1x_1 + b_2x_2 + \cdots + b_nx_n + e \qquad (2.3)$$

where $x_1, x_2, \dots, x_n$ are independent variables and y is the response variable. $b_0, b_1, b_2, \dots, b_n$ are regression coefficients and a (e) residual error.

2.2.1.3 Logistic Regression

Linear regression is only appropriate when the response variable is continuous, so it is not useful for categorical response variables. Logistic regression is used for describing the relationship between a categorical response and a set of predictor variables [14, 16]. The logistic regression has mathematical [14] form as below:

$$P(y = 1) = \frac{1}{1 + e^{-(b_0+b_1x_1+b_2x_2+\cdots+b_kx_k)}} \qquad (2.4)$$

2.2.2 Classification

In data mining, classification is used for predicting or assigning data points (often called objects) to one of several predefined categories [17]. Classification uses both categorical or a mixture of continuous numeric and categorical data. The difference between classification and regression is that the response is a categorical variable for classification while the response is a continuous variable for regression. It is the process of learning a target function f which maps each attribute set X to the predefined class labels Y [18]. This technique is capable of processing a wider variety of data to provide more information in detail than a regression model [19]. There are different types of classification like rule based classifiers, Bayesian classifiers, decision tree and artificial neural network, explained further below.

2.2.2.1 Rule Based Classifiers

Rule based classifiers are used to provide knowledge in terms of a set of rules, which tell us what should be concluded in different situations. According to [20] it is a technique for classification which is collection of a set of *IF-THEN* rules. An *IF-THEN* rule is represented in the form of "*IF* condition *THEN* conclusion". The rules for the model are represented in a disjunctive normal form

$$R = r_1 v \, r_2 v \dots r_k \tag{2.5}$$

where R is known as the set of rules and r_i 's are the component classification rules and v is the "OR" operation.

For example in medical domain, the medical decision making rules are mainly designed by medical professionals rather than by algorithms [21]. A particular example is the patient's risk of heart failure which is defined and determined by the following set of rules;

Rule 1: *IF* blood pressure is likely to be high
THEN risk of heart failure is high

Rule 2: *IF* blood pressure is likely to be low
THEN risk of heart failure is low

Rule 3: *IF* alcohol consumption is high
AND patient salt intake is high
THEN blood pressure is likely to be high

Rule 4: *IF* alcohol consumption is low
AND patient salt intake is low

THEN blood pressure is likely to be low

Rule 5: *IF* units of alcohol per week are > 30

THEN alcohol consumption is high

Rule 6: *IF* units of alcohol per week are < 20

THEN alcohol consumption is low

Rule 7: *IF* units of alcohol per week are >= 20 *AND* <= 30

THEN alcohol consumption is average

2.2.2.2 Bayesian Classifiers

Naïve Bayes Classifiers are based on Bayes Theorem which enables statistical classification for combining prior knowledge from classes with new evidence gathered from data [20]. Bayes Theorem is a statistical measure to compute conditional posterior probability from evidence of an event to understand other events [22]. According to Myatt and Johnson [14] Bayes theorem is used to compute probabilities of class membership, given specific evidence. Statistical methods are widely used for classification but success of using such methods depends on the both the size of datasets and on previous knowledge about the dataset. If A and B are random variables then the conditional and joint probability of A and B be used in the mathematical representation as described by Larose [16] as follows:

$$P\left({}^{A}/_{B}\right) = \frac{P(A \cap B)}{P(B)} \tag{2.6}$$

$$= \frac{Number\ of\ outcome\ in\ both\ A\ and\ B}{number\ of\ outcomes\ in\ B}$$

$$P(A \cap B) = P\left({}^{A}/_{B}\right) * P(B) \tag{2.7}$$

$$P(A, B) = P\left({}^{A}/_{B}\right) * P(B) = P\left({}^{B}/_{A}\right) * P(A) \tag{2.8}$$

Also,
$$P\left({}^{B}/_{A}\right) = \frac{P(A \cap B)}{P(A)} \tag{2.9}$$

$$P(A \cap B) = P\left({}^{B}/_{A}\right) * P(A) \tag{2.10}$$

$$P\left({}^{A}/_{B}\right) * P(B) = P\left({}^{B}/_{A}\right) * P(A) \tag{2.11}$$

By rearranging the above equations the following formula is obtained which is known as Bayes Theorem:

$$P\left({}^{A}/_{B}\right) = \frac{P\left({}^{B}/_{A}\right) * P(A)}{P(B)} \tag{2.12}$$

2.2.2.3 Decision Trees

A decision tree is a collection of decision nodes which are connected by branches, moving downward from the root node (decision of choice) through a path of interval nodes and finishing in leaf nodes [23].

Root node: The top level node is called root node or parent node. It consists of zero or more outgoing edges but no incoming edges.

Internal node: It is also known as non-leaf node which has only one incoming edge and two or more than two outgoing edges.

Leaf or terminal node: It is also known as external node which has zero child nodes and only one incoming edge but no outgoing edges.

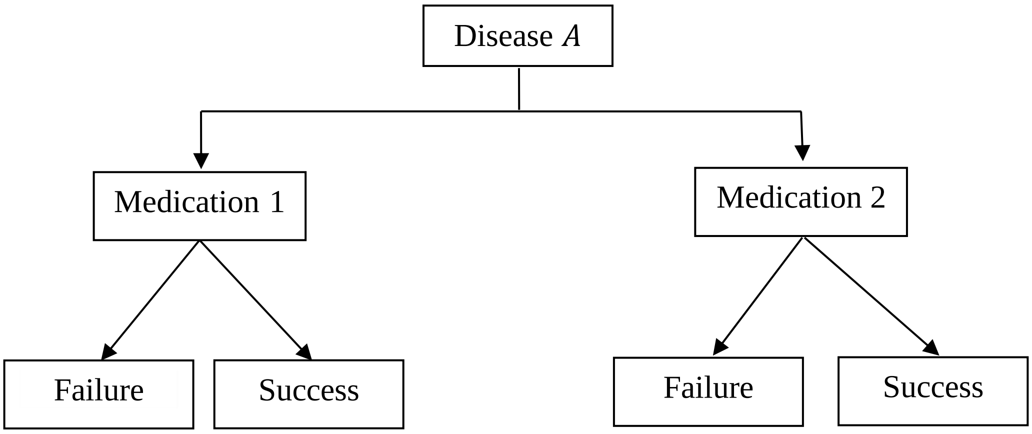


Figure 2.2: Basic structure and components of a decision tree.

2.2.2.4 Artificial Neural Networks

The study of artificial neural networks (ANN) was inspired by biological thinking systems such as how brains process information [23]. An artificial neural network is a computational and mathematical model based on biological neural networks. It was originally developed by neurobiologists and psychologists who sought to develop and test computational analogues of neurons, using a set of connected input/output units where each connection has a weight associated with it [20]. Artificial neural networks are used in data mining tasks to build nonlinear predictive models that learn through training [24]. According to [25] the simplest form of ANN is the perceptron which is a linear combination of the measurements in x that is represented by the equation:

$$f(x) = \sum_{i=1}^{p} w_i x_i \tag{2.13}$$

where w_i , $1 \leq i \leq p$ are the weight parameters of the model.

The most common form of ANN is the multilayered perceptron (MLP) which uses neurons arranged as layers (input, hidden and output layers) [26].

$$y = f\left(\sum_{j=1}^{n}\left(v_j * f\left(\sum_{i=0}^{p} w_{ij} * x_i\right)\right)\right) \tag{2.14}$$

In neural networks Larose [16], the input layers uses the input values from the training dataset along with the target set of variables and the output layers to compute the output value. Then the error is the difference between the output value and the actual value, by which the sum of squared errors can be computed. There are many different types of artificial neural networks and neural network algorithms but the most famous neural network algorithm is back-propagation. The algorithm in this approach has two phases, the forward phase and the backward phase, to compute the error of an output node. In the forward phase weights are computed in the forward direction from the weights obtained in the previous iteration, to affect the output value of every neuron in the network. So the outputs of neurons at position x are calculated prior to calculating the outputs at position $x + 1$. In the backward direction the updated weight formula is applied in reverse that is weights at position $x + 1$ are updated before the weights at position x are updated. The weights in the back-propagation are proportionally decreased or increased depending upon the direction (either forward or backward) of the error as it works its way through the system of nodes. Once all the weights have been recomputed, the input for another case is entered into the network and this process is repeated exhaustively to make the best prediction through all of the input data patterns during the training phase [18, 27].

2.3 Unsupervised Learning

In data mining unsupervised learning has no pre-defined labels (classes) and is similar to exploratory data analysis which aims to find the hidden information and relations among the variables. This approach has no target (predictor) and response variables to determine the prediction values [13]. It includes factor analysis, principal components, association rules, cluster analysis. In this section these will be described below.

2.3.1 Factor Analysis

Factor analysis is a generic term for the family of multivariate statistical techniques for the reduction of a set of observable variables into a much smaller number of latent factors. The primary purpose of factor analysis is data reduction and summarization [28]. It is used to find a set of hidden factors or latent attributes from the original set of variables, often as a linear combination [18]. This technique was originally developed to investigate human intelligence [28] for exploring the relations among observed data to assess underlying factors that may not be observed directly [29]. A mechanism and mathematical model for determining a set of hidden factors [26] by performing linear transformations on observed variables, $x_1, x_2, \dots, x_p$ to determine the set of factors $f_1, f_2, \dots, f_p$, such that

$$\begin{cases} x_1 - \mu_1 = l_{11}f_1 + l_{12}f_2 + \cdots + l_{1m}f_m + \varepsilon_1 \\ x_2 - \mu_2 = l_{21}f_1 + l_{22}f_2 + \cdots + l_{2m}f_m + \varepsilon_2 \\ \quad \vdots \qquad\quad \vdots \qquad\quad \vdots \qquad\qquad\quad \vdots \qquad \vdots \\ x_p - \mu_p = l_{p1}f_1 + l_{p2}f_2 + \cdots + l_{pm}f_m + \varepsilon_p \end{cases} \tag{2.15}$$

where, $\mu_1, \mu_2, \dots, \mu_p$ are the means of the variables $x_1, x_2, \dots, x_p$ and the terms $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_p$ represent the unobservable part of variables $x_1, x_2, \dots, x_p$ which are also called specificfactors. The terms l_{ij}, $i = 1,2, \dots, p$ and $j = 1,2, \dots, m$ are known as the loadings. The factors $f_1, f_2, \dots, f_m$ are known as the common factors. This can be written in matrix form as follows:

$$X - \mu = LF + \varepsilon \tag{2.16}$$

Given the observed variables X, along with their means μ, we attempt to find the set of factors F and the associated loadings.

2.3.2 Principal Component Analysis

Principal component analysis (PCA) is a standard tool in modern data analysis which was introduced for data reduction or dimension reduction for multidimensional data [28]. It is a technique to find a new set of dimensions that represents the variability of the new data in better way [18]. The basic idea of principal component analysis is to determine a set of linear transformations of a large number of correlated variables such that the new set of variables could provide most of the variance in a relatively smaller number of uncorrelated variables [30]. The mathematical formulations for PCA in [26] is described as; suppose if $x_1, x_2, \dots, x_p$ is a set of p variables and there are N observations of these variables, then the mean vector μ is the vector whose p components are defined as:

$$\mu_i = \frac{1}{N}\sum_{j=1}^{N} x_{ij}\,, \qquad i = 1,2,\dots,p \tag{2.17}$$

The unbiased $p \times p$ variance–covariance matrix of this sample is defined as

$$S = \frac{1}{N-1}\sum_{j=1}^{N}(x_j - \mu)(x_j - \mu)' \tag{2.18}$$

Finally, the $p \times p$ correlation matrix R of this sample is defined as

$$R = D^{-\frac{1}{2}} S D^{\frac{1}{2}} \tag{2.19}$$

where the matrix $D^{\frac{1}{2}}$ is the sample standard deviation matrix, which is calculated from the covariance S as the square root of its diagonal elements, while the matrix $D^{-\frac{1}{2}}$ is the inverse of $D^{\frac{1}{2}}$

$$D^{\frac{1}{2}} = \begin{bmatrix} \sqrt{S_{11}} & 0 & \cdots & 0 \\ 0 & \sqrt{S_{22}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{S_{pp}} \end{bmatrix} \tag{2.20}$$

2.3.3 Association Rules

In data mining, use of association rules is an unsupervised learning process where no a priori information being used. It is a process for determining important relationships between variables, which is also known as affinity analysis or basket analysis [31-33]. Each association rule is in the form of “if antecedent, then consequent” together with a degree of the support and confidence associated with each of the rules [23, 34]. These two terms, support and confidence, are very important in measuring the strength of the association (relationship) rule for the products or items. To measure the association rules two terms need to be define, support in which we determine how often a rule is applicable in the given dataset while confidence determines how frequent items I appeared in transactions T found to be true [35-37]. In [32] association rules are explained such as, suppose $I = I_1, I_2, \ldots, I_m$ are set of various items and $T = T_1, T_2, \ldots, T_n$ are transactions in such a way that any transaction is a subset of items taken from I i.e. $T \subseteq I$. Then an association rule is an implication of the form $A \Longrightarrow B$ where, A and B are a disjoint set of items i.e. $A \cap B = \emptyset$ where A is known as antecedent and B consequent. In [38] a study was carried out to describe association rules for both categorical and quantitative variables for large data. One of the most popular a priori rules, which allows that any subset of frequent items must be frequent is known as “co-occur”. Most of the forms of association rule algorithms such as Apriori, Charm, FP-growth, Partition and DIC and MagnumOpus [39] are based on this.

2.3.4 Cluster Analysis

Cluster analysis is the process of grouping a set of data values (often called objects) into classes of similar objects. It results in groups of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters [20]. There is a wide variety of clustering algorithms and the behaviour of every algorithm is different. Some algorithms may produce different results on the same dataset on different occasions, while different algorithms may lead to different results on the same dataset. Cluster analysis has been utilized successfully in various types of fields and problems: for example, in medicine, clustering cures for diseases, or symptoms of diseases can lead to very useful deductions of relatedness; in psychiatry a better way of therapy may be based on clustering symptoms such as paranoia, schizophrenia, etc. and in archeology clustering may establish taxonomies of stone tools, funeral objects, etc. [40, 41]. There are various types of cluster techniques that are divided into different categories such as partitioning, hierarchical, density based and model based methods, discussed in the sections below.

2.3.4.1 Partitioning Methods

In cluster analysis, partitioning algorithms divide the dataset into clusters based on some criterion applied to simple cluster statistics, such as means, modes and medoids. This is a very simple, basic and iterative approach to determine clusters, which partition the N number of objects in the $\mathscr{D}$ dataset into k number of clusters with $(k \leq N)$. In this section some well-known partitioning clustering algorithms (*k-means, k-medoids* (PAM), *k-modes,* CLARA and CLARANS) are described below:

2.3.4.1.1 K-Means

k-means clustering is a technique that classifies a given set of data into clusters, which are represented by their centroids in such a way that objects within a group are more similar to each other than objects in different groups [42] and this is regarded as one of the simplest clustering techniques [43]. In the *k-means* method as described by Han and Kamber [20], N number of objects are partitioned into k number of clusters starting with k initial centroid guesses, where k is the number of desired clusters specified by the user. Each point is assigned to the closest centroid and each collection of points assigned to a centroid defines a cluster. For each cluster the centroid is updated based on the points assigned to the cluster, and then the algorithm repeats the assigning and update process until no point changes between clusters and consequently all centroids remain the same. The steps in the *k-means* algorithm are as follows**:**

Input:

k : The number of clusters,

$\mathscr{D}$: Dataset containing N objects.

Output:

A set of k clusters.

Method:

1) Arbitrarily choose k objects from $\mathcal{D}$ as the initial cluster centers.
2) Repeat
3) Reassign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;
4) Update the cluster means, i.e., calculate the mean value of the reassigned objects for each cluster;
5) Until no change;

k-means is more efficient and effective for dealing with large datasets than some other clustering algorithms, and its overall computational complexity is $O(Nkt)$, where N = number of objects in the dataset, k = number of clusters and t = number of iterations. It is not appropriate for determining clusters with non-convex shapes and clusters of very different size. It is also sensitive to noise and outlier objects which may impact the convergence of the mean values.

2.3.4.1.2 K-Medoids or Partition Around Medoid (PAM)

k-medoids clustering is also a partitioning based clustering algorithm: it is a modified form of *k-means* that partitions the data based on medoids. It is also known as PAM (partition around medoids) described in [44] and its process is closely related to *k-means*. A problem in *k-means* clustering is that it is very sensitive to the outliers and there may be no objects close to the mean (or centroid) in the clusters [45]. Due to this issue the medoid object is chosen from the data to represent the cluster, which is a better choice than the centroid as it is still a central object in the cluster but is less sensitive to others.

The steps in the *k-medoids* algorithm are as follows**:**

Input:

k: The number of clusters,

$\mathscr{D}$: Data set containing Nobjects.

Output:

A set of k clusters.

Method:

1) Choose k objects from $\mathscr{D}$ reprehensive objects as the initial medoids.
2) Repeat
3) Reassign each object to the cluster with nearest object.
4) For each representative object randomly select a non-representative object.
5) Compute the total dissimilarity cost by swapping representative object with randomly non-representative object.
6) If total cost < 0 then replace representative object with non-representative object.
7) Until no change.

Experimental results using the PAM algorithm showed satisfactory performance for small datasets (e.g., 100 objects in 5 clusters) [44], while *k-medoids* was costly and inefficient for large datasets [46]. The computation complexity for each iteration is $O(k(N-k)^2)$ and for large values of N and k computation will be much more expensive than *k-means*.

2.3.4.1.3 CLARA (Clustering LARge Application)

CLARA clustering is designed and proposed by Kaufman and Rousseeuw [44] for partitioning larger datasets than would be desirable when using PAM, and is based on sampling. In this approach, instead of finding representative objects as medoids for the whole dataset, it draws a random sample of objects from the dataset and then applies PAM on this sample to find the candidate medoids. If the sample is drawn in a sufficiently random way the sample medoids will represent the entire dataset well enough. For a better approximation CLARA draws multiple random samples and applies the PAM to each sample to find the best clustering partitions as output. The clustering measure can be based on the average dissimilarity of the objects for the entire dataset and not only for those objects in the random samples. The steps in the CLARA algorithm are as follows**:**

Input:

k: The number of clusters,

$\mathcal{D}$: Data set containing Nobjects.

Output:

A set of k clusters.

Method:

1) For $i = 1$ to 5 , repeat the following steps:
2) Draw a sample of 40+2k objects randomly from the entire dataset, apply PAM algorithm to find the medoids of the sample.
3) For each object in the entire dataset, determine which of the k medoids is the most similar to object.
4) Calculate the average dissimilarity of clustering obtained in the previous step. If this value is less than the current minimum, use this value and retain the k medoids found in step (2) as the best medoids obtained so far.
5) Return to step (1) to start next iteration.

Experiments results reported in [44] show that five samples of size 40 +2k give satisfactory results in a dataset of size 1000 observations in 10 clusters. However, as

CLARA is based on sampling to find the best k medoids it will not necessarily find the best clustering, and if the random sampling is biased it will degrade clustering results for the whole dataset using this approach. In [47] the proposed solution to handle this problem is to draw several samples and use these to cluster the entire dataset several times, and finally select the results with minimum average dissimilarity. The computational complexity for each iteration is $O(ks^2 + k(N - k))$, where $s =$ size of sample, $k =$ number of clusters and $N =$ number of objects.

2.3.4.1.4 CLARANS (CLustering Algorithm based on RANdomized Search)

CLARANS is an efficient medoids based clustering algorithm. It is used for spatial data mining to find the interesting relationships and characteristics which may be exist implicitly in large and spatial datasets. It is a combination of PAM and CLARA, but the key difference between PAM and CLARANS is that the former only searches a subset of neighbours node i.e., a set of k mediods (set of objects) to define the cluster [48]. It is an optimization algorithm which draws a random sample of arbitrary node to check and find the maximum number of neighbours of node (maxneighbour), where random sample node is specified by the user. Here, the clustering process implies every node is a potential solution. The clustering obtained after replacing a medoid is called the neighbour of the current clustering (current node). Once a better neighbour is located with lower error, CLARANS moves to the neighbour's node and starts the process again. If a better neighbour is not located current clustering provides (numlocal) a local minimum and the algorithm begins with newly selected nodes searching for a new local minimum. Once a user specified numbers of local minima are searched, the algorithm stops and outputs the best local minimum with lowest error (mincost).

The steps in the CLARANS algorithm are as follows**:**

Input:

k: The number of clusters,

$\mathscr{D}$: Data set containing Nobjects.

Output:

A set of k clusters.

Method:

1) Input parameters maxneighbour, mumlocal. Initialize i to 1 and mincost to a large number.
2) Set current to arbitrary node.
3) Set j to 1.
4) Consider a random neighbour S of current, and calculate the cost differential of the two nodes.
5) If S has a lower cost, set current to S and go to step 3. Otherwise increment j by 1. If $j \leq$ maxneighbour go to step 4.
6) When $j >$ maxneibhbour, compare the cost of current with mincost. If the former is less than mincost, set mincost to cost of current and best node to current.
7) Increment i by one. If $i >$ numlocal, ouput best node and stop, otherwise, go to Step 2.

The performance of this approach is more efficient, effective and scalable than PAM and CLARA in terms of quality of clustering and running time with computational complexity $O(N^2)$ [20, 49, 50] for each iteration where as above N is the number of objects.

2.3.4.1.5 K-Modes and K-Prototypes

The partitioning algorithms using numerical values defined above are based on taking the mean, medoid or sample of data for medoids of the object as an initial reference object for computation, which is regarded as the most centrally located object in a cluster. However, in the case of handling categorical data *k-modes* and for mixture of data *k-prototypes* were proposed by Huang [47]. *k-modes* is a frequency-based method to update modes as representatives of clusters. New modes for minimizing the clustering cost function are computed by using dissimilarity measures such as simple mismatches [44]: a smaller number of mismatches indicates objects are more similar. The steps in the *k-modes* algorithm are as follows**:**

Input:

k: The number of clusters,

$\mathcal{D}$: Data set containing Nobjects.

Output:

A set of k clusters.

Method:

1) Choose k initial modes from a dataset $\mathcal{D}$, for each cluster.
2) Repeat
3) Assign each object in $\mathcal{D}$ to a cluster whose mode is the nearest one to this object. Update the mode of the cluster after each assigning.
4) After all objects have been assigned to a cluster, recalculate the similarity of objects against the new modes. If an object is discovered such that its nearest mode belongs to another cluster rather than its current one, reassign this object to that cluster and update the mode of each cluster.
5) Until no object has changed cluster membership.

Another approach called *k-prototypes* applies to a mixture of categorical and numerical data as described in [47, 52], using combined dissimilarity measures such as Euclidean distance and simple matching dissimilarity measures for numeric and categorical variables respectively. The steps for the *k-prototypes* algorithm are as follows**:**

Input:

k: The number of clusters,

$\mathscr{D}$: Data set containing Nobjects.

Output:

A set of k clusters.

Method:

1) Choose k initial prototypes from a dataset $\mathscr{D}$, for each cluster.
2) Repeat
3) Assign each object in $\mathscr{D}$ to a cluster whose prototype is the nearest one to this object. Update the prototype of the cluster after each assigning.
4) After all objects have been assigned to a cluster, recalculate the similarity of objects against the current prototypes. If an object is discovered such that its nearest prototype belongs to another cluster rather than its current one, reassign this object to that cluster and update the prototypes of both clusters.
5) Until no object has changed clusters after a full cycle test of $\mathscr{D}$.

It is claimed in [51] that *k-modes* is computationally much slower than *k-means* but faster than *k-medoids*, while Huang [47] claimed that *k-modes* algorithm is faster than *k-means* and *k-prototype* as it requires less number of iteration to converge.

2.3.4.2 Hierarchical Clustering

The hierarchical clustering technique produces a multi-scale hierarchical cluster structure in either a top down or bottom up fashion, creating a hierarchy of overlapping clusters from small to large and its complexity is at least $O(N^2)$ [27, 53, 54]. It is a tree-like diagram (also known as dendrogram) built through recursive partitioning (divisive cluster) or combining (agglomerative cluster) described by Larose [23] which are two different types of hierarchical clustering algorithms [55] defined as:

Agglomerative clustering methods construct a hierarchy of clusters in bottom up fashion starting with each data point being assigned to its own cluster [27]. Many authors described and used this method with details. According to [26, 60] agglomerative clustering is considering initially each object as a cluster by itself. Then using some distance metric, the pairs of clusters are merged to obtain a single all-inclusive cluster. AGNES (AGglomerative NESting) described in [44] is an example of agglomerative clustering. Initially, it places each object into its own cluster and finally these clusters are merged using distance metric. In [61] mentioned agglomerative algorithms are the single-link, the average-link and ward's method.

Divisive clustering uses a top-down approach to construct the hierarchy of clusters, which begins with all of the objects in the same cluster. In each successive iteration, a cluster is split up into smaller clusters, until each object is in one cluster, or until a termination condition holds [20]. A number of clustering algorithms based on the hierarchical concept have been developed such as BIRCH [56], CURE [57] and ROCK [58]. Hierarchical clustering has been used in many applications. In [59] the development of a wireless sensor network based on hierarchical clustering is described which will save on the communication cost and energy utilization. For labeling documents, a bottom up and top down hierarchical clustering is developed in [60]. Even though a hierarchical clustering algorithm is used in many fields there is no common technique which can be applied universally. The main advantage of the hierarchical clustering is that it provides the ordering and information of the object while disadvantage is high computational cost [61]. Further, creating a poor partition in the early steps may group the objects incorrectly and this is not easily backtracked [62].

2.3.4.3 Density Based Clustering

This technique discerns clusters of arbitrary shape in datasets with noise [22]. It determines the high density regions of the objects in the data space, which are separated by other regions of low density. The algorithm for density based clustering defines core points, border points and noise points. It has two input parameters, MinPts and Eps, where MinPts is the minimum number of data points in any cluster and Eps is the threshold or maximum radius of any cluster. Any two core points which are close to one another are put in the same cluster. Similarly, any border point which is close to a core point is assigned to the same core point cluster, while the noise points are eliminated [18]. Density based cluster analysis plays an important role for providing useful information and knowledge and is widely used in the areas of earth and science, biomedical image segmentation, molecular biology, astronomy and geographical data clustering [63].

2.3.4.4 Grid-Based Clustering

Grid based clustering is a space driven approach that uses a multiresolution grid data structure. In multiresolution grid data, objects quantize into a finite number of cells that form the grid structure on which all of the operations for clustering are performed [64]. In this clustering approach [14] the individual objects are not used but a finite number of cells are utilizes and this makes the algorithm executed faster.

There are different types of grid-based clustering algorithms available such as STING [65], CLIQUE [66] and WaveCluster [67]. These three algorithms utilize a uniform grid mesh to cover the whole problem space. For problems with highly irregular data distributions, the resolution of the grid mesh must be fine enough to obtain a good clustering quality. According to [68] due to the efficiency of grid partitioning computation, this approach has a fast processing time.

2.3.4.5 Model Based Clustering

In the previous sections different types of heuristic clustering algorithms were described, which rely on computations testing each object individually. Model based clustering assumes that the data is based on the generation of a mixture of ρ probability distributions (e.g. multivariate normal distributions) [69, 70] and cluster memberships of the dataset are not known. In model based clustering the purpose is to find the parameters of the cluster distributions by maximum likelihood estimation and Bayesian information criteria to determine the most likely model [71, 72]. It is an approach to optimize the fit between the data and some mathematical model, where every cluster is identified by one of the distributions. In the mixture likelihood approach, the Expectation-Maximization (EM) algorithm [73] is widely used for estimating parameters of a finite mixture probability density. The Expectation-Maximization algorithm is used to find a maximum likelihood estimate (MLE) of parameters in the model through an iterative process [22]. The EM algorithm adjusts an initial k clusters with two steps of iteration. Expectation steps (E-steps) for assigning the object to cluster with a centre that is close to the object and Maximization steps (M-steps) for estimating the model parameters [64]. The final model can be found by using the Bayesian Information Criterion (BIC) [74, 75] where the highest BIC value is used to determine the best model (for more description and information about BIC refer to Schwarz [74]).

2.4 Fundamental Steps in Data Preparation

Pre-processing is very fundamental and an important first step to prepare the data for analysis before applying any data mining method. It has been said [76] that pre-processing (cleaning) to deal with missing or incorrect values can take 80% of the total analysis time. To undertake data preparation, it is necessary to know the structure of data such as types of variables, their statistics and visualization, before applying an appropriate approach of data mining (e.g. clustering, regression analysis etc.). Because in the datasets variables are of different types (binary, categorical, nominal, ordinal, quantitative, interval and ratio), only those data mining techniques appropriate, based on the variable type, should be used. Discussion of these aspects is provided in this section.

2.4.1 Variable Types

In real world applications information is stored using different types of variables such as binary, categorical, nominal, ordinal, quantitative, interval and ratio. Not all of these variables are suitable for a particular data mining technique. For example *k-means* clustering can take only numerical variables as its name indicates, while *k-modes* and *k-prototypes* can take a mixture of numeric and categorical variables as discussed above in section 2.3.4.1.5. Discussion of these variable types follows:

Binary: A binary variable has only possible of two values, e.g. true or false, or presence or absence. Binary variables are sometimes also known as dichotomous variables.

Categorical: A categorical variable is also known as a discrete or qualitative variable and has two or more categories. It is further divided into two variants, nominal and ordinal. These variables are typically coded as numerical values but should not usually be analyzed as though they are numeric.

Nominal: This is one of the forms of a categorical variable where an object is assigned to an unordered category. This type of variable may be coded in numeric form but these numerical values have no mathematical interpretation and are just labeling to denote categories. For example, gender may be coded as 1 and 2, the colours black, red and white may be also coded as 1, 2 and 3.

Ordinal: The ordinal variable is also a type of categorical variable in which there is strict monotonic order. For example, human height can be (small, medium and high) which can be coded into numbers small = 1, medium = 2, high = 3.

Quantitative: A quantitative variable has any numerical continuous value (positive or negative) within a finite interval (e.g. blood pressure, age, weight, height and temperature etc.). This variable has also two different types, interval and ratio.

Interval: It is a variable in which the interval between values has meaning and there is no true zero value.

Ratio: It is variable that has a true value of zero and represents the total absence of the variable being measured. For example, it makes sense to say a Kelvin temperature of 100 is twice as hot as a Kelvin temperature of 50 because it represents twice as much the thermal energy (unlike Fahrenheit temperatures of 100 and 50).

2.4.2 Data Visualization

Data visualization is an extremely useful way of exploring and understanding data using human visual skills rather than computational analysis. It is not only used with data mining methods for knowledge discovery and analysis, but also is important for selecting variables to appropriate data mining technique. The main benefit of this approach is that the human expert is directly and visually involved in the data analysis. For example, to explore the results obtained by clustering algorithms, visualization can be used to reorder the data points of a similarity matrix for visualizing into very low dimensions showing the effect of the clusters, which is very useful for improving clustering results by merging or splitting clusters. In the visualization, it may be very easy to conceptualise the data into two or three dimensions. Many advanced data visualization approaches are available to extend this scope, but generally do not support more than five dimensions [16]. Here the question arises, how we can find the relationships between hundreds of variables in the large datasets? For this reason there are a number of dimension reduction methods (e.g. principal components and factor analysis) available to reduce the number of components for visualizing. Although dimensions reductions are useful, [77] notes that it is also important to find which parameters and groups of parameters are most strongly affecting any split. Almost any visualization technique for multidimensional data can also be used for cluster visualization. A number of visualization methods have been described in [78] and developed during the last decades for visualizing complex and large datasets. A recent study [79] proposed and described an approach of integrating multiple visualization methods for exploring data. By using a visualization approach, data within a cluster can be summarized into a series of graphics, which can be very useful and informative to understand the cluster profiles in a better way. Visualization techniques by Keim [80] are classified using three criteria which are the technique itself, interaction and distortion. In [75] these are described in two categories: nonlinear (Sammon's mapping, multidimensional scaling and self-organizing maps) and linear (class preserving, parallel coordinates and tree maps). According to [81] information visualization and visual data mining can be helpful for dealing with a massive amount of information.

Information visualization classification and visual data mining approaches are described based on the data type to be visualized, and the visualization technique to be applied. The purpose of these techniques is to reduce the number of components by mapping high multidimensional (i.e. hundreds of dimensions) data to low dimensional (two or three dimensions) visually. It is easier to summarize the data using only a few dimensions.

2.4.3 Data Cleansing

Data cleansing is important to carry out before the analysis phase. It is the process of improving the quality of data in the data source by removing or supplying missing, incorrect, or improper values. In real time and large datasets, the issue often occurs that values are missing in number of variables or a number of records. Data quality is a serious issue in real data as compared to simulated datasets, due to complex structure, larger number of observations and variables, which all have the possibility for missing or incorrect values. In many situations, there are different ways to handle the data for quality improvement, such as the Naive Bayesian [82] to find the missing values in the data sets. It is necessary to take any data cleansing activity into account to assess the effectiveness and performance of any data mining technique in real world datasets. The outlier or incorrect values are the data entry values that are at more distance from the other data values. The easiest way to identify and remove these values by scatter or bar plot and using interquartile range.

2.4.4 Variable Selection and Scaling

Often databases have different types and formats including millions of observations and many hundreds of variables for any particular large dataset. There are numerous different variables in these databases, which may include mixed types of variables (qualitative and quantitative) as described above. Even though all the variables in the dataset may have importance, it is necessary to select the right variables with respect to the intended analysis. Before selecting the variables we need to identify the outliers and scale the variables into appropriate form to find better underlying structures for the dataset. It is important to choose relevant variables for this purpose. Therefore, a number of studies have been carried out for variables selection before applying data mining technique. For example, according to [83] the clustering algorithms may not succeed completely to identify clear structure in datasets due to wrong selection of variables. It is suggested [83, 84] that different types of mechanisms (e.g. forward, backward, hierarchical and stepwise) need to be used to select the important variables for clustering. In [85, 86] approaches are described based on Bayesian methods and ranking for empirical correlation between the variables to find better structure. Generally, in cluster analysis the algorithms use some distance measure and only apply to quantitative continuous variables [87]. The different selected variables in the dataset to be analysed may have a different range of values and this may invalidate some distance measures. Therefore, to overcome the skewed relative distance effect it is necessary to scale or normalise the data, otherwise highest values of variables will tend to dominate [88]. Both the variable selections and scaling for clustering algorithms can affect the clusters that formed by the clustering algorithms.

There are different of types of scaling available such as mean, standard deviation, z-score, values [0, 1] and log ratio. The most commonly used is the z-score in [13] which has the mean value 0 and standard deviation 1 and computed as follows:

$$Z = \frac{x_i - \bar{x}}{s} \tag{2.21}$$

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{N} \tag{2.22}$$

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{N - 1}} \tag{2.23}$$

where, x is variable, $\bar{x}$ = mean and s = standard deviation and $i = 1,2, \ldots, N$ number of observations.

There are still problems even if the data are scaled. For example, by scaling the data to 0 mean and standard deviation to 1 small size clusters may be eliminated.

2.4.5 Measures of Similarity and Dissimilarity

In data mining measuring similarity and dissimilarity between objects is a special concern for clustering. For example, in the *k-means* algorithm a number of common similarity and dissimilarity measures are used to find how similar (close) and how dissimilar (different) objects are from each other, typically based on some distance (e.g. Euclidian, City Block, Mahalanobis and Minkowski etc.) metric [88, 89]. The choosing of these measures directly affects the outcomes of clustering algorithms. In clustering, different types of similarity and dissimilarity measures may need to be used to characterise objects for quantitative, categorical and binary variables. This choice is heuristic in any clustering algorithm because the performance of clustering algorithm is based on the distance metric in an unspecified way [90-92].

Applying one distance metric for clustering objects may result in some being close to each other but further away in another distance metric [93]. These measures are used in algorithms to assess two different factors: how close objects are from each other in a cluster, and how far a cluster is from other clusters. Choosing a suitable distance metric is not a systematic process, despite much research having been undertaken on comparing distance metrics. Ideally the distance metric for clusters can also provide meaningful proximity indication between the clusters [94]. Different distance metrics may lead to different proximity [95], but the Euclidean distance metric is regarded as natural and conventionally used for many real datasets [91, 96]. There are also several types of similarity and dissimilarity measure used for different types of variables and clustering algorithms. Eventually, in [97] the measures are divided into three different types of coefficients for distance, association and correlation. For example, for binary variables common measures are Jaccard, Dice, Pearson, Yule, Hamann and simple matching, as explained in [75]. In this work, the *k-means* clustering algorithm has been adopted, which mostly uses the numerical values variables from the data. For this particular algorithm Euclidean, Manhattan, Maximum, Average, Minkowski and Mahalanobis distance measures are commonly used [98]. These measures are appropriate for numerical continuous data and their computations are described in [75, 99].

Using these measures, the distance function of any two or more objects in the space of $x, y \in D$ is denoted by $d_{(x,y)}$distance with the following properties [98]:

$$\begin{cases} d_{(x,y)} \geq 0 & \\ d_{(x,y)} = 0 & \text{(if and only if } x = y\text{)} \\ d_{(x,y)} = d_{(y,x)} & \text{(Symmetric)} \\ d_{(x,y)} \leq d_{(x,y)} + d_{(y,z)} & \text{(Triangle inequality)} \end{cases} \tag{2.24}$$

The Manhattan and Euclidean distance metrics are commonly used for partitional clustering algorithms for a multi-dimensional data space. These metrics give good results very well with many kinds of datasets, while drawbacks may occur with variables which include extreme values [93]. The general purpose of Euclidean distance in clustering is to make the distance of an object within a cluster from its centroid, smaller than the distance from different clusters. The Manhattan distance, also known as city block distance, approximates this by forming the sum of the distances for all attributes [75]. Minkowski distance, which allows a wider range of choice for the exponent than 2 as in Euclidean distance, performs better for datasets with compact or isolated clusters [101]. Euclidean, Manhattan and Maximum distances are special cases of Minkowski distance, where r in the formula provides the order of the Minkowski function. For $r = 1, 2$ and ∞ we have the Manhattan, Euclidean and Maximum distances respectively [75]. The behaviour of Mahalanobis distance is used to allow for elliptical contours in the dataset variables and gives the clusters an elliptical shape. It therefore distorts the original variables space in cluster analysis [102] and includes only the numeric variables [99, 101, 103, 104]. This distance metric computes the correlation by taking the inverse of the variance-covariance matrix of the dataset [104]. The Maximum distance is also known as the “sup” distance, which is defined as the maximum value of the distances between the variables. In [105], it is noted that when the values of variables are not available or very small, the Euclidean distance is very small. In that situation average distance is more useful, which is a modified form of Euclidean distance. The most commonly distance metrics formulas used for cluster analysis are in [75, 99], which are listed below:

Metrics	**Formulas**
Euclidean distance	$d(x,y) = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2}$
Manhattan distance	$d(x,y) = \sum_{i=1}^{N} \lvert x_i - y_i \rvert$
Mahalanobis distance	$d_{(x,y)} = \sqrt{(x-y)^T \Sigma^{-1}(x-y)}$
Minkowski distance	$d_{(x,y)} = (\sum_{i=1}^{n} \lvert x_i - y_i \rvert^r)^{1/r}$,$r \geq 1$
Maximum distance	$d(x,y) = \max_{1 \leq i \leq N} \lvert x_i - y_i \rvert$
Average distance	$d_{(x,y)} = \left(\frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2\right)^{\frac{1}{2}}$

Table 2.1: Computing similarity and dissimilarity metrics.

where Σ denotes the covariance matrix and x and y are numeric values of the dataset in N dimension space.

2.5 Application of Data Mining in Health

Today medical records are widely stored and retrieved electronically and provide the source of information for many purposes which include systematic information about the patients, their medical history such as illnesses, allergies, detail consultation, medications prescribed, procedures carried out, tests ordered and their results [106]. There are many different types and formats of medical records for diverse purposes such as for maintaining the patient history and evaluating the quality of care provided by the health organization. Medical records are the property of the hospital or the practice attended by the patient and usually there is a confidentiality agreement in place between patients and doctors. Some variants in digital representations of stored medical records are electronic health records (EHR) maintained by the health service provider, and personal health records (PHR) maintained by the patient or their agent.

However, to improve and maintain our health in better ways, research on health data makes use of different statistical and data mining techniques to find patterns in the cohort or population scale. As storage density increases and cost decreases exponentially, more and more transactional data is being collected in health. For example there are currently 5.7 million hospital admissions, 210 million doctor's visits and a similar number of prescribed medicines dispensed that are captured electronically annually in Australia [107]. Unfortunately the data is not being fully utilized to provide useful knowledge as a basis for future medical practice. Today research in health administration, adverse drug reactions and drug safety, population health, epidemiology and disease diagnostics is being carried out by using different types of data mining techniques, in different manners. In this section the application of some of these will be described. The intention is not to provide a comprehensive review of this broad topic but to give some representative examples so that the types of underlying problem amenable to this approach are clarified, and the value in using data mining can be appreciated.

2.5.1 Drug Usage

To determine the associations between particular drug usage and impact of its use for specified diseases without prior knowledge, and to identify the factors that increase the risk of some adverse drug reactions, researchers have used classification and association analysis [107]. For example, a recent study has used data mining software application from 42 pharmacies of 2449 patient records for finding of the study that led to improvement of asthma [108].

2.5.2 Epidemiology

Preterm births are continuously increasing and create complex health care problems [109]. Data mining methods used included logistic regression, linear regression and neural networks on 19970 observations of varied ethnic background of pregnant women, with 1622 variables. Preterm birth predictors were evaluated and compared using traditional statistical methods. Outcomes were investigated for all these methods for all datasets and it was found that these did not significantly differ. For improving birth outcomes prediction, [110] applied four different types of data mining tools: inductive methods, neural networks, classification and regression trees and logistic regression. They compared the results from these multiple analyses with a receiver operating characteristics (ROC) technique that produces a graph in which the area under curve (AUC = 1.0 perfect prediction) is an easy way to interpret outcomes visually. In analysis of multi-marker studies [111], logistic regression and classification and regression tree (CART) methods were used that aimed to assess and compare the value of different laboratory parameters, to predict the need for intensive care treatment in unselected emergency room patients.

2.5.3 Hospital Utilization

A study by [112] to investigate and compare hospital utilization among Australian born and 8 different refugee source countries, showed people born in refugee countries have lower or similar rates of hospitalization compared with Australian born. A random sample of 100,000 admissions of Australian born people for each year was compared with total number of admissions from 8 refugee born countries in the same year; a total of 49,835 hospital admissions from these countries were recorded between 1998 to 2004. The dataset used for this study was over six financial years from the statewide hospital discharge of all patients admitted to public and private hospitals in Victoria.

Similar research by [113] to determine the effect of hospital utilization in Denmark between ethnic background and Danish born people by using multiple regression analysis shows that in some diagnoses patients born in Denmark stay longer in hospitals than immigrants. This study included 5310 persons discharged as inpatients, outpatients or emergency room patients born outside the Nordic countries and is compared with 10,000 random sample of all patients born in Denmark in 1997 from Bispebjerg Hospital. Another study for emergency hospital services utilization in Spain by [114] among immigrant and Spanish born people was carried out to examine how emergency hospital services (EHS) were utilized by people. The data set included patients between 15 to 64 years old and covered 96,916 visits during the years 2004 to 2005 in public hospitals. They used descriptive statistical analysis and logistic regression techniques that showed people born outside Spain use EHS differently and more frequently than native born people in Spain.

In Portugal [115] another study investigated health care utilization by immigrants. The data sample in this research included 1513 migrants. The study showed that age, length of stay, legal status and economic situation are interrelated in health services usage, by using logistic regression and odds ratio at 95% confidence intervals. Another Australian study by [116] examined people with mental health problems who frequently attend emergency department (ED) in tertiary referral metropolitan hospitals. The data was collected for 12 consecutive months between 2002 to 2003 year for 45,671 patients from which 869 psychiatric patients were identified, and 1076 presentations of these patients. They found a significant difference for age and diagnosis and younger people appeared more prominently in the frequent presenters group, and also experienced more anxiety mood diagnosis than other group. A study in Spain was reported by [117] for 11,578 admissions to psychiatric emergency services in a tertiary hospital. Data collected includes socio demographic and clinical information that was used to determine the relationship between homeless persons and to identify the difference between homeless and non-homeless patients. The method used was multivariate logistic regression analysis to compute odds ratios for the factor associated with homelessness and decision to hospitalization. The study found that patients associated to homelessness were male which had more psychosis and drug abuse disorders, a higher risk of being danger and frequent assistance of hospitalization than non-homeless patients. Data mining methods both supervised and unsupervised learning are discussed in this chapter using data mining tools introduced and implemented by various software packages. The most popular data mining tools are SAS, R, Matlab, Stata and SPSS which are available and commonly used to research, explore and uncover important patterns and data properties in different domain especially hospitals, pharmaceuticals and other health datasets.

2.6 Summary

This chapter gave an overview for supervised and unsupervised different data mining techniques that are used to discover knowledge from many different data applications to find interesting patterns. A brief explanation of all of the techniques was discussed including different clustering algorithms particularly partitioning algorithms like *k-means*, *PAM* etc. The *k-means* clustering algorithm is the main focus for the remainder of this thesis, with the development of a new approach for clustering results evaluation to find the best estimated number of clusters based on the results obtained from *k-means*. In addition to this, some fundamental steps for data preparation were presented which are not only applicable for cluster analysis but also used for any data mining technique before its application. Furthermore, different types of variables, data visualization, data cleansing and similarity/dissimilarity measures were discussed in detail. Finally, some examples have been provided using data mining techniques especially in health related fields.

Chapter 3

Clustering Evaluation

A brief description of how different clustering algorithms work has been provided in the previous chapter. However, a challenge in using the clustering algorithms is to find the optimal number of clusters giving well-structured or quality clusters without prior knowledge of the data. In this chapter we will provide an overview of clustering results using a range of different existing validation indexes which are used for selecting the number of clusters and their quality.

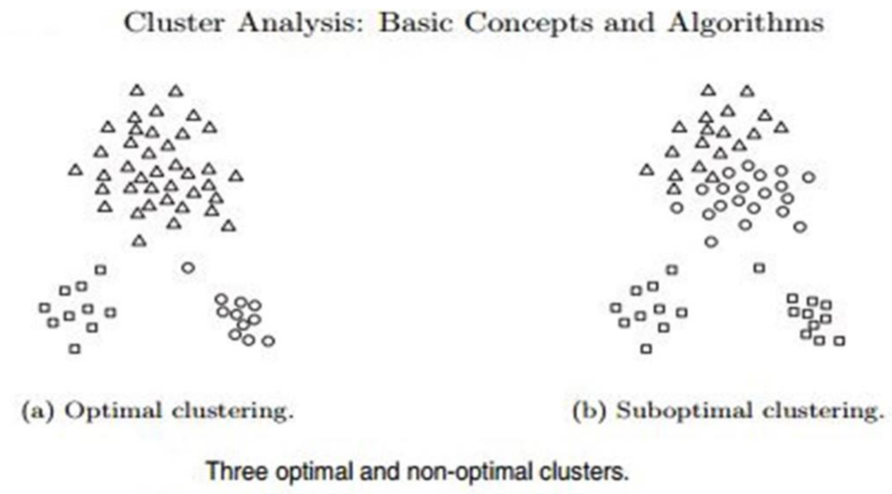
3.1 Introduction

The purpose of clustering algorithms is to find similar objects within a dataset according to their characteristics: this is usually accomplished by using some of the distance measures (e.g. Euclidean, Maximum, Manhattan) discussed in the previous chapter. Using these distance measures allows the dataset to be partitioned into clusters for determining the similarity and dissimilarity between different objects. These show the characteristics of objects in the same cluster are more similar to the objects in other clusters. In clustering, the *k-means* algorithm is a popular and simple technique that groups a dataset into a given k number of clusters, in such a way that objects within a cluster (group) minimize the normalised least squares distance between each other, over the entire dataset [42]. As the characteristics of objects are similar intra (within) cluster they are termed *homogenous* objects, and dissimilarity of characteristics inter (between) clusters is termed *heterogeneous* objects [20]. **Intra-cluster:** distance is a measure of the sum of distances between cluster objects to the centroid of their cluster. Minimizing the sum of squared intra-cluster distances leads to homogeneity and tightness of the cluster. **Inter-cluster:** distance is a measure of distance between cluster centroids. Maximum inter-cluster distance indicates good cluster separation [118].

Clustering algorithms are classified into two major types: *partitional* and *hierarchical* [48] as discussed in detail in Chapter 2. Although, the aim of clustering algorithms in both types is to group the data into different clusters they can provide different solutions for determining the quality and number of clusters due to their implementation differences. Clustering has been utilized successfully in various types of problems across many fields. For example, in medicine, clustering broad population information on occurrence and progression of diseases can lead to very useful insights on the determinants of health.

3.2 Issues with *K-Means*

The *k-means* algorithm partitions N objects into k clusters by randomly selecting k initial candidate cluster centroids (where *k* is the number of desired clusters specified by the user). Even though *k-means* is an efficient and commonly used algorithm to determine the optimal number of clusters (selection of finding the best or estimated number of a well partitioned set of clusters). However, the main issue in its application is that no consistent solution is available for the optimal number of clusters especially when considering the complexity of the real datasets. Often, in a partitioning algorithm such as *k-means* another major issue is the choice of the initial centroids. This choice can affect the results significantly [88], and normally it is chosen randomly. There are heuristic based approaches for selecting the centroid found in [119, 120] to fix these issues but [121] found these can slow the process. The figure below represents the effect of randomly chosen initial centroids. It shows three clusters labelled with square, circle and triangle. Figure 3.1 (a) clearly shows three optimal clusters fully separated. Figure 3.1 (b) shows the difference with incorrectly selected initial.



(a) Optimal clustering.

(b) Suboptimal clustering.

Three optimal and non-optimal clusters.

Figure 3.1: Three clusters with initial centroids randomly selected [122].

Another issue is that parameters and variables chosen by the algorithm or by users affect the performance of the algorithm differently and result in different optimal values. Due to these effects, a critical question arises: what value of k should be used to construct well defined homogenous and separated clusters with the lack of a priori knowledge (as it is an unsupervised technique)? For this reason different types of validation approaches to evaluate resultant clusters from an algorithm have been presented by authors to determine which k is preferable for cluster quality.

3.3 Cluster Quality

In the circumstances discussed above, determining the correct number and quality of clusters is essential. In order to gain these [123, 124] have explained the criteria for quality of cluster. These criteria are divided into three categories clustering - compactness, connectedness and spatial separation, which are defined as follows:

Compactness: the process of keeping the intra cluster distance to a minimum. The algorithms which perform well in this category are *k-means*, average linkage agglomerative clustering and self-organizing maps.

Connectedness: the process of clustering neighbouring elements which have properties suggesting they should share the same cluster e.g. density and single linkage agglomerative clustering.

Spatial separation: the process that enables maximizing inter cluster separation i.e. clusters should have wide spaces between each other. Spatial separation algorithms consist of three different types - complete linkage, single linkage and comparison of centroids. In order to judge algorithm performance for these 3 categories clustering validation indexes are used for evaluating the clustering results. These indexes are explained in details in the next section.

3.4 Evaluating Clusters

The common approach to evaluate clustering results is known as cluster validity or validation. The purpose of validation is to determine the optimal clustering giving a sensible structure of clusters for better understating data [125, 126]. For this purpose, there are three different approaches, *external*, *relative* and *internal* validations that can be used to evaluate the results of clustering algorithm [125]. The characteristics of these three validation indexes are described as follows:

External validation: the process of evaluating clustering results based on the pre-specified (prior information) structure of the dataset.

Relative validation: the process of evaluating clustering structure by comparing between multiple clustering schemes.

Internal validation: the process of evaluating result of clustering algorithm without prior information and based on quantities that involve vectors of the dataset themselves [124, 125, 127].

A number of popular and commonly used cluster validation indexes including *Dunn, Duda* and *Hart, Calinsiki* and *Harabasz, Davies-Bouldin, Silhouette, SD, Gap* statistics and *Cubic Clustering Criterion (CCC). Duda* and *Hart* [128] described two main basic issues in cluster validation:

- How many clusters are present in the data?
- How good is the clustering itself ?

An examination of 30 clustering validation indexes for determining the optimal number of clusters was carried out by Milligan and Cooper [129] using simulated data with well separated clusters. They found there is no perfect approach to determine the optimal number of clusters. However, the *Calinsiki* & *Harabasz* and *Duda* & *Hart* indexes work well for the data used by them to make comparisons with other indexes. One of the popular indexes is the *Cubic Clustering Criteria* suggested by Sarle [130]. This has been shown to be an appropriate choice for many circumstances, but may not perform well for irregular and elongated clusters. A validity measure for colour image segmentation proposed by Siddheswar and Rose [131], is based on the ratio of intra and inter cluster distances for determining a minimum value of the index. A comparative study [126, 127] to determine the optimal number of clusters found that the *Dunn* and *Davies-Bouldin* worked better with well separated clusters in a simulated dataset. According to Chou [132] the *Dunn* index does not perform well with a mixture of different shapes and density clusters. In another study [133] compare the 15 indexes to determine the number of clusters and determine that the *Davies-Bouldin* and *Calinsiki* and *Harabasz* indexes gave the best result for simulated binary data. A study presented by [134] focuses on comparing 11 indexes using simulated data involving a mixture of noise, low, high density for five different scenarios.

Generally, the optimizing criteria in *k-means* clustering minimises intra cluster distance, e.g. the sum of squared distances between elements in the cluster from their centroid and similarly maximizes inter cluster distance using squared Euclidean distance. A key concept with using some of the indexes e.g. minimizing sums of squares is to plot a graph with calculated values of the index against the number of clusters and analyse this plotted graph to see where a change occurs (e.g. a bend in the plot). However, a problem using these graphs is that it is often difficult to spot the minimum value for a bend. The case of no clear bend points may indicate absences of cluster structure or multiple optimal values.

These indexes find maximum or minimum values, or the values relative to some critical values such *Gap* and *Duda* and *Hart* uses critical values for determining the optimal number of clusters. Deficiencies with some of these indexes is that they are computationally expensive or are unable to discover the optimal number of homogenous and well separated clusters in some specific and large datasets [135]. In this study, we will limit the discussion to internal validation only. The next section will explain in detail internal validation indexes.

3.5 Internal Validations

The process of evaluating the quality of clustering results from an algorithm is called clustering validation. There are many internal validation indexes [128, 129, 136-138] to evaluate the quality of clusters. A purpose of these indexes is to find the clustering algorithms which result in compact and well separated clusters.

The internal validations involve of taking the clustering and the underlying dataset as an input and using intrinsic information for assigning the quality of clustering as in [123]. According to [133] internal validation is divided into three groups: The first group is the sum of squares within (SSW) and between (SSB) the clusters which are used to compute the intra and inter clusters dispersion. The second group uses the overall scatter matrix of data points and the sum of scatter matrices in each cluster.

The third group is independent of the first two. The following is a brief explanation and formulation of eight well known and widely used internal validation indexes.

3.5.1 Dunn ($\boldsymbol{Dunn_{index}}$) Index

Dunn index($Dunn_{index}$) is used to define separation and compactness of clusters. It is computed as the ratio between minimum intra cluster distance to maximum inter cluster distance [137]. It combines dissimilarity between clusters and their diameters to estimate the most reliable number of clusters and it requires the definition of at least two clusters [139]. *Dunn* index implementation is quite easy with low complexity [140] but it still requires a reasonable amount of time for computation and is sensitive to a noisy and sparse dataset [124]. The *Dunn* index is defined as

$$Dunn_{index} = \frac{Dist_{min}}{Dist_{max}} \tag{3.1}$$

where $Dist_{min}$ is minimum intra cluster distance, $Dist_{max}$ is maximum inter cluster distance. The maximum value for the index $Dunn_{index}$ gives the best k value.

3.5.2 Duda and Hart ($\boldsymbol{DH_{index}}$) Index

The *Duda* and *Hart* index [128, 129] uses a hypothesis test concerning a null hypothesis for the number of specifying k cluster are homogeneous and an alternate hypothesis which consists of two clusters. The test is based on the ratio between the sum of squared errors within clusters when the data set is partitioned and the squared error when the data is not partitioned. Mathematically it will be computed as

$$(DH)_{(index)} = \frac{Je(2)}{Je(1)} \tag{3.2}$$

where $Je(1)$ = Error sum of squares (ESS) before the data is partitioned and $Je(2)$=Error sum of squares (ESS) after partitioning the dataset using a clustering algorithm e.g. (*k-means*). The value of this index is compared with its critical value.

A high value of the index shows distinct cluster structure and that at certain values k is optimal and the clusters are well separated.

3.5.3 Calinski and Harabasz (CH_{index}) Index

The *CH* index (CH_{index}) proposed in [136] measures the between cluster sum of squares and within cluster sum of squares. The CH_{index} is defined as

$$CH_{index} = \frac{BCSS_k/(k-1)}{WCSS_k/(n-k)} \tag{3.3}$$

where $BCSS_k$ = Between cluster sum of squares for k clusters, $WCSS_k$ = Within cluster sum of squares for k clusters. The maximum value of the CH_{index} gives the optimal number of clusters in the dataset.

3.5.4 Silhouette (Sil_{index}) Index

The *Silhouette* index [141] (Sil_{index}) uses average dissimilarity between points to show the structure of the data and consequently its possible clusters. The purpose of this index is to calculate average dissimilarity between points in the same cluster and different cluster to illustrate the structure of the data [78, 142]. It is formally defined as

$$Sil_{index} = \frac{b_{(i)} - a_{(i)}}{max\{a_{(i)}, b_{(i)}\}} \tag{3.4}$$

where $a_{(i)}$ is the average distance for the i^{th} object to all the objects in the same cluster while $b_{(i)}$ is the minimum average distance for the i^{th} object to all objects in different clusters. The value of this index lies $-1 \leq Sil_{index} \leq 1$. The maximum value of Sil_{index} indicates the optimal number of clusters in the dataset. The best value is 1 and the worst value is -1 while values close to 0 indicate clusters are overlapping.

3.5.5 Davies-Bouldin(DB_{index}) Index

The concepts of dispersion of a cluster and dissimilarity between clusters are used to compute the *Davies-Bouldin* index (DB_{index}) [143]. It is defined as the ratio of the sum of the within cluster dispersions to the between cluster separation. Similarly to *Dunn*'s index, the *Davies-Bouldin* index needs at least two clusters.

$$(DB)_{index} = \frac{1}{k} \sum_{i=1, i \neq j}^{k} max \left(\frac{averdis_i + averdis_j}{dis(c_i, c_j)} \right) \tag{3.5}$$

where k is the number of clusters, $averdis_i$ is the average distance of all objects in cluster i from the c_i , $averdis_j$ is the average distance of all objects in cluster j from the cluster centre c_j and $dis(c_i, c_j)$ is the distance between cluster centres c_i and c_j. A small value of DB_{index} will indicate the optimal number of clusters that are well separated and compact. According to [135] it is a favourable index due to its simple calculation.

3.5.6 SD (SD_{index}) Index

The *SD* validity index proposed in [125, 144] is based on the average scattering of clustering (intra cluster distance) and the total separation of clusters (inter cluster distance). The average scattering for clusters is defined as

$$Scatt(n_c) = \frac{1}{n_c} \sum_{i=1..n_c} \frac{||\sigma(v_i)||}{||\sigma(X)||} \tag{3.6}$$

while the total separation between clusters is defined as

$$Dis(n_c) = \frac{D_{max}}{D_{min}} \sum_{k=1...n_c} \left[\sum_{z=1...n_c} ||v_k - v_z|| \right]^{-1} \tag{3.7}$$

where $D_{max} = \max(||v_i - v_j||)$, $\forall i, j \in \{1,2,3,.........n_c\}$ is the maximum distance between cluster centres and $D_{min} = \min(||v_i - v_j||)$, $\forall i, j \in \{1,2,3,.........n_c\}$ is the minimum distance between cluster centres. Given (3.6) and (3.7):

$$\left(SD(n_c)\right)_{index} = a.Scat(n_c) + Dis(n_c) \tag{3.8}$$

where a is a weighting factor equal to $Dis(c_{max})$ and c_{max} is the maximum number of input clusters. To determine the best number of clusters that fits the dataset we need to calculate the above *SD* index value. The minimum value of this index will determine and confirm the degree of cluster separation and hence if it is the best number of clusters.

3.5.7 Gap Statistics (Gap_{index}) Index

Determining the optimal number of cluster, using the gap statistics is described in [145]. This index selects the number of clusters by relating the changes in within-cluster $logW_k$ dispersion using the null hypothesis for the uniform distribution that indicates one cluster (no obvious clusters) and an alternative of k clusters. The best number of k is the value where $logW_k$ falls farthest below a reference curve. The steps involve for computing the *Gap* statistics are as follows:

Suppose the dataset $\mathcal{D}$ is partitioned into k clusters with n_k objects in each cluster. First compute the intra cluster distance for each cluster and the sum of the pairwise distances for all the points in each of the k clusters using:

$$D_k = \sum_{x_i \in C_k} \sum_{x_j \in C_k} \left\|x_i - x_j\right\|^2 = 2n_k \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 \tag{3.9}$$

$$W_k = \sum_{k=1}^{K} \frac{1}{2n_k} D_k \tag{3.10}$$

where W_k is the pooled within cluster sum of squares. As the Gap statistic uses a reference distribution for hypothesis testing, create B reference datasets and compute the within cluster dispersion W_{kb}^*, $b = 1,2, \dots, B, k = 1,2, \dots, K$. The information is stored using the following formula for the gap statistics:

$$Gap(k) = \left(\frac{1}{B}\right)\sum\nolimits_{b=1}^{B} log(W_{kb}^{*}) - log(W_k) \tag{3.11}$$

Compute the standard deviation as follows

$$sd_k = \sqrt{\left(\frac{1}{B}\right)\sum_{b=1}^{B}(log(W_{kb}^{*}) - \bar{l})^2} \tag{3.12}$$

$$\bar{l} = \left(\frac{1}{B}\right)\sum_{b=1}^{B} log(W_{kb}^{*}). \tag{3.13}$$

The estimated number of optimal clusters can be determined through finding the smallest k such that

$$Gap(k) \geq Gap(k+1) - s_{k+1} \tag{3.14}$$

where $S_k = sd_k\sqrt{1 + {}^{1}\!/_{B}}$, and sd_k is the standard deviation, B is the number of reference datasets generated using the uniform distribution and W_{kb}^{*} is the within-dispersion matrix.

3.5.8 Cubic Clustering Criterion (CCC_{index}) Index

The *Cubic Clustering Criterion* uses extensive simulation for its development and is based on the assumption that clusters obtained from the uniform distribution are hypercubes of the same size. In most cases the assumption of strictly hypercube structure is false, but it is generally conservative unless the number of clusters is large [130]. This algorithm may be considered as testing the assumption (uniform distribution) using a null hypothesis with *CCC* as an approximate test statistic. The test statistic formula is:

$$CCC_{index} = ln\left[\frac{1 - E(R^2)}{1 - R^2}\right]\frac{\sqrt{\frac{np*}{2}}}{0.001 + E(R^2))^{1.2}} \tag{3.15}$$

$$R^2 = 1 - \frac{trace(W)}{trace(T)} \tag{3.16}$$

T = $X'X$ is the total-sample sum-of-squares and crossproducts ($SSCP$) matrix ($p\ x\ p$)

W=$T - B$ is the within-cluster $SSCP$ matrix ($p\ x\ p$)

$B\ =\ \bar{X}'Z'Z\bar{X}$ is between-cluster $SSCP$ matrix ($p\ x\ p$)

$\bar{X} =\ (Z'Z)^{-1}Z'X$

where Z is a cluster indicator matrix ($n\ x\ q$) with element $z_{ik} = 1$ if the i^{th} observation belongs to the k^{th} cluster, 0 otherwise.

$$E(R)^2 = 1 - \left[\frac{\sum_{j=1}^{p*} \frac{1}{n + u_j} + \sum_{j=p*+1}^{p} \frac{u_j^2}{n + u_j}}{\sum_{j=i}^{p} u_j^2}\right] * \left[\frac{(n-q)^2}{n}\right] * \left[1 + \frac{4}{n}\right] \qquad (3.17)$$

Where, $u_{j = \frac{s_j}{c}}$, s_j= square root of the j^{th} eigenvalue of $\frac{T}{(n-1)}$, $c = \left(\frac{v*}{q}\right)^{\frac{1}{p*}}$, $v^* = \prod_{j=1}^{p^*} s_j$ .p is chosen to be the largest integer less than q such that u_p^* is >=1.

The maximum value (CCC > 2) across the hierarchy level is used to indicate the optimal number of clusters while values between 0 and 2 may indicate possible clusters. A very negative CCC will indicate the risk of outliers is low and this may indicate a small number of clusters are optimal.

3.6 Summary

Determining the best number of clusters is an important feature of cluster analysis. The aim of this chapter was to provide reasons for use of clustering validation measures and a brief discussion of such validation indexes was provided. Even though these existing indexes are applicable to any clustering algorithm, the performance of these indexes may depend on the structure of the clustering algorithm and dataset. Wrongly chosen initial centroids can produce poor results and there may be a need to run a computer algorithm multiple times to determine the optimal number of clusters. Therefore, to overcome this problem there is a need to develop an improved index or approach. In the next chapter, the development and explanation of the new approach will be presented.

Chapter 4

New Approach

4.1 Introduction

To estimate the best number of clusters is a challenging and major issue in cluster analysis. Often to achieve this requires a large number of steps particularly when iterative or exhaustive search is applied. For this reason a number of validation indexes (approaches) have been developed based on the use of heuristics, such as to minimize the distance from centroid of a cluster to each element in the cluster, or to maximize the distance between clusters (intra and inter cluster distances), or to use an average scatter of clusters and total separation between clusters. These approaches have been reviewed in more detail in Chapter 3. They are to some extend limited in effectiveness by the assumptions made about what constitutes best clustering as there is no or very little prior information available in cases of large and complex datasets.

According to [146] attempting to find the best clustering algorithm is fruitless and cluster selection should be seen as a subjective part of the process of exploratory data analysis. Even though many clustering algorithms and validation indexes are available but none of them can be demonstrated to be always the best. Also, when using different initial k values, the outcome of these existing indexes (*Dunn, Duda* and *Hart, Calinski and Harabasz, Silhouette* and *Davies-Bouldin, SD, Gap Statistics Cubic Clustering Criterion*) may result in determining different best k values. This may change the profile (structure) of clusters and confound the existing approaches to estimate the best number of clusters. Therefore, it is necessary to reapply the existing approaches many times with independent runs of the clustering algorithm and clustering evaluation approaches, to find a reasonable outcome. In conflicting circumstances, the elements between different clusters may form other clusters due to similarity among them, or remain belonging to one of the clusters which expands and contracts as k increases or decreases. This makes it difficult to choose the best k clustering structure.

Another issue in clustering particularly for the *k-means* clustering algorithm, is choosing the initial centroid positions which are a sensitive aspect for the computation and may make it difficult to find the best k. Due to this effect different elements may belong to different clusters when running the algorithm with the same or different k. When running the algorithm repeatedly for the same k, initial centroids may be chosen differently and this can lead to *k-means* clusters being different. According to [88] the *k-means* algorithm is sensitive to the selection of the initial centroids and so the algorithm may not converge if the initial centroid is not chosen properly. For addressing this issue, a number of studies have been described in [120, 147, 148]. Moreover, existing approaches are less able to provide information about the appropriateness of k, based on whether the clusters formed are fully or partially separated. In addition, when having partially separated clusters there may also be isolated clusters which lead to different amounts of overlap among the clusters.

To overcome these issues a new approach is proposed and developed that allows exploration of the clustering structure based on the elements in the set of clusters, as inter cluster (elements between clusters) mapping of the common elements while increasing and decreasing k. This approach is an extension and improvement of the original MMM (based on only the adjacent mapping of elements between clusters) concept described in [149, 150] to derive a criterion (signature function) based on changes in cluster membership over a range of successive k values. We define forward and backward mapping around k as k to $k + r$ and $k + r$ to k to find the inter cluster mapping of common elements in a sequence. In the work presented here it is assumed that minimum k is 2 ($k_{min} = 2$), and maximum k is 16 ($k_{max} \leq 16$) and $1 \leq r \leq 14$ (limiting k and therefore r to a practical finite range). The new approach compares cluster properties at adjacent ($r = 1$) and more distant ($1 < r \leq 14$) cluster structure for a range of several r values regarded by the user as a reasonable number. This new approach provides a more efficient solution, independent of cluster characteristics such as choosing of the initial centroid, variances between and within clusters, and more consistent in its behaviour across a wider range of potential k values.

4.2 Overview of the New Approach

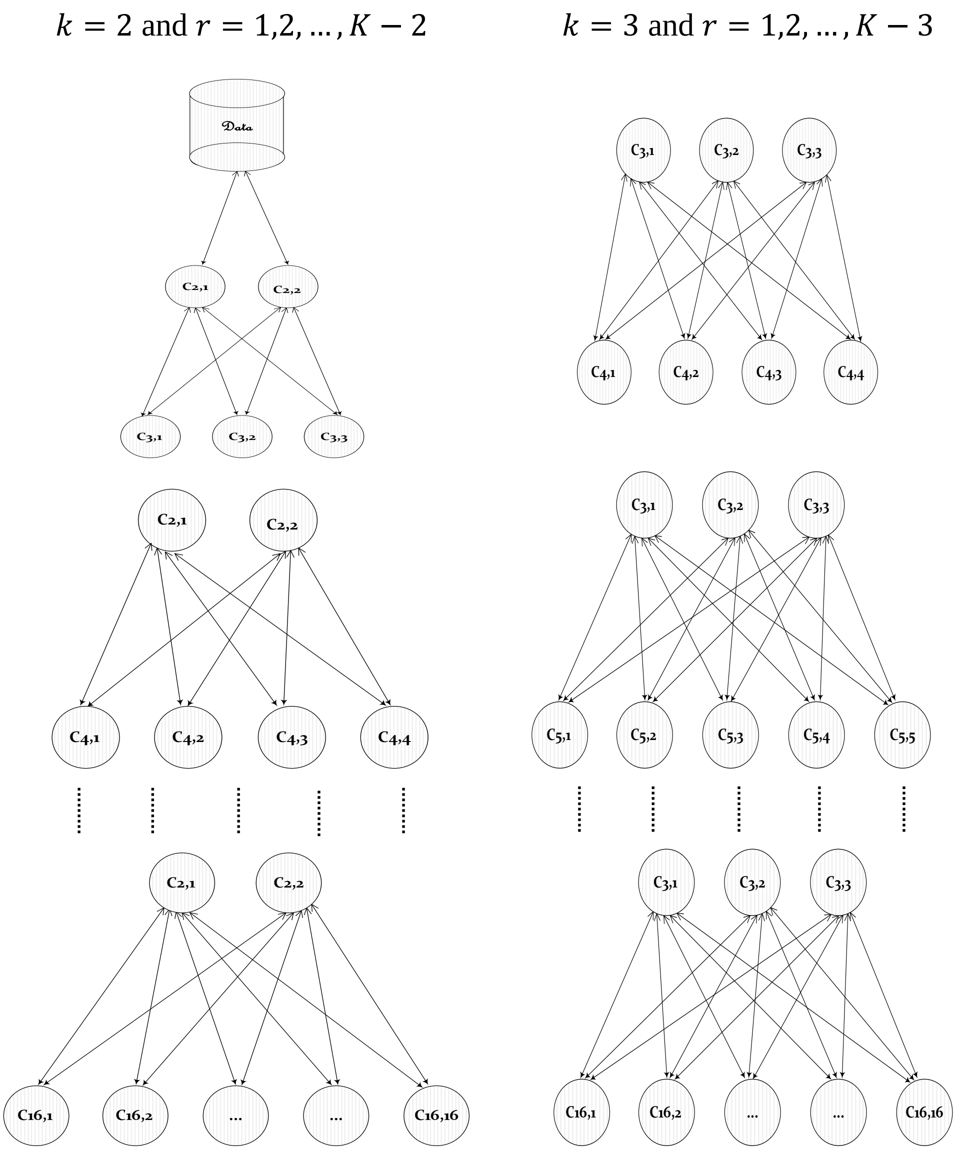
The underlying idea behind the MMM approach is the analysis of inter cluster mapping of common elements (objects) between corresponding clusters that provides an indication of an extent to which elements are common between such clusters, considered either forwards from k to $k + r$ and backwards from $k + r$ to k clusters also described in [151]. This forward and backward mapping of common elements will form an indicator of the proportion of common elements between the clusters. Further, this mapping information will be used to determine the mutual similarity of combined mapped proportion clusters using the inner product of the matrices of these forward and backward proportions of elements.

To achieve this approach a detailed explanation and description of its computational development will be provided in the next sections. At first, the forward and backward membership mapping of common elements will be constructed and their proportion will be expressed around k groups in the form of rectangular matrices (k to $k + r$

and $k + r$ to k) from different $k + r$ mapping distances around k. The rows of these matrices indicate vectors of inter cluster mapped common elements which transform to rows sum scale proportion of common elements between clusters. It is essential to know that clusters will split in forward mapping of common elements between clusters while these will be collapsing in the backward mapping. Second, the combined similarity proportion will be computed, for which purpose the use of a matrix inner product is very simple and a natural form of a similarity measure. Recently, inner product measure has been used in different areas of research such as social networks [152], dimension reduction [153, 154], speak recognition [155], self-organized similarity in fuzzy clustering [156] and document mapping [157]. In clustering, a study proposed in [158] found efficiently the maximum best match using the inner product metric. Cheng [159] has presented a divide-and-merge approach efficiently and effectively using the inner product to compute similarity for optimal clustering. Auvolat [160] has described and developed a maximum inner product search approach based on *k-means* variant clustering such as spherical *k-means*. In another approach a criteria for maximizing the trace is suggested [161] as the product of between groups dispersion matrix and inverse of within groups dispersion matrix.

Thus, we use this similarity measure for the inter cluster proportion of the common elements from forward and backward matrices to obtain $k \times k$ combined mapped proportion matrices for different $k + r$ distances (such as $k \times (k + r)$ and $(k + r) \times k$). These combined mapped proportion matrices provide the combined similarity at each entry of the diagonal, while off diagonal entries are the dissimilarity or overlap proportions of the clusters.

We have explained above the construction of combined mapped proportion matrices. We define combined mapped elements by multiplying the combined mapped proportion matrices for each $k+r$ distance by the sizes of k clusters, converting them into $[\boldsymbol{k}]$ matrices. The traces (sum of diagonal elements) of these combined mapped elements matrices indicate the similarity of clusters while off diagonal entries indicate elements overlap (dissimilarity) of clusters. By using these matrices, we can compute average similarity (average of traces) and average dissimilarity or overlap (off diagonal sum average) at each k for different $k+r$ distances when k is fixed and r is sequence moving from 1 to 14. Finally, we compute the dispersion coefficient of variation (CV) from traces of combined mapped matrices. It is a useful measure for determining the degree of dispersion at each k for different $k+r$ mapped distances. The Figure 4.1 shows the forward and backward inter cluster elements mapping when $k=2$ and $k=3$ with different $k+r$, where $r=1,2,\ldots,14$, with the bold and light arrowheads, in the figure, indicating the forward and backward directions respectively.



$k = 2$ and $r = 1,2, \dots, K - 2$

$k = 3$ and $r = 1,2, \dots, K - 3$

Figure 4.1: Forward and backward mapping of elements when $k = 2$ and $k = 3$ for different $k + r$ distant and $r = 1,2, \dots, K - k$.

4.3 Development of the Approach

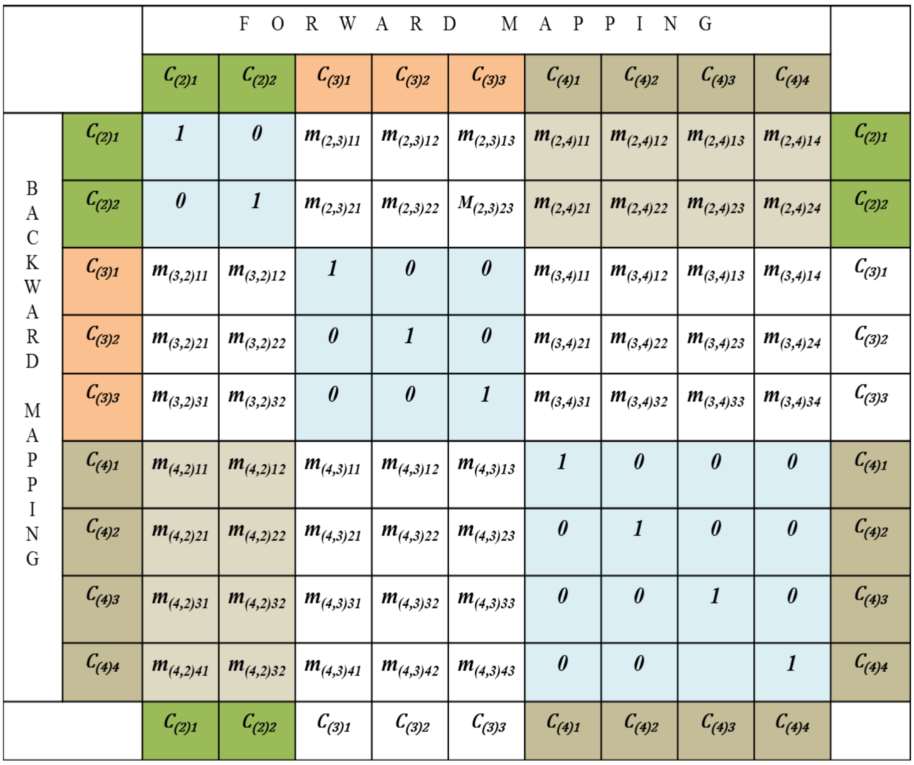
The new approach determines the best k based on the inter cluster forward/backward mapping of common elements and inner products of matrices for finding similarity, overlap and coefficient of variation. This approach provides a more systematic process for evaluating the results of clustering algorithms, which would be a valuable criterion to determine the correct number of clusters by optimizing similarity of elements between clusters. In the next sections it will be explained how to compute and develop the new approach using the features and quantities inherited from the clusters and dataset to understand how closely elements are mapped and how much is separation between the clusters.

4.3.1 Forward and Backward Elements Mapping

To develop a practical implementation of this approach we need to consider different resultant k number of clusters from the *k-means* algorithm, over a range of successive k values always starting with minimum number of clusters $k_{min} = 2\ (k_2)$ to maximum $k_{max} = K,$ where $K = N - 1$ and $N=$ total number of elements (objects or observations) in the dataset $\mathscr{D}$, and here we will fix $k_{max} = K = 16$ to allow better understanding of the clustering structure with this manageable number of clusters. Following are the notations and details used to define and describe the approach.

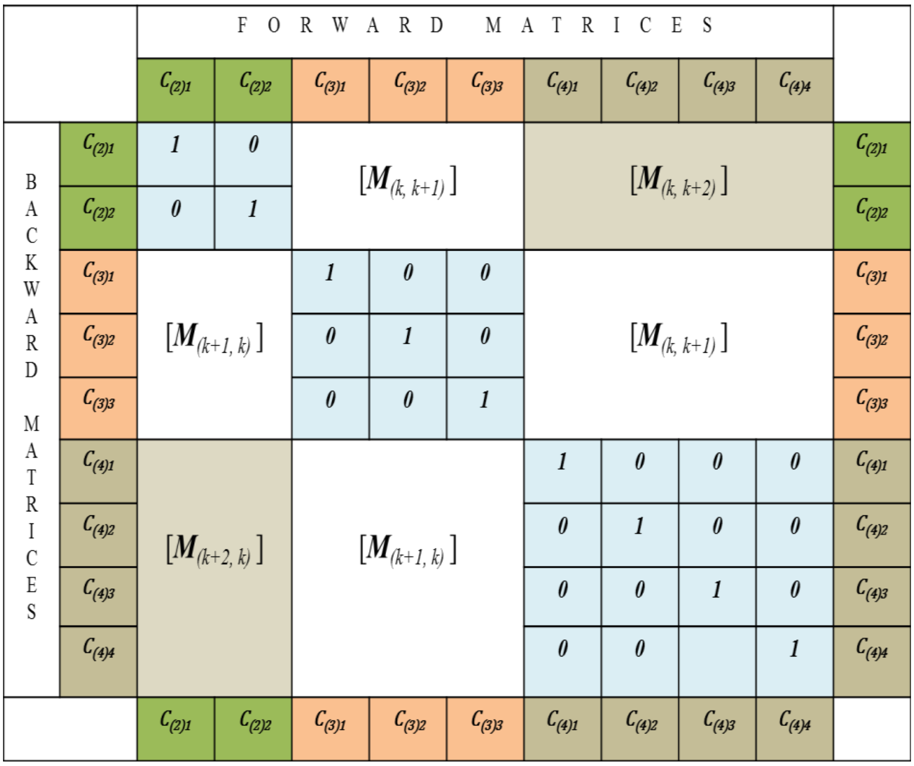
Assume we have dataset $\mathscr{D}$ containing N number of elements (observations). Each observation may have number of variables that may be used to determine information about the relationship between them. Using a *k-means* clustering algorithm on $\mathscr{D}$ with a certain set of standard parameters to control the behaviour of the *k-means* algorithm, such as choosing variables, number of variables, number of clusters (centres), number of maximum iterations, number of initial random starting sets of seeds and algorithm, a partitioning of $\mathscr{D}$ into k number of clusters, $k \leq K$ is obtained.

Suppose at any k number of clusters, we define the first set of clusters to be $\{C_{(k)i}\}$, where $k \in \{2,3,\dots,K\},\ i = 1,\dots,k$. We define another set of clusters $\{C_{(k+r)j}\}$, where $r \in \{1,2,\dots,K-k\}$ and $j = 1,2,\dots k+r$. These $K - k$ set of clusters may contain fewer elements per cluster than the preceding set of clusters as we increase (centres) k until reaching one element in each $K, (N-1)$ cluster. We define $m_{(k,k+r)ij}$ number of common elements in the forward inter cluster mapping, from the source cluster $C_{(k)i}$ to the target cluster $C_{(k+r)j}$ as the set of all those elements which are a member (common) of both $C_{(k)i}$ and $C_{(k+r)j}$ i.e. $C_{(k)i} \cap C_{(k+r)j}$. These forward mapped numbers of elements are used to construct forward mapped inter cluster $\left[M_{(k,k+r)ij}\right]$ matrices for $i = 1,2, \dots, k$ and $j = 1,2, \dots, k+r$ showing the mapped elements from a particular cluster $C_{(k)i}$ at k to all different sets of target clusters $C_{(k+r)j}$ at $k+r$. Similarly, for backward inter cluster mapping, from the target cluster $C_{(k+r)j}$ to the source cluster $C_{(k)i}$, is defined as $m_{(k+r,k)ji}$ the numbers of common elements, which would be used to construct backward mapped inter cluster $\left[M_{(k+r,k)ji}\right]$ matrices. These backward mapped inter cluster matrices are also known as transpose of the M forward inter cluster mapped matrices. These M matrices are rectangular of size $k \times (k+r)$ and $(k+r) \times k$ respectively. The Figure 4.2 represents forward and backward mapping of m elements from source clusters to target clusters while Figure 4.3 shows the construction of M matrices when $k = 2$ for $k+r$ and $r = 1,2$.



		FORWARD MAPPING									
		$C_{(2)1}$	$C_{(2)2}$	$C_{(3)1}$	$C_{(3)2}$	$C_{(3)3}$	$C_{(4)1}$	$C_{(4)2}$	$C_{(4)3}$	$C_{(4)4}$	
BACKWARD MAPPING	$C_{(2)1}$	1	0	$m_{(2,3)11}$	$m_{(2,3)12}$	$m_{(2,3)13}$	$m_{(2,4)11}$	$m_{(2,4)12}$	$m_{(2,4)13}$	$m_{(2,4)14}$	$C_{(2)1}$
	$C_{(2)2}$	0	1	$m_{(2,3)21}$	$m_{(2,3)22}$	$M_{(2,3)23}$	$m_{(2,4)21}$	$m_{(2,4)22}$	$m_{(2,4)23}$	$m_{(2,4)24}$	$C_{(2)2}$
	$C_{(3)1}$	$m_{(3,2)11}$	$m_{(3,2)12}$	1	0	0	$m_{(3,4)11}$	$m_{(3,4)12}$	$m_{(3,4)13}$	$m_{(3,4)14}$	$C_{(3)1}$
	$C_{(3)2}$	$m_{(3,2)21}$	$m_{(3,2)22}$	0	1	0	$m_{(3,4)21}$	$m_{(3,4)22}$	$m_{(3,4)23}$	$m_{(3,4)24}$	$C_{(3)2}$
	$C_{(3)3}$	$m_{(3,2)31}$	$m_{(3,2)32}$	0	0	1	$m_{(3,4)31}$	$m_{(3,4)32}$	$m_{(3,4)33}$	$m_{(3,4)34}$	$C_{(3)3}$
	$C_{(4)1}$	$m_{(4,2)11}$	$m_{(4,2)12}$	$m_{(4,3)11}$	$m_{(4,3)12}$	$m_{(4,3)13}$	1	0	0	0	$C_{(4)1}$
	$C_{(4)2}$	$m_{(4,2)21}$	$m_{(4,2)22}$	$m_{(4,3)21}$	$m_{(4,3)22}$	$m_{(4,3)23}$	0	1	0	0	$C_{(4)2}$
	$C_{(4)3}$	$m_{(4,2)31}$	$m_{(4,2)32}$	$m_{(4,3)31}$	$m_{(4,3)32}$	$m_{(4,3)33}$	0	0	1	0	$C_{(4)3}$
	$C_{(4)4}$	$m_{(4,2)41}$	$m_{(4,2)32}$	$m_{(4,3)41}$	$m_{(4,3)42}$	$m_{(4,3)43}$	0	0		1	$C_{(4)4}$
		$C_{(2)1}$	$C_{(2)2}$	$C_{(3)1}$	$C_{(3)2}$	$C_{(3)3}$	$C_{(4)1}$	$C_{(4)2}$	$C_{(4)3}$	$C_{(4)4}$	

Figure 4.2: Forward and Backward mapping of common m elements.



		FORWARD MATRICES									
		$C_{(2)1}$	$C_{(2)2}$	$C_{(3)1}$	$C_{(3)2}$	$C_{(3)3}$	$C_{(4)1}$	$C_{(4)2}$	$C_{(4)3}$	$C_{(4)4}$	
BACKWARD MATRICES	$C_{(2)1}$	1	0	$[M_{(k, k+1)}]$			$[M_{(k, k+2)}]$				$C_{(2)1}$
	$C_{(2)2}$	0	1								$C_{(2)2}$
	$C_{(3)1}$	$[M_{(k+1, k)}]$		1	0	0	$[M_{(k, k+1)}]$				$C_{(3)1}$
	$C_{(3)2}$			0	1	0					$C_{(3)2}$
	$C_{(3)3}$			0	0	1					$C_{(3)3}$
	$C_{(4)1}$	$[M_{(k+2, k)}]$		$[M_{(k+1, k)}]$			1	0	0	0	$C_{(4)1}$
	$C_{(4)2}$						0	1	0	0	$C_{(4)2}$
	$C_{(4)3}$						0	0	1	0	$C_{(4)3}$
	$C_{(4)4}$						0	0		1	$C_{(4)4}$
		$C_{(2)1}$	$C_{(2)2}$	$C_{(3)1}$	$C_{(3)2}$	$C_{(3)3}$	$C_{(4)1}$	$C_{(4)2}$	$C_{(4)3}$	$C_{(4)4}$	

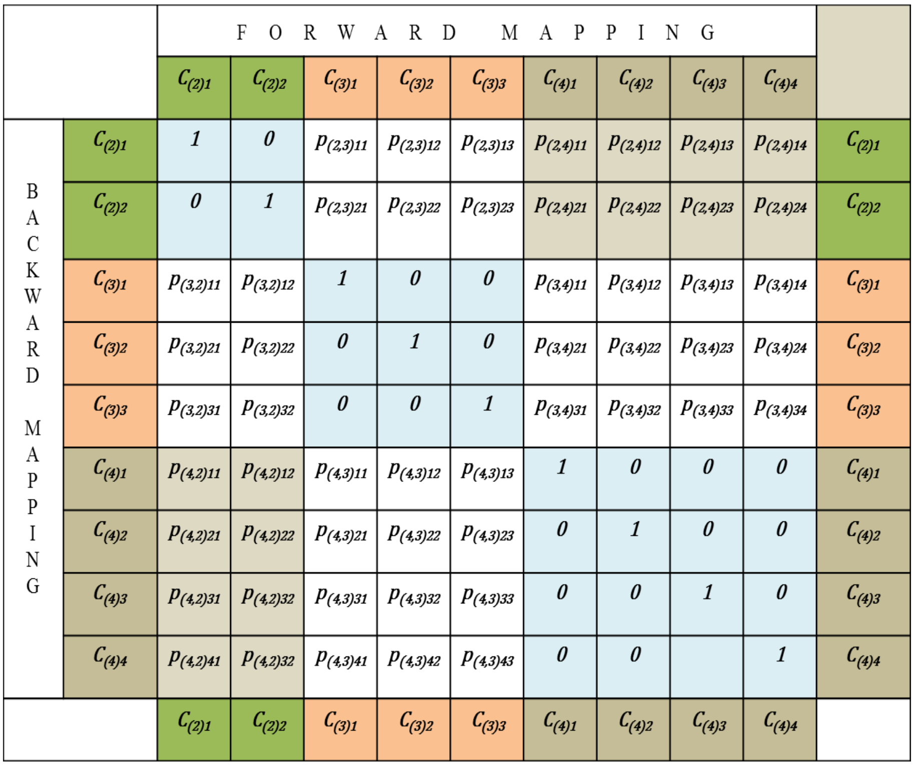
Figure 4.3: Forward and Backward M matrices.

In the backward mapping of elements, the target $C_{(k+r)j}$ cluster is swapped to source $C_{(k)i}$ and source $C_{(k)i}$ to target $C_{(k+r)j}$ clusters. Each row of these forward and backward matrices is represented as the transpose of a vector which shows the elements mapped from the particular cluster to its corresponding cluster. We define forward inter cluster proportion of elements $\left[P_{(k,k+r)ij}\right]$ matrices from particular source cluster $C_{(k)i}$ to all sets of target clusters $C_{(k+r)j}$ in $\left[M_{(k,k+r)ij}\right]$ matrices. Each row (vector) in M is normalized to a unit of 1 which is also known as the row sum to 1 proportion and can be computed as

$$p_{(k,\ \ k+r)ij} = \frac{m_{(k,\ \ k+r)ij}}{\sum_{j=1}^{k+r} m_{(k,\ \ k+r)ij}} \qquad (4.1)$$

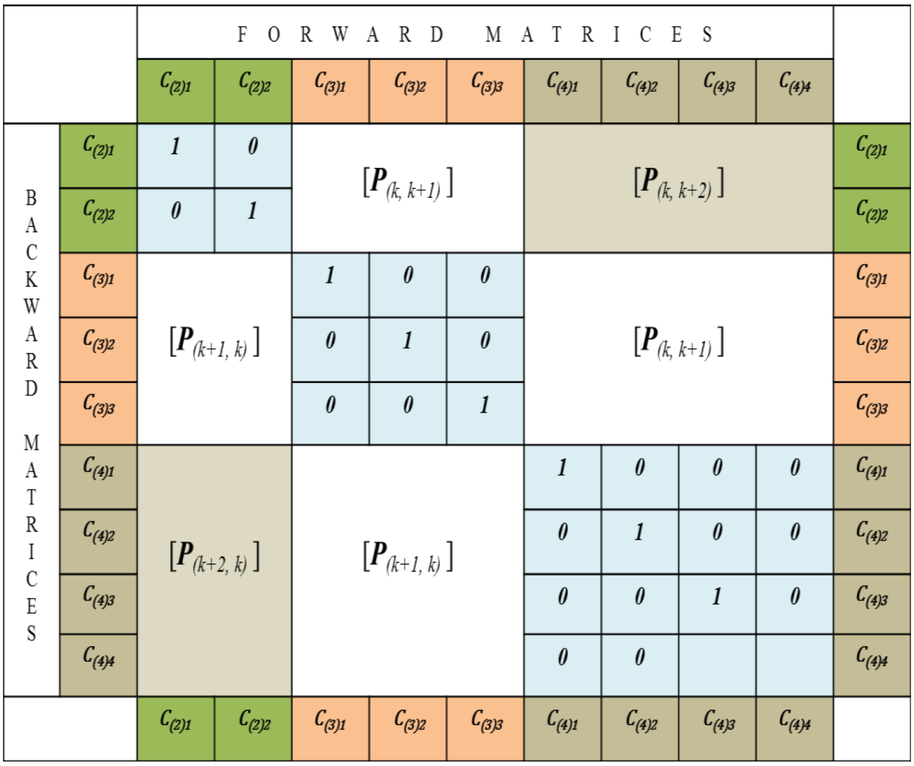
Equation 4.1 simply shows the inter cluster proportion of the elements mapped from the source cluster $C_{(k)i}$ to the target cluster$C_{(k+r)j}$. Similarly, we can obtain the backward inter clusters proportion $\left[P_{(k+r,k)ji}\right]$ matrices.

The Figures 4.4 shows forward and backward proportion p of elements from source cluster to target clusters and Figure 4.5 represents proportion P matrices when $k = 2$, and $r = 1,2$.



		FORWARD MAPPING									
		$C_{(2)1}$	$C_{(2)2}$	$C_{(3)1}$	$C_{(3)2}$	$C_{(3)3}$	$C_{(4)1}$	$C_{(4)2}$	$C_{(4)3}$	$C_{(4)4}$	
BACKWARD MAPPING	$C_{(2)1}$	1	0	$p_{(2,3)11}$	$p_{(2,3)12}$	$p_{(2,3)13}$	$p_{(2,4)11}$	$p_{(2,4)12}$	$p_{(2,4)13}$	$p_{(2,4)14}$	$C_{(2)1}$
	$C_{(2)2}$	0	1	$p_{(2,3)21}$	$p_{(2,3)22}$	$p_{(2,3)23}$	$p_{(2,4)21}$	$p_{(2,4)22}$	$p_{(2,4)23}$	$p_{(2,4)24}$	$C_{(2)2}$
	$C_{(3)1}$	$p_{(3,2)11}$	$p_{(3,2)12}$	1	0	0	$p_{(3,4)11}$	$p_{(3,4)12}$	$p_{(3,4)13}$	$p_{(3,4)14}$	$C_{(3)1}$
	$C_{(3)2}$	$p_{(3,2)21}$	$p_{(3,2)22}$	0	1	0	$p_{(3,4)21}$	$p_{(3,4)22}$	$p_{(3,4)23}$	$p_{(3,4)24}$	$C_{(3)2}$
	$C_{(3)3}$	$p_{(3,2)31}$	$p_{(3,2)32}$	0	0	1	$p_{(3,4)31}$	$p_{(3,4)32}$	$p_{(3,4)33}$	$p_{(3,4)34}$	$C_{(3)3}$
	$C_{(4)1}$	$p_{(4,2)11}$	$p_{(4,2)12}$	$p_{(4,3)11}$	$p_{(4,3)12}$	$p_{(4,3)13}$	1	0	0	0	$C_{(4)1}$
	$C_{(4)2}$	$p_{(4,2)21}$	$p_{(4,2)22}$	$p_{(4,3)21}$	$p_{(4,3)22}$	$p_{(4,3)23}$	0	1	0	0	$C_{(4)2}$
	$C_{(4)3}$	$p_{(4,2)31}$	$p_{(4,2)32}$	$p_{(4,3)31}$	$p_{(4,3)32}$	$p_{(4,3)33}$	0	0	1	0	$C_{(4)3}$
	$C_{(4)4}$	$p_{(4,2)41}$	$p_{(4,2)32}$	$p_{(4,3)41}$	$p_{(4,3)42}$	$p_{(4,3)43}$	0	0		1	$C_{(4)4}$
		$C_{(2)1}$	$C_{(2)2}$	$C_{(3)1}$	$C_{(3)2}$	$C_{(3)3}$	$C_{(4)1}$	$C_{(4)2}$	$C_{(4)3}$	$C_{(4)4}$	

Figure 4.4: Forward and Backward proportion p of mapped common elements.



		FORWARD MATRICES									
		$C_{(2)1}$	$C_{(2)2}$	$C_{(3)1}$	$C_{(3)2}$	$C_{(3)3}$	$C_{(4)1}$	$C_{(4)2}$	$C_{(4)3}$	$C_{(4)4}$	
BACKWARD MATRICES	$C_{(2)1}$	1	0	$[\mathbf{P}_{(k, k+1)}]$			$[\mathbf{P}_{(k, k+2)}]$				$C_{(2)1}$
	$C_{(2)2}$	0	1								$C_{(2)2}$
	$C_{(3)1}$	$[\mathbf{P}_{(k+1, k)}]$		1	0	0	$[\mathbf{P}_{(k, k+1)}]$				$C_{(3)1}$
	$C_{(3)2}$			0	1	0					$C_{(3)2}$
	$C_{(3)3}$			0	0	1					$C_{(3)3}$
	$C_{(4)1}$	$[\mathbf{P}_{(k+2, k)}]$		$[\mathbf{P}_{(k+1, k)}]$			1	0	0	0	$C_{(4)1}$
	$C_{(4)2}$						0	1	0	0	$C_{(4)2}$
	$C_{(4)3}$						0	0	1	0	$C_{(4)3}$
	$C_{(4)4}$						0	0			$C_{(4)4}$
		$C_{(2)1}$	$C_{(2)2}$	$C_{(3)1}$	$C_{(3)2}$	$C_{(3)3}$	$C_{(4)1}$	$C_{(4)2}$	$C_{(4)3}$	$C_{(4)4}$	

Figure 4.5: Forward and Backward proportion P matrices.

In the next section, we will use these proportion matrices to compute the mapped combined mapped proportion and combined mapped elements that will reveal the separation and overlap information.

4.3.2 Combined Proportions and Combined Elements Matrices

Further, we apply the inner product for both forward and backward proportion matrices as computed and constructed in the above section to determine the mutual similarity, as this method is commonly used to measure similarity as discussed in section 4.2. Accordingly, we define $[O_{(k,k)}]$ to be a combined mapped proportion. These are square matrices obtained from multiplying $[P_{(k,k+r)ij}]$ by $[P_{(k+r,k)ji}]$. It is important to note, to determine a similarity maximum from these proportion matrices we will always use the inner product from forward to backward. Therefore, due to the cardinality difference $(k, k+r) \times (k+r, k) \neq (k+r, k) \times (k, k+r)$ backward to forward would be less informative at $k+r$ than forward to backward at k. This may result in loss of information and increase the overlap between clusters. The table below shows the difference between the forward and backward proportion matrices and vice versa $P_{(2,3)} \times P_{(3,2)} \neq P_{(3,2)} \times P_{(2,3)}$ to obtain O matrix. The combined mapped proportions gain in both matrices are not the same e.g. $O_{(2,2)} \neq O_{(3,3)}$.

$P_{(2,3)} = \begin{array}{cc} & \begin{array}{ccc} C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \end{array} \\ \begin{array}{c} C_{(2,1)} \\ C_{(2,2)} \end{array} & \begin{bmatrix} 0.67 & 0.33 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$	$P_{(3,2)} = \begin{array}{cc} & \begin{array}{cc} C_{(2,1)} & C_{(2,2)} \end{array} \\ \begin{array}{c} C_{(3,1)} \\ C_{(3,2)} \\ C_{(3,3)} \end{array} & \begin{bmatrix} 1.0 & 0.0 \\ 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} \end{array}$
$O_{(2,2)} = P_{(2,3)} \times P_{(3,2)} = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$	$O_{(3,3)} = P_{(3,2)} \times P_{(2,3)} = \begin{bmatrix} 0.67 & 0.33 & 0.0 \\ 0.67 & 0.33 & 0.0 \\ 0,0 & 0.0 & 1.0 \end{bmatrix}$

Table 4.1: Forward, backward and combined mapped proportions matrices with opposite cardinality.

The forward $P_{(2,3)}$ and backward $P_{(3,2)}$ are inter cluster proportions while $O_{(2,2)}$ and $O_{(3,3)}$ are combined mapped proportions. The diagonal o_{ii} entries of O matrix represents the similarity between the mapped sets of $C_{(2)i}$ and $C_{(2+1)j}$ clusters for the forward and backward proportions, while off-diagonal o_{ij} entries represent proportion of overlap or dissimilarity between clusters.

Finally, we define and compute "combined mapped elements" $[Q_{(k,k)}]$ matrices by finding the inner product of $[\boldsymbol{k}]$ and $\left[O_{(k,k)}\right]$ matrices for each k and $(k+r)$ distance. The $\left[Q_{(k,k)}\right]$ matrices for each k with different $k+r$ distances show the number of elements belonging to the same cluster (within cluster) similarity at diagonals $q_{(ii)}$ while off diagonal $q_{(ij)}$ are the number of elements belonging to different clusters (overlap). A simple example is described here to show the mapping of elements and proportion between clusters below:

Example: This is an illustration to show how we find the inter cluster mapping of common elements and common proportion of elements between source clusters to all the set of target clusters resulting from the *k-mean*s algorithm at $k = 2$ and $k = 3 = k + 1$ clusterings is discussed as follows: Suppose we have dataset $\mathscr{D}$ with $N = 10$ number of elements and each element has a number of variables that show relationships. We apply the *k-means* algorithm to get the partition of $\mathscr{D}$ into k clusters e.g. for $k_2 (k = 2)$ and $\mathscr{D}$ is partitioned into $C_{(2)1}$ and $C_{(2)2}$ while for $k_3 (k = 3)$ it is partitioned into $C_{(3)1}$, $C_{(3)2}$ and $C_{(3)3}$. Figure 4.6 shows adjacent $(r = 1)$ inter cluster mapping of common, $m_{(k,k+1)ij}$ and $m_{(k+1,k)ji}$, number of elements, $p_{(k,k+1)ij}$ and $p_{(k,k+1)ji}$ proportion forward and backward, when $k = 2$ and $r = 1$ for $k + r$.

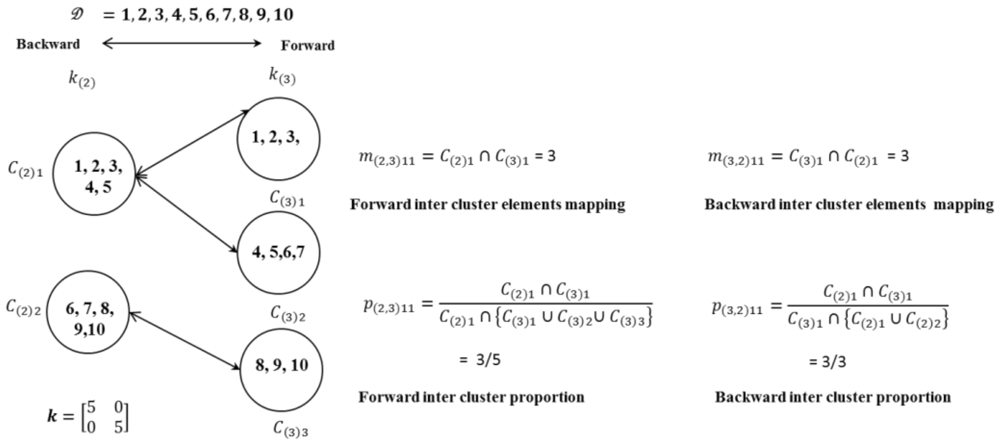

$\mathcal{D}$ = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Backward
Forward
$k_{(2)}$
$k_{(3)}$
$C_{(2)1}$
1, 2, 3, 4, 5
$C_{(2)2}$
6, 7, 8, 9,10
1, 2, 3,
$C_{(3)1}$
4, 5,6,7
$C_{(3)2}$
8, 9, 10
$C_{(3)3}$
$k = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$
$m_{(2,3)11} = C_{(2)1} \cap C_{(3)1} = 3$
Forward inter cluster elements mapping
$m_{(3,2)11} = C_{(3)1} \cap C_{(2)1} = 3$
Backward inter cluster elements mapping
$p_{(2,3)11} = \frac{C_{(2)1} \cap C_{(3)1}}{C_{(2)1} \cap \{C_{(3)1} \cup C_{(3)2} \cup C_{(3)3}\}} = 3/5$
Forward inter cluster proportion
$p_{(3,2)11} = \frac{C_{(2)1} \cap C_{(3)1}}{C_{(3)1} \cap \{C_{(2)1} \cup C_{(2)2}\}} = 3/3$
Backward inter cluster proportion

Figure 4.6: Forward and backward inter cluster mapping of common elements.

It is important to note that since, we have fixed k and r the question is to what extent forward and backward mapping can be made between clusters. In the first place, we can map clusters for $k + r$ adjacent mapping distances $(r = 1)$ as $k = 2,3, \dots, K - 1$ and it would allow us to map maximum $K = 16$ for $k + 1$. As a result, we can map only $k = 15$ to $k = 15 + 1 = K$, for adjacent $(r = 1)$ and similarly, for adjacent to more mapping distances we can only map to the limit of k and r. Let us start at $k = 2$, we are mapping elements in cluster from range $k = 2$ to $k = 16$ clusters with different $k + r$ mapped distant and $r \in (1,2, \dots, K - k)$ e.g. k to $(k + 1, k + 2, \dots, k + 14)$ in forward sequence mapping and $(k + 1, k + 2, \dots, k + 14)$ to k in backward sequence mapping. As a result, for different $k + r$ we can construct 14 different (2×2) combined mapped proportion matrices $[O_{(2,2)}], [O_{(2,2)}], \dots, [O_{(2,2)}]$. As we increase mapping from $k = 2$ to $k = 3$ with different $k + r$ ultimately the number of O matrices will be decreased by one O matrix, which will construct of 13 O matrices. Eventually, mapping at $k = 15$, only one O matrix will be obtained. As combined mapped elements Q matrices are the outcome of O, this would yield the same number of Q matrices at each k with different $k + r$, which can be seen below in Figure 4.7. In the circumstances $k = 15$ and $r = 1$, for $k + r$ mapping distance, only 1 trace value from the Q matrix will be obtained. Hence, the Q matrix will be ignored at $k = 15$ as it cannot be used to compute the average traces or the coefficient of variation. The inter cluster mapping m elements and p proportions at fixed k for different $k + r$ is represented in the tables below. These also show different M and their proportion P matrices when k is fixed and $r = 2,3, \ldots, K - k$ move in sequence.

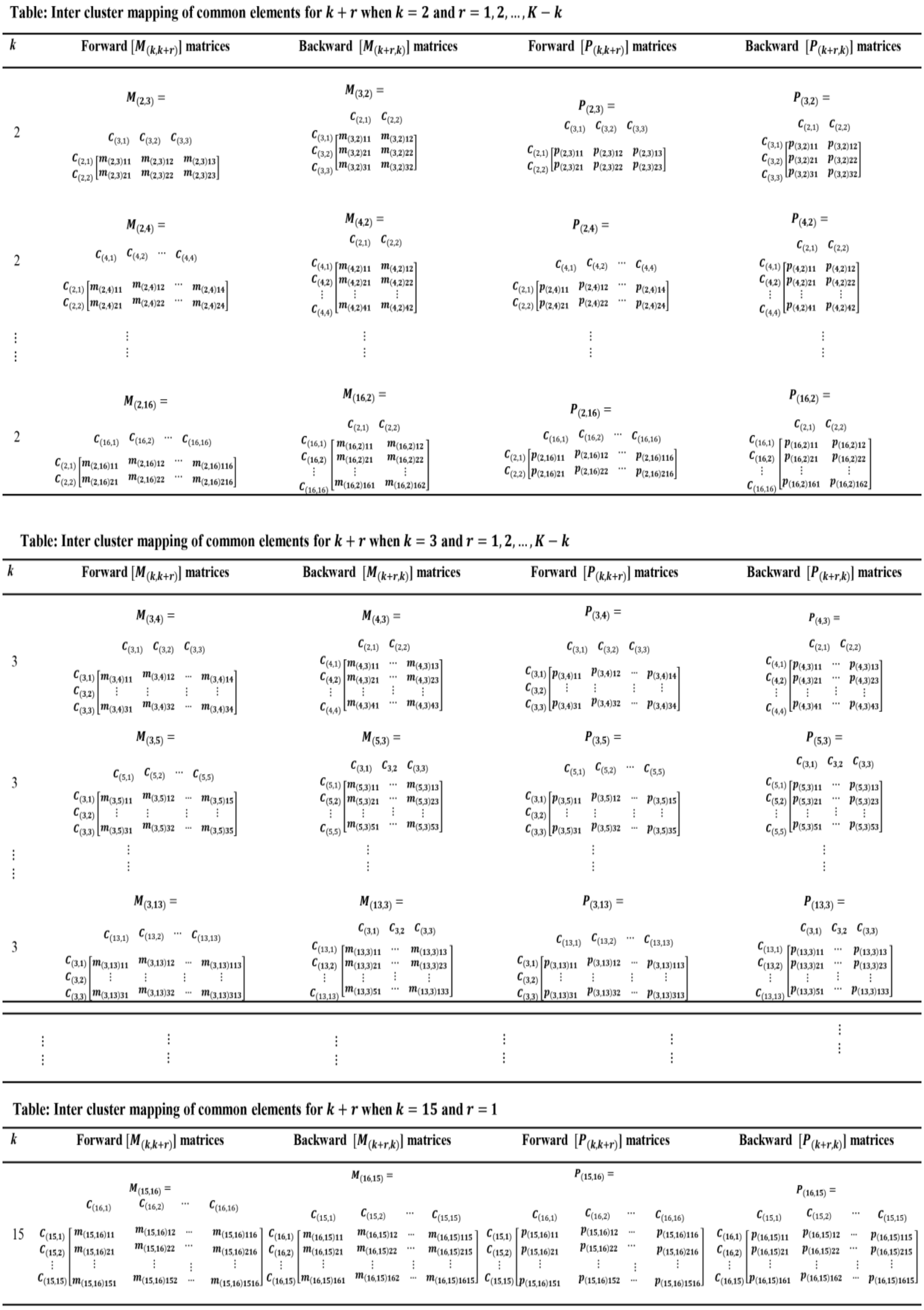


Table: Inter cluster mapping of common elements for $k + r$ when $k = 2$ and $r = 1, 2, \ldots, K - k$

k	Forward $[M_{(k,k+r)}]$ matrices	Backward $[M_{(k+r,k)}]$ matrices	Forward $[P_{(k,k+r)}]$ matrices	Backward $[P_{(k+r,k)}]$ matrices
2	$M_{(2,3)} = \begin{array}{cc} & \begin{matrix} C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \end{matrix} \\ \begin{matrix} C_{(2,1)} \\ C_{(2,2)} \end{matrix} & \begin{bmatrix} m_{(2,3)11} & m_{(2,3)12} & m_{(2,3)13} \\ m_{(2,3)21} & m_{(2,3)22} & m_{(2,3)23} \end{bmatrix} \end{array}$	$M_{(3,2)} = \begin{array}{cc} & \begin{matrix} C_{(2,1)} & C_{(2,2)} \end{matrix} \\ \begin{matrix} C_{(3,1)} \\ C_{(3,2)} \\ C_{(3,3)} \end{matrix} & \begin{bmatrix} m_{(3,2)11} & m_{(3,2)12} \\ m_{(3,2)21} & m_{(3,2)22} \\ m_{(3,2)31} & m_{(3,2)32} \end{bmatrix} \end{array}$	$P_{(2,3)} = \begin{array}{cc} & \begin{matrix} C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \end{matrix} \\ \begin{matrix} C_{(2,1)} \\ C_{(2,2)} \end{matrix} & \begin{bmatrix} p_{(2,3)11} & p_{(2,3)12} & p_{(2,3)13} \\ p_{(2,3)21} & p_{(2,3)22} & p_{(2,3)23} \end{bmatrix} \end{array}$	$P_{(3,2)} = \begin{array}{cc} & \begin{matrix} C_{(2,1)} & C_{(2,2)} \end{matrix} \\ \begin{matrix} C_{(3,1)} \\ C_{(3,2)} \\ C_{(3,3)} \end{matrix} & \begin{bmatrix} p_{(3,2)11} & p_{(3,2)12} \\ p_{(3,2)21} & p_{(3,2)22} \\ p_{(3,2)31} & p_{(3,2)32} \end{bmatrix} \end{array}$
2	$M_{(2,4)} = \begin{array}{cc} & \begin{matrix} C_{(4,1)} & C_{(4,2)} & \cdots & C_{(4,4)} \end{matrix} \\ \begin{matrix} C_{(2,1)} \\ C_{(2,2)} \end{matrix} & \begin{bmatrix} m_{(2,4)11} & m_{(2,4)12} & \cdots & m_{(2,4)14} \\ m_{(2,4)21} & m_{(2,4)22} & \cdots & m_{(2,4)24} \end{bmatrix} \end{array}$	$M_{(4,2)} = \begin{array}{cc} & \begin{matrix} C_{(2,1)} & C_{(2,2)} \end{matrix} \\ \begin{matrix} C_{(4,1)} \\ C_{(4,2)} \\ \vdots \\ C_{(4,4)} \end{matrix} & \begin{bmatrix} m_{(4,2)11} & m_{(4,2)12} \\ m_{(4,2)21} & m_{(4,2)22} \\ \vdots & \vdots \\ m_{(4,2)41} & m_{(4,2)42} \end{bmatrix} \end{array}$	$P_{(2,4)} = \begin{array}{cc} & \begin{matrix} C_{(4,1)} & C_{(4,2)} & \cdots & C_{(4,4)} \end{matrix} \\ \begin{matrix} C_{(2,1)} \\ C_{(2,2)} \end{matrix} & \begin{bmatrix} p_{(2,4)11} & p_{(2,4)12} & \cdots & p_{(2,4)14} \\ p_{(2,4)21} & p_{(2,4)22} & \cdots & p_{(2,4)24} \end{bmatrix} \end{array}$	$P_{(4,2)} = \begin{array}{cc} & \begin{matrix} C_{(2,1)} & C_{(2,2)} \end{matrix} \\ \begin{matrix} C_{(4,1)} \\ C_{(4,2)} \\ \vdots \\ C_{(4,4)} \end{matrix} & \begin{bmatrix} p_{(4,2)11} & p_{(4,2)12} \\ p_{(4,2)21} & p_{(4,2)22} \\ \vdots & \vdots \\ p_{(4,2)41} & p_{(4,2)42} \end{bmatrix} \end{array}$
⋮ ⋮	⋮ ⋮	⋮ ⋮	⋮ ⋮	⋮ ⋮
2	$M_{(2,16)} = \begin{array}{cc} & \begin{matrix} C_{(16,1)} & C_{(16,2)} & \cdots & C_{(16,16)} \end{matrix} \\ \begin{matrix} C_{(2,1)} \\ C_{(2,2)} \end{matrix} & \begin{bmatrix} m_{(2,16)11} & m_{(2,16)12} & \cdots & m_{(2,16)116} \\ m_{(2,16)21} & m_{(2,16)22} & \cdots & m_{(2,16)216} \end{bmatrix} \end{array}$	$M_{(16,2)} = \begin{array}{cc} & \begin{matrix} C_{(2,1)} & C_{(2,2)} \end{matrix} \\ \begin{matrix} C_{(16,1)} \\ C_{(16,2)} \\ \vdots \\ C_{(16,16)} \end{matrix} & \begin{bmatrix} m_{(16,2)11} & m_{(16,2)12} \\ m_{(16,2)21} & m_{(16,2)22} \\ \vdots & \vdots \\ m_{(16,2)161} & m_{(16,2)162} \end{bmatrix} \end{array}$	$P_{(2,16)} = \begin{array}{cc} & \begin{matrix} C_{(16,1)} & C_{(16,2)} & \cdots & C_{(16,16)} \end{matrix} \\ \begin{matrix} C_{(2,1)} \\ C_{(2,2)} \end{matrix} & \begin{bmatrix} p_{(2,16)11} & p_{(2,16)12} & \cdots & p_{(2,16)116} \\ p_{(2,16)21} & p_{(2,16)22} & \cdots & p_{(2,16)216} \end{bmatrix} \end{array}$	$P_{(16,2)} = \begin{array}{cc} & \begin{matrix} C_{(2,1)} & C_{(2,2)} \end{matrix} \\ \begin{matrix} C_{(16,1)} \\ C_{(16,2)} \\ \vdots \\ C_{(16,16)} \end{matrix} & \begin{bmatrix} p_{(16,2)11} & p_{(16,2)12} \\ p_{(16,2)21} & p_{(16,2)22} \\ \vdots & \vdots \\ p_{(16,2)161} & p_{(16,2)162} \end{bmatrix} \end{array}$

Table: Inter cluster mapping of common elements for $k + r$ when $k = 3$ and $r = 1, 2, \ldots, K - k$

k	Forward $[M_{(k,k+r)}]$ matrices	Backward $[M_{(k+r,k)}]$ matrices	Forward $[P_{(k,k+r)}]$ matrices	Backward $[P_{(k+r,k)}]$ matrices
3	$M_{(3,4)} = \begin{array}{cc} & \begin{matrix} C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \end{matrix} \\ \begin{matrix} C_{(3,1)} \\ C_{(3,2)} \\ C_{(3,3)} \end{matrix} & \begin{bmatrix} m_{(3,4)11} & m_{(3,4)12} & \cdots & m_{(3,4)14} \\ \vdots & \vdots & \vdots & \vdots \\ m_{(3,4)31} & m_{(3,4)32} & \cdots & m_{(3,4)34} \end{bmatrix} \end{array}$	$M_{(4,3)} = \begin{array}{cc} & \begin{matrix} C_{(2,1)} & C_{(2,2)} \end{matrix} \\ \begin{matrix} C_{(4,1)} \\ C_{(4,2)} \\ \vdots \\ C_{(4,4)} \end{matrix} & \begin{bmatrix} m_{(4,3)11} & \cdots & m_{(4,3)13} \\ m_{(4,3)21} & \cdots & m_{(4,3)23} \\ \vdots & \vdots & \vdots \\ m_{(4,3)41} & \cdots & m_{(4,3)43} \end{bmatrix} \end{array}$	$P_{(3,4)} = \begin{array}{cc} & \begin{matrix} C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \end{matrix} \\ \begin{matrix} C_{(3,1)} \\ C_{(3,2)} \\ C_{(3,3)} \end{matrix} & \begin{bmatrix} p_{(3,4)11} & p_{(3,4)12} & \cdots & p_{(3,4)14} \\ \vdots & \vdots & \vdots & \vdots \\ p_{(3,4)31} & p_{(3,4)32} & \cdots & p_{(3,4)34} \end{bmatrix} \end{array}$	$P_{(4,3)} = \begin{array}{cc} & \begin{matrix} C_{(2,1)} & C_{(2,2)} \end{matrix} \\ \begin{matrix} C_{(4,1)} \\ C_{(4,2)} \\ \vdots \\ C_{(4,4)} \end{matrix} & \begin{bmatrix} p_{(4,3)11} & \cdots & p_{(4,3)13} \\ p_{(4,3)21} & \cdots & p_{(4,3)23} \\ \vdots & \vdots & \vdots \\ p_{(4,3)41} & \cdots & p_{(4,3)43} \end{bmatrix} \end{array}$
3	$M_{(3,5)} = \begin{array}{cc} & \begin{matrix} C_{(5,1)} & C_{(5,2)} & \cdots & C_{(5,5)} \end{matrix} \\ \begin{matrix} C_{(3,1)} \\ C_{(3,2)} \\ C_{(3,3)} \end{matrix} & \begin{bmatrix} m_{(3,5)11} & m_{(3,5)12} & \cdots & m_{(3,5)15} \\ \vdots & \vdots & \vdots & \vdots \\ m_{(3,5)31} & m_{(3,5)32} & \cdots & m_{(3,5)35} \end{bmatrix} \end{array}$	$M_{(5,3)} = \begin{array}{cc} & \begin{matrix} C_{(3,1)} & C_{3,2} & C_{(3,3)} \end{matrix} \\ \begin{matrix} C_{(5,1)} \\ C_{(5,2)} \\ \vdots \\ C_{(5,5)} \end{matrix} & \begin{bmatrix} m_{(5,3)11} & \cdots & m_{(5,3)13} \\ m_{(5,3)21} & \cdots & m_{(5,3)23} \\ \vdots & \vdots & \vdots \\ m_{(5,3)51} & \cdots & m_{(5,3)53} \end{bmatrix} \end{array}$	$P_{(3,5)} = \begin{array}{cc} & \begin{matrix} C_{(5,1)} & C_{(5,2)} & \cdots & C_{(5,5)} \end{matrix} \\ \begin{matrix} C_{(3,1)} \\ C_{(3,2)} \\ C_{(3,3)} \end{matrix} & \begin{bmatrix} p_{(3,5)11} & p_{(3,5)12} & \cdots & p_{(3,5)15} \\ \vdots & \vdots & \vdots & \vdots \\ p_{(3,5)31} & p_{(3,5)32} & \cdots & p_{(3,5)35} \end{bmatrix} \end{array}$	$P_{(5,3)} = \begin{array}{cc} & \begin{matrix} C_{(3,1)} & C_{3,2} & C_{(3,3)} \end{matrix} \\ \begin{matrix} C_{(5,1)} \\ C_{(5,2)} \\ \vdots \\ C_{(5,5)} \end{matrix} & \begin{bmatrix} p_{(5,3)11} & \cdots & p_{(5,3)13} \\ p_{(5,3)21} & \cdots & p_{(5,3)23} \\ \vdots & \vdots & \vdots \\ p_{(5,3)51} & \cdots & p_{(5,3)53} \end{bmatrix} \end{array}$
⋮ ⋮	⋮ ⋮	⋮ ⋮	⋮ ⋮	⋮ ⋮
3	$M_{(3,13)} = \begin{array}{cc} & \begin{matrix} C_{(13,1)} & C_{(13,2)} & \cdots & C_{(13,13)} \end{matrix} \\ \begin{matrix} C_{(3,1)} \\ C_{(3,2)} \\ C_{(3,3)} \end{matrix} & \begin{bmatrix} m_{(3,13)11} & m_{(3,13)12} & \cdots & m_{(3,13)113} \\ \vdots & \vdots & \vdots & \vdots \\ m_{(3,13)31} & m_{(3,13)32} & \cdots & m_{(3,13)313} \end{bmatrix} \end{array}$	$M_{(13,3)} = \begin{array}{cc} & \begin{matrix} C_{(3,1)} & C_{3,2} & C_{(3,3)} \end{matrix} \\ \begin{matrix} C_{(13,1)} \\ C_{(13,2)} \\ \vdots \\ C_{(13,13)} \end{matrix} & \begin{bmatrix} m_{(13,3)11} & \cdots & m_{(13,3)13} \\ m_{(13,3)21} & \cdots & m_{(13,3)23} \\ \vdots & \vdots & \vdots \\ m_{(13,3)51} & \cdots & m_{(13,3)133} \end{bmatrix} \end{array}$	$P_{(3,13)} = \begin{array}{cc} & \begin{matrix} C_{(13,1)} & C_{(13,2)} & \cdots & C_{(13,13)} \end{matrix} \\ \begin{matrix} C_{(3,1)} \\ C_{(3,2)} \\ C_{(3,3)} \end{matrix} & \begin{bmatrix} p_{(3,13)11} & p_{(3,13)12} & \cdots & p_{(3,13)113} \\ \vdots & \vdots & \vdots & \vdots \\ p_{(3,13)31} & p_{(3,13)32} & \cdots & p_{(3,13)313} \end{bmatrix} \end{array}$	$P_{(13,3)} = \begin{array}{cc} & \begin{matrix} C_{(3,1)} & C_{3,2} & C_{(3,3)} \end{matrix} \\ \begin{matrix} C_{(13,1)} \\ C_{(13,2)} \\ \vdots \\ C_{(13,13)} \end{matrix} & \begin{bmatrix} p_{(13,3)11} & \cdots & p_{(13,3)13} \\ p_{(13,3)21} & \cdots & p_{(13,3)23} \\ \vdots & \vdots & \vdots \\ p_{(13,3)51} & \cdots & p_{(13,3)133} \end{bmatrix} \end{array}$
⋮ ⋮	⋮ ⋮	⋮ ⋮	⋮ ⋮	⋮ ⋮

Table: Inter cluster mapping of common elements for $k + r$ when $k = 15$ and $r = 1$

k	Forward $[M_{(k,k+r)}]$ matrices	Backward $[M_{(k+r,k)}]$ matrices	Forward $[P_{(k,k+r)}]$ matrices	Backward $[P_{(k+r,k)}]$ matrices
15	$M_{(15,16)} = \begin{array}{cc} & \begin{matrix} C_{(16,1)} & C_{(16,2)} & \cdots & C_{(16,16)} \end{matrix} \\ \begin{matrix} C_{(15,1)} \\ C_{(15,2)} \\ \vdots \\ C_{(15,15)} \end{matrix} & \begin{bmatrix} m_{(15,16)11} & m_{(15,16)12} & \cdots & m_{(15,16)116} \\ m_{(15,16)21} & m_{(15,16)22} & \cdots & m_{(15,16)216} \\ \vdots & \vdots & \vdots & \vdots \\ m_{(15,16)151} & m_{(15,16)152} & \cdots & m_{(15,16)1516} \end{bmatrix} \end{array}$	$M_{(16,15)} = \begin{array}{cc} & \begin{matrix} C_{(15,1)} & C_{(15,2)} & \cdots & C_{(15,15)} \end{matrix} \\ \begin{matrix} C_{(16,1)} \\ C_{(16,2)} \\ \vdots \\ C_{(16,15)} \end{matrix} & \begin{bmatrix} m_{(16,15)11} & m_{(16,15)12} & \cdots & m_{(16,15)115} \\ m_{(16,15)21} & m_{(16,15)22} & \cdots & m_{(16,15)215} \\ \vdots & \vdots & \vdots & \vdots \\ m_{(16,15)161} & m_{(16,15)162} & \cdots & m_{(16,15)1615} \end{bmatrix} \end{array}$	$P_{(15,16)} = \begin{array}{cc} & \begin{matrix} C_{(16,1)} & C_{(16,2)} & \cdots & C_{(16,16)} \end{matrix} \\ \begin{matrix} C_{(15,1)} \\ C_{(15,2)} \\ \vdots \\ C_{(15,15)} \end{matrix} & \begin{bmatrix} p_{(15,16)11} & p_{(15,16)12} & \cdots & p_{(15,16)116} \\ p_{(15,16)21} & p_{(15,16)22} & \cdots & p_{(15,16)216} \\ \vdots & \vdots & \vdots & \vdots \\ p_{(15,16)151} & p_{(15,16)152} & \cdots & p_{(15,16)1516} \end{bmatrix} \end{array}$	$P_{(16,15)} = \begin{array}{cc} & \begin{matrix} C_{(15,1)} & C_{(15,2)} & \cdots & C_{(15,15)} \end{matrix} \\ \begin{matrix} C_{(16,1)} \\ C_{(16,2)} \\ \vdots \\ C_{(16,15)} \end{matrix} & \begin{bmatrix} p_{(16,15)11} & p_{(16,15)12} & \cdots & p_{(16,15)115} \\ p_{(16,15)21} & p_{(16,15)22} & \cdots & p_{(16,15)215} \\ \vdots & \vdots & \vdots & \vdots \\ p_{(16,15)161} & p_{(16,15)162} & \cdots & p_{(16,15)1615} \end{bmatrix} \end{array}$

Figure 4.7: Forward and backward mapping of elements and proportion of common elements for each k with different $k + r$.

4.3.3 Computing Cluster Similarity and Overlap (Dissimilarity)

In this section, we further mathematically compute similarity, overlap, average similarity, average overlap and coefficient of variation (CV) from the combined mapped elements Q matrices. We define the similarity as the trace value from each Q matrix , for each k with different $k + r$ distances. Equation 4.2 shows the similarity of Q matrices, where, q_{ii} represents number of elements similarity in the cluster (similar elements) and q_{ij} number of elements (dissimilarity or overlap) belonging to different clusters.

$$Trace\_Q_{(k,k+r)} = \sum_{i=1}^{k} q_{ii} \tag{4.2}$$

In addition, the traces of each k with different $k + r$ mapped distance would be used to define and compute the average similarity at the same k from different $k + r$. The equation 4.3 is representing average similarity at different k.

$$Average\_Trace_{(k)} = \mu_{(k)} = \frac{\sum_{r=1}^{K-k}\left(Trace_{(k,k+r)}\right)}{K-k} \tag{4.3}$$

where $k = 2,3, \ldots K$ and $r = 1,2, \ldots, K - k$.

We define the dissimilarity number of elements overlap between clusters form Q matrices as the number of elements belonging to the others clusters. The overlap between clusters at each k with different $k + r$ can be computed as in equation (4.4).

$$Overlap_{(k,k+r)} = N - Trace_{(k,k+r)} \tag{4.4}$$

Further, to define and compute the average overlaps as the number of elements at - fixed k from $k + r$ distance using the following equation.

$$Average\_Overlap_{(k)} = N - \frac{\sum_{r=1}^{K-k}\left(Trace_{(k,k+r)}\right)}{K-k} \tag{4.5}$$

Finally, we define the best estimated number of clusters $\boldsymbol{K}$ as the maximum average of the traces from equation 4.3 above, i.e. as

$$\boldsymbol{K} = Max\left(Average\_Trace_{(k)}\right) \tag{4.6}$$

Equation 4.6 determines the criteria for the best estimated number of cluster if:

1 Only one average value is a maximum and is not equal to N, the total number of elements in the dataset.

2 More than one of the average values is a maximum and not equal to N. This indicates that if the clusters have potential to split, then the last average value will indicate the best $\boldsymbol{K}$ number after which the average values may start to diminish.

3 Only one average value is a maximum and is equal to N, the total number of elements in the dataset.

4 More than one of the average values is a maximum and equal to N then the best $\boldsymbol{K}$ would be the last one, after which average values will diminish.

For the criteria 3 and 4 the accuracy would be 100 % indicating clusters are fully separated without any overlap of elements between clusters. This would be a special case and the ratio of average similarity to N will be equal to 1. As the average similarity at the best $\boldsymbol{K}$ is equal to N this will also show all $k + r$ mapped distance clusters being fully separated. However, in all of the above circumstances at the best $\boldsymbol{K}$**,** the average overlap will be a minimum.

4.3.4 Cluster Stability

To show the stability of clusters at the best $\boldsymbol{K}$, we compute the coefficient of variation (CV) for each k from the different $k + r$ mapped distances.

$$CV_k = \frac{\sigma_k}{\mu_{(k)}} \tag{4.7}$$

$$\sigma_k = \sqrt{\frac{\sum_{r=1}^{K-k}\left(Trace_{(k,k+r)} - \mu_k\right)}{n-1}} \tag{4.8}$$

where σ_k is the SD of the traces and μ_k is the average similarity from equation 4.3 at different k. The minimum value of CV at the best $\boldsymbol{K}$ will represent clusters which are more stable with small perturbation.

For better understanding to explore clustering structure, we would visualise similarity, average similarity, average overlap and coefficient of variation computed as above by plotting them at each k for different $k + r$. For example, in the situation of similarity from each Q, trace values can be plotted against each k for different $k + r$. Likewise, plots can be made for average similarity, average overlap, overlap and CV values at different k. The purpose of producing these plots is to clearly visualise the value of traces and other calculations at each and different k. The graphs may also be helpful in understanding the structure in the data. As explained above, from these graphs the maximum average similarity value indicates the best value of $\boldsymbol{K}$. In the circumstances that the plotted line indicates a plateau as the maximum average similarity (more than one) is consistent, the point before the plotted value decreases would estimate the best $\boldsymbol{K}$ number of clusters.

The figure below indicate combined mapped proportion O and combined mapped elements Q matrices, traces (similarity) and overlap between clusters at each k from Q matrices with $k + r$ distance.

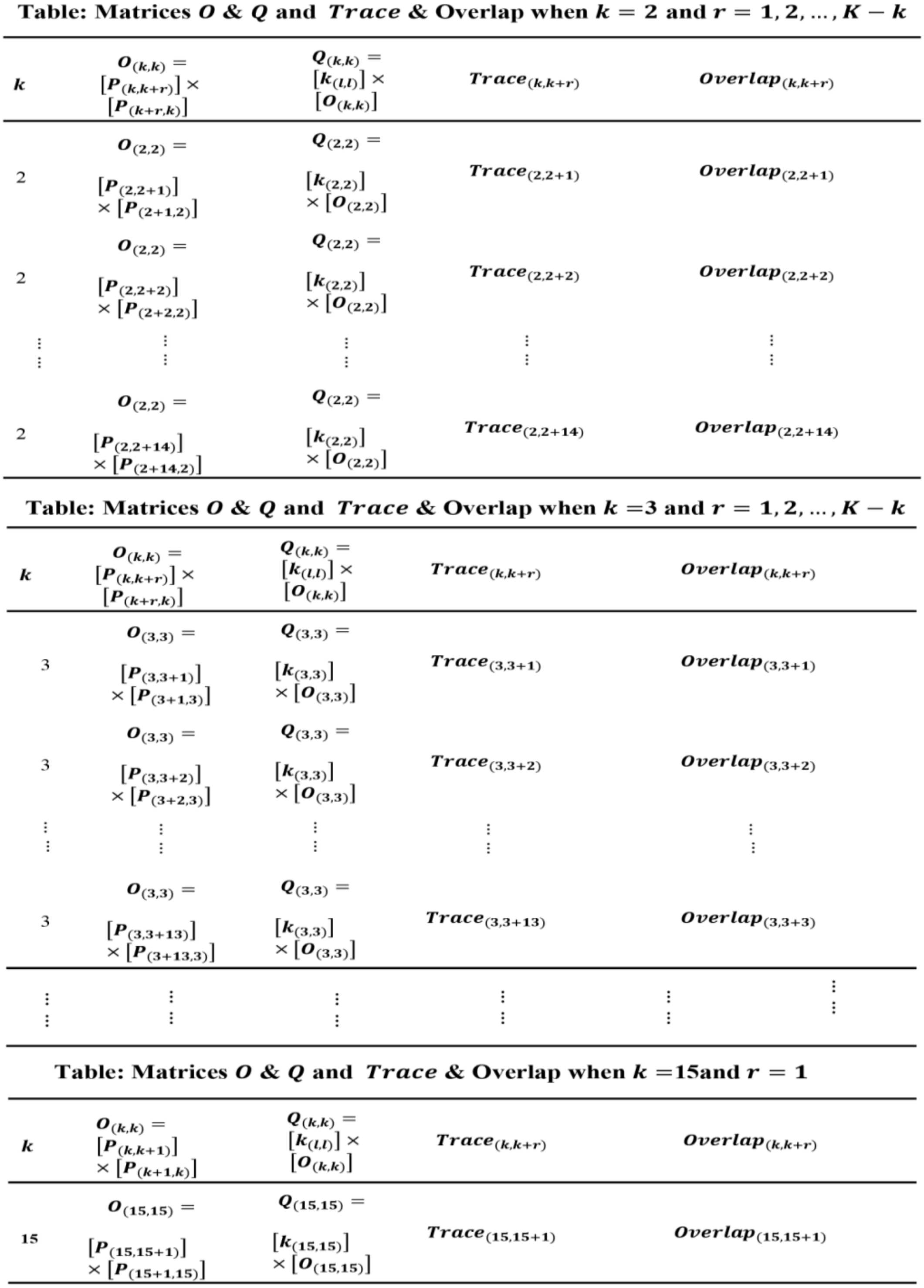


Table: Matrices O & Q and $Trace$ & Overlap when $k = 2$ and $r = 1, 2, \dots, K - k$

k	$O_{(k,k)} =$ $[P_{(k,k+r)}] \times$ $[P_{(k+r,k)}]$	$Q_{(k,k)} =$ $[k_{(l,l)}] \times$ $[O_{(k,k)}]$	$Trace_{(k,k+r)}$	$Overlap_{(k,k+r)}$
2	$O_{(2,2)} =$ $[P_{(2,2+1)}]$ $\times [P_{(2+1,2)}]$	$Q_{(2,2)} =$ $[k_{(2,2)}]$ $\times [O_{(2,2)}]$	$Trace_{(2,2+1)}$	$Overlap_{(2,2+1)}$
2	$O_{(2,2)} =$ $[P_{(2,2+2)}]$ $\times [P_{(2+2,2)}]$	$Q_{(2,2)} =$ $[k_{(2,2)}]$ $\times [O_{(2,2)}]$	$Trace_{(2,2+2)}$	$Overlap_{(2,2+2)}$
⋮	⋮	⋮	⋮	⋮
2	$O_{(2,2)} =$ $[P_{(2,2+14)}]$ $\times [P_{(2+14,2)}]$	$Q_{(2,2)} =$ $[k_{(2,2)}]$ $\times [O_{(2,2)}]$	$Trace_{(2,2+14)}$	$Overlap_{(2,2+14)}$

Table: Matrices O & Q and $Trace$ & Overlap when k =3 and $r = 1, 2, \dots, K - k$

k	$O_{(k,k)} =$ $[P_{(k,k+r)}] \times$ $[P_{(k+r,k)}]$	$Q_{(k,k)} =$ $[k_{(l,l)}] \times$ $[O_{(k,k)}]$	$Trace_{(k,k+r)}$	$Overlap_{(k,k+r)}$
3	$O_{(3,3)} =$ $[P_{(3,3+1)}]$ $\times [P_{(3+1,3)}]$	$Q_{(3,3)} =$ $[k_{(3,3)}]$ $\times [O_{(3,3)}]$	$Trace_{(3,3+1)}$	$Overlap_{(3,3+1)}$
3	$O_{(3,3)} =$ $[P_{(3,3+2)}]$ $\times [P_{(3+2,3)}]$	$Q_{(3,3)} =$ $[k_{(3,3)}]$ $\times [O_{(3,3)}]$	$Trace_{(3,3+2)}$	$Overlap_{(3,3+2)}$
⋮	⋮	⋮	⋮	⋮
3	$O_{(3,3)} =$ $[P_{(3,3+13)}]$ $\times [P_{(3+13,3)}]$	$Q_{(3,3)} =$ $[k_{(3,3)}]$ $\times [O_{(3,3)}]$	$Trace_{(3,3+13)}$	$Overlap_{(3,3+3)}$

⋮	⋮	⋮	⋮	⋮	⋮

Table: Matrices O & Q and $Trace$ & Overlap when k =15and $r = 1$

k	$O_{(k,k)} =$ $[P_{(k,k+1)}]$ $\times [P_{(k+1,k)}]$	$Q_{(k,k)} =$ $[k_{(l,l)}] \times$ $[O_{(k,k)}]$	$Trace_{(k,k+r)}$	$Overlap_{(k,k+r)}$
15	$O_{(15,15)} =$ $[P_{(15,15+1)}]$ $\times [P_{(15+1,15)}]$	$Q_{(15,15)} =$ $[k_{(15,15)}]$ $\times [O_{(15,15)}]$	$Trace_{(15,15+1)}$	$Overlap_{(15,15+1)}$

Figure 4.8: Combined mapped proportion, combined mapped elements, traces and overlap for each k with different $k + r$.

4.4 Computation Illustration for the New Approach

In this section we will compute and demonstrate the inter cluster forward and backward mapping common elements to construct M, P, O, k and Q matrices by an example as follows:

4.4.1 Inter Cluster Mapping when $k = 2$ and $r = 1$

We begin using the notation defined in section 4.3 when $k = 2$ and $r = 1$ $(k = 2 + 1 = 3)$ to compute the forward and backward inter clusters mapping to construct the M, P, O and Q matrices. Suppose we have $k = 2(k_2)$ and $k + r = 2 + 1 = 3 = k(k_3)$ number of resultant clusters from the *k-means* algorithm. The $m_{(2,2+1)ij}$ number of common elements mapped forward between clusters $C_{(2)i}$ from k_2 and $C_{(2+1)j}$ from k_3 are the set of all those elements which are a member (common) of both source cluster $C_{(2)i}$ and target cluster $C_{(2+1)j}$ i.e $C_{(2)i} \cap C_{(2+1)j}$ and vice versa i.e. $m_{(2+1,2)ji}$ number of common elements mapped backward from a particular target $C_{(2+1)j}$ of k_3 to all sets of source clusters $C_{(2)i}$ of k_2 where $i = 1,2$ and $j = 1,2,3$. We construct the forward matrix $\left[M_{(2,2+1)ij}\right]$ for $i = 1,2$ (rows) and $j = 1,..,2 + 1$ (columns) by substituting all mapped entries from a particular cluster $C_{(2)i}$ at $k = 2$ to all the different sets of clusters $C_{(2+1)j}$ at $k = 2 + 1$. Similarly, the backward $\left[M_{(2+1,2)ji}\right]$ matrix is constructed by substituting all $m_{(2+1,2)ji}$ mapped elements. Both of these matrices are rectangular of size $2 \times (2 + 1)$ and $(2 + 1) \times 2$ respectively. In the matrix$\left[M_{(2+1,2)ji}\right]$, we are mapping the smaller sized clusters (less number of elements) $\left\{C_{(2+1)j}\right\}$ to the larger sized clusters (large number of elements) $\left\{C_{(2)i}\right\}$. It is more likely that all elements in a cluster from $\left\{C_{(2+1)i}\right\}$ would map to a cluster from $\left\{C_{(2)j}\right\}$, which indicates complete inter cluster mapping of elements. Next we would compute the forward inter cluster proportion from a particular source cluster $\{C_{(2)i}\}$ to all different set of target clusters $C_{(2+1)j}$ in the $[M_{(2,2+1)ij}]$ matrix using equation 4.1 in section 4.3.1. This is also known as row sum forward inter cluster proportion matrix $[P_{(2,2+1)ij}]$ and is computed as below:

$$p_{(2,\ 2+1)ij} = \frac{m_{(2,\ 2+1)ij}}{\sum_{j=1}^{2+1} m_{(2,\ 2+1)ij}} \tag{4.9}$$

Equation 4.9 simply shows the proportion of the elements mapped from the source cluster $C_{(2)i}$ to the target cluster $C_{(2+1)j}$. Similarly, we can construct the backward inter cluster proportion matrix $[P_{(2+1,2)ji}]$ for $j = 1,..,2+1$ (rows) and $i = 1,2$ (columns) which shows the inter cluster proportion of elements mapping from a particular cluster $C_{(2+1)j}$ at $k = 3$ to all different set of clusters $C_{(2)i}$ at $k = 2$.

4.4.2 Combined Mapped Proportion Matrices

We achieve this step using the inner product both on inter cluster forward and backward proportion matrices computed as $[O_{(2,2)}] = [P_{(2,2+1)ij}] \times [P_{(2+1,2)ji}]$ matrix of 2×2 for $k + 1$ mapped distance. The diagonal elements of $[O_{(2,2)}]$ are o_{ii} where $i = 1,2$ and ith row and column, the combined mapped proportions, while the off-diagonal elements o_{ij} are cluster proportions belonging to the different clusters. The matrix $[O_{(2,2)}]$ would indicate the transition probability matrix where each row sums to 1 as elements of a transition probability matrix should. As each row sum of O is 1 it is clear that O is a row stochastic matrix [162] or probability transition matrix [163]. Up until now, we have explained and computed the $[O_{(2,2)}]$ for $k + 1$. Similarly, we can compute combined proportion matrices in sequence $[O_{(2,2)}], [O_{(2,2)}], \dots, [O_{(2,2)}]$ for different $k + r$ mapped distance, $r = 2,3, \dots, K - k$.

4.4.3 Combined Mapped Elements Matrices

To obtain combined mapped elements $[Q_{(2,2)}]$ by applying the inner product for $[k_{(2,2)}]$ and $[O_{(2,2)}]$ for $k+1$, $k_{(2,2)}$ is constructed form *k-means* clustering results. Similarly, we can compute further when $k=2$ fixed $[Q_{(2,2)}], [Q_{(2,2)}], \dots, [Q_{(2,2)}]$ for all $k+r$ mapped distances. Likewise, forward and backward elements would be mapped for each $k = 3,4, \dots, 15$ and $r = 1,2, \dots, K-k$ with different $k+r$ mapping distances to compute different combined mapped elements matrices $[Q_{(3,3)}], [Q_{(3,3)}], \dots, [Q_{(3,3)}]$, $[Q_{(4,4)}], [Q_{(4,4)}], \dots, [Q_{(4,4)}]$,…, $[Q_{(15,15)}]$ respectively. To illustrate this process we have provided a simple worked example below:

Example: Suppose the dataset $\mathscr{D}$ comprises $\{a,b,c,d,e,f,g,h,i,j,k,l,m,n\}, N = 14$. By applying the *k-means* algorithm $\mathscr{D}$ is partitioned into $k=2$ and $k=3$ clusters. We constructed the M,P and O matrices for $k=2$, and $k=3(k+r)$ when $r=1$. The Figure 4.9 shows the number of elements in cluster $C_{(2)1}$ and $C_{(2)2}$ are 9 and 5 while in the cluster $C_{(3)1}$,$C_{(3)2}$ and $C_{(3)3}$ are $6, 3$ and 5 for $k+1$.

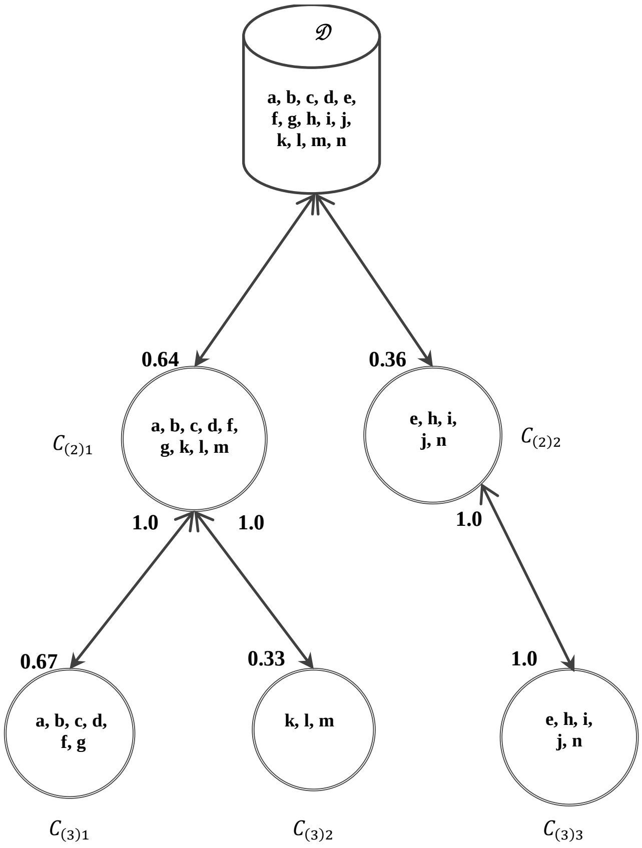


Figure 4.9: Forward and backward inter cluster mapping of common elements at $k = 2$ and $k = 3$ number of clusters.

k+1* value of *k	**Forward mapping of common elements $[M_{(k,k+1)}]$**	**Backward mapping of common elements $[M_{(k+1,k)}]$**	**Forward proportion of elements $[P_{(k,k+1)}]$**	**Backward proportion of elements $[P_{(k+1,k)}]$**
2	$M_{(2,3)} = \begin{array}{c c} & \begin{matrix} C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \end{matrix} \\ \begin{matrix} C_{(2,1)} \\ C_{(2,2)} \end{matrix} & \begin{bmatrix} 6 & 3 & 0 \\ 0 & 0 & 5 \end{bmatrix} \end{array}$	$M_{(3,2)} = \begin{array}{c c} & \begin{matrix} C_{(2,1)} & C_{(2,2)} \end{matrix} \\ \begin{matrix} C_{(3,1)} \\ C_{(3,2)} \\ C_{(3,3)} \end{matrix} & \begin{bmatrix} 6 & 0 \\ 3 & 0 \\ 0 & 5 \end{bmatrix} \end{array}$	$P_{(2,3)} = \begin{array}{c c} & \begin{matrix} C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \end{matrix} \\ \begin{matrix} C_{(2,1)} \\ C_{(2,2)} \end{matrix} & \begin{bmatrix} 0.67 & 0.33 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$	$P_{(3,2)} = \begin{array}{c c} & \begin{matrix} C_{(2,1)} & C_{(2,2)} \end{matrix} \\ \begin{matrix} C_{(3,1)} \\ C_{(3,2)} \\ C_{(3,3)} \end{matrix} & \begin{bmatrix} 1.0 & 0.0 \\ 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} \end{array}$

$\boldsymbol{k}$	$O_{(2,2)} = [P_{(2,2+1)}] \times [P_{(2+1,2)}]$	$Q_{(2,2)} = [k_{(2,2)}] \times [O_{(2,2)}]$	$Trace_{(2,2+1)}$	$Overlap_{(2,2+1)}$	$O_{(3,3)} = [P_{(3,2)}] \times [P_{(2,3)}]$
2	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.67 & 0.33 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$ **Inner product**	$\begin{bmatrix} 9.0 & 0.0 \\ 0.0 & 5.0 \end{bmatrix} =$ $\begin{bmatrix} 9 & 0 \\ 0 & 5 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	14	0	$\begin{bmatrix} 0.67 & 0.33 & 0.0 \\ 0.67 & 0.33 & 0.0 \\ 0,0 & 0.0 & 1.0 \end{bmatrix}$ **Inner product**
$O_{(2,2)} \neq O_{(3,3)}$					

Table 4.2: Shows value of trace & overlap and matrices M, P, O and Q.

From the Table 4.2 we can see the forward inter cluster mapped element from $C_{(2)1}$ to $C_{(3)1}$ is 6 , $C_{(2)1}$ to $C_{(3)2}$ is 3 and $C_{(2)2}$ to $C_{(3)3}$ is 5, while backward elements from

$C_{(3)2}$ to $C_{(2)1}$ and $C_{(3)3}$ to $C_{(2)2}$ are completely mapped. This shows that smaller sized clusters are mapped to their source clusters completely as discussed in section 4.4.1. i.e. $p_{(2+1,2)11} = p_{(2+1,2)21} = p_{(2+1,2)32} = 1$. The Table 4.2 shows the row sum of proportion equal to 1 for O matrices, which form the transition matrix. As discussed 4.3 section we would apply the inner product only in the forward to backward direction to determine the maximum mapped similarity. Accordingly, in the table the last column indicates the difference between using inner products for $P_{(2,3)} \times P_{(3,2)}$ and $P_{(3,2)} \times P_{(2,3)}$ to obtain O. This shows the similarity at the diagonal in the forward direction $P_{(2,3)} \times P_{(3,2)}$ is higher than the opposite direction of $P_{(3,2)} \times P_{(3,2)}$ in the proportion matrices.

4.5 Adjacent and Non-adjacent Mapping

Here, we demonstrate and examine by an example the effect of adjacent and non-adjacent mapping over greater mapping distances and consider average mapping distances for estimating the best number of clusters. This example represents specifically the effect of $k + 1$ adjacent, $k + 2$, and $k + 3$ non-adjacent mapping distances (r) at different k to obtain combined mapped elements matrices. These matrices show the similarity (traces) and overlap of elements between clusters. To illustrate this, a well-known Ruspini dataset is used with four clusters known in advance. It is a commonly used dataset in the literature to illuminate and evaluate clustering structure which was first used and analysed by Ruspini [164] to investigate fuzzy clustering. It is a two dimensional numerical dataset that includes 75 elements in total. Figure 4.11(a) shows the scatter plot of this dataset.

Table 4.3(a) shows the traces and overlaps for adjacent mapping at different k for fixed $k + 1$ in the manner $(k,\ k + 1) \times (k + 1,\ k)$ e.g. $(2, 3 \times 3, 2), (3, 4 \times 4, 3), \dots, (15, 16 \times 16, 15)$ while Table 4.3(b) looks for non-adjacent mapping at different k for fixed $k+2$ in the manner $(k, k+2) \times (k+2, k)$ e.g. $(2, 4 \times 4, 2), (3, 5 \times 5, 3), \dots, (14, 16 \times 16, 14)$. These are also shown using similarity and overlap values at different k, as well as the number of k values 15 and 14 for $k+1$ and $k+2$ respectively as discussed in section 4.3.2.

k	$(k, k+1) \times (k+1, k)$	***Traces (similarity)***	***Overlap***	
2	(2,3×3,2)	**75**	**0**	$k = 2,3, .. 15, r = 1$
3	(3,4×4,3)	**75**	**0**	
4	(4,5×5,4)	**75**	**0**	
5	(5,6×6,5)	**75**	**0**	
6	(6,7×7,6)	**75**	**0**	
7	(7,8×8,7)	**73.21**	**1.79**	
8	(8,9×9,8)	**71.56**	**3.44**	
9	(9,10×10,9)	**75**	**0**	
10	(10,11×11,10)	**64.61**	**10.39**	
11	(11,12×12,11)	**72.35**	**2.65**	
12	(12,13×13,12)	**71.79**	**3.21**	
13	(13,14×14,13)	**73.66**	**1.34**	
14	(14,15×15,14)	**71.6**	**3.4**	
15	(15,16×16,15)	**66.02**	**8.98**	

(a)

k	$(k, k+2) \times (k+2, k)$	***Traces (similarity)***	***Overlap***	
2	(2,4×4,2)	**75**	**0**	$k = 2,3, .. 14, r = 2$
3	(3,5×5,3)	**75**	**0**	
4	(4,6×6,4)	**75**	**0**	
5	(5,7×7,5)	**75**	**0**	
6	(6,8×8,6)	**73.21**	**1.79**	
7	(7,9×9,7)	**69.77**	**5.23**	
8	(8,10×10,8)	**71.56**	**3.44**	
9	(9,11×11,9)	**71.78**	**3.22**	
10	(10,12×12,10)	**70.44**	**4.56**	
11	(11,13×13,11)	**69.14**	**5.86**	
12	(12,14×14,12)	**70.45**	**4.55**	
13	(13,15×15,13)	**70.26**	**4.74**	
14	(14,16×16,14)	**75**	**0**	

(b)

Table 4.3: Summary of traces (similarity) and overlap values at different k for $k+1$ and $k+2$ mapped distances.

The results in Table 4.3(a) show from $k = 2$ to $k = 6$ traces (similarity) values are at the maximum equal to $N = 75$ indicating no overlap between clusters for $k+1$. Table (b) shows a small change for the $k+2$ mapping distance, where the traces are a maximum and equal to $N = 75$ indicating no overlap between clusters from $k = 2$ to $k = 5$. This indicates as mapping distances increase from $k+1$ to $k+2$ a number of elements with similar characteristics merge to form a cluster in this dataset. This data contains clusters which are well separated with few elements in common as seen in Figure 4.10(a) clusters merge as the mapping distance increases. In the situation of $k+1$ and $k+2$ the estimated number of clusters are 6 and 5 respectively while the correct number of clusters is 4. This is one of the scenarios when $k+1$ or $k+2$ mapping distances are unable to detect a correct number of clusters. For typical cluster analysis the structure of data is unknown in respect of shapes and sizes, high or low density and the number of clusters is unknown in advance. Therefore, it is essential to show the behaviour of adjacent and non-adjacent forward and backward mapping distances. To control these structure issues we will map all the clusters at fixed k with different $k + r$ to the limit of K. Then we compute average similarity (traces) at different k for the best and stable set of the clusters. Figure 4.10 shows the difference between traces at different k.

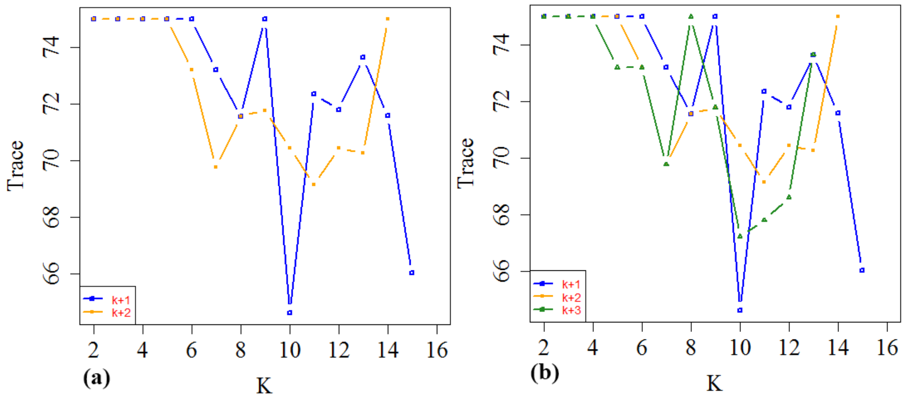


Figure 4.10: Similarity at different k with $k + 1$, $k + 2$ and $k + 3$.

In the figure plot (a) shows the difference between trace values for $k + 1$ and $k + 2$ while plot (b) shows the differences simultaneously from $k + 1$ to $k + 3$ at different k in different colours (blue, orange and green). From these graphs, by considering the $k + 1$ blue line we see the number of clusters is estimated as 6, the $k + 2$ orange line suggests 5 clusters and the $k + 3$ line indicates 4 clusters when the trace values in each mapping distance equal $N = 75$. It is known in advance that the correct number of clusters is 4 while using $k + r$ mapping provides different numbers of clusters. These two Figure 4.10 (a) and (b) plots indicate the number of clusters change as mapping distance increases. Adjacent and non-adjacent mapping distances are not always appropriate for estimating the number of clusters. To avoid this problem we will map the elements in the set of clusters for more distant $k + r$ where $r > 2$ in sequence. Then the average similarity for determining the best $\boldsymbol{K}$ clusters will be computed and the stability of these $\boldsymbol{K}$ clusters checked. Table 4.4 shows traces (similarity) at $k = 2$ and 3 with different $k + r$ sequence mapped distances which will be used to compute the average of the traces.

	k	r	$(k, k + r) \times (k + r, k)$	***Traces (similarity)***	***Overlap***
$k = 2, r = 1,2, \dots, K - k$	2	1	(2,3×3,2)	**75**	**0**
	2	2	(2,4×4,2)	**75**	**0**
	2	3	(2,5×5,2)	**75**	**0**
	2	4	(2,6×6,2)	**75**	**0**
	2	5	(2,7×7,2)	**75**	**0**
	2	6	(2,8×8,2)	**75**	**0**
	2	7	(2,9×9,2)	**75**	**0**
	2	8	(2,10×10,2)	**75**	**0**
	2	9	(2,11×11,2)	**75**	**0**
	2	10	(2,12×12,2)	**75**	**0**
	2	11	(2,13×13,2)	**75**	**0**
	2	12	(2,14×14,2)	**75**	**0**
	2	13	(2,15×15,2)	**75**	**0**
	2	14	(2,16×16,2)	**75**	**0**

(a)

	k	r	$(k, k + r) \times (k + r, k)$	***Traces (similarity)***	***Overlap***
$k = 3, r = 1,2, \dots, K - k$	3	1	(3,4×4,3)	**75**	**0**
	3	2	(3,5×5,3)	**75**	**0**
	3	3	(3,6×6,3)	**75**	**0**
	3	4	(3,7×7,3)	**75**	**0**
	3	5	(3,8×8,3)	**75**	**0**
	3	6	(3,9×9,3)	**75**	**0**
	3	7	(3,10×10,3)	**75**	**0**
	3	8	(3,11×11,3)	**75**	**0**
	3	9	(3,12×12,3)	**75**	**0**
	3	10	(3,13×13,3)	**75**	**0**
	3	11	(3,14×14,3)	**75**	**0**
	3	12	(3,15×15,3)	**75**	**0**
	3	13	(3,16×16,3)	**75**	**0**

(b)

Table 4.4: Summary of traces and overlap when fixed $k = 2$ and $k = 3$ with different $k + r$.

Table 4.4(a) shows for $k = 2$ that the traces (similarity) equal N, while Table 4.4(b) for $k = 3$ shows the similarity is also equal to N. We proceed to compute the average similarity to obtain the best number of clusters and to find coefficients of variation between traces which specifies the stable set of clusters.

Similarity				***Overlap***			
K	***Max Trace***	***Min Trace***	***Average Traces***	***CV***	***Max Overlap***	***Min Overlap***	***Average Overlap***
2	75	75	**75**	**0**	0	0	**0**
3	75	75	**75**	**0**	0	0	**0**
4	75	75	**75**	**0**	0	0	**0**
5	75	73.21	**73.607**	**0.01**	1.79	0	**1.393**
6	75	68.16	**72**	**0.029**	6.84	0	**3**
7	73.21	66.8	**70.449**	**0.032**	8.2	1.79	**4.551**
8	75	68.59	**71.795**	**0.031**	6.41	0	**3.205**
9	75	67.09	**70.297**	**0.038**	7.91	0	**4.703**
10	70.44	64.61	**67.202**	**0.029**	10.39	4.56	**7.798**
11	72.35	65.96	**68.61**	**0.035**	9.04	2.65	**6.39**
12	71.79	68.61	**70.325**	**0.019**	6.39	3.21	**4.675**
13	73.66	70.26	**72.527**	**0.027**	4.74	1.34	**2.473**
14	75	71.6	**73.3**	**0.033**	3.4	0	**1.7**

Table 4.5: Summarises the different values of traces, overlap and CV values at different k.

Table 4.5 shows the values of maximum, minimum, similarity and overlap, average for similarity and overlap with coefficient of variation(CV). For better understanding examine the Figure 4.11.

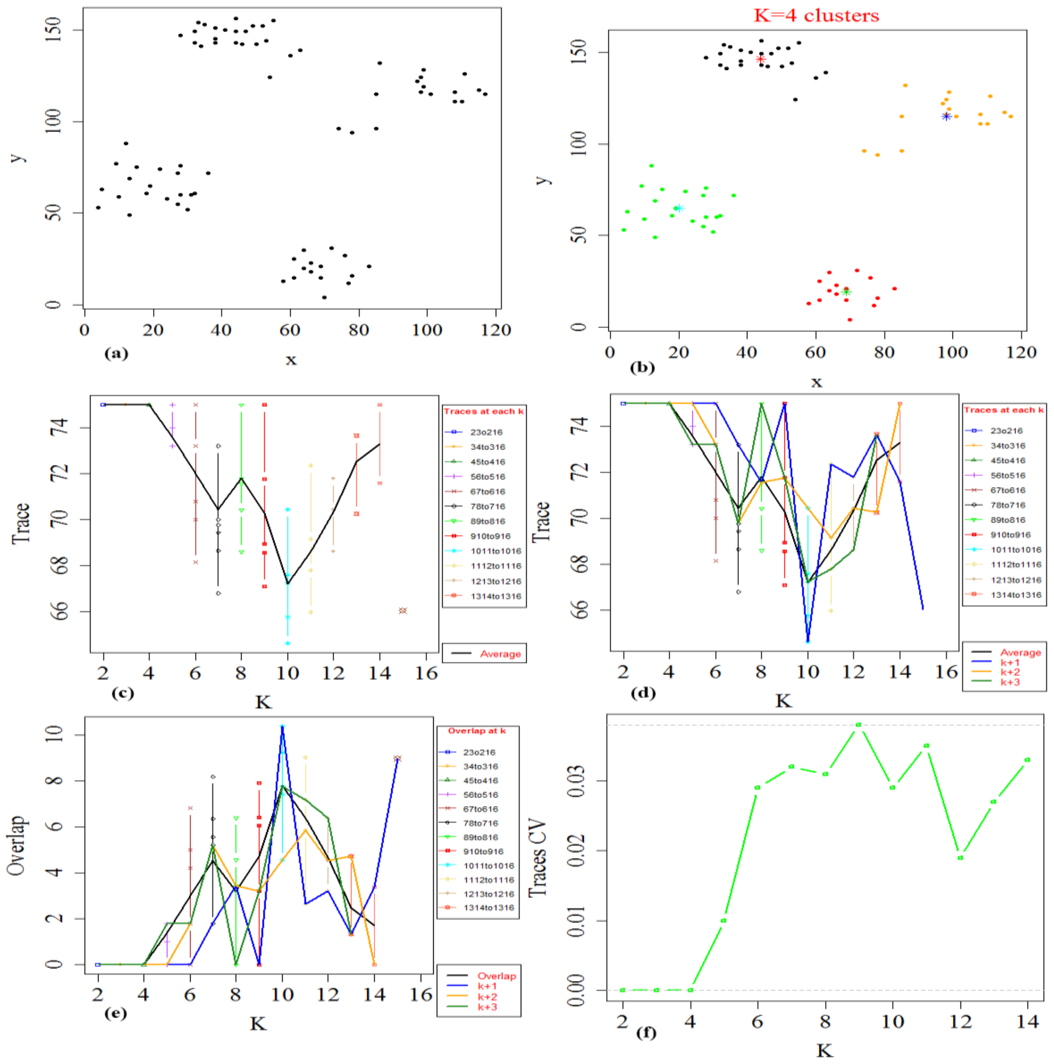


Figure 4.11: Plot (a) shows the scatter plot of the Rusipini dataset. Plot (b) shows the number of clusters obtained by a *k-means* algorithm using $k = 4$ and membership of elements labeled with their centroids in different colours. Plots (c)-(f) show the trace values, overlap and coefficient of variation (CV) at different k for $k + r$ mapped distances.

Figure 4.11(c) shows no variation occurs between traces obtained from $k = 2$ to $k = 4$ with different $k + r$ and the black solid line shows the average similarity values are reaching a maximum until $k = 4$ and then decrease for further k values. The Rusipini data is a well used example for low density and well distinguished clusters that shows the effect of $k + 1$ and $k + 2$ or more mapping distances: for instance at $k = 5$ and $k = 6$ only a small number of elements split to form an extra cluster. Eventually, these smaller clusters completely merge to form unified clusters for $k + 3$ and greater distance mappings. The average of trace (similarity) is the most suitable criterion to determine the best $\boldsymbol{K}$ when the estimated number of clusters becomes more settled and stable. In this case, the average traces (similarity) up to $k = 4$ is a maximum using 4 criteria Chapter 4 section 4.3.3 which indicates the best number of clusters and these clusters are fully separated with no overlaps and very stable as the CV value is 0. Plots (d) and (e) show the differences as composite graphs for the behaviour of traces (similarity), overlap, average traces (similarity) and overlap for $k + r$, while (f) indicates the coefficient of variation, at different k and is 0 at the best $\boldsymbol{K}$.

4.6 Conclusion

In this chapter, a new approach is proposed and developed using results obtained by the *k-means* clustering algorithm to determine the best number of clusters. Clustering validation of resultant clusters is an important issue due to lack of predefined cluster information. However, evaluating the clustering results as shown above can help us to gain better understanding of properties of the data. For this purpose, a new approach for clustering evaluation was developed and implemented, with respect to the features and quantities inherited from the set of clusters, such as cluster membership assigned to the elements (observations) by the clustering algorithm.

In the proposed approach the inter cluster forward and backward mapping of the elements between clusters which are adjacent and non-adjacent (moving sequentially away from the adjacent distance) is carried out until the limit maximum number of clusters are reached. Then these forward and backward proportions of mapped elements are combined using the inner product of matrices to obtain combined mapped proportion and elements to determine the traces (similarity), maximum average traces (similarity), overlap, average overlap and coefficient of variation. Additionally, an example for computing and constructing combined mapped proportion and combined mapped elements matrices is given. Finally, the effect of adjacent and non-adjacent mapping distance was demonstrated by an extra example. This approach is the starting of the statistical and probability approach to analyse and determine clusters best number especially for large and complex data.

In the next chapter, the new approach will be applied on a variety of simulated datasets to show its performance and compare results with different existing approaches for clustering.

Chapter 5

Application to Simulated Data

5.1 Introduction

In this chapter three different types of datasets, each including four different cases of simulated datasets, are used to analyse and evaluate the performance of the new approach presented and described in Chapter 4. With the validated index of the new approach was compared with eight other existing and commonly used cluster validation indexes which were discussed in detail in Chapter 3.

The performance of the new approach is demonstrated by using variety of datasets with different clustering structures (e.g. elliptical and spherical (circular) shapes, sizes and densities etc.). In order to visually illustrate and identify the differences between clusters and to determine the best $\boldsymbol{K}$ (estimated number of clusters) scatter plots will be used. First, two different types of datasets will be generated, called **Type1 and Type2,** each with four different cases (**case1, case2, case3, case4**) associated with different settings of the dataset cluster element generating parameter values $(\mu, \sigma\ )$. There are a different number of clusters in each type. In all these cases and for both these dataset types, the value of the spread (σ) between cluster centroids will be gradually increased to form high, high-to-medium, medium-to-low and low density clusters. Second, another kind of simulated dataset will be sourced from a well-known dataset collection called **Type3**, consisting of four different cases (**case1, case2, case3, case4**) representing different structures having several numbers of clusters. In all of these cases the data is simulated in such a way that the desired number of clusters (clustering structure) is known in advance: this will allow us to validate and check the performance and the best cluster numbers when using existing validation approaches and the new approach. The new approach was developed using the R programming language and utilizing different R packages such as fpc and NbClust. The remainder of this chapter is organised as follows: Section 5.2 presents use of simulated data of three different sizes of spherical clusters with four different cases, each with increased variation (σ) between three clusters. Section 5.3 is a similar situation as in section 5.2 but for five spherical clusters of equal sizes. Section 5.4 checks and compares the performance of mixtures of different shapes, sizes and density of clusters. Section 5.5 provides a summary of this chapter.

5.2 Type1 Datasets: Spherical Clusters of Different Sizes

Type1 (fixed centres (μ) with gradually increasing spread(σ)): Generally, it is best to show the performance of an approach using some well-formed datasets with known properties, for better understanding. Type1 datasets consist of $N = 1500$ elements with three spherical and various sized clusters, having different fixed means (μ) in each case while varying densities (increasing standard deviations). The notion of increasing standard deviation checks at what stage the original structure in a dataset will begin to fail in determining a clear clustering structure. All cases are generated with normal distributions which spread out to gradually overlap as the standard deviation increases for three different selected centroids, $\left(\mu_{1(x,y)} = (0.3, 0.3),\ \mu_{2(x,y)} = (0.6, 0.6),\ \mu_{3(x,y)} = (0.9, 0.9)\right)$. The standard deviations $\sigma_1 = 0.03,\ \sigma_2 = 0.05,\ \sigma_3 = 0.11,\ \sigma_4 = 0.30$ were used to give four different cases.

Details of these cases are shown in the Table 5.1.

Case1	***Case2***
$n = 600, \mu_1 = (0.3{,}0.3), \sigma_1 = 0.03$	$n = 600, \mu_1 = (0.3{,}0.3), \sigma_2 = 0.05$
$n = 400, \mu_2 = (0.6{,}0.6), \sigma_1 = 0.03$	$n = 400, \mu_2 = (0.6{,}0.6), \sigma_2 = 0.05$
$n = 500, \mu_3 = (0.9{,}0.9), \sigma_1 = 0.03$	$n = 500, \mu_3 = (0.9{,}0.9), \sigma_2 = 0.05$
Case3	***Case4***
$n = 600, \mu_1 = (0.3{,}0.3), \sigma_3 = 0.11$	$n = 600, \mu_1 = (0.3{,}0.3), \sigma_4 = 0.30$
$n = 400, \mu_2 = (0.6{,}0.6), \sigma_3 = 0.11$	$n = 400, \mu_2 = (0.6{,}0.6), \sigma_4 = 0.30$
$n = 500, \mu_3 = (0.9{,}0.9), \sigma_3 = 0.11$	$n = 500, \mu_3 = (0.9{,}0.9), \sigma_4 = 0.30$

Table 5.1: Type1 dataset with 3 different centroids and 4 standard deviations: (where n = number of elements in each cluster, μ = mean, σ = standard deviation).

Visualization is a useful tool for understanding the relationships between variables particularly for two dimensional datasets. Therefore, to further illustrate the above four cases a set of scatter plots are shown for each case in the figure below.

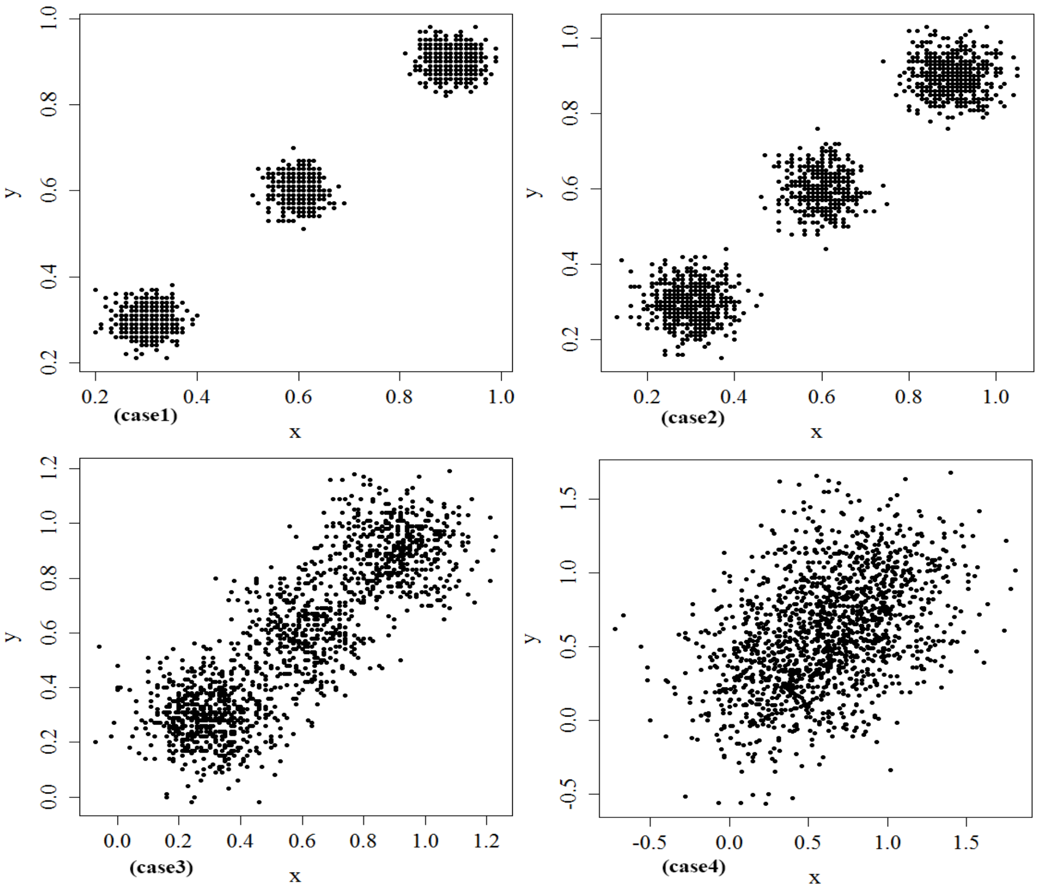


Figure 5.1: Type1 dataset scatter plots for 4 cases.

5.3.1 Case1: High Density Clusters

To analyse the performance of the new approach for case1 of a type1 dataset, the k-means algorithm is used to cluster the dataset to obtain $k = 2,3,\ldots,16$ clusters with different numbers of elements in each. The number of elements in each cluster will be used to construct diagonal $k \times k$ matrices at each k as we can see in the figure below:

$$\boldsymbol{k_2} = \{600,900\}, \boldsymbol{K_3}=\{500,600,400\}, \cdots, \boldsymbol{k_{16}} = \{121,85,\cdots,149\}$$

$$\boldsymbol{k_2} = \begin{array}{c|cc} & \boldsymbol{c_{(2,1)}} & \boldsymbol{c_{(2,2)}} \\ \hline \boldsymbol{c_{(2,1)}} & 600 & 0 \\ \boldsymbol{c_{(2,2)}} & 0 & 900 \end{array}, \boldsymbol{k_3} = \begin{array}{c|ccc} & \boldsymbol{c_{(3,1)}} & \boldsymbol{c_{(3,2)}} & \boldsymbol{c_{(3,3)}} \\ \hline \boldsymbol{c_{(3,1)}} & 500 & 0 & 0 \\ \boldsymbol{c_{(3,2)}} & 0 & 600 & 0 \\ \boldsymbol{c_{(3,3)}} & 0 & 0 & 400 \end{array}, \cdots, \boldsymbol{k_{16}} = \begin{array}{c|cccc} & \boldsymbol{c_{(16,1)}} & \boldsymbol{c_{(16,2)}} & \cdots & \boldsymbol{c_{(16,16)}} \\ \hline \boldsymbol{c_{(16,1)}} & 121 & 0 & \cdots & 0 \\ \boldsymbol{c_{(16,2)}} & 0 & 85 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ \boldsymbol{c_{(16,16)}} & 0 & 0 & \cdots & 149 \end{array}$$

Figure 5.2: Represents clusters sizes on the diagonal from $\boldsymbol{k_2}$ to $\boldsymbol{k_{16}}$.

Now we construct the forward and backward common elements M, their proportion P, and combine mapped proportion O, matrices at each k for different $k + r$ mapping distances. For example, at fixed k and different $k + r$ mapping distances in a sequence the forward mapping of common the elements matrices $M_{(2,\ 2+1)}, M_{(2,\ 2+2)}, \ldots, M_{(2,\ 2+14)}, M_{(3,\ 3+1)}, M_{(3,\ 3+2)}, \ldots, M_{(3,\ 3+13)}, \cdots, M_{(15,\ 15+1)}$ calculated. These $M_{(k,k+r)}$ matrices are converted into corresponding proportion matrices $P_{(k,k+r)}$ with row sum scaling, i.e. $P_{(2,\ 2+1)}, P_{(2,\ 2+2)}, \ldots, P_{(2,\ 2+14)}$, $P_{(3,\ 3+1)}, P_{(3,\ 3+2)}, \ldots, P_{(3,\ 3+13)}$,…, $P_{(15,\ 15+1)}$ are calculated. Similarly, backward mapping $M_{(k+r,k)}$ and proportion $P_{(k+r,k)}$ matrices are constructed. These proportion (P) matrices are used to compute combined mapped proportion (O) matrices by applying the inner product of forward to backward matrices. The diagonal of these O matrices indicates the proportion of elements which are similar within each cluster, while off diagonal elements indicate the proportion of elements which overlap between clusters as described in Chapter 4. The combined mapped elements Q

matrices are achieved using the inner product of the O and $\boldsymbol{k}$ matrices. In the Q matrices diagonal values represent the non-overlapping elements mapped (similarity) within clusters while off diagonal values represent the elements which overlap between clusters. The figure below represents forward and backward M, P, combined mapped O and Q matrices, where $O = P_{(k,k+r)} \times P_{(k+r,k)}$ and $Q = \boldsymbol{k} \times O$, $k = 2,3, \dots 16$, $r = 1,2, \dots 14$.

For $k = 2$ and $r = 1,2, \dots, K - k$ in $k + r$

Forward M and P

Forward *M*

$$M_{23}= \begin{array}{c|ccc} & C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \\ \hline C_{(2,1)} & 0 & 600 & 0 \\ C_{(2,2)} & 500 & 0 & 400 \end{array}$$

$$M_{24}= \begin{array}{c|cccc} & C_{(4,1)} & C_{(4,2)} & C_{(4,3)} & C_{(4,4)} \\ \hline C_{(2,1)} & 319 & 0 & 281 & 0 \\ C_{(2,2)} & 0 & 400 & 0 & 500 \end{array}$$

$$\vdots$$

$$M_{216}= \begin{array}{c|ccccc} & C_{(16,1)} & C_{(16,2)} & \cdots & \cdots & C_{(16,16)} \\ \hline C_{(2,1)} & 0 & 85 & \cdots & \cdots & 149 \\ C_{(2,2)} & 121 & 0 & \cdots & \cdots & 0 \end{array}$$

Forward *P*

$$P_{23}= \begin{array}{c|ccc} & C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \\ \hline C_{(2,1)} & 0.0 & 1.0 & 0.0 \\ C_{(2,2)} & 0.56 & 0.0 & 0.44 \end{array}$$

$$P_{24}= \begin{array}{c|cccc} & C_{(4,1)} & C_{(4,2)} & C_{(4,3)} & C_{(4,4)} \\ \hline C_{(2,1)} & 0.53 & 0.0 & 0.47 & 0.0 \\ C_{(2,2)} & 0 & 0.44 & 0.0 & 0.56 \end{array}$$

$$\vdots$$

$$P_{216}= \begin{array}{c|ccccc} & C_{(16,1)} & C_{(16,2)} & \cdots & \cdots & C_{(16,16)} \\ \hline C_{(2,1)} & 0.0 & 0.14 & \cdots & \cdots & 0.25 \\ C_{(2,2)} & 0.13 & 0.0 & \cdots & \cdots & 0.0 \end{array}$$

Backward M and P

Backward *M*

$$M_{32}= \begin{array}{c|cc} & C_{(2,1)} & C_{(2,2)} \\ \hline C_{(3,1)} & 0 & 500 \\ C_{(3,2)} & 600 & 0 \\ C_{(3,3)} & 0 & 400 \end{array}$$

$$M_{42}= \begin{array}{c|cc} & C_{(2,1)} & C_{(2,2)} \\ \hline C_{(4,1)} & 319 & 0 \\ C_{(4,2)} & 0 & 400 \\ C_{(4,3)} & 281 & 0 \\ C_{(4,4)} & 0 & 500 \end{array}$$

$$\vdots$$

$$M_{162}= \begin{array}{c|cc} & C_{(2,1)} & C_{(2,2)} \\ \hline C_{(16,1)} & 0 & 121 \\ C_{(16,2)} & 85 & 0 \\ \vdots & \vdots & \vdots \\ C_{(16,16)} & 149 & 0 \end{array}$$

Backward *P*

$$P_{32}= \begin{array}{c|cc} & C_{(2,1)} & C_{(2,2)} \\ \hline C_{(3,1)} & 0.0 & 1.0 \\ C_{(3,2)} & 1.0 & 0.0 \\ C_{(3,3)} & 0.0 & 1.0 \end{array}$$

$$P_{42}= \begin{array}{c|cc} & C_{(2,1)} & C_{(2,2)} \\ \hline C_{(4,1)} & 1.0 & 0.0 \\ C_{(4,2)} & 0.0 & 1.0 \\ C_{(4,3)} & 1.0 & 0.0 \\ C_{(4,4)} & 0.0 & 1.0 \end{array}$$

$$\vdots$$

$$P_{162}= \begin{array}{c|cc} & C_{(2,1)} & C_{(2,2)} \\ \hline C_{(16,1)} & 0.0 & 1.0 \\ C_{(16,2)} & 1.0 & 0.0 \\ \vdots & \vdots & \vdots \\ C_{(16,16)} & 1.0 & 0.0 \end{array}$$

***k*+*r* Forward and Backward movement *M* and *P* proportion matrices , when *k*=2 and 1 ≤ *r* ≥ 14**

O and Q

$$P_{23} \times P_{32} = O_{(2,2)}= \begin{array}{c|cc} & C_{(2,1)} & C_{(2,2)} \\ \hline C_{(2,1)} & 1.0 & 0.0 \\ C_{(2,2)} & 0.0 & 1.0 \end{array} \qquad k_2 \times O_{(2,2)}= Q_{(2,2)} = \begin{array}{c|cc} & C_{(2,1)} & C_{(2,2)} \\ \hline C_{(2,1)} & 600 & 0 \\ C_{(2,2)} & 0 & 900 \end{array}$$

$$P_{24} \times P_{42} = O_{(2,2)}= \begin{array}{c|cc} & C_{(2,1)} & C_{(2,2)} \\ \hline C_{(2,1)} & 1.0 & 0.0 \\ C_{(2,2)} & 0.0 & 1.0 \end{array} \qquad k_2 \times O_{(2,2)}= Q_{(2,2)} = \begin{array}{c|cc} & C_{(2,1)} & C_{(2,2)} \\ \hline C_{(2,1)} & 600 & 0 \\ C_{(2,2)} & 0 & 900 \end{array}$$

$$\vdots \qquad\qquad \vdots$$

$$P_{216} \times P_{162} = O_{(2,2)}= \begin{array}{c|cc} & C_{(2,1)} & C_{(2,2)} \\ \hline C_{(2,1)} & 1.0 & 0.0 \\ C_{(2,2)} & 0.0 & 1.0 \end{array} \qquad k_2 \times O_{(2,2)}= Q_{(2,2)} = \begin{array}{c|cc} & C_{(2,1)} & C_{(2,2)} \\ \hline C_{(2,1)} & 600 & 0 \\ C_{(2,2)} & 0 & 900 \end{array}$$

***O* Combined proportion matrices at *k*=2** **_Q_ Combined elements matrices at *k*=2**

Figure 5.3: Summary of elements for forward and backward mapping, proportion and combined matrices.

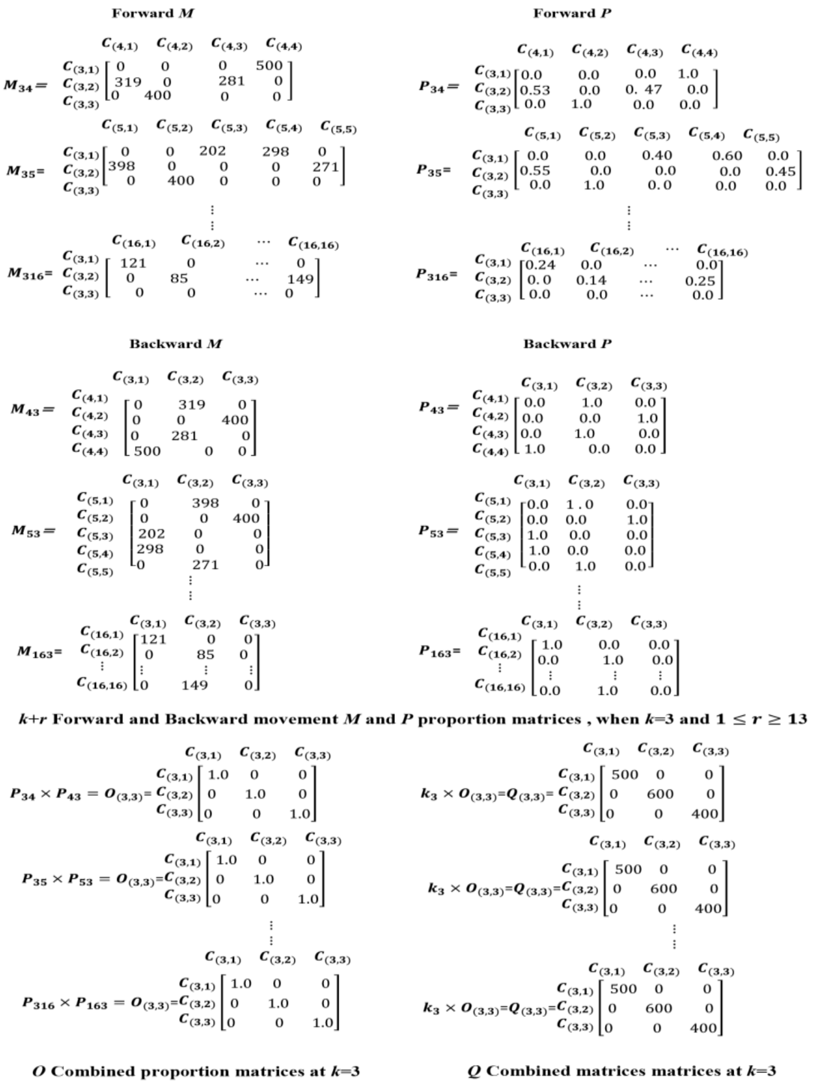
The Figure 5.3 above shows for inter cluster forward and backward mapping of common elements in each forward mapping clusters split while backward mapping involves merges (collapses) as seen in M and P matrices. The figure also represents that for combined mapped elements Q matrices, the number of elements on the diagonal are 600 and 900 while on the off diagonal there are zero elements. This shows that clusters are mapped with no overlaps when $k = 2$ and $r \in (1,2, \dots, K - k)$. Furthermore, to compute forward and backward M, P, O and Q matrices at each k for different $k + r$ mapping distances we repeated the process for different k and $k + r$. The forward and backward M, proportion P, and combined mapped O and Q at $k = 3$ and $r = 1,2, \dots, K - k$, $k = 14$ and $r = 1,2, \dots, K - k$, $k = 15$ and $r = K - k$ with different $k + r$ can be seen in the Figure 5.4.

For $k = 3$ and $r = 1,2, \dots , K - k$ in $k + r$

Forward M and P

Backward M and P

O and Q



Forward *M*

$$M_{34}= \begin{array}{c|cccc} & C_{(4,1)} & C_{(4,2)} & C_{(4,3)} & C_{(4,4)} \\ \hline C_{(3,1)} & 0 & 0 & 0 & 500 \\ C_{(3,2)} & 319 & 0 & 281 & 0 \\ C_{(3,3)} & 0 & 400 & 0 & 0 \end{array}$$

$$M_{35}= \begin{array}{c|ccccc} & C_{(5,1)} & C_{(5,2)} & C_{(5,3)} & C_{(5,4)} & C_{(5,5)} \\ \hline C_{(3,1)} & 0 & 0 & 202 & 298 & 0 \\ C_{(3,2)} & 398 & 0 & 0 & 0 & 271 \\ C_{(3,3)} & 0 & 400 & 0 & 0 & 0 \end{array}$$

$\vdots$

$$M_{316}= \begin{array}{c|cccc} & C_{(16,1)} & C_{(16,2)} & \cdots & C_{(16,16)} \\ \hline C_{(3,1)} & 121 & 0 & \cdots & 0 \\ C_{(3,2)} & 0 & 85 & \cdots & 149 \\ C_{(3,3)} & 0 & 0 & \cdots & 0 \end{array}$$

Forward *P*

$$P_{34}= \begin{array}{c|cccc} & C_{(4,1)} & C_{(4,2)} & C_{(4,3)} & C_{(4,4)} \\ \hline C_{(3,1)} & 0.0 & 0.0 & 0.0 & 1.0 \\ C_{(3,2)} & 0.53 & 0.0 & 0.\ 47 & 0.0 \\ C_{(3,3)} & 0.0 & 1.0 & 0.0 & 0.0 \end{array}$$

$$P_{35}= \begin{array}{c|ccccc} & C_{(5,1)} & C_{(5,2)} & C_{(5,3)} & C_{(5,4)} & C_{(5,5)} \\ \hline C_{(3,1)} & 0.0 & 0.0 & 0.40 & 0.60 & 0.0 \\ C_{(3,2)} & 0.55 & 0.0 & 0.0 & 0.0 & 0.45 \\ C_{(3,3)} & 0.0 & 1.0 & 0.\ 0 & 0.0 & 0.0 \end{array}$$

$\vdots$

$$P_{316}= \begin{array}{c|cccc} & C_{(16,1)} & C_{(16,2)} & \cdots & C_{(16,16)} \\ \hline C_{(3,1)} & 0.24 & 0.0 & \cdots & 0.0 \\ C_{(3,2)} & 0.\ 0 & 0.14 & \cdots & 0.25 \\ C_{(3,3)} & 0.0 & 0.0 & \cdots & 0.0 \end{array}$$

Backward *M*

$$M_{43}= \begin{array}{c|ccc} & C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \\ \hline C_{(4,1)} & 0 & 319 & 0 \\ C_{(4,2)} & 0 & 0 & 400 \\ C_{(4,3)} & 0 & 281 & 0 \\ C_{(4,4)} & 500 & 0 & 0 \end{array}$$

$$M_{53}= \begin{array}{c|ccc} & C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \\ \hline C_{(5,1)} & 0 & 398 & 0 \\ C_{(5,2)} & 0 & 0 & 400 \\ C_{(5,3)} & 202 & 0 & 0 \\ C_{(5,4)} & 298 & 0 & 0 \\ C_{(5,5)} & 0 & 271 & 0 \end{array}$$

$\vdots$

$$M_{163}= \begin{array}{c|ccc} & C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \\ \hline C_{(16,1)} & 121 & 0 & 0 \\ C_{(16,2)} & 0 & 85 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ C_{(16,16)} & 0 & 149 & 0 \end{array}$$

Backward *P*

$$P_{43}= \begin{array}{c|ccc} & C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \\ \hline C_{(4,1)} & 0.0 & 1.0 & 0.0 \\ C_{(4,2)} & 0.0 & 0.0 & 1.0 \\ C_{(4,3)} & 0.0 & 1.0 & 0.0 \\ C_{(4,4)} & 1.0 & 0.0 & 0.0 \end{array}$$

$$P_{53}= \begin{array}{c|ccc} & C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \\ \hline C_{(5,1)} & 0.0 & 1\ .\ 0 & 0.0 \\ C_{(5,2)} & 0.0 & 0.0 & 1.0 \\ C_{(5,3)} & 1.0 & 0.0 & 0.0 \\ C_{(5,4)} & 1.0 & 0.0 & 0.0 \\ C_{(5,5)} & 0.0 & 1.0 & 0.0 \end{array}$$

$\vdots$

$$P_{163}= \begin{array}{c|ccc} & C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \\ \hline C_{(16,1)} & 1.0 & 0.0 & 0.0 \\ C_{(16,2)} & 0.0 & 1.0 & 0.0 \\ \vdots & \vdots & \vdots & \vdots \\ C_{(16,16)} & 0.0 & 1.0 & 0.0 \end{array}$$

***k+r* Forward and Backward movement *M* and *P* proportion matrices , when *k*=3 and 1 ≤ *r* ≥ 13**

$$P_{34} \times P_{43} = O_{(3,3)} = \begin{array}{c|ccc} & C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \\ \hline C_{(3,1)} & 1.0 & 0 & 0 \\ C_{(3,2)} & 0 & 1.0 & 0 \\ C_{(3,3)} & 0 & 0 & 1.0 \end{array}$$

$$P_{35} \times P_{53} = O_{(3,3)} = \begin{array}{c|ccc} & C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \\ \hline C_{(3,1)} & 1.0 & 0 & 0 \\ C_{(3,2)} & 0 & 1.0 & 0 \\ C_{(3,3)} & 0 & 0 & 1.0 \end{array}$$

$\vdots$

$$P_{316} \times P_{163} = O_{(3,3)} = \begin{array}{c|ccc} & C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \\ \hline C_{(3,1)} & 1.0 & 0 & 0 \\ C_{(3,2)} & 0 & 1.0 & 0 \\ C_{(3,3)} & 0 & 0 & 1.0 \end{array}$$

$$k_3 \times O_{(3,3)} = Q_{(3,3)} = \begin{array}{c|ccc} & C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \\ \hline C_{(3,1)} & 500 & 0 & 0 \\ C_{(3,2)} & 0 & 600 & 0 \\ C_{(3,3)} & 0 & 0 & 400 \end{array}$$

$$k_3 \times O_{(3,3)} = Q_{(3,3)} = \begin{array}{c|ccc} & C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \\ \hline C_{(3,1)} & 500 & 0 & 0 \\ C_{(3,2)} & 0 & 600 & 0 \\ C_{(3,3)} & 0 & 0 & 400 \end{array}$$

$\vdots$

$$k_3 \times O_{(3,3)} = Q_{(3,3)} = \begin{array}{c|ccc} & C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \\ \hline C_{(3,1)} & 500 & 0 & 0 \\ C_{(3,2)} & 0 & 600 & 0 \\ C_{(3,3)} & 0 & 0 & 400 \end{array}$$

***O* Combined proportion matrices at *k*=3**

***Q* Combined matrices matrices at *k*=3**

⋮ ⋮ ⋮ ⋮

⋮ ⋮ ⋮ ⋮

⋮ ⋮ ⋮ ⋮

For $k = 14$ and $r = 1,2, \dots , K - k$ in $k + r$

Forward M and P

Forward $\boldsymbol{M}$

$$\boldsymbol{M_{1415}} = \begin{array}{ccccc} & C_{(15,1)} & C_{(15,2)} & \cdots & C_{(15,15)} \\ C_{(14,1)} & 0 & 0 & \cdots & 0 \\ C_{(14,2)} & 0 & 27 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ C_{(14,14)} & 0 & 66 & \cdots & 0 \end{array}$$

Forward $\boldsymbol{P}$

$$\boldsymbol{P_{1415}} = \begin{array}{ccccc} & C_{(15,1)} & C_{(15,2)} & \cdots & C_{(15,15)} \\ C_{(14,1)} & 0.0 & 0.0 & \cdots & 0.0 \\ C_{(14,2)} & 0.0 & 0.22 & \cdots & 0.0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ C_{(14,14)} & 0.0 & 0.47 & \cdots & 0 \end{array}$$

$$\boldsymbol{M_{1416}} = \begin{array}{ccccc} & C_{(16,1)} & C_{(16,2)} & \cdots & C_{(16,16)} \\ C_{(14,1)} & 0 & 0 & \cdots & 0 \\ C_{(14,2)} & 0 & 61 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ C_{(14,14)} & 0 & 0 & \cdots & 0 \end{array}$$

$$\boldsymbol{P_{1416}} = \begin{array}{ccccc} & C_{(16,1)} & C_{(16,2)} & \cdots & C_{(16,16)} \\ C_{(14,1)} & 0.0 & 0.0 & \cdots & 0.0 \\ C_{(14,2)} & 0.0 & 0.49 & \cdots & 0.0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ C_{(14,14)} & 0.0 & 0.0 & \cdots & 0.0 \end{array}$$

Backward M and P

Backward $\boldsymbol{M}$

$$\boldsymbol{M_{1514}} = \begin{array}{ccccc} & C_{(14,1)} & C_{(14,2)} & \cdots & C_{(14,14)} \\ C_{(15,1)} & 0 & 0 & \cdots & 0 \\ C_{(15,2)} & 0 & 27 & \cdots & 66 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ C_{(15,15)} & 0 & 0 & \cdots & 0 \end{array}$$

Backward $\boldsymbol{P}$

$$\boldsymbol{P_{1514}} = \begin{array}{ccccc} & C_{(14,1)} & C_{(14,2)} & \cdots & C_{(14,14)} \\ C_{(15,1)} & 0.0 & 0.0 & \cdots & 0.0 \\ C_{(15,2)} & 0.0 & 0.12 & \cdots & 0.29 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ C_{(15,15)} & 0.0 & 0.0 & \cdots & 0.0 \end{array}$$

$$\boldsymbol{M_{1614}} = \begin{array}{ccccc} & C_{(14,1)} & C_{(14,2)} & \cdots & C_{(14,14)} \\ C_{(16,1)} & 0 & 0 & \cdots & 0 \\ C_{(16,2)} & 0 & 61 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ C_{(16,14)} & 0 & 0 & \cdots & 0 \end{array}$$

$$\boldsymbol{P_{1614}} = \begin{array}{ccccc} & C_{(14,1)} & C_{(14,2)} & \cdots & C_{(14,14)} \\ C_{(16,1)} & 0.0 & 0.0 & \cdots & 0.0 \\ C_{(16,2)} & 0.0 & 0.43 & \cdots & 0.0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ C_{(16,14)} & 0.\ 0 & 0.0 & \cdots & 0.0 \end{array}$$

***k*+*r* Forward and Backward movement *M* and *P* proportion matrices , when *k*=14 and $1 \leq r \geq 2$**

O and Q

$$\boldsymbol{P_{1415}} \times \boldsymbol{P_{1514}} = \boldsymbol{O_{(14,14)}} = \begin{array}{ccccc} & C_{(14,1)} & C_{(14,2)} & \cdots & C_{(14,14)} \\ C_{(14,1)} & 0.\ 82 & 0.0 & \cdots & 0.0 \\ C_{(14,2)} & 0.0 & 0.47 & \cdots & 0.4 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ C_{(14,14)} & 0.0 & 0.35 & \cdots & 0.36 \end{array}$$

$$\boldsymbol{k_{14}} \times \boldsymbol{O_{(14,14)}} = \boldsymbol{Q_{(14,14)}} = \begin{array}{ccccc} & C_{(14,1)} & C_{(14,2)} & \cdots & C_{(14,14)} \\ C_{(14,1)} & 163.18 & 0 & \cdots & 0 \\ C_{(14,2)} & 0 & 58.\ 28 & \cdots & 55.6 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ C_{(14,14)} & 0 & 44.60 & \cdots & 50.04 \end{array}$$

$$\boldsymbol{P_{1416}} \times \boldsymbol{P_{1614}} = \boldsymbol{O_{(14,14)}} = \begin{array}{ccccc} & C_{(14,1)} & C_{(14,2)} & \cdots & C_{(14,14)} \\ C_{(14,1)} & 0.84 & 0.0 & \cdots & 0.0 \\ C_{(14,2)} & 0.0 & 0.48 & \cdots & 0.24 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ C_{(14,14)} & 0.0 & 0.21 & 0 & 0.54 \end{array}$$

$$\boldsymbol{k_{14}} \times \boldsymbol{O_{(14,14)}} = \boldsymbol{Q_{(14,14)}} = \begin{array}{ccccc} & C_{(14,1)} & C_{(14,2)} & \cdots & C_{(14,14)} \\ C_{(14,1)} & 167.16 & 0.0 & \cdots & 0.0 \\ C_{(14,2)} & 0.0 & 59.52 & \cdots & 33.36 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ C_{(14,14)} & 0.0 & 26.\ 04 & \cdots & 75.06 \end{array}$$

***O* Combined proportion matrices at *k*=14**

***Q* Combined elements matrices at *k*=14**

For $k = 15$ and $r = K - k = 16 - 15 = 1$ in $k + r$

Forward

Forward *M*

$$\boldsymbol{M_{1516}} = \begin{array}{ccccc} & C_{(16,1)} & C_{(16,2)} & \cdots & C_{(16,16)} \\ C_{(15,1)} & 0 & 0 & \cdots & 0 \\ C_{(15,2)} & 0 & 107 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ C_{(15,15)} & 125 & 0 & \cdots & 0 \end{array}$$

Forward *P*

$$\boldsymbol{P_{1516}} = \begin{array}{ccccc} & C_{(16,1)} & C_{(16,2)} & \cdots & C_{(16,16)} \\ C_{(15,1)} & 0.0 & 0.0 & \cdots & 0.0 \\ C_{(15,2)} & 0.0 & 0.47 & \cdots & 0.0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ C_{(15,15)} & 1.0 & 0.0 & \cdots & 0.0 \end{array}$$

Backward

Backward *M*

$$\boldsymbol{M_{1615}} = \begin{array}{ccccc} & C_{(15,1)} & C_{(15,2)} & \cdots & C_{(15,15)} \\ C_{(16,1)} & 0 & 0 & \cdots & 125 \\ C_{(16,2)} & 0 & 107 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ C_{(16,16)} & 0 & 0 & \cdots & 0 \end{array}$$

Backward *P*

$$\boldsymbol{P_{1615}} = \begin{array}{ccccc} & C_{(15,1)} & C_{(15,2)} & \cdots & C_{(15,15)} \\ C_{(16,1)} & 0.0 & 0.0 & \cdots & 1.0 \\ C_{(16,2)} & 0.0 & 0.76 & \cdots & 0.0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ C_{(16,16)} & 0.0 & 0.0 & \cdots & 0.0 \end{array}$$

***k*+*r* Forward and Backward movement *M* and *P* proportion matrices , when *k*=15 and *r* =1**

O and Q

$$\boldsymbol{P_{1516}} \times \boldsymbol{P_{161}} = \boldsymbol{O_{(15,15)}} = \begin{array}{ccccc} & C_{(15,1)} & C_{(15,2)} & \cdots & C_{(15,15)} \\ C_{(15,1)} & 0.0 & 0.\ 0 & \cdots & 0.\ 0 \\ C_{(15,2)} & 0.82 & 0.18 & \cdots & 0.0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ C_{(15,15)} & 0.0 & 0.0 & \cdots & 1.0 \end{array}$$

$$\boldsymbol{k_{15}} \times \boldsymbol{O_{(15,15)}} = \boldsymbol{Q_{(15,15)}} = \begin{array}{ccccc} & C_{(15,1)} & C_{(15,2)} & \cdots & C_{(15,15)} \\ C_{(15,1)} & 134 & 0 & \cdots & 0 \\ C_{(15,2)} & 0 & 188.6 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ C_{(15,15)} & 0 & 0 & \cdots & 125 \end{array}$$

***O* Combined proportion matrices *k*=15**

***Q* Combined elements matrices at *k*=15**

Figure 5.4: Forward/backward M, P and combined mapped O, Q matrices, when $k = 3$ and $r = 1,2,.., K - k$, ..., $k = 15$ and $r = K - k = 1$.

The next step is to calculate the similarity (traces), the amount of overlapping elements (off diagonal sum) for each k with different $k + r$. Finally, we computed average similarity (traces), average overlap and coefficient of variation (CV) at different k. All these calculations from Q matrices are based on the definitions given in Chapter 4 section 4.3. The results details are shown in the table below;

	k	r	$(k,k+r)\times(k+r,k)$	***Traces (similarity)***	***Overlap***
$k=2,r=1,2,\dots,K-k$	2	1	(2,3×3,2)	**1500**	**0**
	2	2	(2,4×4,2)	**1500**	**0**
	2	3	(2,5×5,2)	**1500**	**0**
	2	4	(2,6×6,2)	**1500**	**0**
	2	5	(2,7×7,2)	**1500**	**0**
	2	6	(2,8×8,2)	**1500**	**0**
	2	7	(2,9×9,2)	**1500**	**0**
	2	8	(2,10×10,2)	**1500**	**0**
	2	9	(2,11×11,2)	**1500**	**0**
	2	10	(2,12×12,2)	**1500**	**0**
	2	11	(2,13×13,2)	**1500**	**0**
	2	12	(2,14×14,2)	**1500**	**0**
	2	13	(2,15×15,2)	**1500**	**0**
	2	14	(2,16×16,2)	**1500**	**0**
	k	r	$(k,k+r)\times(k+r,k)$	***Traces (similarity)***	***Overlap***
$k=3,r=1,2,\dots,K-k$	3	1	(3,4×4,3)	**1500**	**0**
	3	2	(3,5×5,3)	**1500**	**0**
	3	3	(3,6×6,3)	**1500**	**0**
	3	4	(3,7×7,3)	**1500**	**0**
	3	5	(3,8×8,3)	**1500**	**0**
	3	6	(3,9×9,3)	**1500**	**0**
	3	7	(3,10×10,3)	**1500**	**0**
	3	8	(3,11×11,3)	**1500**	**0**
	3	9	(3,12×12,3)	**1500**	**0**
	3	10	(3,13×13,3)	**1500**	**0**
	3	11	(3,14×14,3)	**1500**	**0**
	3	12	(3,15×15,3)	**1500**	**0**
	3	13	(3,16×16,3)	**1500**	**0**
	⋮	⋮	⋮	⋮	⋮
	k	r	$(k,k+r)\times(k+r,k)$	***Traces (similarity)***	***Overlap***
$k=15,r=1$	15	1	(15,16×16,15)	**896.79**	**603.21**

(a)

k	$(k,k+r)\times(k+r,k)$	***Traces (similarity)***	***Overlap***	
2	(2,3×3,2)	**1500**	**0**	$k=2,3,\dots15,r=1$
3	(3,4×4,3)	**1500**	**0**	
4	(4,5×5,4)	**1482**	**18**	
5	(5,6×6,5)	**1398.48**	**101.52**	
6	(6,7×7,6)	**1231.18**	**268.82**	
7	(7,8×8,7)	**1303.69**	**196.31**	
8	(8,9×9,8)	**1302.74**	**197.26**	
9	(9,10×10,9)	**1151.03**	**348.97**	
10	(10,11×11,10)	**1116.63**	**383.37**	
11	(11,12×12,11)	**1012.01**	**487.99**	
12	(12,13×13,12)	**926.65**	**573.35**	
13	(13,14×14,13)	**934.96**	**565.04**	
14	(14,15×15,14)	**991.32**	**508.68**	
15	(15,16×16,15)	**896.79**	**603.21**	
k	$(k,k+r)\times(k+r,k)$	***Traces (similarity)***	***Overlap***	
2	(2,4×4,2)	**1500**	**0**	$k=2,3,\dots15,r=2$
3	(3,5×5,3)	**1500**	**0**	
4	(4,6×6,4)	**1440.38**	**59.62**	
5	(5,7×7,5)	**1413.87**	**86.13**	
6	(6,8×8,6)	**1146.81**	**353.19**	
7	(7,9×9,7)	**1249.34**	**250.66**	
8	(8,10×10,8)	**1168.41**	**331.59**	
9	(9,11×11,9)	**1031.01**	**468.99**	
10	(10,12×12,10)	**995.75**	**504.25**	
11	(11,13×13,11)	**908.42**	**591.58**	
12	(12,14×14,12)	**1116.7**	**383.3**	
13	(13,15×15,13)	**1017.7**	**482.3**	
14	(14,16×16,14)	**931.01**	**568.99**	
⋮	⋮	⋮	⋮	
k	$(k,k+r)\times(k+r,k)$	***Traces (similarity)***	***Overlap***	
2	(2,16×16,2)	**1500**	**0**	$k=2,r=14$

(b)

Similarity				***Overlap***			
K	***Max Trace***	***Min Trace***	***Average Traces***	***CV***	***Max Overlap***	***Min Overlap***	***Average Overlap***
2	1500	1500	**1500**	**0**	0	0	**0**
3	1500	1500	**1500**	**0**	0	0	**0**
4	1482	1365.57	**1399.912**	**0.024**	134.43	18	**100.088**
5	1413.87	1286.02	**1330.836**	**0.031**	213.98	86.13	**169.164**
6	1372.02	1136.97	**1213.149**	**0.061**	363.03	127.98	**286.851**
7	1316.08	1106.05	**1216.572**	**0.057**	393.95	183.92	**283.428**
8	1302.74	1029.1	**1142.757**	**0.069**	470.9	197.26	**357.242**
9	1161.92	1020.14	**1078.596**	**0.052**	479.86	338.08	**421.404**
10	1164.71	995.75	**1058.65**	**0.063**	504.25	335.29	**441.35**
11	1012.01	908.42	**957.084**	**0.039**	591.58	487.99	**542.916**
12	1116.7	926.65	**1024.215**	**0.076**	573.35	383.3	**475.785**
13	1017.7	923.69	**958.783**	**0.054**	576.31	482.3	**541.217**
14	991.32	931.01	961.165	0.044	568.99	508.68	538.835

(c)

Table 5.2: Summary of the values computed from Q matrices for case1.

In Table 5.2: (a) represents similarity and overlap at fixed k for different $k+r$, (b) represents the similarity and overlap at different k with fixed $k+r$, and (c) shows the values of maximum, minimum similarity and overlap, average similarity and overlap with coefficient of variation (CV). Since the average similarity at $k=2$ and

$k = 3$ equals the maximum possible and the average similarity is equal to $N = 1500$ (total number of elements), there is no overlap (0 elements overlap) and so the clusters at $k = 2$ and 3 are fully separated. Hence the best number of clusters is three as in the situation of more than one maximum average peak the last maximum peak will indicate optimal number of clusters (mentioned in Chapter 4 section 4.3.3 with criteria 4). For $k = 2$ the average similarity is the maximum possible and this indicates there is potential to split the clusters and in fact the clusters keep splitting until $k = 3$. Average similarity for further k values decreased strongly indicating that clusters are overlapping, which can be seen in the Figure 5.5 (b) with the black solid line.

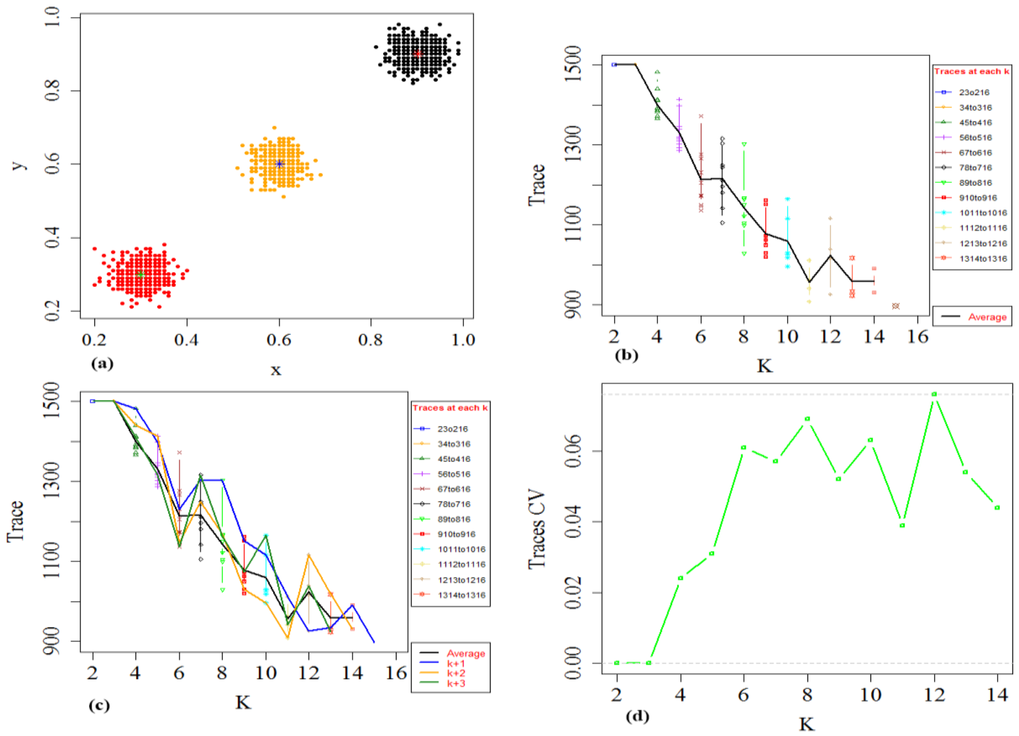


Figure 5.5: Plot (a) represents *k-means* clusters with elements labelled by three different colours for each cluster. Plots (b) - (d) show similarity, average similarity, average overlap and CV values respectively.

In the above figure plot (a) identifies three clusters with their centroids in different colours obtained by applying the *k-means* algorithm. These are fully separated clusters. Plot (b) shows the values of similarities from Table 5.2 while the legend beside the figure indicates similarities in different colours at different k. The black solid line is the average similarity at different k from Table 5.2 (c) and is maximum until $k = 3$ and decreasing beyond that k. Plot (c) of the figure is a composite graph from Table 5.2 that shows difference between similarities at different k for only $k + 1$, $k + 2$ and $k + 3$ and the average similarity in various colours. Plot (d) represents the coefficient of variation (CV) is zero for $k = 3$ which indicate clusters are stable at the best $\boldsymbol{K}$. In the above Table 5.3(a) represents the values of each index from $k = 2$ to $k = 13$ with the optimal number of clusters in bold values. Table 5.3 (b) summarises the optimal number of clusters for each index and shows the optimal number of clusters varies with different indexes and suggests $k = 2$ to $k = 4$ based on 5 runs.

1^{st} run:	***Case1:* Existing indexes values**									
K	***Dunn***	***DH***	***DH Critical***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***Gap Critical***	***CCC***
2	**0.471**	0.013	0.604	5545.984	0.702	**0.388**	15.119	0.693	-2.379	72.553
3	0.004	**0.667**	0.581	2801.287	**0.873**	1.109	**6.261**	0.329	-2.438	**123.635**
4	0.015	1.734	0.538	**41365.16**	0.685	0.808	48.728	**2.772**	0.044	102.01
5	0.007	2.393	0.513	35069.96	0.458	1.163	53.764	2.721	0.101	90.569
6	0.007	2.103	0.496	33542.57	0.501	1.043	55.304	2.569	-0.067	81.629
7	0.007	0.302	0.558	29883.97	0.54	1.074	65.419	2.581	0.275	69.302
8	0.007	2.625	0.49	26002.47	0.5	1.197	42.677	2.588	0.106	71.223
9	0.006	2.04	0.504	29178.7	0.46	0.961	60.217	2.497	-0.086	64.299
10	0.007	0.841	0.496	28604.16	0.315	1.1	85.559	2.487	-0.024	68.088
11	0.011	2.669	0.397	24280.7	0.317	1.085	114.785	2.565	0.018	63.047
12	0.008	0.809	0.486	28015.18	0.311	1.057	62.407	2.549	0.463	62.948
13	0.011	2.57	0.402	27179	0.316	1.065	60.92	2.558	0.038	61.307

(a)

Number of runs		***Dunn***	***DH***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***CCC***
2nd	***K***	**3**	**2**	**4**	**2**	**3**	**2**	**4**	**4**
	Values	**1.151**	**1.874**	**42463.08**	**0.702**	**0.199**	**20.161**	**2.761**	**103.3**
3rd	***K***	**3**	**3**	**3**	**3**	**3**	**2**	**3**	**3**
	Values	**1.151**	**0.998**	**55298.21**	**0.873**	**0.199**	**13.926**	**3.108**	**123.635**
4th	***K***	**2**	**2**	**3**	**3**	**3**	**3**	**4**	**4**
	Values	**0.471**	**45.419**	**55298.21**	**0.873**	**0.199**	**6.539**	**2.79**	**103.3**
5th	***K***	**3**	**3**	**3**	**4**	**3**	**3**	**3**	**3**
	Values	**1.151**	**1.598**	**55298.21**	**0.724**	**0.199**	**6.532**	**3.102**	**123.635**

(b)

Table 5.3: The summary of eight different existing indexes with 5 simulated runs. The optimal number of clusters is highlighted with bold values.

Even though the indexes for a few runs suggest fairly similar numbers of clusters, clearly there is not complete agreement about the best $\boldsymbol{K}$. As discussed in Chapter 4 section 4.1 the *k-means* algorithm is sensitive to its initial nominated centroid and so different choices may provide inconsistent results. Our approach overcomes this dependency on initial centroid choice when mapping common elements from adjacent to non-adjacent clusters (sequential mapping away from adjacent distances). Furthermore, our approach also determines the best $\boldsymbol{K}$ and gives various details such as degree of separation, overlap, fully separated and a stable set of clusters at the best $\boldsymbol{K}$. None of the other indexes provide such detailed information associated with the choice of the best $\boldsymbol{K}$.

5.3.2 Case2: High-to-Medium Density Clusters

The case2 dataset includes more spread than the previous case: in Figure 5.1 see the scatter plot for case2. Table 5.4(a) in the figure below represents the traces (similarity) and overlap at fixed k for different $k + r$ mapping distance. It shows how much change in traces occur at $k = 2$ for different $k + r$ e.g. different traces values. These traces are less than $N = 1500$ (total number of elements) and indicate clusters overlap as compared to case1 at $k = 2$ (see Table 5.2 (a)). In Table 5.4 (a) $k = 3$ traces are higher and equal to $N = 1500$ than the $k = 2$ traces. This shows the effect of spread and represents cluster overlap for case2 as compared to case1 (see the differences at $k = 2$ for case1 and case2, Table 5.2 (a) and Table 5.4 (a)). Thus, at $k = 3$ the average trace is a maximum where the set of clusters are more stable (no change in CV value occurs). The new approach when using criterion 3 mentioned in Chapter 4 section 4.3.3 gives the best $\boldsymbol{K}$ to be 3. Table 5.4(b) shows the number of elements at different k with fixed $k + r$ distance, and at $k = 3$ the trace is greater than all other k values. Table 5.4(c) presents different values of similarity and overlap with coefficient of variation at different k values.

	k	r	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***
$k = 2, r = 1,2, \dots, K-k$	2	1	(2,3×3,2)	**1470**	**30**
	2	2	(2,4×4,2)	**1470**	**30**
	2	3	(2,5×5,2)	**1470**	**30**
	2	4	(2,6×6,2)	**1470**	**30**
	2	5	(2,7×7,2)	**1470**	**30**
	2	6	(2,8×8,2)	**1470**	**30**
	2	7	(2,9×9,2)	**1470**	**30**
	2	8	(2,10×10,2)	**1478.85**	**21.15**
	2	9	(2,11×11,2)	**1478.85**	**21.15**
	2	10	(2,12×12,2)	**1478.85**	**21.15**
	2	11	(2,13×13,2)	**1478.85**	**21.15**
	2	12	(2,14×14,2)	**1478.85**	**21.15**
	2	13	(2,15×15,2)	**1470**	**30**
	2	14	(2,16×16,2)	**1470**	**30**
	k	r	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***
$k = 3, r = 1,2, \dots, K-k$	3	1	(3,4×4,3)	**1500**	**0**
	3	2	(3,5×5,3)	**1500**	**0**
	3	3	(3,6×6,3)	**1500**	**0**
	3	4	(3,7×7,3)	**1500**	**0**
	3	5	(3,8×8,3)	**1500**	**0**
	3	6	(3,9×9,3)	**1500**	**0**
	3	7	(3,10×10,3)	**1500**	**0**
	3	8	(3,11×11,3)	**1500**	**0**
	3	9	(3,12×12,3)	**1500**	**0**
	3	10	(3,13×13,3)	**1500**	**0**
	3	11	(3,14×14,3)	**1500**	**0**
	3	12	(3,15×15,3)	**1500**	**0**
	3	13	(3,16×16,3)	**1500**	**0**
	⋮	⋮	⋮	⋮	⋮
	k	r	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***
$k = 15, r = 1$	15	1	(15,16×16,15)	**1055.39**	**444.61**

(a)

k	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***	
2	(2,3×3,2)	**1470**	**30**	$k = 2,3,..15, r = 1$
3	(3,4×4,3)	**1500**	**0**	
4	(4,5×5,4)	**1432.26**	**67.74**	
5	(5,6×6,5)	**1393.61**	**106.39**	
6	(6,7×7,6)	**1293.74**	**206.26**	
7	(7,8×8,7)	**1182.63**	**317.37**	
8	(8,9×9,8)	**1142.4**	**357.6**	
9	(9,10×10,9)	**1055.36**	**444.64**	
10	(10,11×11,10)	**1121.99**	**378.01**	
11	(11,12×12,11)	**1091.34**	**408.66**	
12	(12,13×13,12)	**1052.36**	**447.64**	
13	(13,14×14,13)	**933.66**	**566.34**	
14	(14,15×15,14)	**1026.91**	**473.09**	
15	(15,16×16,15)	**1055.39**	**444.61**	
k	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***	
2	(2,4×4,2)	**1470**	**30**	$k = 2,3,..15, r = 2$
3	(3,5×5,3)	**1500**	**0**	
4	(4,6×6,4)	**1370.52**	**129.48**	
5	(5,7×7,5)	**1373.93**	**126.07**	
6	(6,8×8,6)	**1205.21**	**294.79**	
7	(7,9×9,7)	**1152.56**	**347.44**	
8	(8,10×10,8)	**1025.02**	**474.98**	
9	(9,11×11,9)	**1059.9**	**440.1**	
10	(10,12×12,10)	**1155.38**	**344.62**	
11	(11,13×13,11)	**1056.06**	**443.94**	
12	(12,14×14,12)	**971.8**	**528.2**	
13	(13,15×15,13)	**916.6**	**583.4**	
14	(14,16×16,14)	**1204.39**	**295.61**	
⋮	⋮	⋮	⋮	
k	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***	
2	(2,16×16,2)	**1470**	**30**	$k = 2, r = 14$

(b)

Similarity				***Overlap***			
K	***Max Trace***	***Min Trace***	***Average Traces***	***CV***	***Max Overlap***	***Min Overlap***	***Average Overlap***
2	1478.85	1470	**1473.161**	**0.003**	30	21.15	**26.839**
3	1500	1500	**1500**	**0**	0	0	**0**
4	1491.42	1335.36	**1396.885**	**0.029**	164.64	8.58	**103.115**
5	1393.61	1229.54	**1317.581**	**0.034**	270.46	106.39	**182.419**
6	1293.74	1151.21	**1232.877**	**0.037**	348.79	206.26	**267.123**
7	1200.48	1114.83	**1164.69**	**0.026**	385.17	299.52	**335.31**
8	1142.4	1021.28	**1088.633**	**0.041**	478.72	357.6	**411.367**
9	1093.33	1024.9	**1059.279**	**0.024**	475.1	406.67	**440.721**
10	1155.38	938.63	**1043.49**	**0.078**	561.37	344.62	**456.51**
11	1091.34	1033.02	**1059.928**	**0.024**	466.98	408.66	**440.072**
12	1124.97	971.8	**1049.87**	**0.06**	528.2	375.03	**450.13**
13	1006.39	916.6	**952.217**	**0.05**	583.4	493.61	**547.783**
14	1204.39	1026.91	**1115.65**	**0.112**	473.09	295.61	**384.35**

(c)

Table 5.4: A collection of tables represent summary of values computed from Q matrices of case2.

The Figure 5.6 represents the difference between number of clusters (i.e. the number of different colours) obtained at $k = 2$ and $k = 3$ from *k-means* while other parts of the figure show plots of the Table 5.4 values.

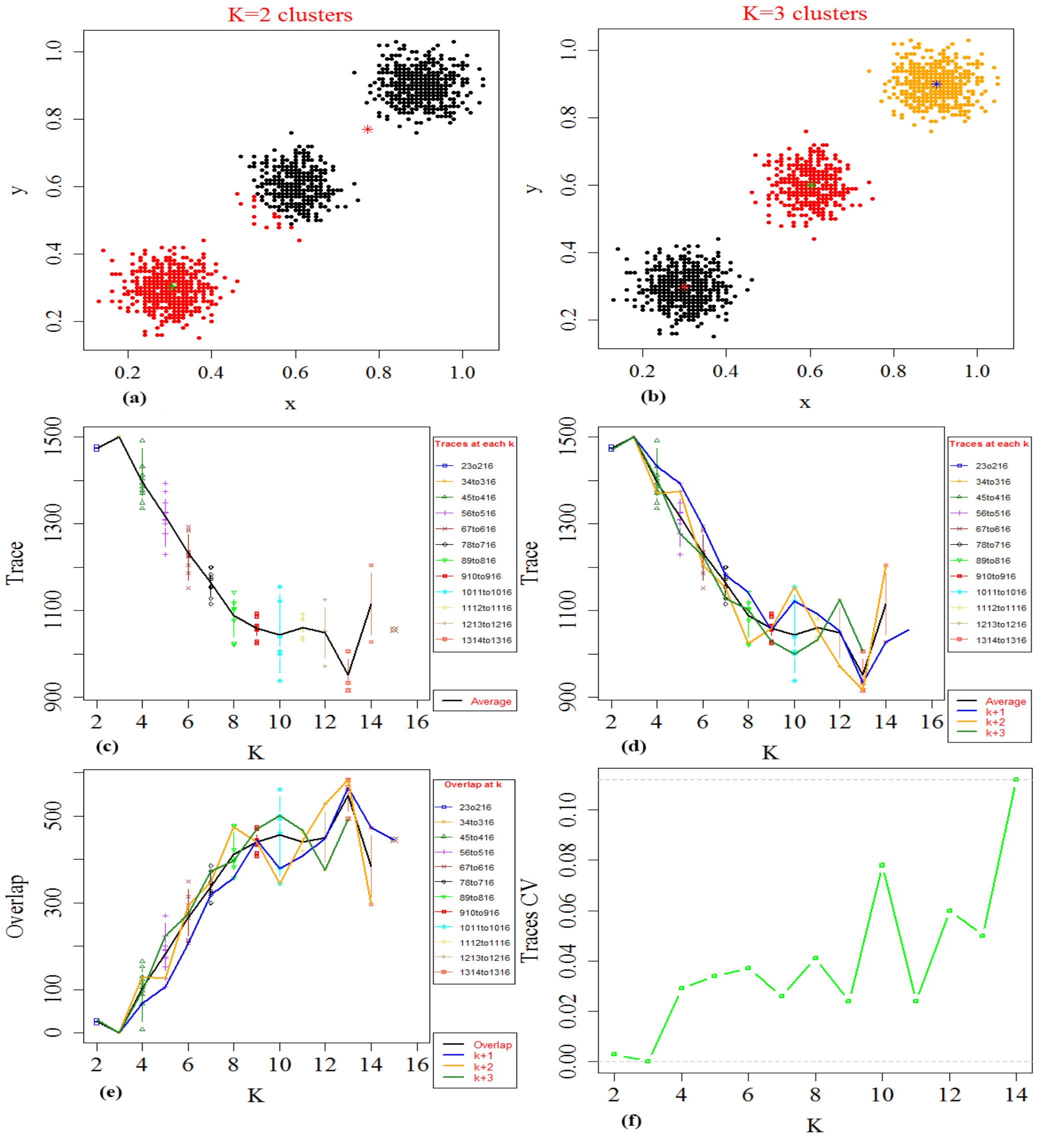


Figure 5.6: The memberships of the clusters obtained from *k-means* are labelled by different colours with their centroid in (a) and (b). Plots (c)-(f) show similarity, average similarity, overlap, average overlap and CV values.

The difference between clusters at $k = 2$ and $k = 3$ is clearly visualised in plots (a) and (b). Plot (c) shows traces and average traces. The black solid line is a maximum at $k = 3$ which is equal to $N = 1500$, which means clusters are fully separated (as can be seen in plot (b)). The plots (d) and (e) are composite plots that show the difference for similarity, overlap only for $k+1$, $k+2$, $k+3$, average similarity and overlap from the Table 5.4. Plot (f) shows the value of CV is 0 and clusters are stable at the best $\boldsymbol{K}$.

1st run:				***Case2:* Existing indexes values**						
K	***Dunn***	***DH***	***DH Critical***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***Gap Critical***	***CCC***
2	0.011	0.129	0.599	5070.723	0.649	0.482	15.276	0.977	-1.365	72.622
3	**0.36**	**0.764**	0.571	**19987.57**	**0.787**	**0.331**	**7.035**	**2.371**	0.338	**92.296**
4	0.009	1.285	0.571	15394.83	0.591	0.864	35.343	2.059	0.086	74.258
5	0.003	1.716	0.533	12356.47	0.469	1.19	30.35	2.003	0.09	64.136
6	0.003	2.731	0.524	10696.4	0.472	1.044	28.348	1.945	0.054	56.475
7	0.003	0.585	0.558	9815.304	0.321	0.982	34.387	1.722	-0.077	47.065
8	0.006	1.767	0.471	10603.21	0.32	1.215	28.94	1.711	0.128	47.926
9	0.011	1.456	0.451	10691.17	0.315	1.169	43.131	1.65	-0.076	43.521
10	0.007	2.544	0.508	10275.86	0.319	1.183	36.276	1.876	0.085	40.983
11	0.006	9.843	0.434	9268.546	0.315	1.132	35.932	1.725	-0.025	32.252
12	0.007	10.25	0.412	9786.605	0.32	1.079	48.985	1.843	0.427	38.404
13	0.011	0.512	0.496	9748.096	0.327	1.04	47.784	1.745	-0.013	27.588

(a)

Number of runs		***Dunn***	***DH***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***CCC***
2nd	***K***	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
	Values	**0.36**	**1.206**	**19987.57**	**0.787**	**0.331**	**6.923**	**2.368**	**92.296**
3rd	***K***	**3**	**2**	**3**	**3**	**3**	**3**	**3**	**3**
	Values	**0.36**	**0.783**	**19987.57**	**0.787**	**0.331**	**6.873**	**2.353**	**92.296**
4th	***K***	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
	Values	**0.36**	**1.109**	**19987.57**	**0.787**	**0.331**	**7.127**	**2.369**	**92.296**
5th	***K***	**3**	**2**	**3**	**3**	**3**	**3**	**3**	**3**
	Values	**0.36**	**0.783**	**19987.57**	**0.787**	**0.331**	**7.815**	**2.376**	**92.296**

(b)

Table 5.5: The summary of eight different indexes with 5 multiple runs. The optimal number of clusters is highlighted with bold values.

From the above tables the DH index is inconsistent while the remaining indexes perform well to identify the correct number of clusters. The results show all the indexes perform better for high-to-medium density spherical clusters. The new approach like the other indexes give the correct number of cluster as three. In addition to this the new approach also identifies clusters are fully separated and stable with CV value 0 at the best $\boldsymbol{K}$.

5.3.3 Case3: Medium-to-Low Density Clusters

The scatter plot of a case3 dataset is shown in Figure 5.1 which represents more spread compared to first two cases and clusters have no clearly visible boundaries.

$k = 2, r = 1,2,\ldots,K-k$

k	r	$(k, k+r) \times (k+r, k)$	*Traces (similarity)*	*Overlap*
2	1	(2,3×3,2)	**1305**	**195**
2	2	(2,4×4,2)	**1335**	**165**
2	3	(2,5×5,2)	**1342.55**	**157.45**
2	4	(2,6×6,2)	**1395**	**105**
2	5	(2,7×7,2)	**1357.45**	**142.55**
2	6	(2,8×8,2)	**1335**	**165**
2	7	(2,9×9,2)	**1350**	**150**
2	8	(2,10×10,2)	**1350**	**150**
2	9	(2,11×11,2)	**1380**	**120**
2	10	(2,12×12,2)	**1380**	**120**
2	11	(2,13×13,2)	**1365**	**135**
2	12	(2,14×14,2)	**1395**	**105**
2	13	(2,15×15,2)	**1395**	**105**
2	14	(2,16×16,2)	**1380**	**120**

$k = 3, r = 1,2,\ldots,K-k$

k	r	$(k, k+r) \times (k+r, k)$	*Traces (similarity)*	*Overlap*
3	1	(3,4×4,3)	**1412.84**	**87.16**
3	2	(3,5×5,3)	**1391.56**	**108.44**
3	3	(3,6×6,3)	**1387.71**	**112.29**
3	4	(3,7×7,3)	**1410.74**	**89.26**
3	5	(3,8×8,3)	**1387.71**	**112.29**
3	6	(3,9×9,3)	**1406.89**	**93.11**
3	7	(3,10×10,3)	**1396.78**	**103.22**
3	8	(3,11×11,3)	**1417**	**83**
3	9	(3,12×12,3)	**1426.07**	**73.93**
3	10	(3,13×13,3)	**1395.74**	**104.26**
3	11	(3,14×14,3)	**1429.92**	**70.08**
3	12	(3,15×15,3)	**1415.96**	**84.04**
3	13	(3,16×16,3)	**1420.85**	**79.15**
⋮	⋮	⋮	⋮	⋮

$k = 15, r = 1$

k	r	$(k, k+r) \times (k+r, k)$	*Traces (similarity)*	*Overlap*
15	1	(15,16×16,15)	**1189.75**	**310.25**

(a)

$k = 2,3,..15, r = 1$

k	$(k, k+r) \times (k+r, k)$	*Traces (similarity)*	*Overlap*
2	(2,3×3,2)	**1305**	**195**
3	(3,4×4,3)	**1412.84**	**87.16**
4	(4,5×5,4)	**1418.28**	**81.72**
5	(5,6×6,5)	**952.49**	**547.51**
6	(6,7×7,6)	**1160.29**	**339.71**
7	(7,8×8,7)	**1196.37**	**303.63**
8	(8,9×9,8)	**1282.79**	**217.21**
9	(9,10×10,9)	**1245.37**	**254.63**
10	(10,11×11,10)	**944.98**	**555.02**
11	(11,12×12,11)	**1072.13**	**427.87**
12	(12,13×13,12)	**994.75**	**505.25**
13	(13,14×14,13)	**1142.4**	**357.6**
14	(14,15×15,14)	**1208.98**	**291.02**
15	(15,16×16,15)	**1189.75**	**310.25**

$k = 2,3,..15, r = 2$

k	$(k, k+r) \times (k+r, k)$	*Traces (similarity)*	*Overlap*
2	(2,4×4,2)	**1335**	**165**
3	(3,5×5,3)	**1391.56**	**108.44**
4	(4,6×6,4)	**1128.35**	**371.65**
5	(5,7×7,5)	**1096.66**	**403.34**
6	(6,8×8,6)	**1112.34**	**387.66**
7	(7,9×9,7)	**1173.51**	**326.49**
8	(8,10×10,8)	**1150.42**	**349.58**
9	(9,11×11,9)	**1131.77**	**368.23**
10	(10,12×12,10)	**1212.33**	**287.67**
11	(11,13×13,11)	**961.01**	**538.99**
12	(12,14×14,12)	**986.71**	**513.29**
13	(13,15×15,13)	**1185.1**	**314.9**
14	(14,16×16,14)	**1181**	**319**
⋮	⋮	⋮	⋮

$k = 2, r = 14$

k	$(k, k+r) \times (k+r, k)$	*Traces (similarity)*	*Overlap*
2	(2,16×16,2)	**1380**	**120**

(b)

Similarity				*Overlap*			
K	***Max Trace***	***Min Trace***	***Average Traces***	***CV***	***Max Overlap***	***Min Overlap***	***Average Overlap***
2	1395	1305	**1361.786**	**0.02**	195	105	**138.214**
3	1429.92	1387.71	**1407.675**	**0.01**	112.29	70.08	**92.325**
4	1418.28	1128.35	**1318.076**	**0.055**	371.65	81.72	**181.924**
5	1326.96	952.49	**1203.25**	**0.085**	547.51	173.04	**296.75**
6	1176.5	1052.44	**1126.577**	**0.032**	447.56	323.5	**373.423**
7	1196.37	1051.46	**1116.78**	**0.052**	448.54	303.63	**383.22**
8	1282.79	1059.09	**1124.074**	**0.065**	440.91	217.21	**375.926**
9	1245.37	1008.71	**1108.661**	**0.072**	491.29	254.63	**391.339**
10	1212.33	944.98	**1064.005**	**0.089**	555.02	287.67	**435.995**
11	1072.13	961.01	**1028.31**	**0.041**	538.99	427.87	**471.69**
12	1075.82	986.71	**1029.178**	**0.044**	513.29	424.18	**470.822**
13	1185.1	1134.35	**1153.95**	**0.024**	365.65	314.9	**346.05**
14	1208.98	1181	**1194.99**	**0.017**	319	291.02	**305.01**

(c)

Table 5.6: A collection of tables represents the values calculated from Q matrices at different k with $k+r$ mapping distance for case3.

The results in Table 5.6 (a) show the similarity values are higher at $k = 3$ with less variation than $k = 2$ for different $k+r$ mapping distances. Table 5.6(b) shows similarity and overlap at different k for $k + r$ mapped distances. Table 5.6(c) clearly shows the average similarity is a maximum at $k = 3$ where there is also minimum average overlap. Also at $k = 3$ clusters are stable with minimum CV value.

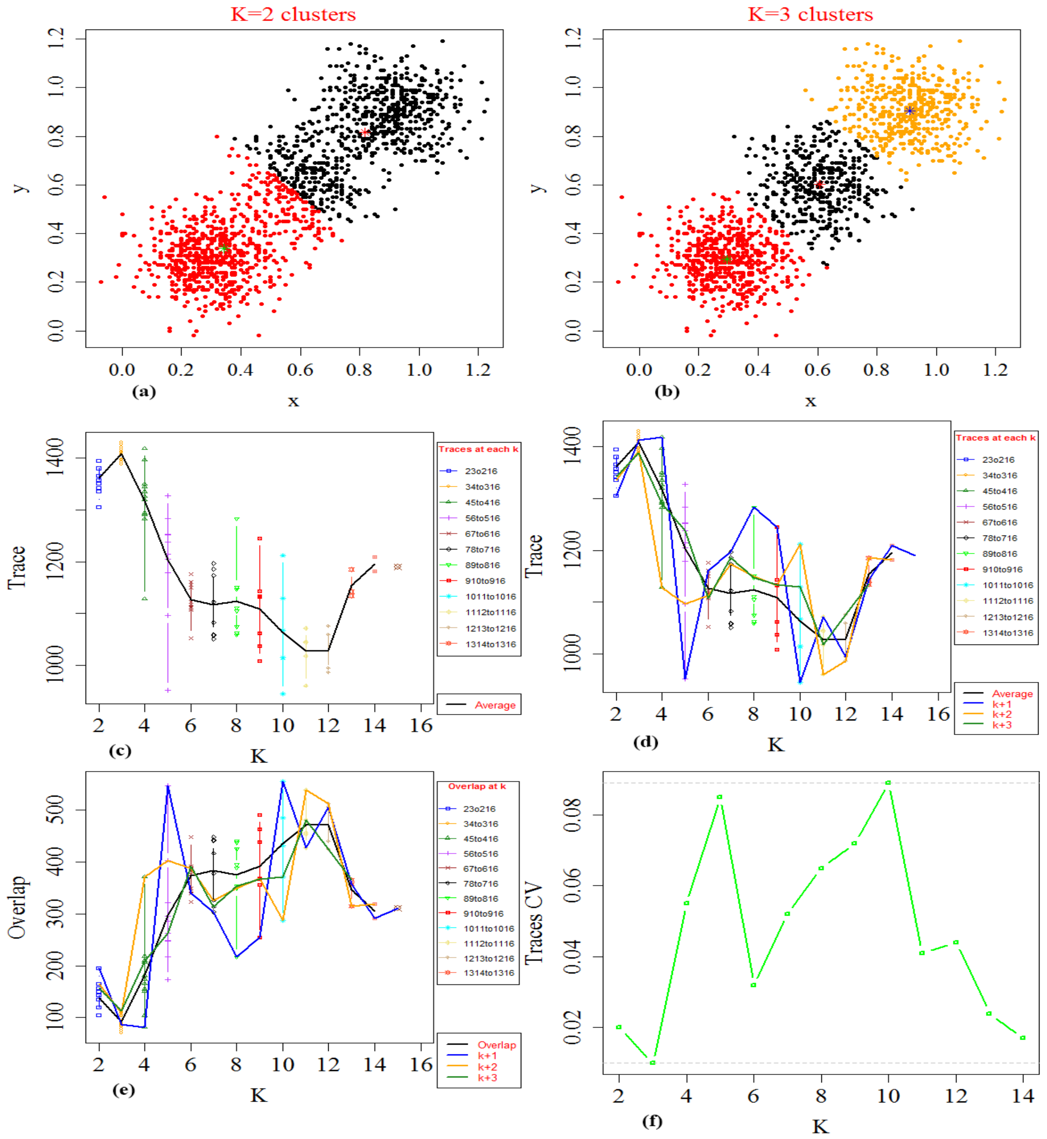


Figure 5.7: The memberships of the clusters obtained from *k-means* are labelled by different colours with their centroid in (a) and (b). Parts (c) to (f) are plots from the values of combined Q matrices.

In the Figure 5.7 plot (a) shows there is almost no gap between clusters centroids and boundaries at $k = 2$, while at $k = 3$ there is one extra cluster as seen in plot (b). Plot (c) shows the optimum estimated number of clusters is at $k = 3$ (mentioned in Chapter 4 section 4.3.3 criteria 1). At this value of k average similarity is a maximum with minimum average overlap as can be seen using the black solid line in plot (c). Plots (d) and (e) show the difference between similarity, overlap, average similarity and overlap at different k for$k + r$ mapping distances. The plot (f) shows clusters are stable with minimum CV value at $k = 3$.

1^st^ run:	***Case3:* Existing indexes values**									
K	***Dunn***	***DH***	***DH Critical***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***Gap Critical***	***CCC***
2	0.004	**0.595**	0.59	3759.045	**0.578**	**0.631**	8.621	1.266	-0.2	**61.75**
3	**0.012**	1.122	0.57	**4397.264**	0.544	0.705	**8.358**	**1.521**	0.33	48.056
4	0.007	2.46	0.533	3431.51	0.427	1.031	15.126	1.218	0.089	34.477
5	0.01	1.456	0.529	3116.247	0.34	1.226	14.566	1.098	-0.014	27.099
6	0.002	1.429	0.534	2847.155	0.304	1.134	15.092	1.115	0.054	23.531
7	0.011	1.923	0.515	2774.952	0.318	1.19	15.94	1.101	0.04	21.891
8	0.005	1.799	0.499	2778.347	0.312	1.087	14.254	1.092	0.065	22.303
9	0.006	1.564	0.484	2663.111	0.32	1.055	15.144	1.063	0.05	30.306
10	0.004	1.051	0.501	2654.405	0.329	1.038	15.34	1.031	0.042	30.051
11	0.008	1.03	0.485	2586.488	0.322	1.009	17.099	1.043	0.043	29.899
12	0.011	1.665	0.491	2551.804	0.335	1.059	16.838	0.982	-0.017	28.395
13	0.004	2.033	0.428	2517.767	0.317	1.01	17.28	1.022	0.044	28.32

(a)

Number of runs		***Dunn***	***DH***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***CCC***
2nd	***K***	**3**	**2**	**3**	**2**	**2**	**2**	**3**	**2**
	Values	**0.012**	**0.7**	**4397.264**	**0.578**	**0.631**	**8.338**	**1.505**	**61.751**
3rd	***K***	**3**	**2**	**3**	**2**	**2**	**2**	**3**	**2**
	Values	**0.012**	**0.666**	**4397.264**	**0.578**	**0.631**	**8.386**	**1.498**	**61.751**
4th	***K***	**16**	**2**	**3**	**2**	**2**	**2**	**3**	**2**
	Values	**0.013**	**0.715**	**4397.264**	**0.578**	**0.631**	**7.967**	**1.495**	**61.751**
5th	***K***	**3**	**2**	**3**	**2**	**2**	**2**	**3**	**2**
	Values	**0.012**	**0.7**	**4397.264**	**0.578**	**0.631**	**7.876**	**1.501**	**61.751**

(b)

Table 5.7: The summary of values computed from the different existing indexes with optimal number of clusters is highlighted with bold values.

Table (a) represents the values of each index from $k = 2$ to $k = 13$ with optimal number of clusters in bold while (b) summarises the optimal number of clusters for each index. The *Dunn*, *CH* and *Gap* indexes are similar to the new approach as 3 as determining estimated number of clusters while the other indexes indicate 2.

5.3.4 Case4: Low Density Clusters

This is the final case with more extreme variation; a case4 scatter plot is shown in Figure 5.1 and perhaps is the simplest case for identifying the clusters easily. In this case shows the original structure has disappeared due to a large amount of variation.

Table 5.8 (a) and (b) show the similarity and overlap at different k for $k + r$ mapping distances. Table (c) shows the minimum, maximum, similarity average similarity and overlap with CV values. Average similarity is a maximum at $k = 2$ which is the estimated number of clusters, using criteria 1 Chapter 4 section 4.3.3.

$k = 2, r = 1,2, \dots, K - k$

k	r	$(k, k+r) \times (k+r, k)$	*Traces (similarity)*	*Overlap*
2	1	(2,3×3,2)	**1260**	**240**
2	2	(2,4×4,2)	**1147.64**	**352.36**
2	3	(2,5×5,2)	**1267.64**	**232.36**
2	4	(2,6×6,2)	**1395**	**105**
2	5	(2,7×7,2)	**1342.64**	**157.36**
2	6	(2,8×8,2)	**1350**	**150**
2	7	(2,9×9,2)	**1290**	**210**
2	8	(2,10×10,2)	**1305**	**195**
2	9	(2,11×11,2)	**1350**	**150**
2	10	(2,12×12,2)	**1342.64**	**157.36**
2	11	(2,13×13,2)	**1365**	**135**
2	12	(2,14×14,2)	**1365**	**135**
2	13	(2,15×15,2)	**1350**	**150**
2	14	(2,16×16,2)	**1380**	**120**

$k = 3, r = 1,2, \dots, K - k$

k	r	$(k, k+r) \times (k+r, k)$	*Traces (similarity)*	*Overlap*
3	1	(3,4×4,3)	**1177.16**	**322.84**
3	2	(3,5×5,3)	**1272.7**	**227.3**
3	3	(3,6×6,3)	**1132.16**	**367.84**
3	4	(3,7×7,3)	**1181.98**	**318.02**
3	5	(3,8×8,3)	**1136.5**	**363.5**
3	6	(3,9×9,3)	**1185.6**	**314.4**
3	7	(3,10×10,3)	**1255.96**	**244.04**
3	8	(3,11×11,3)	**1322.16**	**177.84**
3	9	(3,12×12,3)	**1247.52**	**252.48**
3	10	(3,13×13,3)	**1253.72**	**246.28**
3	11	(3,14×14,3)	**1243.06**	**256.94**
3	12	(3,15×15,3)	**1243.6**	**256.4**
3	13	(3,16×16,3)	**1306.14**	**193.86**
⋮	⋮	⋮	⋮	⋮

$k = 15, r = 1$

k	r	$(k, k+r) \times (k+r, k)$	*Traces (similarity)*	*Overlap*
15	1	(15,16×16,15)	**1044.81**	**455.19**

(a)

$k = 2,3, \dots 15, r = 1$

k	$(k, k+r) \times (k+r, k)$	*Traces (similarity)*	*Overlap*
2	(2,3×3,2)	**1260**	**240**
3	(3,4×4,3)	**1177.16**	**322.84**
4	(4,5×5,4)	**1118.99**	**381.01**
5	(5,6×6,5)	**1025.97**	**474.03**
6	(6,7×7,6)	**1176.35**	**323.65**
7	(7,8×8,7)	**1118.44**	**381.56**
8	(8,9×9,8)	**905.97**	**594.03**
9	(9,10×10,9)	**1142.65**	**357.35**
10	(10,11×11,10)	**1229.96**	**270.04**
11	(11,12×12,11)	**1024.75**	**475.25**
12	(12,13×13,12)	**1065.91**	**434.09**
13	(13,14×14,13)	**1268.67**	**231.33**
14	(14,15×15,14)	**1284.48**	**215.52**
15	(15,16×16,15)	**1044.81**	**455.19**

$k = 2,3, \dots 15, r = 2$

k	$(k, k+r) \times (k+r, k)$	*Traces (similarity)*	*Overlap*
2	(2,4×4,2)	**1147.64**	**352.36**
3	(3,5×5,3)	**1272.7**	**227.3**
4	(4,6×6,4)	**1079.51**	**420.49**
5	(5,7×7,5)	**1046.3**	**453.7**
6	(6,8×8,6)	**1071.46**	**428.54**
7	(7,9×9,7)	**1042.41**	**457.59**
8	(8,10×10,8)	**950.66**	**549.34**
9	(9,11×11,9)	**1099.84**	**400.16**
10	(10,12×12,10)	**1103.88**	**396.12**
11	(11,13×13,11)	**1018.04**	**481.96**
12	(12,14×14,12)	**945.77**	**554.23**
13	(13,15×15,13)	**1263.79**	**236.21**
14	(14,16×16,14)	**1029.78**	**470.22**
⋮	⋮	⋮	⋮

$k = 2, r = 14$

k	$(k, k+r) \times (k+r, k)$	*Traces (similarity)*	*Overlap*
2	(2,16×16,2)	**1380**	**120**

(b)

	Similarity			*Overlap*			
K	***Max Trace***	***Min Trace***	***Average Traces***	***CV***	***Max Overlap***	***Min Overlap***	***Average Overlap***
2	1395	1147.64	**1322.183**	**0.049**	352.36	105	**177.817**
3	1322.16	1132.16	**1227.558**	**0.049**	367.84	177.84	**272.442**
4	1268.36	968.15	**1119.875**	**0.084**	531.85	231.64	**380.125**
5	1131.5	1005.7	**1066.285**	**0.045**	494.3	368.5	**433.715**
6	1176.35	948.29	**1066.406**	**0.059**	551.71	323.65	**433.594**
7	1118.44	935.37	**1017.889**	**0.055**	564.63	381.56	**482.111**
8	1009.8	905.97	**944.751**	**0.035**	594.03	490.2	**555.249**
9	1142.65	965.3	**1039.977**	**0.069**	534.7	357.35	**460.023**
10	1229.96	889.4	**1016.39**	**0.125**	610.6	270.04	**483.61**
11	1024.75	894.48	**983.33**	**0.054**	605.52	475.25	**516.67**
12	1065.91	865.54	**973.683**	**0.09**	634.46	434.09	**526.317**
13	1268.67	962.2	**1164.887**	**0.151**	537.8	231.33	**335.113**
14	1284.48	1029.78	1157.13	0.156	470.22	215.52	342.87

(c)

Table 5.8: Summary of the values computed from combined mapped elements Q matrices.

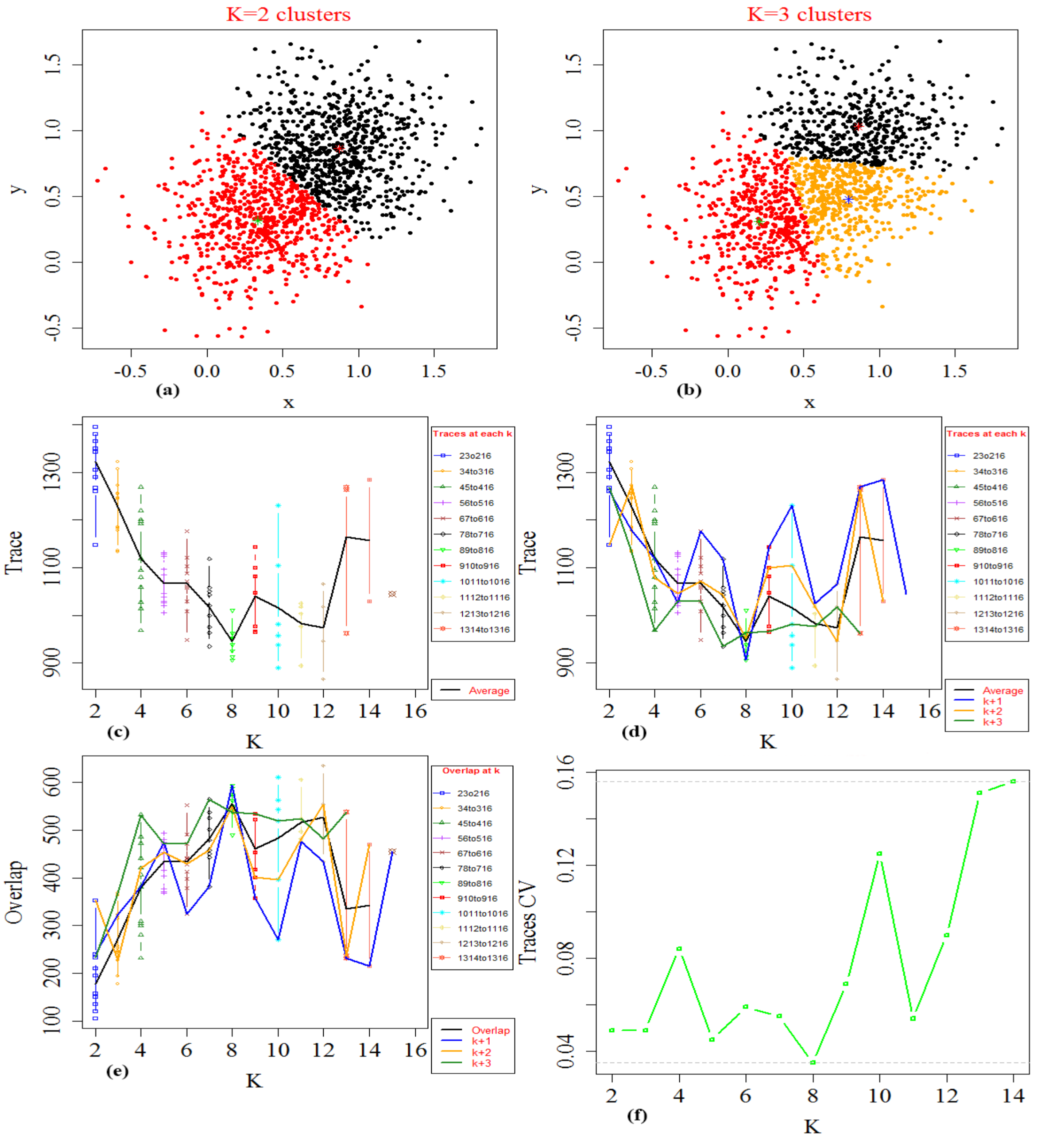


Figure 5.8: The membership of the clusters obtained from *k-means* are labelled by different colours with their centroid in (a) and (b). Plots (c) - (f) show similarity, average similarity, overlap, average overlap and CV values.

Plots (a) and (b) in the above figure show the difference between clusters at $k = 2$ and $k = 3$. The plots show that the clustering structure disappears. Plot (c) shows the similarity with different colours while average similarity is shown by the solid black line and indicates average similarity is a maximum at $k = 2$. The plots (d) and (f) represents differences in a composite graph of Table 5.8. Plot (f) shows clusters are stable with minimum CV value at $k = 2$. Even CV at $k = 3$ is also at the minimum but average similarity is not maximum.

1^{st} run: *Case4:* **Existing indexes values**

K	*Dunn*	*DH*	*DH Critical*	*CH*	*Sil*	*DB*	*SD*	*Gap*	*Gap Critical*	*CCC*
2	0.008	**0.97**	0.591	**1402.888**	**0.409**	1.033	7.051	**1.274**	0.224	**30.762**
3	0.005	0.958	0.569	1140.394	0.321	1.17	7.224	1.122	0.187	15.266
4	0.006	1.392	0.548	1150.05	0.347	0.976	**6.391**	0.949	0.029	20.303
5	0.005	1.345	0.534	1087.634	0.329	1.082	6.743	0.944	0.036	16.87
6	0.009	1.299	0.519	1069.545	0.326	1.023	6.447	0.879	-0.022	15.388
7	0.007	1.658	0.509	1092.583	0.327	0.977	6.959	0.922	0.078	15.491
8	0.007	1.444	0.496	1060.537	0.308	0.99	7.109	0.869	0.046	13.13
9	0.009	1.81	0.488	1053.981	0.323	0.968	7.936	0.84	0.033	12.598
10	**0.011**	2.051	0.448	1017.279	0.321	0.936	7.173	0.86	0.048	13.008
11	0.009	1.59	0.463	1040.223	0.327	0.956	7.808	0.836	0.049	12.581
12	0.007	1.395	0.428	989.55	0.311	0.998	8.735	0.807	0.065	12.181
13	0.004	2.147	0.376	1040.967	0.319	**0.928**	7.883	0.808	0.026	11.969

(a)

Number of runs		*Dunn*	*DH*	*CH*	*Sil*	*DB*	*SD*	*Gap*	*CCC*
2nd	*K*	15	2	2	2	12	4	2	2
	Values	0.011	1.075	1402.888	0.409	0.922	6.391	1.34	30.762
3rd	*K*	16	2	2	2	14	4	2	2
	Values	0.014	1.029	1402.888	0.409	0.929	6.383	1.272	30.762
4th	*K*	10	2	2	2	13	6	2	2
	Values	0.011	0.97	1402.888	0.409	0.917	6.326	1.279	30.762
5th	*K*	10	2	2	2	15	4	2	2
	Values	0.013	1.013	1402.888	0.409	0.91	6.266	1.318	30.762

(b)

Table 5.9: The summary of values computed from the different existing indexes with the optimal number of clusters highlighted in bold.

Table (a) represents the values of each index from $k = 2$ to $k = 13$ with optimal number of clusters in bold. Table (b) summarises the optimal number of clusters for each index. The results show two is the estimated number of clusters by most indexes (*DH*, *CH*, *Sil*, *Gap* and *CCC*). This is similar to the new approach result. The remaining three (*Dunn*, *DB* and *SD*) indexes vary for determining cluster number values from $k = 4$ to $k = 16$. This shows these indexes may not perform well in cases of severe noise.

5.3 Type2 Datasets: Spherical Clusters Equal Sizes

Type2 dataset (fixed centres (μ) with increasing spread (σ)): Type2 datasets consist of five equal sized spherical clusters constructed in such a way that centroids of three clusters are represented as vertices of a triangle while the other two clusters are further away and on top of each other. This type of dataset also consists of four different cases and each case has 5 identical centres (μ) but standard deviation (σ) values ranging from $\sigma = 0.25$ to $\sigma = 1.6$. Details of type2 datasets are represented in the table below.

Case1	***Case2***
$n = 400, \mu_1 = (2{,}6), \sigma_1 = 0.25$	$n = 400, \mu_1 = (2{,}6), \sigma_2 = 0.35$
$n = 400, \mu_2 = (4{,}6), \sigma_1 = 0.25$	$n = 400, \mu_2 = (4{,}6), \sigma_2 = 0.35$
$n = 400, \mu_3 = (3{,}8), \sigma_1 = 0.25$	$n = 400, \mu_3 = (3{,}8), \sigma_2 = 0.35$
$n = 400, \mu_4 = (8{,}4), \sigma_1 = 0.25$	$n = 400, \mu_4 = (8{,}4), \sigma_2 = 0.35$
$n = 400, \mu_5 = (8{,}6), \sigma_1 = 0.25$	$n = 400, \mu_5 = (8{,}6), \sigma_2 = 0.35$
Case3	***Case4***
$n = 400, \mu_1 = (2{,}6), \sigma_3 = 0.45$	$n = 400, \mu_1 = (2{,}6), \sigma_4 = 1.6$
$n = 400, \mu_2 = (4{,}6), \sigma_3 = 0.45$	$n = 400, \mu_2 = (4{,}6), \sigma_4 = 1.6$
$n = 400, \mu_3 = (3{,}8), \sigma_3 = 0.45$	$n = 400, \mu_3 = (3{,}8), \sigma_4 = 1.6$
$n = 400, \mu_4 = (8{,}4), \sigma_3 = 0.45$	$n = 400, \mu_4 = (8{,}4), \sigma_4 = 1.6$
$n = 400, \mu_5 = (8{,}6), \sigma_3 = 0.45$	$n = 400, \mu_5 = (8{,}6), \sigma_4 = 1.6$

Table 5.10: Details of Type2 datasets (where n = number of observations in each clusters, $\mu =$ mean, $\sigma =$ standard deviation).

For these four cases scatter plots are shown in the Figure 5.9.

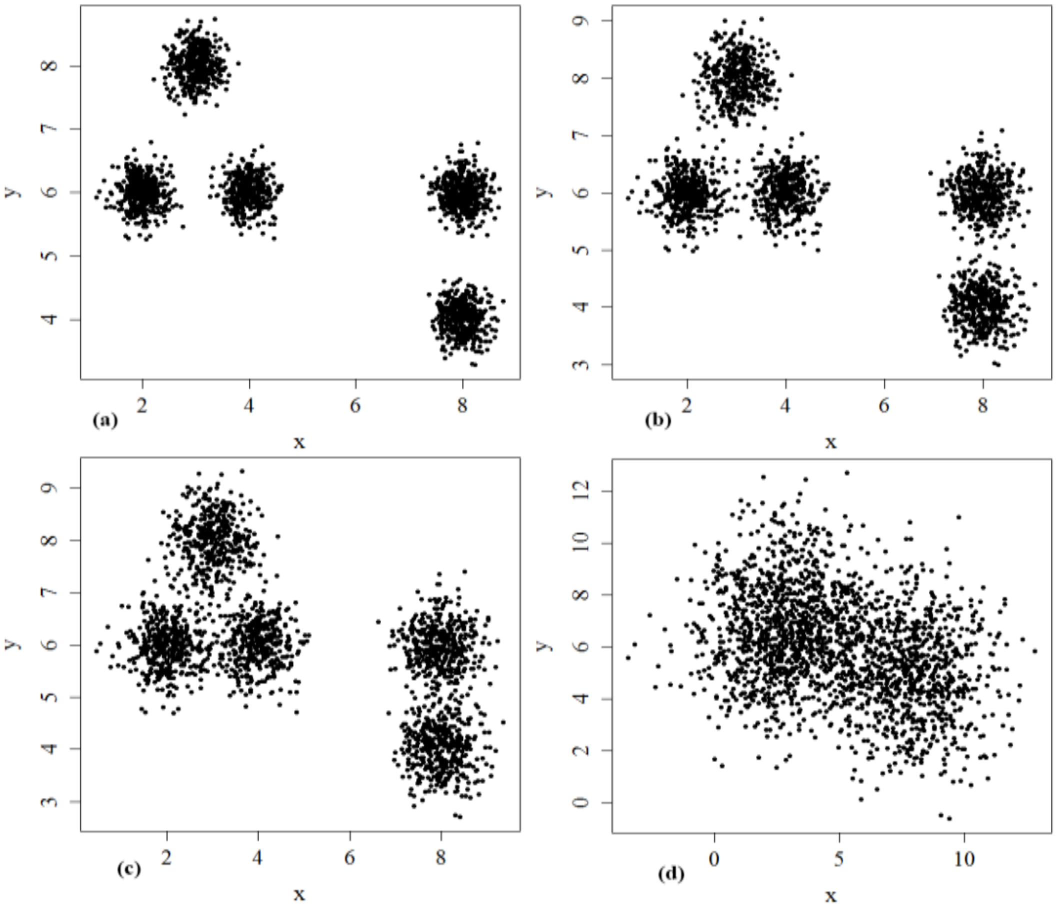


Figure 5.9: Type2 dataset scatter plots for four cases.

5.3.1 Case1: High Density Clusters

Case1 shows clearly 5 clusters as seen in the Figure 5.9 (a). Table (5.11) below represents the values computed from the new approach. Table (a) represents similarity and overlap values and the similarity is equal to $N = 2000$ at each k for different $k + r$ mapping distances. Table (b) is the summary of similarity and overlap at different k for $k + r$ and from $k = 2$ to $k = 5$ and clusters are completely separated while overlap begins as k increases. Table (c) specifies maximum, minimum, average similarity and overlap with coefficient of variation (CV). The average similarity is a maximum for $k = 2$ to $k = 5$ and is equal to N with average overlap of 0 elements between clusters. This shows using the new approach that clusters have the potential to split until $k = 5$. The average traces maximum increases until $k = 5$. According to criterion 4 in Chapter 4 section 4.3.3 the best number of clusters is five.

	k	r	$(k,k+r)\times(k+r,k)$	*Traces (similarity)*	*Overlap*
$k = 2, r = 1,2,\ldots,K-k$	2	1	(2,3×3,2)	**2000**	**0**
	2	2	(2,4×4,2)	**2000**	**0**
	2	3	(2,5×5,2)	**2000**	**0**
	2	4	(2,6×6,2)	**2000**	**0**
	2	5	(2,7×7,2)	**2000**	**0**
	2	6	(2,8×8,2)	**2000**	**0**
	2	7	(2,9×9,2)	**2000**	**0**
	2	8	(2,10×10,2)	**2000**	**0**
	2	9	(2,11×11,2)	**2000**	**0**
	2	10	(2,12×12,2)	**2000**	**0**
	2	11	(2,13×13,2)	**2000**	**0**
	2	12	(2,14×14,2)	**2000**	**0**
	2	13	(2,15×15,2)	**2000**	**0**
	2	14	(2,16×16,2)	**2000**	**0**
	k	r	$(k,k+r)\times(k+r,k)$	*Traces (similarity)*	*Overlap*
$k = 3, r = 1,2,\ldots,K-k$	3	1	(3,4×4,3)	**2000**	**0**
	3	2	(3,5×5,3)	**2000**	**0**
	3	3	(3,6×6,3)	**2000**	**0**
	3	4	(3,7×7,3)	**2000**	**0**
	3	5	(3,8×8,3)	**2000**	**0**
	3	6	(3,9×9,3)	**2000**	**0**
	3	7	(3,10×10,3)	**2000**	**0**
	3	8	(3,11×11,3)	**2000**	**0**
	3	9	(3,12×12,3)	**2000**	**0**
	3	10	(3,13×13,3)	**2000**	**0**
	3	11	(3,14×14,3)	**2000**	**0**
	3	12	(3,15×15,3)	**2000**	**0**
	3	13	(3,16×16,3)	**2000**	**0**
	⋮	⋮	⋮	⋮	⋮
	k	r	$(k,k+r)\times(k+r,k)$	*Traces (similarity)*	*Overlap*
$k = 15, r = 1$	15	1	(15,16×16,15)	**1654.39**	**345.61**

(a)

k	$(k,k+r)\times(k+r,k)$	*Traces (similarity)*	*Overlap*	
2	(2,3×3,2)	**2000**	**0**	$k = 2,3,..15, r = 1$
3	(3,4×4,3)	**2000**	**0**	
4	(4,5×5,4)	**2000**	**0**	
5	(5,6×6,5)	**2000**	**0**	
6	(6,7×7,6)	**1982.23**	**17.77**	
7	(7,8×8,7)	**1813.12**	**186.88**	
8	(8,9×9,8)	**1702.48**	**297.52**	
9	(9,10×10,9)	**2000**	**0**	
10	(10,11×11,10)	**1734.86**	**265.14**	
11	(11,12×12,11)	**1696.29**	**303.71**	
12	(12,13×13,12)	**1579.99**	**420.01**	
13	(13,14×14,13)	**1353.85**	**646.15**	
14	(14,15×15,14)	**1388.97**	**611.03**	
15	(15,16×16,15)	**1654.39**	**345.61**	
k	$(k,k+r)\times(k+r,k)$	*Traces (similarity)*	*Overlap*	
2	(2,4×4,2)	**2000**	**0**	$k = 2,3,..15, r = 2$
3	(3,5×5,3)	**2000**	**0**	
4	(4,6×6,4)	**2000**	**0**	
5	(5,7×7,5)	**2000**	**0**	
6	(6,8×8,6)	**1806.76**	**193.24**	
7	(7,9×9,7)	**1888.24**	**111.76**	
8	(8,10×10,8)	**1702.48**	**297.52**	
9	(9,11×11,9)	**1734.86**	**265.14**	
10	(10,12×12,10)	**1812.26**	**187.74**	
11	(11,13×13,11)	**1760.92**	**239.08**	
12	(12,14×14,12)	**1308.87**	**691.13**	
13	(13,15×15,13)	**1643.28**	**356.72**	
14	(14,16×16,14)	**1518.96**	**481.04**	
⋮	⋮	⋮	⋮	
k	$(k,k+r)\times(k+r,k)$	*Traces (similarity)*	*Overlap*	
2	(2,16×16,2)	**2000**	**0**	$k = 2, r = 14$

(b)

Similarity				***Overlap***			
K	***Max Trace***	***Min Trace***	***Average Traces***	***CV***	***Max Overlap***	***Min Overlap***	***Average Overlap***
2	2000	2000	**2000**	**0**	0	0	**0**
3	2000	2000	**2000**	**0**	0	0	**0**
4	2000	2000	**2000**	**0**	0	0	**0**
5	2000	2000	**2000**	**0**	0	0	**0**
6	1982.23	1806.76	**1899.826**	**0.029**	193.24	17.77	**100.174**
7	1912.24	1813.12	**1877.369**	**0.021**	186.88	87.76	**122.631**
8	1898.53	1700.39	**1780.977**	**0.046**	299.61	101.47	**219.022**
9	2000	1734.86	**1862.969**	**0.052**	265.14	0	**137.031**
10	1830.27	1611.65	**1754.977**	**0.047**	388.35	169.73	**245.023**
11	1760.92	1540.92	**1623.358**	**0.061**	459.08	239.08	**376.642**
12	1579.99	1308.87	**1475.648**	**0.079**	691.13	420.01	**524.352**
13	1643.28	1353.85	**1528.337**	**0.101**	646.15	356.72	**471.663**
14	1518.96	1388.97	**1453.965**	**0.063**	611.03	481.04	**546.035**

(c)

Table 5.11: A collection of tables representing the values computed from Q matrices.

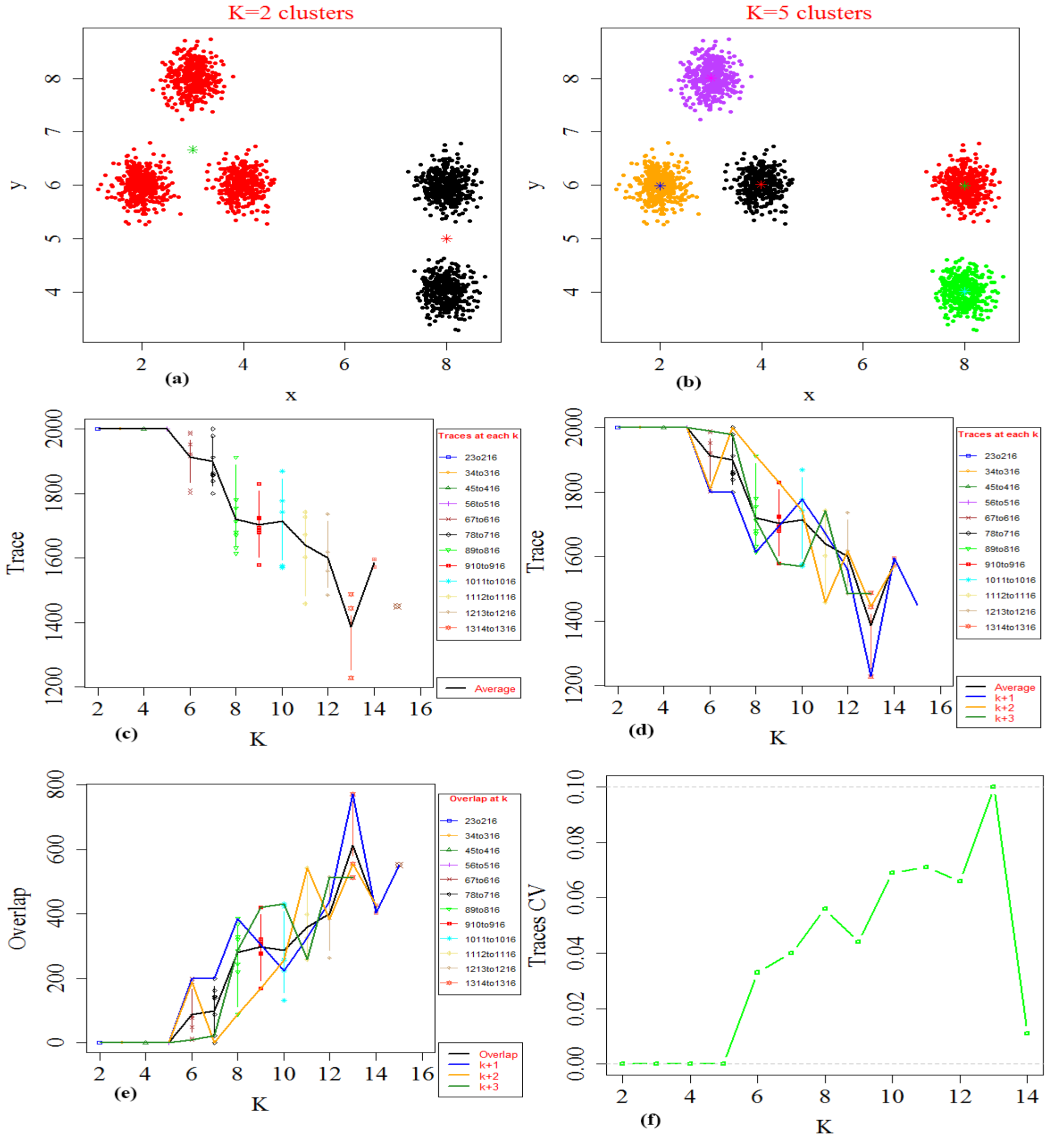


Figure 5.10: Plots (a)-(b) represents clusters obtained from *k-means* and elements are labelled in different colours for each cluster with centroids. Plots (c)-(f) show similarity, overlap, average similarity, average overlap and CV.

Plot (a) represents two clusters (red and black colours) obtained by the *k-means* algorithm. The two centroids (green and red stars) are in middle positions. Plot (b) represents five clusters but now the centroid of each cluster is internal. Plot (c) indicates similarity using different colours and average similarity as a black solid line. Plot (d) is a composite graph that shows the difference between similarity for $k+1$, $k+2$, $k+3$ and average similarity in different colours. Similarly, plot (e) is also a composite graph showing the difference in overlap. Plot (f) indicates the values of CV at different k and that clusters are stable at the best $\boldsymbol{K} = 5$. From the above figure we can easily visualise that for $k = 5$ the average similarity is a maximum and equal to total number of elements $N = 2000$ (i.e. mapping of 100% and 0 overlap) which indicates all the clusters are fully separated at $k = 5$.

1st run:	***Case1:* Existing indexes values**								
K	***Dunn***	***DH***	***DH Critical***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***Gap Critical***	***CCC***
2	**0.689**	**2.354**	0.594	229.038	0.727	2.897	1.649	0.932	-0.118	136.144
3	0.084	0.114	0.594	7896.669	0.647	**0.344**	2.028	0.915	-0.332	109.547
4	0.005	0.036	0.594	4462.014	0.752	0.866	1.336	**1.249**	0.205	159.514
5	0.005	2.033	0.505	**32211.57**	**0.779**	0.778	**1.079**	0.832	0.019	**215.483**
6	0.004	0.796	0.5	27518.08	0.678	1.069	6.174	0.099	-2.132	129.732
7	0.006	18.385	0.558	5024.723	0.686	0.972	7.364	2.255	0.096	196.922
8	0.01	1.293	0.506	23316.68	0.586	0.941	5.651	0.618	-1.526	191.203
9	0.003	1.422	0.489	20361.73	0.568	0.886	5.346	2.142	0.184	185.977
10	0.005	0.648	0.463	20653.56	0.503	1.2	4.789	2.062	0.01	183.771
11	0.006	1.205	0.477	19629.31	0.401	1.136	4.719	2.073	0.072	180.69
12	0.008	4.851	0.456	19528.78	0.41	1.106	6.172	2.088	0.098	177.249
13	0.006	0.37	0.482	18863.19	0.325	1.203	4.943	1.907	-0.022	88.089

(a)

Number of runs		***Dunn***	***DH***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***CCC***
2nd	***K***	**2**	**2**	**6**	**2**	**5**	**3**	**3**	**9**
	Values	**0.689**	**1.302**	**27579.79**	**0.727**	**0.344**	**1.606**	**1.066**	**186.338**
3rd	***K***	**2**	**3**	**6**	**7**	**5**	**3**	**4**	**8**
	Values	**0.689**	**19.728**	**27478.7**	**0.686**	**0.344**	**1.45**	**1.266**	**186.529**
4th	***K***	**2**	**2**	**6**	**4**	**4**	**4**	**3**	**6**
	Values	**0.689**	**1.062**	**27856.43**	**0.752**	**0.338**	**1.271**	**0.952**	**203.835**
5th	***K***	**2**	**5**	**8**	**4**	**3**	**3**	**5**	**6**
	Values	**0.691**	**17.9**	**22314.63**	**0.752**	**0.344**	**1.501**	**0.84**	**204.13**

(b)

Table 5.12: The summary of eight different existing indexes with 5 multiple runs. The optimal number of clusters is highlighted in bold.

Table (a) shows a summary of each index values from $k = 2$ to $k = 13$ with the optimal number of clusters highlighted in bold. Table (b) summarises the optimal number of clusters for each index. *Dunn* index represents 2 clusters constantly. *DH* gives 2 to 5. *DB*, *SD* and *Gap* give between 3 and 5. *CH* and *CCC* are contrary and suggest the best number of clusters is between 5 and 9. *Sil* is unable to find correct k. From Figure 5.10 (a) the centroid is empty and none of clusters is represented with centroids. This reveals that the variance or distance between the centroids is not a minimum at $k = 2$ while at $k = 5$ all the centroids belong within each cluster and the minimum variance is at $k = 5$. The new approach indicates clearly $k \ = 5$ is the best number of clusters where each clusters is fully separated and the set of clusters are disjoint which means 0 overlap while other indexes are unable to find the correct number of clusters. The coefficient of variation is 0 and clusters are stable at the best value of $\boldsymbol{K}$. The next 3 cases are the extension of case1. They increase variation gradually.

5.3.2 Case2: High-to-Medium Density Clusters

The scatter plot of case2 dataset is depicted in Figure 5.9 (b). For the dataset in this particular case it is not only of interest to detect the optimal number of clusters but also to explore the structure for different k. Table 5.13 (a) and (b) show the similarity and overlap at different k with different $k + r$ mapping distances. Table 5.13(c) represents the different values of similarity and overlap with CV values at different k. Choosing $k = 2$ gives a special case in which average similarity has a maximum equal to N as shown in Table 5.13(c). The second choice at $k = \ 5$ is better as average similarity has maximum, minimum (number of elements) overlap between clusters and minimum CV value better than those at $k = 6$ and higher k values. The difference between average similarity, overlap and CV from $k = 2$ to $k = 5$ is slight with a small perturbation when 8.36 average number of elements belong to different clusters at $k = 5$ while for $k = 2$ there is full separation.

	k	r	$(k,k+r)\times(k+r,k)$	***Traces (similarity)***	***Overlap***
$k=2,r=1,2,\dots,K-k$	2	1	(2,3×3,2)	**2000**	**0**
	2	2	(2,4×4,2)	**2000**	**0**
	2	3	(2,5×5,2)	**2000**	**0**
	2	4	(2,6×6,2)	**2000**	**0**
	2	5	(2,7×7,2)	**2000**	**0**
	2	6	(2,8×8,2)	**2000**	**0**
	2	7	(2,9×9,2)	**2000**	**0**
	2	8	(2,10×10,2)	**2000**	**0**
	2	9	(2,11×11,2)	**2000**	**0**
	2	10	(2,12×12,2)	**2000**	**0**
	2	11	(2,13×13,2)	**2000**	**0**
	2	12	(2,14×14,2)	**2000**	**0**
	2	13	(2,15×15,2)	**2000**	**0**
	2	14	(2,16×16,2)	**2000**	**0**

	k	r	$(k,k+r)\times(k+r,k)$	***Traces (similarity)***	***Overlap***
$k=3,r=1,2,\dots,K-k$	3	1	(3,4×4,3)	**1995.99**	**4.01**
	3	2	(3,5×5,3)	**1995.99**	**4.01**
	3	3	(3,6×6,3)	**1995.99**	**4.01**
	3	4	(3,7×7,3)	**1995.99**	**4.01**
	3	5	(3,8×8,3)	**1995.99**	**4.01**
	3	6	(3,9×9,3)	**2000**	**0**
	3	7	(3,10×10,3)	**2000**	**0**
	3	8	(3,11×11,3)	**1995.99**	**4.01**
	3	9	(3,12×12,3)	**1995.99**	**4.01**
	3	10	(3,13×13,3)	**1995.99**	**4.01**
	3	11	(3,14×14,3)	**2000**	**0**
	3	12	(3,15×15,3)	**1995.99**	**4.01**
	3	13	(3,16×16,3)	**1995.99**	**4.01**
	⋮	⋮	⋮	⋮	⋮

	k	r	$(k,k+r)\times(k+r,k)$	***Traces (similarity)***	***Overlap***
$k=15,r=1$	15	1	(15,16×16,15)	**1618.73**	**381.27**

(a)

k	$(k,k+r)\times(k+r,k)$	***Traces (similarity)***	***Overlap***	
2	(2,3×3,2)	**2000**	**0**	$k=2,3,..15,r=1$
3	(3,4×4,3)	**1995.99**	**4.01**	
4	(4,5×5,4)	**2000**	**0**	
5	(5,6×6,5)	**1996**	**4**	
6	(6,7×7,6)	**1996**	**4**	
7	(7,8×8,7)	**1970.49**	**29.51**	
8	(8,9×9,8)	**1584.99**	**415.01**	
9	(9,10×10,9)	**1785.88**	**214.12**	
10	(10,11×11,10)	**1653.87**	**346.13**	
11	(11,12×12,11)	**1375.63**	**624.37**	
12	(12,13×13,12)	**1528.21**	**471.79**	
13	(13,14×14,13)	**1571.85**	**428.15**	
14	(14,15×15,14)	**1490.65**	**509.35**	
15	(15,16×16,15)	**1618.73**	**381.27**	

k	$(k,k+r)\times(k+r,k)$	***Traces (similarity)***	***Overlap***	
2	(2,4×4,2)	**2000**	**0**	$k=2,3,..15,r=2$
3	(3,5×5,3)	**1995.99**	**4.01**	
4	(4,6×6,4)	**1996**	**4**	
5	(5,7×7,5)	**1992**	**8**	
6	(6,8×8,6)	**1980.6**	**19.4**	
7	(7,9×9,7)	**1783.7**	**216.3**	
8	(8,10×10,8)	**1585.59**	**414.41**	
9	(9,11×11,9)	**1611.53**	**388.47**	
10	(10,12×12,10)	**1339.66**	**660.34**	
11	(11,13×13,11)	**1357.65**	**642.35**	
12	(12,14×14,12)	**1386.38**	**613.62**	
13	(13,15×15,13)	**1500.23**	**499.77**	
14	(14,16×16,14)	**1399.4**	**600.6**	
⋮	⋮	⋮	⋮	

k	$(k,k+r)\times(k+r,k)$	***Traces (similarity)***	***Overlap***	
2	(2,16×16,2)	**2000**	**0**	$k=2,r=14$

(b)

Similarity				***Overlap***			
K	***Max Trace***	***Min Trace***	***Average Traces***	***CV***	***Max Overlap***	***Min Overlap***	***Average Overlap***
2	2000	2000	**2000**	**0**	0	0	**0**
3	2000	1995.99	**1996.915**	**0.001**	4.01	0	**3.085**
4	2000	1980	**1992.668**	**0.003**	20	0	**7.332**
5	2000	1980	**1991.638**	**0.002**	20	0	**8.362**
6	1996	1802.49	**1886.89**	**0.04**	197.51	4	**113.11**
7	1970.49	1723.81	**1815.568**	**0.045**	276.19	29.51	**184.432**
8	1873.03	1584.99	**1712.527**	**0.06**	415.01	126.97	**287.473**
9	1785.88	1468.39	**1672.027**	**0.062**	531.61	214.12	**327.973**
10	1690.44	1339.66	**1579.055**	**0.081**	660.34	309.56	**420.945**
11	1695.62	1357.65	**1492.374**	**0.1**	642.35	304.38	**507.626**
12	1613.33	1386.38	**1526.443**	**0.065**	613.62	386.67	**473.558**
13	1571.85	1472.42	**1514.833**	**0.034**	527.58	428.15	**485.167**
14	1490.65	1399.4	**1445.025**	**0.045**	600.6	509.35	**554.975**

(c)

Table 5.13: A collection of tables for case2 which represent the partial results computed from combined mapped elements Q matrices.

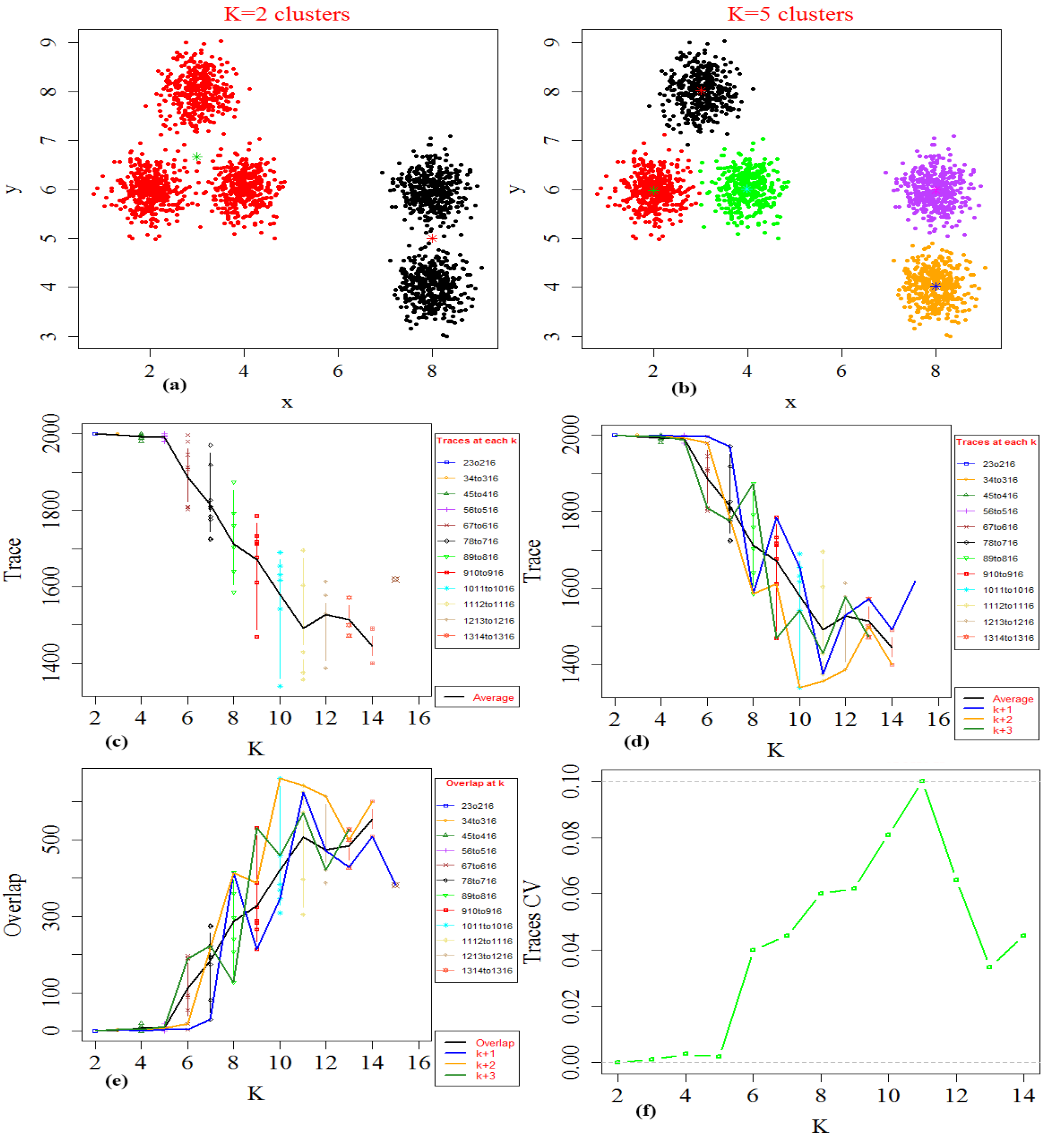


Figure 5.11: The memberships of the clusters obtained from *k-means* are labelled by different colours with their centroid in (a) and (b). Plots (c)-(f) show similarity, average similarity, overlap, average overlap and CV values.

In Figure 5.11 Plots (a) and (b) represent the difference between clusters when $k = 2$ and $k = 5$ from a typical *k-means* algorithm. By considering plots(c)-(f) with different $k + r$ mapped distances at different k the segments of lines in each plot are more stable until $k = 5$ which indicates five is the best number of clusters. Average similarity, overlap and CV show high variation between values for higher k values. Suppose we consider only adjacent $k + 1$ mapped distances to find the estimated clusters as shown in Plot (d). The solid blue line indicates maximum similarity until $k = 6$ but Figure 5.11(b) clearly shows five clusters. This situation has been described in Chapter 4 section 4.5 with an example to show the effect of adjacent mapping distance would not be appropriate for estimating the number of clusters. It seems average similarity and not maximum similarity is the best criterion for estimating the number of clusters and stability between clusters.

1^{st} run:					***Case2:* Existing indexes values**					
K	***Dunn***	***DH***	***DH Critical***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***Gap Critical***	***CCC***
2	**0.469**	**1.366**	0.594	8479.417	**0.711**	**0.464**	1.189	**1.038**	0.063	132.544
3	0.054	0.875	0.594	7025.793	0.672	0.787	1.617	1.145	-0.076	99.222
4	0.007	0.196	0.594	8017.842	0.687	0.65	4.576	0.625	-1.333	149.013
5	0.004	0.129	0.594	6184.954	0.687	0.832	**1.12**	1.97	1.144	**181.071**
6	0.003	1.29	0.558	5044.704	0.609	0.778	3.955	1.811	0.058	120.598
7	0.006	1.504	0.51	**12827.94**	0.532	0.996	4.229	1.745	0.074	163.542
8	0.009	2.965	0.518	11385.82	0.512	0.948	3.771	1.632	-0.049	159.101
9	0.004	1.471	0.508	10745.47	0.452	0.886	3.825	1.649	0.067	154.104
10	0.008	1.024	0.522	10700.36	0.38	1.07	4.346	1.637	0.15	145.459
11	0.007	2.967	0.409	10295.63	0.38	1.014	4.08	1.543	0.073	150.858
12	0.006	0.405	0.558	10098.01	0.312	1.12	4.202	1.525	0.17	146.083
13	0.005	1.398	0.484	7618.638	0.468	1.139	5.123	1.423	-0.004	147.73

(a)

Number of runs		***Dunn***	***DH***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***CCC***
2nd	***K***	**2**	**5**	**5**	**2**	**2**	**2**	**2**	**6**
	Values	**0.469**	**11.684**	**16446**	**0.711**	**0.464**	**1.156**	**1.047**	**170.722**
3rd	***K***	**2**	**3**	**6**	**2**	**4**	**2**	**2**	**6**
	Values	**0.469**	**1.101**	**14008.42**	**0.711**	**0.45**	**1.275**	**1.044**	**170.641**
4th	***K***	**2**	**3**	**6**	**2**	**3**	**2**	**2**	**5**
	Values	**0.469**	**1.353**	**14068.97**	**0.711**	**0.452**	**1.139**	**1.05**	**181.071**
5th	***K***	**2**	**2**	**7**	**2**	**4**	**2**	**4**	**7**
	Values	**0.467**	**1.113**	**12364.27**	**0.711**	**0.45**	**1.155**	**1.058**	**163.891**

(b)

Table 5.14: The summary of eight different indexes with 5 multiple runs. The optimal number of clusters is highlighted in bold.

The results from the above tables show *DH*, *CH*, *DB* and *CCC* indexes indicate the estimated number of clusters varies between $k = 2$ and $k = 7$ while other indexes *Dunn*, *Sil*, *SD* and *Gap* frequently show $k = 2$ as the number of clusters. These alternative indexes rarely show $k = 5$ as optimal and are inconsistent in their choice.

5.3.3 Case3: Medium-to-Low Density Clusters

The scatter plot is shown in Figure 5.9 (c) and indicates more variation than the first two cases discussed earlier. The results in the Table 5.15 (a) show at fixed $k = 2$ the value of similarities are maximum and equal to N. Table (b) shows similarities and overlaps at different k with fixed $k + r$. Table (c) indicates at $k = 2$ average similarity is equal to $N = 2000$ which show the clusters are fully separated with 0 average overlap. This case is very similar to case2 for selecting either at $k = 2$ as fully separated clusters or $k = 5$ as the estimated number of clusters with minimum average overlaps of 43 elements between clusters seen in Table (c).

$k = 2, r = 1,2,\dots,K-k$

k	r	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***
2	1	(2,3×3,2)	**2000**	**0**
2	2	(2,4×4,2)	**2000**	**0**
2	3	(2,5×5,2)	**2000**	**0**
2	4	(2,6×6,2)	**2000**	**0**
2	5	(2,7×7,2)	**2000**	**0**
2	6	(2,8×8,2)	**2000**	**0**
2	7	(2,9×9,2)	**2000**	**0**
2	8	(2,10×10,2)	**2000**	**0**
2	9	(2,11×11,2)	**2000**	**0**
2	10	(2,12×12,2)	**2000**	**0**
2	11	(2,13×13,2)	**2000**	**0**
2	12	(2,14×14,2)	**2000**	**0**
2	13	(2,15×15,2)	**2000**	**0**
2	14	(2,16×16,2)	**2000**	**0**

$k = 3, r = 1,2,\dots,K-k$

k	r	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***
3	1	(3,4×4,3)	**1980.02**	**19.98**
3	2	(3,5×5,3)	**1980.02**	**19.98**
3	3	(3,6×6,3)	**1968.02**	**31.98**
3	4	(3,7×7,3)	**1968.02**	**31.98**
3	5	(3,8×8,3)	**1968.02**	**31.98**
3	6	(3,9×9,3)	**1956.02**	**43.98**
3	7	(3,10×10,3)	**1956.02**	**43.98**
3	8	(3,11×11,3)	**1972.01**	**27.99**
3	9	(3,12×12,3)	**1980.02**	**19.98**
3	10	(3,13×13,3)	**1964.03**	**35.97**
3	11	(3,14×14,3)	**1972.01**	**27.99**
3	12	(3,15×15,3)	**1968.02**	**31.98**
3	13	(3,16×16,3)	**1984.01**	**15.99**
⋮	⋮	⋮	⋮	⋮

$k = 15, r = 1$

k	r	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***
15	1	(15,16×16,15)	**1687.77**	**312.23**

(a)

$k = 2,3,..15, r = 1$

k	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***
2	(2,3×3,2)	**2000**	**0**
3	(3,4×4,3)	**1980.02**	**19.98**
4	(4,5×5,4)	**2000**	**0**
5	(5,6×6,5)	**1976.04**	**23.96**
6	(6,7×7,6)	**1770.61**	**229.39**
7	(7,8×8,7)	**1676.78**	**323.22**
8	(8,9×9,8)	**1964.83**	**35.17**
9	(9,10×10,9)	**1885.58**	**114.42**
10	(10,11×11,10)	**1709.32**	**290.68**
11	(11,12×12,11)	**1313.07**	**686.93**
12	(12,13×13,12)	**1438.15**	**561.85**
13	(13,14×14,13)	**1455.16**	**544.84**
14	(14,15×15,14)	**1655.12**	**344.88**
15	(15,16×16,15)	**1687.77**	**312.23**

$k = 2,3,..15, r = 2$

k	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***
2	(2,4×4,2)	**2000**	**0**
3	(3,5×5,3)	**1980.02**	**19.98**
4	(4,6×6,4)	**1976.04**	**23.96**
5	(5,7×7,5)	**1952.02**	**47.98**
6	(6,8×8,6)	**1984**	**16**
7	(7,9×9,7)	**1655.46**	**344.54**
8	(8,10×10,8)	**1892.79**	**107.21**
9	(9,11×11,9)	**1656.62**	**343.38**
10	(10,12×12,10)	**1343.84**	**656.16**
11	(11,13×13,11)	**1334**	**666**
12	(12,14×14,12)	**1486.75**	**513.25**
13	(13,15×15,13)	**1819.74**	**180.26**
14	(14,16×16,14)	**1674.6**	**325.4**
⋮	⋮	⋮	⋮

$k = 2, r = 14$

k	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***
2	(2,16×16,2)	**2000**	**0**

(b)

	Similarity			***Overlap***			
K	***Max Trace***	***Min Trace***	***Average Traces***	***CV***	***Max Overlap***	***Min Overlap***	***Average Overlap***
2	2000	2000	**2000**	**0**	0	0	**0**
3	1984.01	1956.02	**1970.48**	**0.005**	43.98	15.99	**29.52**
4	2000	1943.96	**1973.338**	**0.007**	56.04	0	**26.662**
5	1979.98	1919.96	**1957.096**	**0.008**	80.04	20.02	**42.904**
6	1984	1757.82	**1876.42**	**0.04**	242.18	16	**123.58**
7	1825.68	1655.46	**1759.589**	**0.039**	344.54	174.32	**240.411**
8	1964.83	1603.63	**1750.245**	**0.071**	396.37	35.17	**249.755**
9	1885.58	1519.43	**1646.236**	**0.071**	480.57	114.42	**353.764**
10	1709.32	1343.84	**1515.877**	**0.082**	656.16	290.68	**484.123**
11	1629.08	1313.07	**1440.726**	**0.09**	686.93	370.92	**559.274**
12	1609.4	1438.15	**1505.605**	**0.048**	561.85	390.6	**494.395**
13	1819.74	1455.16	**1650.687**	**0.111**	544.84	180.26	**349.313**
14	1674.6	1655.12	**1664.86**	**0.008**	344.88	325.4	**335.14**

(c)

Table 5.15: A collection of tables summarises the values calculated from the combined elements Q matrices at different k for $k + r$ mapped distances.

In Figure 5.12 plots (c) to (f) show the bend point at $k = 5$ where average similarities are a maximum with minimum overlaps. For higher k the average similarities decrease extremely. At $k = 2$ the clusters are fully separated but the centroids of the clusters are not within the clusters. However at $k = 5$ the centroids are within the clusters. See Figures 5.12(a) and (b).

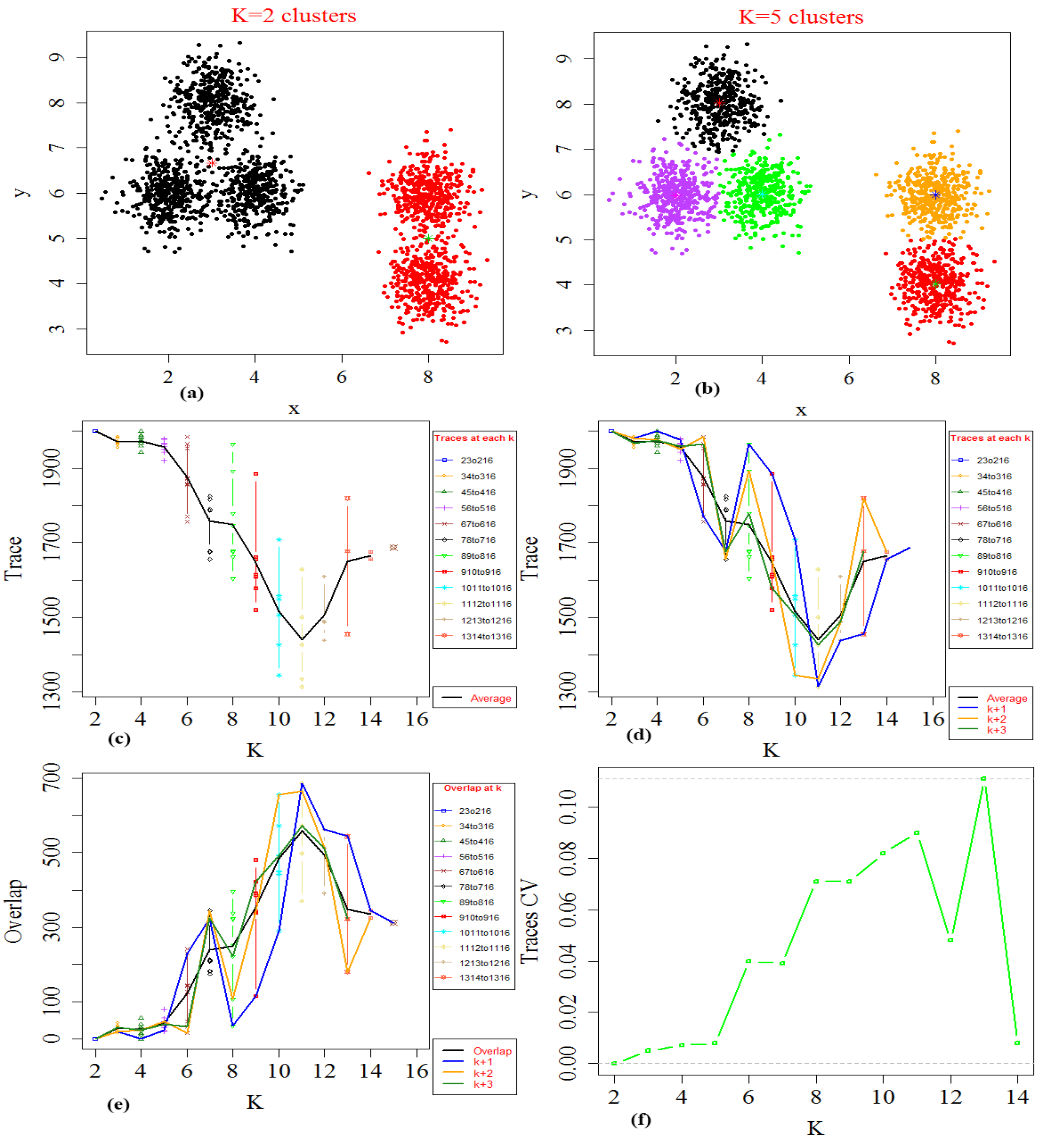


Figure 5.12: Plots (a) and (b) are the results from a *k-means* algorithm. Plots (c)-(f) show similarity, average similarity, overlap, average overlap and CV values.

In Figure 5.12 (d) and (f) are composite plots that show the difference between similarity and overlap for $k + r$ and average similarity and overlap with black solid lines at different k. Plot (f) shows the effect of CV and that clusters are stable until $k = 5$ with small perturbation.

1st run:			***Case3:* Existing indexes values**							
K	***Dunn***	***DH***	***DH Critical***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***Gap Critical***	***CCC***
2	**0.305**	0.287	0.594	7708.421	**0.693**	**0.488**	**1.01**	**1.118**	0.099	128.15
3	0.015	**1.65**	0.563	5254.897	0.502	0.557	1.5	1.069	-0.084	94.472
4	0.017	1.961	0.559	6428.824	0.625	0.554	1.428	1.195	0.75	137.595
5	0.005	5.614	0.492	4263.734	0.449	0.612	3.24	1.65	0.151	123.198
6	0.014	1.196	0.557	**8818.558**	0.531	0.868	2.873	0.812	-0.629	**146.846**
7	0.009	2.276	0.494	7932.678	0.461	1.034	2.876	1.46	0.074	140.398
8	0.007	2.28	0.51	7373.758	0.471	0.98	2.789	1.374	0.01	135.98
9	0.005	1.531	0.549	6975.85	0.411	1.043	2.899	1.393	0.093	131.231
10	0.011	1.53	0.557	6823.005	0.369	1.04	3.249	1.274	0.032	130.327
11	0.008	4.982	0.459	6274.201	0.366	0.966	4.313	1.25	-0.021	128.951
12	0.004	2.498	0.495	6087.401	0.37	1.105	3.067	1.179	0.005	122.306
13	0.008	2.543	0.458	6368.106	0.378	1.136	3.129	1.14	-0.142	125.014

(a)

Number of runs		***Dunn***	***DH***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***CCC***
2nd	***K***	**2**	**3**	**6**	**2**	**2**	**2**	**2**	**7**
	Values	**0.305**	**1.31**	**8742.476**	**0.693**	**0.488**	**1.009**	**1.122**	**140.065**
3rd	***K***	**2**	**2**	**5**	**2**	**2**	**2**	**7**	**5**
	Values	**0.305**	**0.993**	**10191.74**	**0.693**	**0.488**	**0.982**	**1.462**	**156.711**
4th	***K***	**2**	**3**	**5**	**2**	**2**	**2**	**5**	**6**
	Values	**0.305**	**1.442**	**10191.74**	**0.693**	**0.488**	**1.044**	**1.655**	**146.935**
5th	***K***	**2**	**2**	**5**	**2**	**2**	**2**	**2**	**6**
	Values	**0.306**	**2.326**	**10190.49**	**0.693**	**0.488**	**1.07**	**1.121**	**147.21**

(b)

Table 5.16: The summary of eight different indexes with 5 multiple runs. The optimal number of clusters is highlighted in bold.

From the above table estimated number of clusters specified by *Dunn*, *Sil*, *DB* and *SD* indexes are $= 2$, *CH* and *CCC* show 5 to 7 clusters , *DH* give 2 or 3 clusters and *Gap* values from $k = 2$ to $k = 7$. The new approach yields $k = 2$ or $k = 5$ as the best choice. It is up to the researcher to decide on selecting a full separated set of clusters or clusters with a small number of overlap. The new approach shows for this case the dataset has the potential to split the clusters with minimum overlap until $k = 5$.

5.3.4 Case4: Low Density Clusters

The scatter plot for the final case4 can be seen in the Figure 5.9(d) and shows no clustering structure due to the increase of extreme variation (standard deviation).

$k = 2, r = 1,2,\dots,K-k$

k	r	$(k,k+r)\times(k+r,k)$	Traces (similarity)	Overlap
2	1	(2,3×3,2)	1931.26	68.74
2	2	(2,4×4,2)	1745.04	254.96
2	3	(2,5×5,2)	1773.78	226.22
2	4	(2,6×6,2)	1822.52	177.48
2	5	(2,7×7,2)	1891.26	108.74
2	6	(2,8×8,2)	1842.52	157.48
2	7	(2,9×9,2)	1931.26	68.74
2	8	(2,10×10,2)	1813.78	186.22
2	9	(2,11×11,2)	1822.52	177.48
2	10	(2,12×12,2)	1862.52	137.48
2	11	(2,13×13,2)	1842.52	157.48
2	12	(2,14×14,2)	1882.52	117.48
2	13	(2,15×15,2)	1862.52	137.48
2	14	(2,16×16,2)	1882.52	117.48

$k = 3, r = 1,2,\dots,K-k$

k	r	$(k,k+r)\times(k+r,k)$	Traces (similarity)	Overlap
3	1	(3,4×4,3)	1636.53	363.47
3	2	(3,5×5,3)	1638.9	361.1
3	3	(3,6×6,3)	1689.49	310.51
3	4	(3,7×7,3)	1633.45	366.55
3	5	(3,8×8,3)	1547.76	452.24
3	6	(3,9×9,3)	1720.01	279.99
3	7	(3,10×10,3)	1590.84	409.16
3	8	(3,11×11,3)	1631.55	368.45
3	9	(3,12×12,3)	1718.66	281.34
3	10	(3,13×13,3)	1747.75	252.25
3	11	(3,14×14,3)	1735.58	264.42
3	12	(3,15×15,3)	1729.65	270.35
3	13	(3,16×16,3)	1741.35	258.65
⋮	⋮	⋮	⋮	⋮

$k = 15, r = 1$

k	r	$(k,k+r)\times(k+r,k)$	Traces (similarity)	Overlap
15	1	(15,16×16,15)	1674.24	325.76

(a)

$k = 2,3,..15, r = 1$

k	$(k,k+r)\times(k+r,k)$	Traces (similarity)	Overlap
2	(2,3×3,2)	1931.26	68.74
3	(3,4×4,3)	1636.53	363.47
4	(4,5×5,4)	1609.52	390.48
5	(5,6×6,5)	1588.33	411.67
6	(6,7×7,6)	1596.05	403.95
7	(7,8×8,7)	1539.08	460.92
8	(8,9×9,8)	1127.66	872.34
9	(9,10×10,9)	1132.08	867.92
10	(10,11×11,10)	1713.71	286.29
11	(11,12×12,11)	1413.45	586.55
12	(12,13×13,12)	1601.8	398.2
13	(13,14×14,13)	1654.05	345.95
14	(14,15×15,14)	1530.81	469.19
15	(15,16×16,15)	1674.24	325.76

$k = 2,3,..15, r = 2$

k	$(k,k+r)\times(k+r,k)$	Traces (similarity)	Overlap
2	(2,4×4,2)	1745.04	254.96
3	(3,5×5,3)	1638.9	361.1
4	(4,6×6,4)	1462.2	537.8
5	(5,7×7,5)	1462.67	537.33
6	(6,8×8,6)	1395.34	604.66
7	(7,9×9,7)	1466.52	533.48
8	(8,10×10,8)	1305.38	694.62
9	(9,11×11,9)	1269.48	730.52
10	(10,12×12,10)	1268.41	731.59
11	(11,13×13,11)	1549	451
12	(12,14×14,12)	1449.41	550.59
13	(13,15×15,13)	1496.09	503.91
14	(14,16×16,14)	1663.31	336.69
⋮	⋮	⋮	⋮

$k = 2, r = 14$

k	$(k,k+r)\times(k+r,k)$	Traces (similarity)	Overlap
2	(2,16×16,2)	1663.31	336.69

(b)

Similarity				Overlap			
K	Max Trace	Min Trace	Average Traces	CV	Max Overlap	Min Overlap	Average Overlap
2	1931.26	1745.04	1850.467	0.029	254.96	68.74	149.533
3	1747.75	1547.76	1673.963	0.039	452.24	252.25	326.037
4	1732.98	1380.34	1556.387	0.073	619.66	267.02	443.613
5	1602.19	1341.95	1495.288	0.068	658.05	397.81	504.712
6	1596.05	1314.34	1463.978	0.054	685.66	403.95	536.022
7	1539.08	1294.42	1390.814	0.057	705.58	460.92	609.186
8	1361.91	1127.66	1290.807	0.056	872.34	638.09	709.192
9	1338.39	1132.08	1254.973	0.054	867.92	661.61	745.027
10	1713.71	1268.41	1419.633	0.108	731.59	286.29	580.367
11	1549	1396.15	1454.294	0.041	603.85	451	545.706
12	1601.8	1283.44	1405.15	0.108	716.56	398.2	594.85
13	1654.05	1496.09	1549.867	0.058	503.91	345.95	450.133
14	1663.31	1530.81	1597.06	0.059	469.19	336.69	402.94

(c)

Table 5.17: A collection of tables summarise the values calculated from Q matrices at different k with different $k + r$.

Table (a) above shows the similarity values at fixed $k = 2$ is higher with less variation than the values at fixed $k = 3$ for different $k + r$ mapping distances. Table (b) shows the values at different k with fixed $k + r$ and similarity values at $k = 2$ is a maximum with minimum overlap for $k + 1$ mapped distances compared to any other k values. From the table (c) the average similarity is a maximum with minimum average overlap when $k = 2$. The clusters at $k = 2$ indicate CV is a minimum and clusters are stable. In this case the new approach suggests cluster structure is varied and reduced as the standard deviation increases. Thus $k = 5$ is not the best option as a large number of elements overlap.

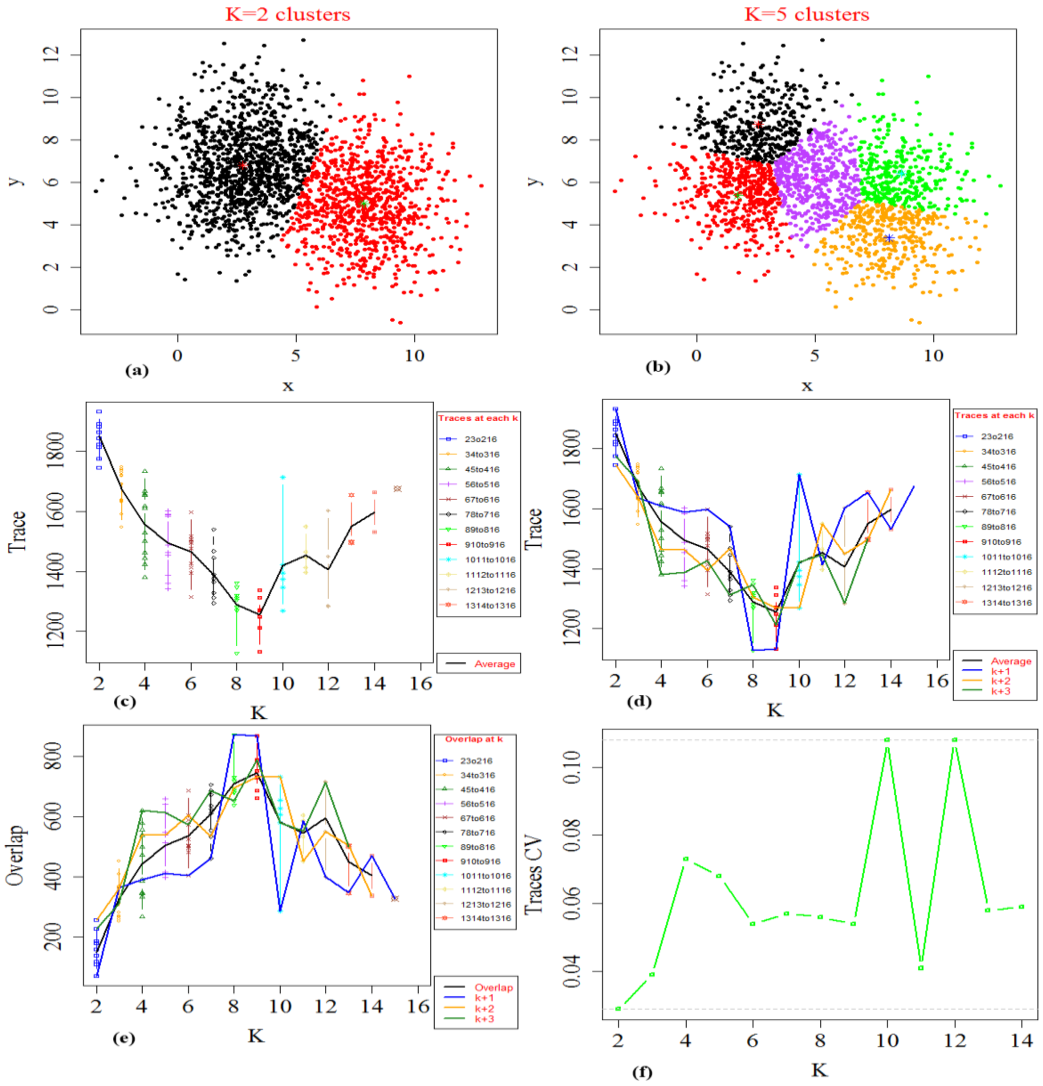


Figure 5.13: Plots (a) and (b) show the number of clusters obtained from a *k-means* clustering algorithm at $k = 2$ and $k = 5$. Plots (c)-(f) show values computed from Q matrices.

In the Figure 5.13 plots (a) and (b) show clusters when $k = 2$ and $k = 5$ obtained from a *k-means* algorithm and memberships of element and centroids of clusters are labelled in different colours. The similarity values at each k in plot (c) for different $k + r$ are indicated with different colours while the black solid line shows the average similarity at different k which is a maximum at $k = 2$ with minimum average overlap and CV. Using 1 criterion mentioned Chapter 4 section 4.3.3 the estimated number of clusters is 2. Plots (d) and (e) illustrate and show differences in a composite graph at different k for $k + r$ mapped distances while (f) indicates CV.

1^{st} run:	*Case4:* **Existing indexes values**								
K	***Dunn***	***DH***	***DH Critical***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***Gap Critical***	***CCC***
2	0.004	**0.91**	0.598	**2462.012**	**0.472**	**0.892**	0.917	**1.249**	0.149	**75.886**
3	0.005	1.622	0.577	1899.543	0.383	1.107	1.055	1.131	0.189	75.781
4	0.004	1.26	0.573	1654.886	0.333	1.124	0.982	0.951	-0.012	67.255
5	0.007	1.325	0.564	1787.546	0.338	1.04	**0.91**	0.97	0.067	64.105
6	0.009	1.265	0.55	1722.885	0.338	1.062	0.996	0.913	0.014	59.848
7	0.002	1.594	0.527	1603.9	0.328	1.062	1.006	0.916	0.038	57.529
8	**0.01**	2.221	0.513	1669.218	0.329	1.017	1.038	0.897	0.081	55.085
9	0.005	1.032	0.525	1619.699	0.315	0.978	1.082	0.846	0.011	53.945
10	0.01	1.406	0.488	1629.253	0.33	0.93	1.072	0.856	0.062	53.563
11	0.006	1.164	0.515	1579.431	0.316	1.012	1.117	0.807	0.034	52.653
12	0.01	1.492	0.52	1617.854	0.328	0.937	1.252	0.842	0.025	52.965
13	0.006	1.881	0.491	1592.435	0.328	0.954	1.253	0.815	0.06	51.724

(a)

Number of runs		***Dunn***	***DH***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***CCC***
2nd	***K***	**14**	**2**	**2**	**2**	**2**	**5**	**2**	**3**
	Values	**0.012**	**1.386**	**2462.012**	**0.472**	**0.892**	**0.947**	**1.25**	**76.064**
3rd	***K***	**15**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
	Values	**0.01**	**0.962**	**2462.012**	**0.472**	**0.892**	**0.882**	**1.266**	**75.886**
4th	***K***	**16**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
	Values	**0.012**	**1.37**	**2462.012**	**0.472**	**0.892**	**0.897**	**1.289**	**75.886**
5th	***K***	**13**	**2**	**2**	**2**	**2**	**2**	**2**	**3**
	Values	**0.008**	**1.31**	**2462.203**	**0.472**	**0.892**	**0.876**	**1.246**	**76.068**

(b)

Table 5.18: The summary of eight different indexes with 5 multiple runs. The optimal number of clusters is highlighted in bold.

For case4 results *Dunn* varies from $k = 8$ to $k = 16$ clusters, *CCC* and *SD* suggest clusters $k = 2$ to $k = 5$ clusters while *DH*, *CH*, *Sil*, *DB* and *Gap* consistently suggest $k = 2$. These later indexes agree with the new approach suggesting $k = 2$. The other indexes do not make this choice and do not work well with severe noise or large datasets.

5.4 Type3 Datasets: Mixture of Clusters

In this section four more cases (case1, case2, case3 and case4) with different and less constrained structure of datasets are presented to strengthen the case for the proposed approach. These datasets are collected from two different sources where clusters in all the cases are of different sizes, shapes, densities and structures. The first two cases are sourced from [165, 132] and used to find the clustering structure based on a new non-metric symmetry distance [165] and to determine the best number of clusters when there are widely different sizes and densities. The latter two cases are sourced from [166] and generated in R using random normal variates with different means and standard deviations to form varied size and shape of clusters. The performance of the new approach is also compared with other existing approaches based on 5 simulated runs. The case1 and case2 datasets are a mixture of spherical and elliptical clusters and include a total of 577 and 850 elements (N) in each case respectively. The case3 dataset consists of seven well separated groups in spherical shape each with different size and having high and medium density with a total of 1400 elements, while case4 has a total of 1350 elements and nine square shaped clusters that are connected at their corners.

5.4.1 Case1: Three Clusters Mixture of Spherical and Elliptical Shapes

The scatter plot of a case1 dataset in Figure 5.14(a) shows a mixture of spherical and elliptical clusters of medium density. Table 5.19(a) shows small variation between similarity at $k = 2$ and $k = 3$ with different $k + r$ mapped distances. Table 5.19(b) shows the similarity and overlap at different k for some $k + r$ $(k + 1 \,\&\, k + 2)$ adjacent and non-adjacent mapping. At $k = 3$ and $k + 1$ adjacent the similarity is a maximum with minimum overlap. Table 5.19(c) indicates maximum, minimum, similarity, overlap, average similarity and overlap with coefficient of variation (CV).

The average similarity maximum up to $k = 3$ with minimum average overlap indicates 3 is the best number of clusters. In this case results show when clusters are of medium density and different size adjacent mapping of clusters also provides better understating of the cluster structure. For example at $k = 3$ with $k + 1$ mapping similarity is $565.3/577 = 97.9$ % with minimum overlap between clusters.

$k = 2, r = 1,2, \dots, K - k$

k	r	$(k, k + r) \times (k + r, k)$	***Traces (similarity)***	***Overlap***
2	1	(2,3×3,2)	**559.91**	**17.09**
2	2	(2,4×4,2)	**554.14**	**22.86**
2	3	(2,5×5,2)	**546.52**	**30.48**
2	4	(2,6×6,2)	**538.9**	**38.1**
2	5	(2,7×7,2)	**538.9**	**38.1**
2	6	(2,8×8,2)	**538.9**	**38.1**
2	7	(2,9×9,2)	**544.67**	**32.33**
2	8	(2,10×10,2)	**544.67**	**32.33**
2	9	(2,11×11,2)	**554.14**	**22.86**
2	10	(2,12×12,2)	**554.14**	**22.86**
2	11	(2,13×13,2)	**554.14**	**22.86**
2	12	(2,14×14,2)	**559.91**	**17.09**
2	13	(2,15×15,2)	**554.14**	**22.86**
2	14	(2,16×16,2)	**559.91**	**17.09**

$k = 3, r = 1,2, \dots, K - k$

k	r	$(k, k + r) \times (k + r, k)$	***Traces (similarity)***	***Overlap***
3	1	(3,4×4,3)	**565.3**	**11.7**
3	2	(3,5×5,3)	**540.29**	**36.71**
3	3	(3,6×6,3)	**544**	**33**
3	4	(3,7×7,3)	**547.9**	**29.1**
3	5	(3,8×8,3)	**547.93**	**29.07**
3	6	(3,9×9,3)	**547.93**	**29.07**
3	7	(3,10×10,3)	**549.77**	**27.23**
3	8	(3,11×11,3)	**549.99**	**27.01**
3	9	(3,12×12,3)	**551.83**	**25.17**
3	10	(3,13×13,3)	**551.83**	**25.17**
3	11	(3,14×14,3)	**549.99**	**27.01**
3	12	(3,15×15,3)	**549.99**	**27.01**
3	13	(3,16×16,3)	**563.43**	**13.57**
⋮	⋮	⋮	⋮	⋮

$k = 15, r = 1$

k	r	$(k, k + r) \times (k + r, k)$	***Traces (similarity)***	***Overlap***
15	1	(15,16×16,15)	**477.7**	**99.3**

(a)

$k = 2,3, \dots 15, r = 1$

k	$(k, k + r) \times (k + r, k)$	***Traces (similarity)***	***Overlap***
2	(2,3×3,2)	**559.91**	**17.09**
3	(3,4×4,3)	**565.3**	**11.7**
4	(4,5×5,4)	**548.45**	**28.55**
5	(5,6×6,5)	**547.1**	**29.9**
6	(6,7×7,6)	**529.81**	**47.19**
7	(7,8×8,7)	**526.62**	**50.38**
8	(8,9×9,8)	**526.9**	**50.1**
9	(9,10×10,9)	**549.04**	**27.96**
10	(10,11×11,10)	**522.17**	**54.83**
11	(11,12×12,11)	**535.03**	**41.97**
12	(12,13×13,12)	**482.19**	**94.81**
13	(13,14×14,13)	**479.16**	**97.84**
14	(14,15×15,14)	**526.33**	**50.67**
15	(15,16×16,15)	**477.7**	**99.3**

$k = 2,3, \dots 15, r = 2$

k	$(k, k + r) \times (k + r, k)$	***Traces (similarity)***	***Overlap***
2	(2,4×4,2)	**554.14**	**22.86**
3	(3,5×5,3)	**540.29**	**36.71**
4	(4,6×6,4)	**553.89**	**23.11**
5	(5,7×7,5)	**533.37**	**43.63**
6	(6,8×8,6)	**502.32**	**74.68**
7	(7,9×9,7)	**480.12**	**96.88**
8	(8,10×10,8)	**498.94**	**78.06**
9	(9,11×11,9)	**496.05**	**80.95**
10	(10,12×12,10)	**487.44**	**89.56**
11	(11,13×13,11)	**523.69**	**53.31**
12	(12,14×14,12)	**550.49**	**26.51**
13	(13,15×15,13)	**463.6**	**113.4**
14	(14,16×16,14)	**479.28**	**97.72**
⋮	⋮	⋮	⋮

$k = 2, r = 14$

k	$(k, k + r) \times (k + r, k)$	***Traces (similarity)***	***Overlap***
2	(2,16×16,2)	**559.91**	**17.09**

(b)

Similarity				***Overlap***			
K	***Max Trace***	***Min Trace***	***Average Traces***	***CV***	***Max Overlap***	***Min Overlap***	***Average Overlap***
2	559.91	538.9	**550.214**	**0.014**	38.1	17.09	**26.786**
3	565.3	540.29	**550.783**	**0.012**	36.71	11.7	**26.217**
4	554.47	513.59	**531.353**	**0.029**	63.41	22.53	**45.647**
5	547.1	476.9	**514.487**	**0.045**	100.1	29.9	**62.513**
6	529.81	445.58	**489.545**	**0.055**	131.42	47.19	**87.455**
7	526.62	452.01	**490.011**	**0.043**	124.99	50.38	**86.989**
8	526.9	476.12	**496.485**	**0.04**	100.88	50.1	**80.515**
9	549.04	445.97	**472.713**	**0.08**	131.03	27.96	**104.287**
10	522.17	443.79	**473.725**	**0.059**	133.21	54.83	**103.275**
11	535.03	460.51	**504.354**	**0.059**	116.49	41.97	**72.646**
12	550.49	482.19	**510.373**	**0.063**	94.81	26.51	**66.628**
13	479.16	449.7	**464.153**	**0.032**	127.3	97.84	**112.847**
14	526.33	479.28	**502.805**	**0.066**	97.72	50.67	**74.195**

(c)

Table 5.19: A collection of tables show the values calculated form Q matrices at different k with $k + r$ mapping distance.

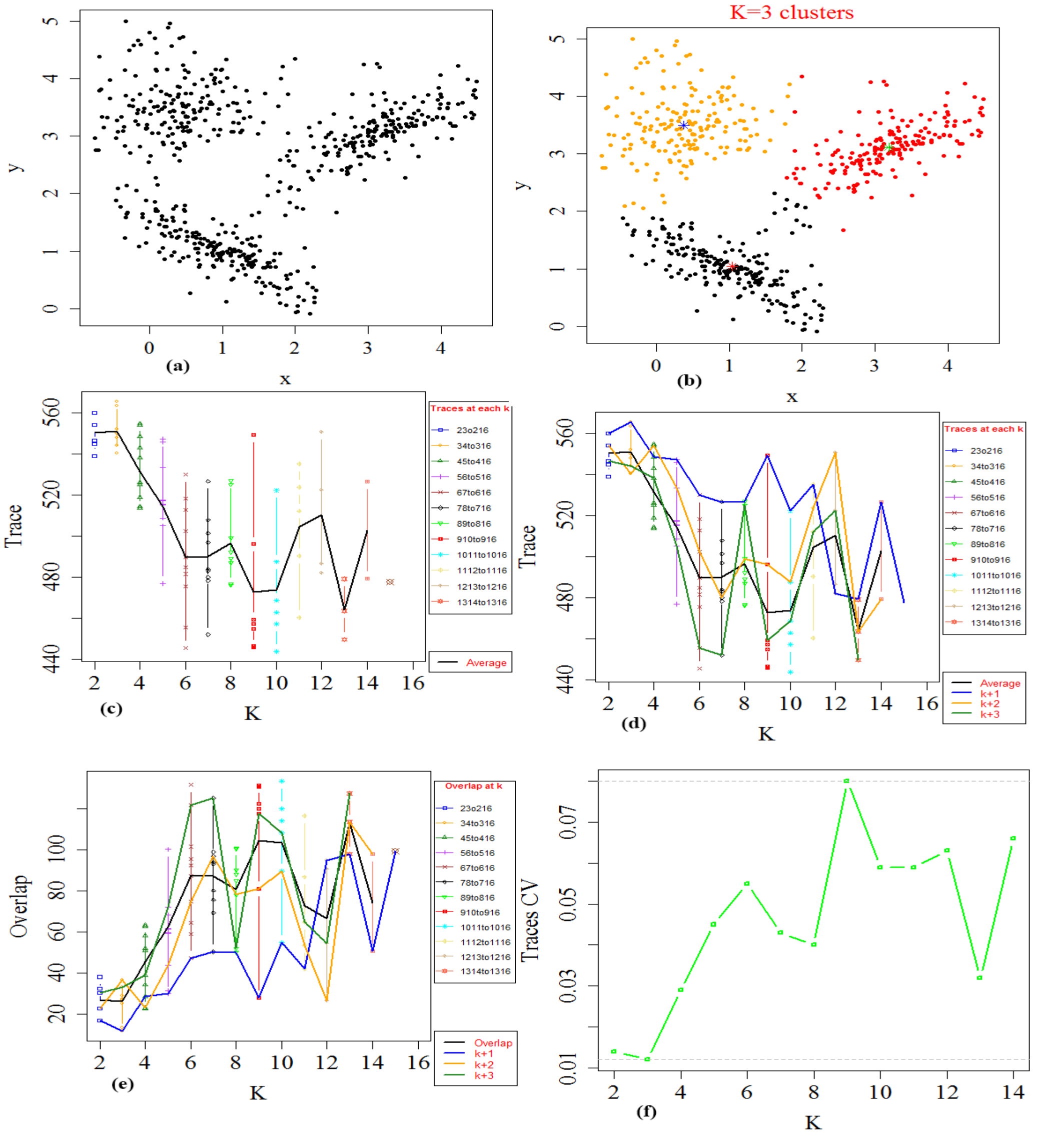


Figure 5.14: Plot (a) shows the scatter plot for case1 dataset while (b) shows $k = 3$ clusters obtained from a *k-means* algorithm. Membership of different clusters is shown by different colours. Plots (c)-(f) show values computed from combined mapped elements Q matrices.

In the figure above plot (c) indicates similarity at different k for $k + r$ mapped distance while the solid black line in the plot shows average similarity at different k and the estimated number of clusters is three using criterion 2 mentioned in Chapter 4 section 4.3.3. Plot (d) represents similarity with adjacent and non-adjacent $k + r$ for $(r = 1{,}2{,}3)$ with average similarity. The blue line in plot (d) with $k + 1$ mapped distance indicates $k = 3$ is a maximum for estimating the number of clusters in this case. Plot (e) shows the difference between overlap and average overlap between clusters at different k for $k + r$. Plot (f) shows the coefficient of variation is a minimum and clusters are stable at the best $\boldsymbol{K}$.

1				*Case1:* **Existing indexes values**						
K	***Dunn***	***DH***	***DH Critical***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***Gap Critical***	***CCC***
2	0.021	0.283	0.557	515.504	0.474	0.856	2.932	0.489	-0.491	21.605
3	**0.065**	**0.657**	**0.501**	**1204.32**	**0.617**	**0.599**	**1.589**	**1.042**	**0.262**	**46.128**
4	0.022	1.136	0.497	1035.799	0.525	0.748	2.565	0.717	0.032	40.113
5	0.008	1.555	0.448	1074.467	0.492	0.834	2.296	0.912	0.034	34.08
6	0.018	2.462	0.434	976.105	0.405	0.932	2.594	0.904	0.049	38.317
7	0.012	6.118	0.336	994.602	0.394	0.897	2.541	0.845	-0.028	36.595
8	0.006	0.451	0.441	1036.139	0.421	0.955	3.258	0.767	0.016	36.716
9	0.011	0.421	0.424	994.073	0.398	0.895	3.074	0.789	-0.026	35.658
10	0.013	3.489	0.198	1105.32	0.393	0.941	2.94	0.744	-0.148	37.345
11	0.014	1.305	0.301	1121.637	0.383	0.9	4.421	0.951	0.082	33.212
12	0.015	0.674	0.322	984.951	0.391	0.929	3.909	0.889	-0.015	35.554
13	0.015	1.246	0.444	1040.214	0.369	0.915	3.72	0.904	-0.006	36.676

(a)

Number of runs		***Dunn***	***DH***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***CCC***
2nd	***K***	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
	Values	**0.065**	**0.657**	**1204.32**	**0.617**	**0.599**	**1.554**	**1.007**	**46.128**
3rd	***K***	**3**	**2**	**3**	**3**	**3**	**3**	**3**	**3**
	Values	**0.065**	**2.207**	**1204.32**	**0.617**	**0.599**	**1.475**	**1.026**	**46.128**
4th	***K***	**3**	**2**	**3**	**3**	**3**	**3**	**3**	**3**
	Values	**0.065**	**2.225**	**1204.32**	**0.617**	**0.599**	**1.362**	**1.001**	**46.128**
5th	***K***	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
	Values	**0.065**	**1.934**	**1204.32**	**0.617**	**0.599**	**1.691**	**1.031**	**46.128**

(b)

Table 5.20: Represents the optimal numbers of clusters with their values highlighted in bold from different indexes with 5 simulated runs.

The results in the above tables show for each run only *DH* varies and suggests 2 or 3 is best number of cluster while all other indexes agree with each other that $k = 3$ is the correct number of clusters. The new approach determines the correct number of clusters as $k = 3$ as discussed above, and in agreement with all the indexes except *DH* based on 5 runs.

5.4.2 Case2: Five Clusters Mixture of Spherical and Elliptical Shapes

This case includes a 273 extra elements compared to the previous case of type3 with mixed densities, different sizes and shapes. The scatter plot in Figure 5.15(a) shows the dataset has a mixture of elliptical and spherical clusters. Table 5.21(a) shows the similarity and overlap at fixed k for different $k + r$ mapped distances while Table 5.21(b) represents these values at different k for fixed $k + r$ e.g. $k + 1$ etc. Table 5.21(c) shows the maximum, minimum, similarity and overlap, average similarity and overlap with CV values.

$k = 2, r = 1,2, \dots, K - k$

k	r	$(k, k+r) \times (k+r, k)$	*Traces (similarity)*	*Overlap*
2	1	(2,3×3,2)	**774.76**	**75.24**
2	2	(2,4×4,2)	**671.66**	**178.34**
2	3	(2,5×5,2)	**791.76**	**58.24**
2	4	(2,6×6,2)	**791.76**	**58.24**
2	5	(2,7×7,2)	**791.76**	**58.24**
2	6	(2,8×8,2)	**791.76**	**58.24**
2	7	(2,9×9,2)	**791.76**	**58.24**
2	8	(2,10×10,2)	**795.38**	**54.62**
2	9	(2,11×11,2)	**808.76**	**41.24**
2	10	(2,12×12,2)	**808.76**	**41.24**
2	11	(2,13×13,2)	**808.76**	**41.24**
2	12	(2,14×14,2)	**808.76**	**41.24**
2	13	(2,15×15,2)	**812.38**	**37.62**
2	14	(2,16×16,2)	**808.76**	**41.24**

$k = 3, r = 1,2, \dots, K - k$

k	r	$(k, k+r) \times (k+r, k)$	*Traces (similarity)*	*Overlap*
3	1	(3,4×4,3)	**644.42**	**205.58**
3	2	(3,5×5,3)	**751.83**	**98.17**
3	3	(3,6×6,3)	**760.33**	**89.67**
3	4	(3,7×7,3)	**765.7**	**84.3**
3	5	(3,8×8,3)	**757.39**	**92.61**
3	6	(3,9×9,3)	**759.44**	**90.56**
3	7	(3,10×10,3)	**773.31**	**76.69**
3	8	(3,11×11,3)	**789.34**	**60.66**
3	9	(3,12×12,3)	**789.34**	**60.66**
3	10	(3,13×13,3)	**789.34**	**60.66**
3	11	(3,14×14,3)	**789.34**	**60.66**
3	12	(3,15×15,3)	**798.81**	**51.19**
3	13	(3,16×16,3)	**796.76**	**53.24**
⋮	⋮	⋮	⋮	⋮

$k = 15, r = 1$

k	r	$(k, k+r) \times (k+r, k)$	*Traces (similarity)*	*Overlap*
15	1	(15,16×16,15)	**764.35**	**85.65**

(a)

$k = 2,3, ..15, r = 1$

k	$(k, k+r) \times (k+r, k)$	*Traces (similarity)*	*Overlap*
2	(2,3×3,2)	**774.76**	**75.24**
3	(3,4×4,3)	**644.42**	**205.58**
4	(4,5×5,4)	**803.5**	**46.5**
5	(5,6×6,5)	**822.11**	**27.89**
6	(6,7×7,6)	**816.26**	**33.74**
7	(7,8×8,7)	**817.76**	**32.24**
8	(8,9×9,8)	**822**	**28**
9	(9,10×10,9)	**784.12**	**65.88**
10	(10,11×11,10)	**807.92**	**42.08**
11	(11,12×12,11)	**850**	**0**
12	(12,13×13,12)	**819.19**	**30.81**
13	(13,14×14,13)	**846.03**	**3.97**
14	(14,15×15,14)	**631.89**	**218.11**
15	(15,16×16,15)	**764.35**	**85.65**

$k = 2,3, ..15, r = 2$

k	$(k, k+r) \times (k+r, k)$	*Traces (similarity)*	*Overlap*
2	(2,4×4,2)	**671.66**	**178.34**
3	(3,5×5,3)	**751.83**	**98.17**
4	(4,6×6,4)	**794.7**	**55.3**
5	(5,7×7,5)	**789.04**	**60.96**
6	(6,8×8,6)	**783.93**	**66.07**
7	(7,9×9,7)	**815.16**	**34.84**
8	(8,10×10,8)	**762.98**	**87.02**
9	(9,11×11,9)	**756.86**	**93.14**
10	(10,12×12,10)	**807.92**	**42.08**
11	(11,13×13,11)	**846.37**	**3.63**
12	(12,14×14,12)	**814.6**	**35.4**
13	(13,15×15,13)	**691.39**	**158.61**
14	(14,16×16,14)	**758.66**	**91.34**
⋮	⋮	⋮	⋮

$k = 2, r = 14$

k	$(k, k+r) \times (k+r, k)$	*Traces (similarity)*	*Overlap*
2	(2,16×16,2)	**808.76**	**41.24**

(b)

Similarity				*Overlap*			
K	***Max Trace***	***Min Trace***	***Average Traces***	***CV***	***Max Overlap***	***Min Overlap***	***Average Overlap***
2	812.38	671.66	**789.77**	**0.045**	178.34	37.62	**60.23**
3	798.81	644.42	**766.565**	**0.052**	205.58	51.19	**83.435**
4	830.55	794.7	**821.562**	**0.013**	55.3	19.45	**28.438**
5	822.11	789.04	**800.747**	**0.012**	60.96	27.89	**49.253**
6	816.26	769.18	**788.794**	**0.024**	80.82	33.74	**61.206**
7	817.76	752.83	**781.319**	**0.035**	97.17	32.24	**68.681**
8	822	730.77	**759.651**	**0.043**	119.23	28	**90.349**
9	789.87	751.59	**767.82**	**0.021**	98.41	60.13	**82.18**
10	807.92	762.22	**796.205**	**0.022**	87.78	42.08	**53.795**
11	850	723.04	**802.956**	**0.075**	126.96	0	**47.044**
12	819.19	703.41	**765.903**	**0.078**	146.59	30.81	**84.097**
13	846.03	691.39	**764.97**	**0.101**	158.61	3.97	**85.03**
14	758.66	631.89	**695.275**	**0.129**	218.11	91.34	**154.725**

(c)

Table 5.21: A collection of tables show the values calculated form Q matrices at different k with $k + r$ mapped distances.

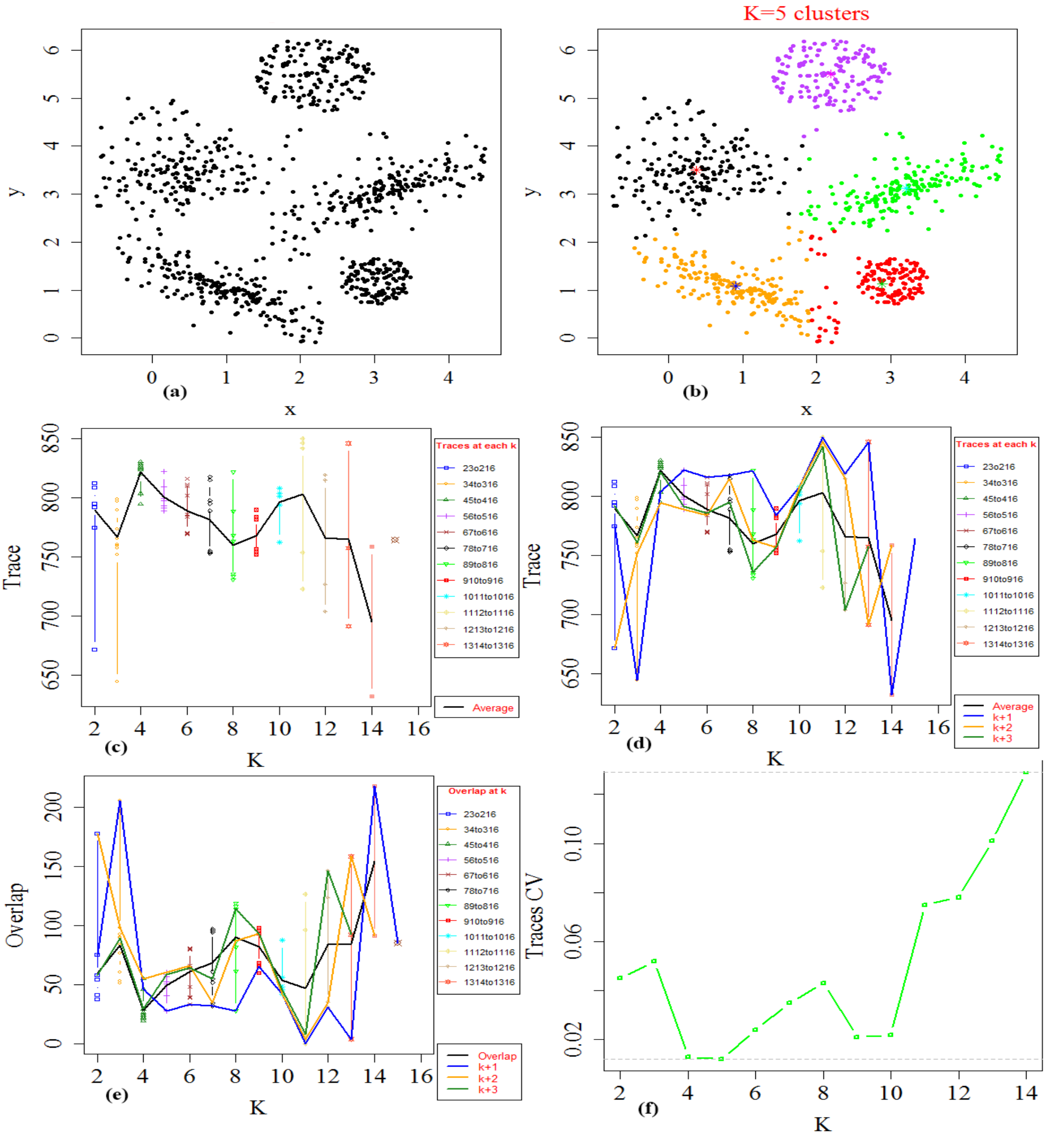


Figure 5.15: Plot (a) shows the scatter plot for case2 dataset while (b) shows the clusters obtained from a *k-means* algorithm and labels the membership of these clusters with different colours. Plots (c)-(f) show values computed from combined mapped elements Q matrices.

In the figure above Plot (c) shows the similarity at each k with different $k + r$ and the solid black line shows the values for average similarity at different k. Plots (d) and (e) show the difference between similarity average similarity and likewise overlap and average overlap values at different k with coloured $k + r$ mapped distances. Plot (f) shows coefficients of variation at different k.

1st run: *Case2:* **Existing indexes values**

K	*Dunn*	*DH*	*DH Critical*	*CH*	*Sil*	*DB*	*SD*	*Gap*	*Gap Critical*	*CCC*
2	0.016	**0.652**	0.572	711.829	0.409	1.045	3.03	0.281	-0.067	27.37
3	0.017	1.099	0.506	822.691	0.451	0.85	1.691	0.36	-0.124	31.739
4	0.007	2.686	0.482	1162.633	0.525	0.7	1.565	0.504	-0.318	38.801
5	0.014	1.796	0.499	**1663.781**	**0.581**	**0.622**	**1.49**	**0.858**	0.04	**48.19**
6	0.016	0.666	0.423	1500.568	0.392	0.697	2.478	0.731	-0.149	46.596
7	0.011	0.919	0.498	1325.797	0.489	0.735	3.139	0.738	-0.054	44.584
8	0.019	2.622	0.458	1162.806	0.49	0.925	3.162	0.586	-0.043	46.299
9	0.012	0.922	0.481	1537.884	0.494	0.876	2.532	0.867	0.042	45.156
10	0.015	3.841	0.404	1268.363	0.439	0.952	3.187	0.551	-0.261	42.571
11	**0.019**	2.744	0.346	1389.929	0.378	0.999	3.211	0.801	0.069	42.898
12	0.011	1.085	0.424	1563.898	0.435	0.961	3.614	0.792	0.137	43.442
13	0.019	2.507	0.352	1508.67	0.429	0.971	2.996	0.816	-0.045	42.471

(a)

Number of runs		*Dunn*	*DH*	*CH*	*Sil*	*DB*	*SD*	*Gap*	*CCC*
2nd	*K*	4	2	5	5	4	5	5	5
	Values	0.036	0.837	1663.781	0.581	0.679	1.367	0.866	48.19
3rd	*K*	14	3	5	5	5	4	5	5
	Values	0.024	2.33	1663.781	0.581	0.622	1.418	0.853	48.19
4th	*K*	15	4	5	5	5	5	2	15
	Values	0.024	1.122	1663.781	0.581	0.622	1.424	0.365	44.945
5th	*K*	4	2	5	5	5	5	2	5
	Values	0.036	0.82	1663.781	0.581	0.622	1.495	0.292	48.19

(b)

Table 5.22: The summary of eight different indexes with 5 simulated runs. The optimal number of clusters is highlighted in bold.

In the above tables results show *Dunn* and *DH* indexes differ from the other indexes. *Dunn* suggests between 4 and 15 while *DH* suggests between 2 and 3 clusters. The *CH* and *Sil* determine 5 as the correct number of clusters and this agrees with the known number. *DB, SD* and *CCC* indexes frequently find the correct number of clusters while *Gap* estimate number of clusters 2 or 5. These results show these indexes were inconsistent. This implies the need to run these indexes multiple times for determining the best number of clusters. The result obtained by the new approach represents values of average similarity is a maximum at $k = 4$ while $k = 5$ gives the second highest value for these two k as shown in the Figure 5.15(c).This indicates clusters are stable with small perturbation at $k = 4$ and $k = 5$. This agrees with the minimum CV values as can be seen in the Figure 5.15(f). In this case, we examine the plots carefully with the values of similarity and average similarity at $k = 4$ and $k = 5$ (criteria 2 from Chapter 4 in section 4.3.3). Similarity is a maximum with minimum overlap and minimum coefficient of variation at $k = 4$ or $k = 5$ compared to other values of k clearly can see in the Figure 5.15 (d) and (f). The average similarity at $k = 4$ and $k = 5$ is sligthty different due to the effect of large and small variation between clusters as seen in the Figure 5.15 (b). Here we find similarity at $k = 5$ is a maximum and this is an indication of the number of clusters with the adjacent $k + 1$ mapped distance. Although for higher k some trace (similarity) values with $k + 1$ are high this shows in forward and backward mapping that just changing a few elements between clusters can increase the similarity for the higher k. This indicates that smaller size clusters (few elements) may represent noise in data as seen in Figure 5.15(b), cluster with memberships in red colour belongs to clusters memberships in orange and green colours. However, in these circumstances, we should also consider minimum difference between average similarity and similarity for (adjacent and non-adjacent) mapped distances. The plots in the Figure 5.15(d) visually showed the difference between similarity (blue solid line) and average similarity (black solid line) at $k = 4$ and $k = 5$ with $k + 1$ is quite small. Minimum CV also supports the conclusion that the dataset consists of 4 or 5 clusters which are stable at the best $\boldsymbol{K}$. This is one scenario which may need to be treated cautiously by examining the plots in Figure 5.15 for exploring and understanding the dataset.

5.4.3 Case3: Seven Spherical Clusters of Different Sizes and Density

The dataset in this case is used to illustrate the behaviour of the new approach for a mixture of large and small size spherical clusters with high and medium densities. The dataset consist of $N = 1400$ elements and the scatter plot of case3 can be seen in Figure 5.16(a). Table 5.23(a) shows the similarity and overlap at fixed k with different $k+r$. It shows for the initial value of k the number of elements overlapping is higher than other values of k for $k+r$ as the clusters split the overlap decreases and elements in the clusters are settled at $k=7$. Table 5.23(b) also shows the same situation for similarity and overlap value at different k. Table 5.23(c) represents small perturbations for average similarity and overlap values at $k=4$ to $k=7$, clusters are fully separated at $k=7$ with 0 elements overlapping. This indicates the best $\boldsymbol{K}$ number of clusters (criteria 4 Chapter 4 section 4.3.3) and clusters are stable as the CV value is 0 at $k=7$.

$k = 2, r = 1,2,\dots,K-k$

k	r	$(k,k+r)\times(k+r,k)$	*Traces (similarity)*	*Overlap*
2	1	(2,3×3,2)	**1386**	**14**
2	2	(2,4×4,2)	**1002**	**398**
2	3	(2,5×5,2)	**1386**	**14**
2	4	(2,6×6,2)	**1400**	**0**
2	5	(2,7×7,2)	**1400**	**0**
2	6	(2,8×8,2)	**1400**	**0**
2	7	(2,9×9,2)	**1400**	**0**
2	8	(2,10×10,2)	**1400**	**0**
2	9	(2,11×11,2)	**1400**	**0**
2	10	(2,12×12,2)	**1400**	**0**
2	11	(2,13×13,2)	**1400**	**0**
2	12	(2,14×14,2)	**1400**	**0**
2	13	(2,15×15,2)	**1400**	**0**
2	14	(2,16×16,2)	**1400**	**0**

$k = 3, r = 1,2,\dots,K-k$

k	r	$(k,k+r)\times(k+r,k)$	*Traces (similarity)*	*Overlap*
3	1	(3,4×4,3)	**998.06**	**401.94**
3	2	(3,5×5,3)	**1400**	**0**
3	3	(3,6×6,3)	**1388**	**12**
3	4	(3,7×7,3)	**1388**	**12**
3	5	(3,8×8,3)	**1388**	**12**
3	6	(3,9×9,3)	**1388**	**12**
3	7	(3,10×10,3)	**1388**	**12**
3	8	(3,11×11,3)	**1388**	**12**
3	9	(3,12×12,3)	**1388**	**12**
3	10	(3,13×13,3)	**1388**	**12**
3	11	(3,14×14,3)	**1388**	**12**
3	12	(3,15×15,3)	**1388**	**12**
3	13	(3,16×16,3)	**1388**	**12**
⋮	⋮	⋮	⋮	⋮

$k = 15, r = 1$

k	r	$(k,k+r)\times(k+r,k)$	*Traces (similarity)*	*Overlap*
15	1	(15,16×16,15)	**1118.66**	**281.34**

(a)

$k = 2,3,..15, r = 1$

k	$(k,k+r)\times(k+r,k)$	*Traces (similarity)*	*Overlap*
2	(2,3×3,2)	**1386**	**14**
3	(3,4×4,3)	**998.06**	**401.94**
4	(4,5×5,4)	**1400**	**0**
5	(5,6×6,5)	**1392**	**8**
6	(6,7×7,6)	**1400**	**0**
7	(7,8×8,7)	**1400**	**0**
8	(8,9×9,8)	**1370.54**	**29.46**
9	(9,10×10,9)	**1306.1**	**93.9**
10	(10,11×11,10)	**1237.64**	**162.36**
11	(11,12×12,11)	**1271.62**	**128.38**
12	(12,13×13,12)	**1114.74**	**285.26**
13	(13,14×14,13)	**1159.36**	**240.64**
14	(14,15×15,14)	**1079.74**	**320.26**
15	(15,16×16,15)	**1118.66**	**281.34**

$k = 2,3,..15, r = 2$

k	$(k,k+r)\times(k+r,k)$	*Traces (similarity)*	*Overlap*
2	(2,4×4,2)	**1002**	**398**
3	(3,5×5,3)	**1400**	**0**
4	(4,6×6,4)	**1400**	**0**
5	(5,7×7,5)	**1392**	**8**
6	(6,8×8,6)	**1400**	**0**
7	(7,9×9,7)	**1400**	**0**
8	(8,10×10,8)	**1341.08**	**58.92**
9	(9,11×11,9)	**1330.02**	**69.98**
10	(10,12×12,10)	**1281.16**	**118.84**
11	(11,13×13,11)	**1259.4**	**140.6**
12	(12,14×14,12)	**1107.32**	**292.68**
13	(13,15×15,13)	**1137.8**	**262.2**
14	(14,16×16,14)	**1239.62**	**160.38**
⋮	⋮	⋮	⋮

$k = 2, r = 14$

k	$(k,k+r)\times(k+r,k)$	*Traces (similarity)*	*Overlap*
2	(2,16×16,2)	**1400**	**0**

(b)

	Similarity			*Overlap*			
K	***Max Trace***	***Min Trace***	***Average Traces***	***CV***	***Max Overlap***	***Min Overlap***	***Average Overlap***
2	1400	1002	**1369.571**	**0.077**	398	0	**30.429**
3	1400	998.06	**1358.928**	**0.08**	401.94	0	**41.072**
4	1400	1400	**1400**	**0**	0	0	**0**
5	1392	1392	**1392**	**0**	8	8	**8**
6	1400	1400	**1400**	**0**	0	0	**0**
7	1400	1400	**1400**	**0**	0	0	**0**
8	1370.54	1341.08	**1361.23**	**0.007**	58.92	29.46	**38.77**
9	1400	1302.56	**1349.14**	**0.029**	97.44	0	**50.86**
10	1328.96	1228.92	**1264.313**	**0.031**	171.08	71.04	**135.687**
11	1297.28	1218.32	**1258.12**	**0.024**	181.68	102.72	**141.88**
12	1214.96	1107.32	**1144.555**	**0.043**	292.68	185.04	**255.445**
13	1240.28	1137.8	**1179.147**	**0.046**	262.2	159.72	**220.853**
14	1239.62	1079.74	**1159.68**	**0.097**	320.26	160.38	**240.32**

(c)

Table 5.23: A collection of values calculated from Q matrices at different k with $k+r$ mapping distance.

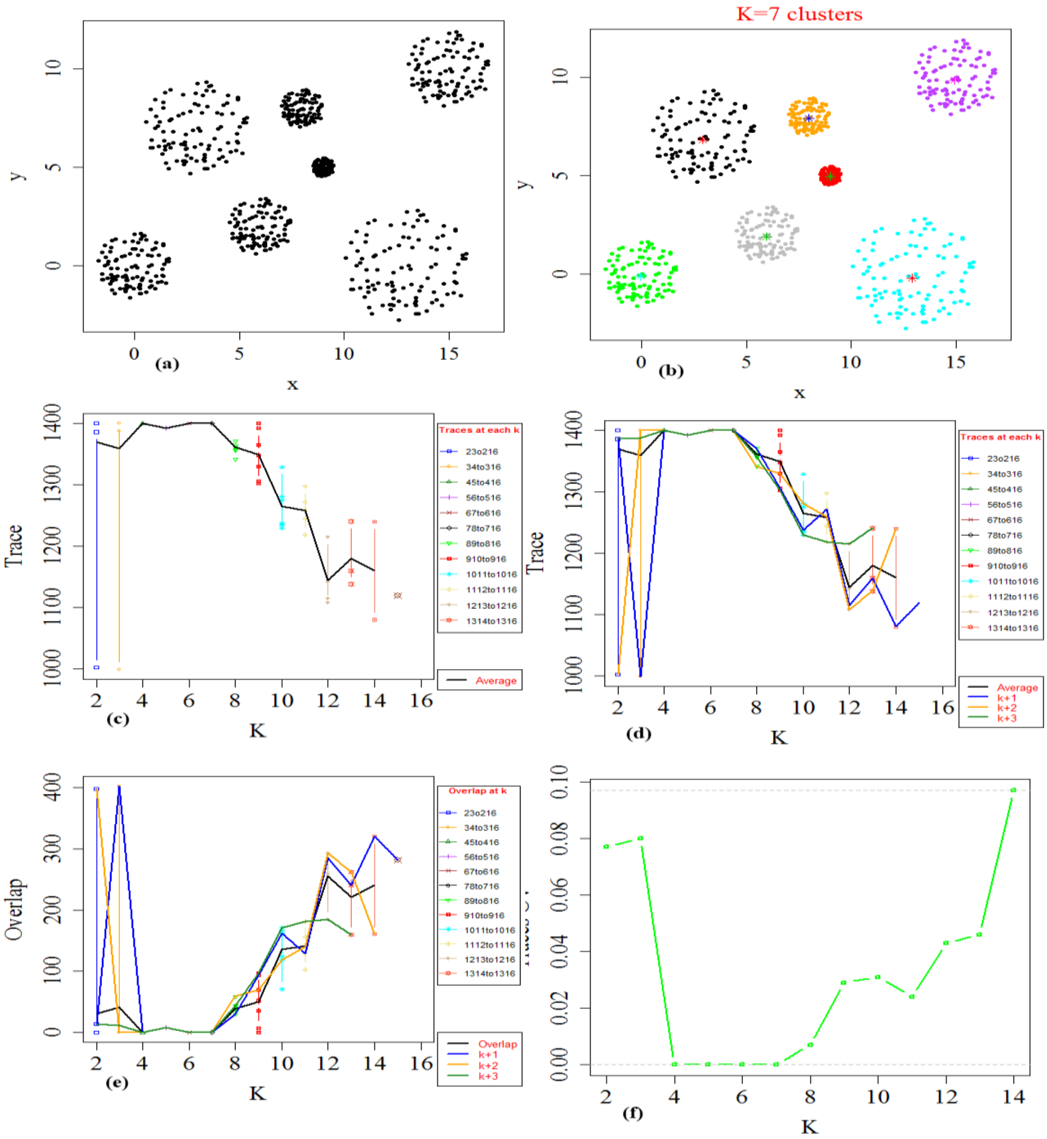


Figure 5.16: Part (a) shows scatter plot of case3 while (b) shows the number of clusters obtained from a *k-means* algorithm and labels the membership of clusters with different colours. Plots (c)-(f) show values computed from combined mapped elements Q matrices.

Plot (c) in Figure 5.16 shows the similarity from Table 5.23(a) and the legend beside the figure indicates similarity in different colours at each k for different $k + r$ distances. The black solid line is average similarity at different k. The flat segment of line shows clusters are splitting till $k = 7$ with maximum average similarity. The elements belonging to different clusters or overlap between clusters is increasing for higher k. Plots (d) and (f) are composite graphs that indicate the difference between average similarity, average overlap, similarity and overlap at different k with different coloured $k + r$. Plot (d) represents a quite small change in CV and CV of zero at $k = 7$ indicates clusters are stable.

1^st^ run:					***Case3:* Existing indexes values**					
K	***Dunn***	***DH***	***DH Critical***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***Gap Critical***	***CCC***
2	0.119	**1.269**	0.581	922.971	0.416	1.033	2.151	**0.203**	0.056	25.219
3	0.196	1.608	0.563	1288.787	0.356	1.035	3.781	0.178	-0.167	15.912
4	0.011	1.075	0.581	1848.786	0.518	0.797	**0.874**	0.37	0.181	35.637
5	0.014	0.208	0.507	2333.147	0.369	0.664	1.071	0.323	-0.094	24.128
6	**0.299**	0.344	0.558	3951.943	0.521	**0.513**	1.831	1.096	-0.135	54.062
7	0.031	11.386	0.507	3551.437	**0.682**	0.716	0.927	0.983	-0.17	**68.584**
8	0.038	1.153	0.409	**4209.358**	0.485	0.628	1.047	0.906	-0.22	24.02
9	0.01	28.473	0.458	3010.957	0.505	0.751	2.645	1.228	0.144	66.56
10	0.013	2.757	0.507	2509.72	0.59	0.756	1.09	1.187	-0.022	65.384
11	0.052	0.008	0.507	3170.3	0.531	0.762	1.627	1.178	0.313	57.044
12	0.008	2.84	0.507	4029.2	0.542	0.686	7.277	0.983	0.08	59.995
13	0.02	1.296	0.419	2840.126	0.612	0.896	5.968	1.058	0.179	63.887

(a)

Number of runs		***Dunn***	***DH***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***CCC***
2nd	***K***	**4**	**2**	**10**	**7**	**6**	**5**	**10**	**9**
	Values	**0.174**	**1.232**	**4309.76**	**0.682**	**0.6**	**0.837**	**1.236**	**66.981**
3rd	***K***	**2**	**2**	**9**	**6**	**7**	**5**	**8**	**7**
	Values	**0.119**	**0.923**	**4586.444**	**0.653**	**0.438**	**0.587**	**1.284**	**68.57**
4th	***K***	**4**	**2**	**11**	**8**	**5**	**4**	**2**	**9**
	Values	**0.307**	**0.986**	**4324.603**	**0.663**	**0.664**	**0.676**	**0.182**	**64.559**
5th	***K***	**6**	**5**	**7**	**7**	**6**	**6**	**2**	**16**
	Values	**0.299**	**3.793**	**4586.515**	**0.682**	**0.513**	**0.728**	**0.173**	**64.919**

(b)

Table 5.24: The summary of eight different indexes with 5 multiple runs. The optimal number of clusters is highlighted in bold.

For this case *Dunn* and *DH* indexes determine $k = 2$ to $k = 6$ optimal clusters and indicates these indexes ignore clusters of small size. The *SD* and *Gap* indexes vary and are also unable to specify correct number of clusters. *CH*, *Sil*, *DB* and *CCC* only detect the correct number of clusters a few times. Results show these indexes over estimate the number of clusters. The new approach has average similarity equal to *N* at $k = 7$. Clusters are fully separated when $k = 7$ as can be seen in Figure 5.16(b) where there are seven clusters in different colours. This indicates clusters stop splitting completely beyond that k and overlap between clusters starts to increase.

5.4.4 Case4: Nine Square Clusters of Medium Density and Equal Sizes

The dataset consists of 1350 total elements with nine square shaped clusters where each one is connected from the corner. The scatter plot is shown in the Figure 5.17(a) for case4. The results in Tables 5.25 (a) and (b) show overlap at initial k is high and a similarity is maximum at $k = 9$ with minimum overlap. Table (c) clearly shows the average similarity is a maximum with minimum average overlap and a low CV value when $k = 9$.

$k = 2, r = 1,2, \dots, K - k$

k	r	$(k, k + r) \times (k + r, k)$	***Traces (similarity)***	***Overlap***
2	1	(2,3×3,2)	**1073.47**	**276.53**
2	2	(2,4×4,2)	**1140.97**	**209.03**
2	3	(2,5×5,2)	**1167.97**	**182.03**
2	4	(2,6×6,2)	**1248.97**	**101.03**
2	5	(2,7×7,2)	**1221.97**	**128.03**
2	6	(2,8×8,2)	**1208.47**	**141.53**
2	7	(2,9×9,2)	**1140.97**	**209.03**
2	8	(2,10×10,2)	**1208.47**	**141.53**
2	9	(2,11×11,2)	**1167.97**	**182.03**
2	10	(2,12×12,2)	**1167.97**	**182.03**
2	11	(2,13×13,2)	**1215**	**135**
2	12	(2,14×14,2)	**1255.5**	**94.5**
2	13	(2,15×15,2)	**1208.47**	**141.53**
2	14	(2,16×16,2)	**1221.97**	**128.03**

$k = 3, r = 1,2, \dots, K - k$

k	r	$(k, k + r) \times (k + r, k)$	***Traces (similarity)***	***Overlap***
3	1	(3,4×4,3)	**1015.35**	**334.65**
3	2	(3,5×5,3)	**1118.31**	**231.69**
3	3	(3,6×6,3)	**1217.38**	**132.62**
3	4	(3,7×7,3)	**1220.27**	**129.73**
3	5	(3,8×8,3)	**1223**	**127**
3	6	(3,9×9,3)	**1174.27**	**175.73**
3	7	(3,10×10,3)	**1214.07**	**135.93**
3	8	(3,11×11,3)	**1217.17**	**132.83**
3	9	(3,12×12,3)	**1226.1**	**123.9**
3	10	(3,13×13,3)	**1221.18**	**128.82**
3	11	(3,14×14,3)	**1222.65**	**127.35**
3	12	(3,15×15,3)	**1199.8**	**150.2**
3	13	(3,16×16,3)	**1227.22**	**122.78**
⋮	⋮	⋮	⋮	⋮

$k = 15, r = 1$

k	r	$(k, k + r) \times (k + r, k)$	***Traces (similarity)***	***Overlap***
15	1	(15,16×16,15)	**1096.76**	**253.24**

(a)

$k = 2,3, \dots 15, r = 1$

k	$(k, k + r) \times (k + r, k)$	***Traces (similarity)***	***Overlap***
2	(2,3×3,2)	**1073.47**	**276.53**
3	(3,4×4,3)	**1015.35**	**334.65**
4	(4,5×5,4)	**932.94**	**417.06**
5	(5,6×6,5)	**1084.25**	**265.75**
6	(6,7×7,6)	**1210.24**	**139.76**
7	(7,8×8,7)	**962.96**	**387.04**
8	(8,9×9,8)	**1245.88**	**104.12**
9	(9,10×10,9)	**1339.5**	**10.5**
10	(10,11×11,10)	**1252.26**	**97.74**
11	(11,12×12,11)	**1244.86**	**105.14**
12	(12,13×13,12)	**1250.9**	**99.1**
13	(13,14×14,13)	**1145.97**	**204.03**
14	(14,15×15,14)	**1182.48**	**167.52**
15	(15,16×16,15)	**1096.76**	**253.24**

$k = 2,3, \dots 15, r = 2$

k	$(k, k + r) \times (k + r, k)$	***Traces (similarity)***	***Overlap***
2	(2,4×4,2)	**1140.97**	**209.03**
3	(3,5×5,3)	**1118.31**	**231.69**
4	(4,6×6,4)	**1108.96**	**241.04**
5	(5,7×7,5)	**1139.93**	**210.07**
6	(6,8×8,6)	**1138.49**	**211.51**
7	(7,9×9,7)	**1174.1**	**175.9**
8	(8,10×10,8)	**1289.24**	**60.76**
9	(9,11×11,9)	**1336.5**	**13.5**
10	(10,12×12,10)	**1251.55**	**98.45**
11	(11,13×13,11)	**1252.56**	**97.44**
12	(12,14×14,12)	**1164.35**	**185.65**
13	(13,15×15,13)	**1152.65**	**197.35**
14	(14,16×16,14)	**1257.69**	**92.31**
⋮	⋮	⋮	⋮

$k = 2, r = 14$

k	$(k, k + r) \times (k + r, k)$	***Traces (similarity)***	***Overlap***
2	(2,16×16,2)	**1221.97**	**128.03**

(b)

Similarity				***Overlap***			
K	***Max Trace***	***Min Trace***	***Average Traces***	***CV***	***Max Overlap***	***Min Overlap***	***Average Overlap***
2	1255.5	1073.47	**1189.153**	**0.041**	276.53	94.5	**160.847**
3	1227.22	1015.35	**1192.059**	**0.051**	334.65	122.78	**157.941**
4	1170.82	932.94	**1104.232**	**0.06**	417.06	179.18	**245.768**
5	1193.87	1071.99	**1149.063**	**0.035**	278.01	156.13	**200.937**
6	1233	1120.5	**1184.825**	**0.031**	229.5	117	**165.175**
7	1272.42	962.96	**1183.576**	**0.076**	387.04	77.58	**166.424**
8	1289.24	1228.66	**1257.359**	**0.018**	121.34	60.76	**92.641**
9	1339.5	1317	**1328.143**	**0.007**	33	10.5	**21.857**
10	1338	1239.53	**1277.077**	**0.033**	110.47	12	**72.923**
11	1252.56	1222.84	**1236.95**	**0.009**	127.16	97.44	**113.05**
12	1250.9	1164.35	**1193.015**	**0.033**	185.65	99.1	**156.985**
13	1152.65	1140.76	**1146.46**	**0.005**	209.24	197.35	**203.54**
14	1257.69	1182.48	**1220.085**	**0.044**	167.52	92.31	**129.915**

(c)

Table 5.25: Summary of the values computed form Q matrices at different k with different $k + r$ mapped distances.

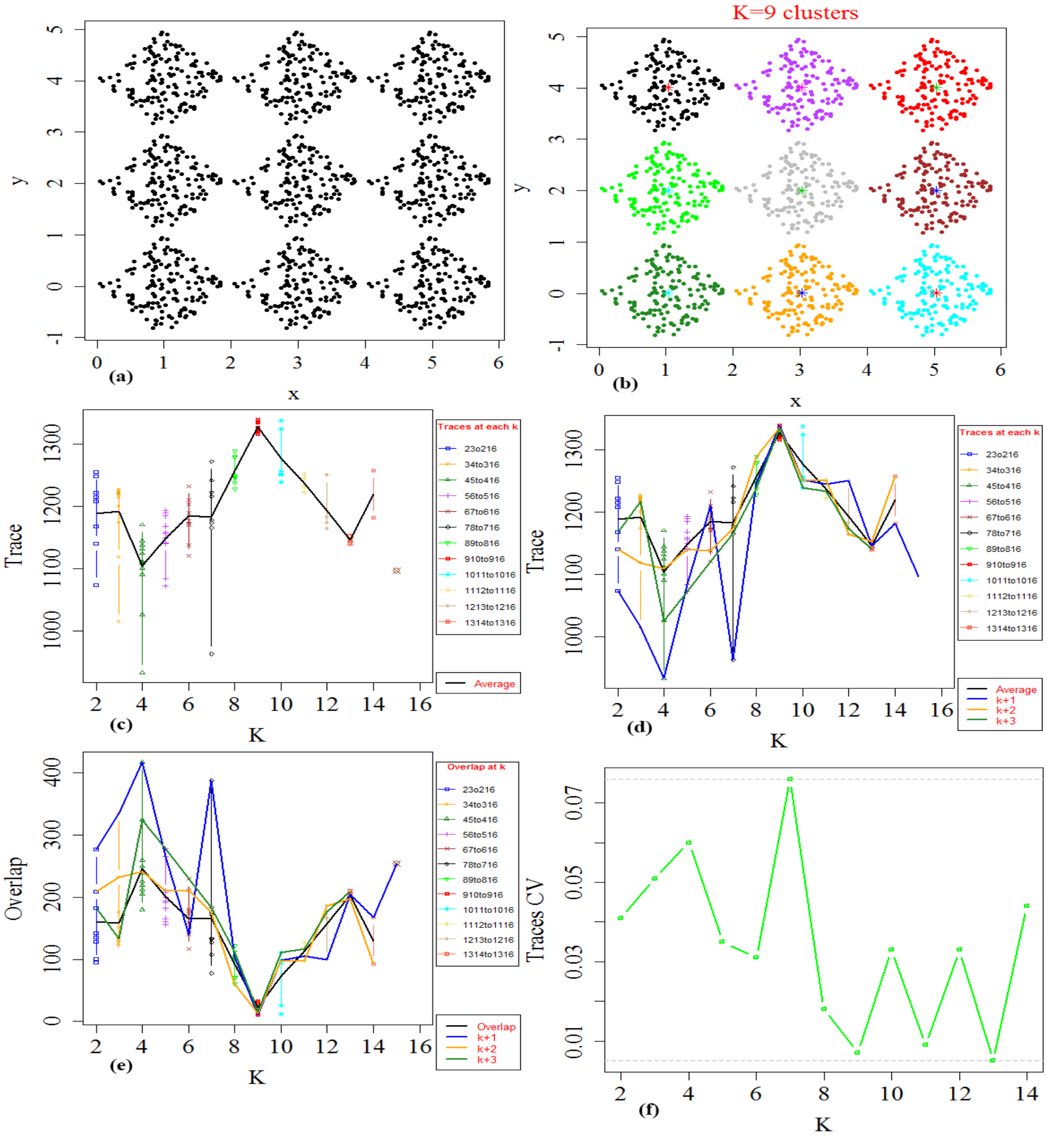


Figure 5.17: Part (a) shows the scatter plot for case4 type3 dataset while (b) shows cluster membership and centroids with different colours. Plots (c)-(f) show values computed from combined mapped elements Q matrices.

Plot (c) in the above figure shows the similarity at different k for $k + r$ mapped distances using various colours and average similarity as the black solid line which is a maximum at $k = 9$. Plots (d) and (e) are composite graphs of similarity and overlap with different $k + r$ coloured lines. For $k = 9$ is the best $\boldsymbol{K}$ number of clusters as the average similarity is high (criteria 1 Chapter 4 section 4.3.3), while average overlap is a minimum. Plot (f) shows at the best $\boldsymbol{K}$, i.e. $k = 9$, clusters are stable as the CV value is a minimum.

1^{st} run:					*Case4:* **Existing indexes values**					
K	***Dunn***	***DH***	***DH Critical***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***Gap Critical***	***CCC***
2	0.006	**0.871**	0.588	800.678	0.349	1.308	2.743	**-0.029**	0.021	29.059
3	0.006	1.042	0.568	1046.461	0.384	0.928	2.462	-0.022	0.051	35.641
4	0.009	1.512	0.533	1287.939	0.396	0.834	1.457	-0.071	-0.051	38.4
5	0.009	2.747	0.521	1326.736	0.417	0.941	1.628	0.024	-0.049	37.193
6	0.014	1.303	0.502	1360.535	0.409	0.888	1.522	0.061	-0.095	36.935
7	0.013	1.188	0.519	1569.578	0.451	0.828	1.379	0.141	-0.165	42.116
8	0.009	0.368	0.51	1939.755	0.501	**0.664**	1.297	0.351	0.16	47.95
9	**0.114**	5.471	0.404	**2707.356**	**0.556**	0.81	**1.184**	0.206	-0.347	**60.691**
10	0.034	1.272	0.48	1575.083	0.526	0.705	2.734	0.57	0.504	57.734
11	0.022	1.999	0.478	2391.978	0.516	0.813	2.96	0.526	0.027	55.25
12	0.022	0.981	0.459	2276.113	0.463	0.944	2.887	0.49	0.047	53.236
13	0.022	3.206	0.343	2209.218	0.454	0.835	2.867	0.44	0.05	51.237

(a)

Number of runs		***Dunn***	***DH***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***CCC***
2nd	***K***	**15**	**2**	**10**	**10**	**9**	**9**	**2**	**9**
	Values	**0.022**	**0.882**	**2519.809**	**0.529**	**0.575**	**1.167**	**-0.026**	**60.691**
3rd	***K***	**9**	**2**	**10**	**9**	**8**	**8**	**2**	**11**
	Values	**0.117**	**0.95**	**2528.22**	**0.556**	**0.664**	**1.264**	**-0.028**	**54.3**
4th	***K***	**9**	**2**	**9**	**10**	**8**	**8**	**2**	**9**
	Values	**0.117**	**0.826**	**2708.512**	**0.524**	**0.668**	**1.264**	**-0.015**	**60.709**
5th	***K***	**11**	**2**	**10**	**10**	**9**	**9**	**2**	**9**
	Values	**0.023**	**0.693**	**2521.233**	**0.529**	**0.574**	**1.159**	**-0.019**	**60.709**

(b)

Table 5.26: The summary of eight different indexes with 5 multiple runs. The optimal number of clusters is highlighted in bold.

From Table 5.26 it is seen that the indexes do not agree on best k. *DH* and *Gap* suggest $k = 2$ while *Dunn* suggests $k = 9$ to $k = 15$. There is no agreement among *CH*, *Sil*, *DB*, *CCC* and *SD* indexes for the right number of clusters. Indexes *DH* and *Gap* were unable to find the correct number of clusters while other indexes perform slightly better but are inconsistent and so there is a need to find the indexes multiple times to determine the best number. The new approach easily identifies the correct number of clusters as 9. See Figure 5.17(c) when average similarity at $k = 9$ is a maximum and average overlap and CV values are at a minimum. For this data the results indicate existing approaches perform poorly when the clusters are of equal size and have a square shape, connected at the corners.

5.5 Summary

Generally, for clustering evaluations two dimensional datasets are used to visually display the results (i.e., how well the clustering structure is discovered for a data). Therefore, we have analysed the behaviour of a new approach by applying it on several varieties of two dimensional datasets for determining the quality of cluster algorithms. The comparison of the new approach with exiting approaches was checked by applying the alternative approaches on these datasets with equal, large and small sized clusters. Data with more complex structure, e.g. square shapes connected from the corner and mixture of different densities and shapes was also used. Existing validation indexes worked well for estimating the correct number of clusters in some cases, but generally these indexes were inconsistent. This makes it difficult to decide which k is better for selecting the best number of clusters as the different examples illustrated.

However the new approach worked well for the same datasets when the aim was to determine the optimum number of clusters and to explore the clustering structure. By plotting the values from the new approach, we found either the maximum average similarity or a segment of average similarity maximum gave the best number of clusters. For example, clearly case1 of type1, case1, case2, case3 of type2 and case1, case3 of type3 datasets used maximum average similarity having a segment of its line in the plot of the similarities to find the best $\boldsymbol{K}$. In other cases, namely case2, case3, case4 of type1, case4 of type2 and case4 of type3 needed only the one maximum average similarity peak to determine the best number of clusters. The experimental results on all the datasets suggest that the new approach works well and determines the best ***K*** value at which either the clusters are mutually exclusive or have minimum overlap. In the next chapter, the efficiency and performance of the new approach will be checked by applying the new approach on some real datasets.

Chapter 6

Application to Real World Datasets

6.1 Introduction

The previous chapter discussed the results and effectiveness of the new approach and compared it with several existing cluster validation indexes using a wide range of various simulated datasets. In this chapter it will be demonstrated how the new approach works on real world datasets. Four different well known datasets are used; two with known clustering structure (Physical activities (Physed) and Wisconsin Breast Cancer (WBC) datasets) and the other two with unknown clustering structure Framingham Heart Study (FHS) and Medical Expenditure Panel Surveys (MEPS), both health datasets with multi-dimensional structure. Prior to applying the *k-means* algorithm for analysing these datasets, missing values and outliers were removed from these datasets.

The remainder of this chapter is organised as follows: In section 6.2 the performance of new approach is tested for the first two datasets where the true clusters of the data are known in advance. Section 6.3 examines the usefulness of the approach to the third and fourth datasets that initially have no a priori knowledge about the number of clusters. Section 6.4 presents a summary of the findings of this chapter.

6.2 Datasets with Known Clustering Structure

In this section two datasets, Physed and WBC, are used to present results which demonstrate the effectiveness of the new approach, and to compare these with eight different clustering validation indexes. These datasets have long been used as a benchmark datasets in cluster analysis [148, 167, 168] as examples where the numbers of cluster or classes are known in advance.

6.2.1 Dataset: Physical Education

This dataset consists of the physical activities of 80 students with four different groups (clusters). These measure different characteristics of body function e.g. flexibility, speed and strength. The Figure 6.1(a) shows scatter plot matrix of the dataset. This was used by Makles [169] to find the best number of clusters. Table 6.1(a) shows only the traces (similarity) values at fixed k for different $k + r$ mapping distances, while (b) shows different k for fixed $k + r$ mapping distances. Table (c) exhibits the values of maximum, minimum, similarity and overlap, average traces (similarity) and overlap and the coefficient of variation (CV). From the set of values in Table 6.1(a) it can be seen that only few elements overlap at $k = 2$ while there is no overlap between clusters at $k = 3$ with different $k + r$. The values in Table (b) shows beyond $k = 7$ clusters overlap while from $k = 3$ to $k = 7$ there is no overlap as trace values are equal to $N = 80$ number of students for the $k + 1$ mapping distance. The situation for $k + 2$ mapping distance shows there is no overlap between clusters from $k = 3$ to $k = 6$. This indicates as the forward and backward mapping distances increase elements from different clusters merge.

	k	r	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***
$k = 2, r = 1,2, \dots, K - k$	2	1	(2,3×3,2)	**77.88**	**2.12**
	2	2	(2,4×4,2)	**77.88**	**2.12**
	2	3	(2,5×5,2)	**77.88**	**2.12**
	2	4	(2,6×6,2)	**77.88**	**2.12**
	2	5	(2,7×7,2)	**78.14**	**1.86**
	2	6	(2,8×8,2)	**78.14**	**1.86**
	2	7	(2,9×9,2)	**78.14**	**1.86**
	2	8	(2,10×10,2)	**78.14**	**1.86**
	2	9	(2,11×11,2)	**78.68**	**1.32**
	2	10	(2,12×12,2)	**78.94**	**1.06**
	2	11	(2,13×13,2)	**78.68**	**1.32**
	2	12	(2,14×14,2)	**78.68**	**1.32**
	2	13	(2,15×15,2)	**78.94**	**1.06**
	2	14	(2,16×16,2)	**78.68**	**1.32**
	k	r	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***
$k = 3, r = 1,2, \dots, K - k$	3	1	(3,4×4,3)	**80**	**0**
	3	2	(3,5×5,3)	**80**	**0**
	3	3	(3,6×6,3)	**80**	**0**
	3	4	(3,7×7,3)	**80**	**0**
	3	5	(3,8×8,3)	**80**	**0**
	3	6	(3,9×9,3)	**80**	**0**
	3	7	(3,10×10,3)	**80**	**0**
	3	8	(3,11×11,3)	**80**	**0**
	3	9	(3,12×12,3)	**80**	**0**
	3	10	(3,13×13,3)	**80**	**0**
	3	11	(3,14×14,3)	**80**	**0**
	3	12	(3,15×15,3)	**80**	**0**
	3	13	(3,16×16,3)	**80**	**0**
	⋮	⋮	⋮	⋮	⋮
	k	r	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***
$k = 15, r = 1$	15	1	(15,16×16,15)	**66.47**	**13.53**

(a)

k	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***	
2	(2,3×3,2)	**77.88**	**2.12**	$k = 2,3, ..15, r = 1)$
3	(3,4×4,3)	**80**	**0**	
4	(4,5×5,4)	**80**	**0**	
5	(5,6×6,5)	**80**	**0**	
6	(6,7×7,6)	**80**	**0**	
7	(7,8×8,7)	**80**	**0**	
8	(8,9×9,8)	**68.55**	**11.45**	
9	(9,10×10,9)	**61.78**	**18.22**	
10	(10,11×11,10)	**58.97**	**21.03**	
11	(11,12×12,11)	**55.88**	**24.12**	
12	(12,13×13,12)	**63.55**	**16.45**	
13	(13,14×14,13)	**66.55**	**13.45**	
14	(14,15×15,14)	**67.15**	**12.85**	
15	(15,16×16,15)	**66.47**	**13.53**	
k	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***	
2	(2,4×4,2)	**77.88**	**2.12**	$k = 2,3, ..15, r = 2$
3	(3,5×5,3)	**80**	**0**	
4	(4,6×6,4)	**80**	**0**	
5	(5,7×7,5)	**80**	**0**	
6	(6,8×8,6)	**80**	**0**	
7	(7,9×9,7)	**70.21**	**9.79**	
8	(8,10×10,8)	**80**	**0**	
9	(9,11×11,9)	**74.78**	**5.22**	
10	(10,12×12,10)	**75.63**	**4.37**	
11	(11,13×13,11)	**70.22**	**9.78**	
12	(12,14×14,12)	**68.95**	**11.05**	
13	(13,15×15,13)	**73.17**	**6.83**	
14	(14,16×16,14)	**71.29**	**8.71**	
⋮	⋮	⋮	⋮	
k	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***	
2	(2,16×16,2)	**78.68**	**1.32**	$k = 2, r = 14$

(b)

Similarity				***Overlap***			
K	***Max Trace***	***Min Trace***	***Average Traces***	***CV***	***Max Overlap***	***Min Overlap***	***Average Overlap***
2	78.94	77.88	**78.334**	**0.005**	2.12	1.06	**1.666**
3	80	80	**80**	**0**	0	0	**0**
4	80	80	**80**	**0**	0	0	**0**
5	80	73.28	**77.709**	**0.038**	6.72	0	**2.291**
6	80	73.28	**77.48**	**0.039**	6.72	0	**2.52**
7	80	70.21	**76.693**	**0.05**	9.79	0	**3.307**
8	80	68.55	**75.456**	**0.056**	11.45	0	**4.544**
9	78.56	61.78	**72.571**	**0.097**	18.22	1.44	**7.429**
10	76.64	58.97	**71.792**	**0.092**	21.03	3.36	**8.208**
11	72.11	55.88	**68.118**	**0.101**	24.12	7.89	**11.882**
12	72.67	63.55	**69.26**	**0.06**	16.45	7.33	**10.74**
13	73.17	66.55	**69.39**	**0.049**	13.45	6.83	**10.61**
14	71.29	67.15	**69.22**	**0.042**	12.85	8.71	**10.78**

(c)

Table 6.1: A collection of tables showing the values computed from Q matrices at different k with different $k + r$ mapping distances.

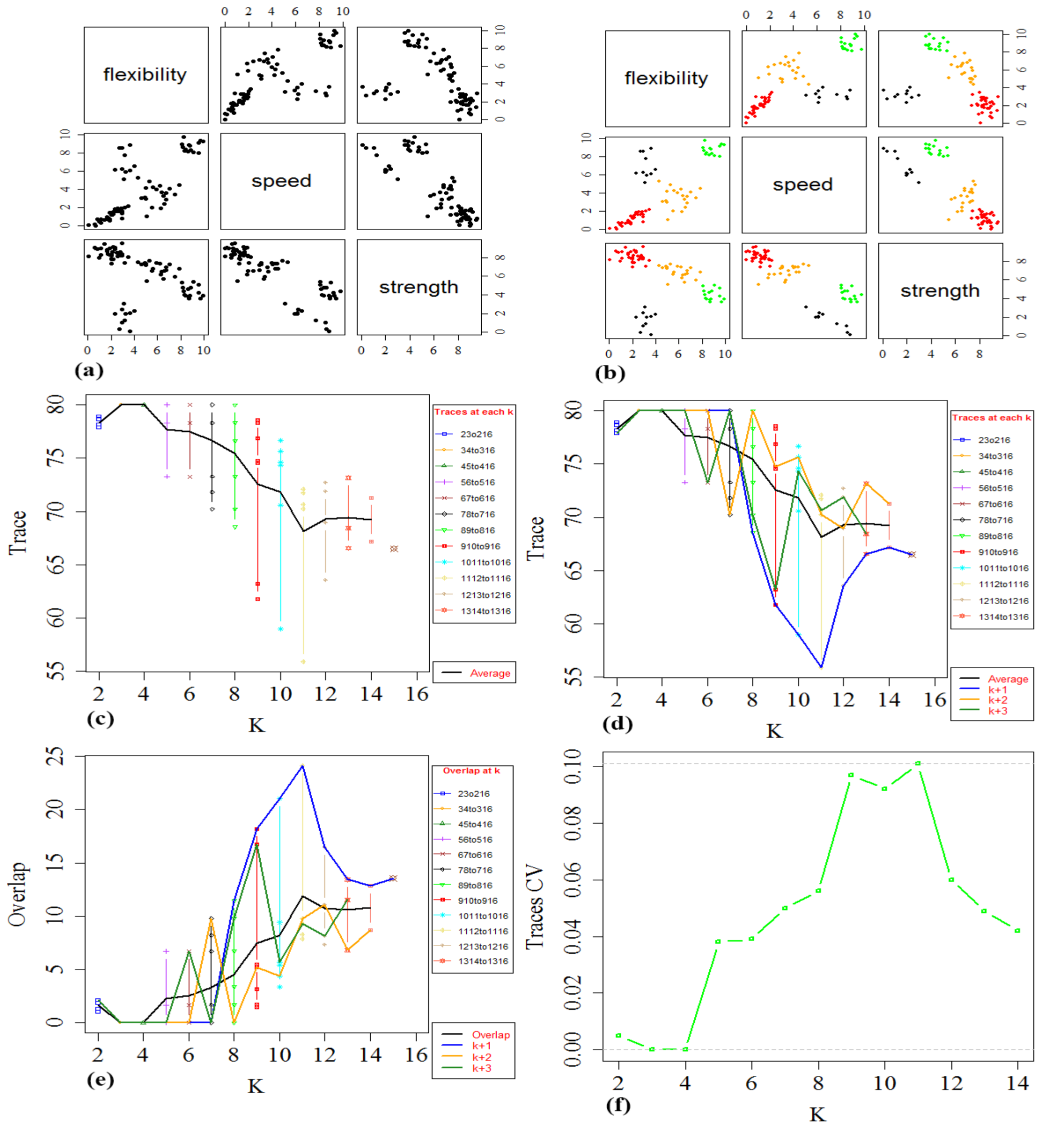


Figure 6.1: Part (a) shows the scatter plot matrix of the physical education dataset. Plot (b) shows the number of clusters obtained by a *k-means* algorithm using $k = 4$ with membership of elements labeled in different colours. Plots (c)-(f) represent the values from Table 6.1 at different k for $k + r$ mapped distances.

For better understanding it is necessary to examine the figures above. Plot (c) shows the similarity in various colours at different k as the legend beside the plot indicates while the black solid line indicates average similarity. This line shows at $k = 3$ and $k = 4$ clusters are fully separated with no overlap between clusters. The plots (d) and (e) show the difference as composite graphs for the behavior of traces (similarity) and overlap at fixed k with different $k + r$ while the black solid line shows average traces (similarity) and overlap at different k. The new approach identifies the correct number of cluster at $k = 4$ as according to the 4 criteria mentioned in Chapter 4 section 4.3.3. Plot (f) indicates the coefficient of variation values and clusters are settled at the best $\boldsymbol{K}$.

1st run:				***Case1:* Existing indexes values**						
K	***Dunn***	***DH***	***DH Critical***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***Gap Critical***	***CCC***
2	0.051	**0.623**	0.421	132.564	0.524	0.75	2.003	0.687	-0.185	16.653
3	**0.189**	7.278	0.41	128.41	**0.628**	0.75	1.287	0.895	-0.895	21.177
4	0.063	0.187	0.484	103.502	0.436	1.038	**0.758**	**0.817**	0.267	16.385
5	0.188	0.666	0.41	261.889	0.547	**0.654**	1.489	0.607	-1.094	**26.212**
6	0.072	1.078	0.23	65.054	0.486	0.851	1.597	1.688	-0.028	23.83
7	0.072	0.343	0.191	219.311	0.526	0.842	2.106	1.855	0.24	21.704
8	0.057	8.634	-0.501	**279.562**	0.522	0.94	2.705	1.648	-0.342	20.035
9	0.057	1.907	-0.123	232.793	0.445	1.027	2.822	1.429	0.125	21.297
10	0.072	0.888	-0.056	223.439	0.372	0.989	2.96	1.62	-0.087	23.779
11	0.068	3.983	-0.21	167.927	0.464	0.844	2.49	1.958	0.069	22.231
12	0.057	11.587	-0.21	181.893	0.474	0.811	3.535	2.136	0.43	22.986
13	0.057	2.292	-0.328	264.827	0.357	0.816	3.695	1.989	0.504	20.526

(a)

Number of runs		***Dunn***	***DH***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***CCC***
2nd	***K***	**4**	**2**	**4**	**4**	**4**	**5**	**4**	**5**
	Values	**0.332**	**2.119**	**337.103**	**0.69**	**0.484**	**1.486**	**1.862**	**25.672**
3rd	***K***	**4**	**2**	**5**	**4**	**4**	**4**	**4**	**7**
	Values	**0.332**	**2.119**	**303.891**	**0.69**	**0.484**	**0.666**	**1.845**	**25.585**
4th	***K***	**4**	**2**	**5**	**2**	**4**	**4**	**4**	**4**
	Values	**0.332**	**0.93**	**303.891**	**0.58**	**0.484**	**0.649**	**1.805**	**28.3**
5th	***K***	**4**	**3**	**5**	**5**	**4**	**4**	**3**	**4**
	Values	**0.332**	**0.223**	**294.562**	**0.613**	**0.484**	**0.659**	**0.973**	**28.3**

(b)

Table 6.2: Shows the optimal numbers of clusters with their values highlighted in bold from eight existing indexes with 5 multiple runs.

Table 6.2(a), first run, shows the values of the indexes from $k = 2$ to $k = 13$ with optimal values highlighted in bold. Table 6.2(b), with four runs, shows only optimal values with the corresponding index. The results show *CH, Sil* and *CCC* indexes overestimate and were rarely able to specify correct numbers of clusters while *DH* indicates 2 as the estimated number of clusters. *Dunn, SD, DB* and *Gap* were frequently able to find the correct number of clusters. We observed using the results from the above tables that the new approach effectively and efficiently performs well giving similar results to *Dunn, SD, DB* and *Gap* indexes. The new approach gives an acceptable solution as it finds four clusters which is the number of clusters or groups known in advance for physical characteristics of different students. Moreover, the new approach also determines that at the best $\boldsymbol{K}$ clusters are fully separated and stable with CV value 0.

6.2.2 Dataset: Wisconsin Breast Cancer

Next, the new approach was applied to the higher dimensional Wisconsin Breast Cancer dataset from UCI [170]. This dataset classified data on patients screened for breast cancer and classified any tumours into two classes: malignant and benign. It was originally collected by Dr. William H. Wolberg at the University of Wisconsin Hospital, Madison. The dataset has been used in the literature [171, 172, 173] for different purposes such as least square modelling and evaluating clustering algorithm performance. It includes $N = 699$ elements (observations) in total and each observation consists of 11 variables. The first variable is the patient ID, the next nine variables have numerical values each from 1 to 10, and the last variable is categorizes the class of breast cancer as benign or malignant. There are 16 missing values which were removed before using the dataset to evaluate the performance of the new approach and compare with other indexes. Table 6.3 shows the names of the variables for the WBC dataset.

Variables	**Variables**
ID	Bare Nuclei
Clump thickness	Bland Chromatin
Uniformity of Cell Size	Normal Nucleoli
Uniformity of Cell Shape	Mitoses
Marginal Adhesion	Class
Single Epithelial Cell Size	

Table 6.3: The set of variables included in the Wisconsin Breast Cancer dataset.

Only the nine numeric variables were selected to obtain the number of clusters $k = 2,3, \dots ,16$ from the *k-means* algorithm, and resultant sets of clusters were used for computing the new approach. Table 6.4(a) shows the similarity and overlap values at fixed k with different $k + r$ mapped distances. Table 6.4(b) shows the results at different k for fixed $k + r$ distances. Table 6.4(c) indicates maximum, minimum, similarity and overlap, average similarity and overlap and the coefficient of variation at different k. Average similarity (traces) of elements between clusters are a maximum (1 criteria Chapter 4 section 4.3.3) with minimum overlap when $k = 2$ so it can be seen, while the minimum CV value for $k = 2$ indicates that clusters are stable.

	k	r	$(k,k+r)\times(k+r,k)$	***Traces (similarity)***	***Overlap***
$k=2,r=1,2,\dots,K-k$	2	1	(2,3×3,2)	**664.74**	**18.26**
	2	2	(2,4×4,2)	**664.74**	**18.26**
	2	3	(2,5×5,2)	**673.87**	**9.13**
	2	4	(2,6×6,2)	**673.87**	**9.13**
	2	5	(2,7×7,2)	**673.87**	**9.13**
	2	6	(2,8×8,2)	**664.74**	**18.26**
	2	7	(2,9×9,2)	**664.74**	**18.26**
	2	8	(2,10×10,2)	**664.74**	**18.26**
	2	9	(2,11×11,2)	**664.74**	**18.26**
	2	10	(2,12×12,2)	**662.44**	**20.56**
	2	11	(2,13×13,2)	**662.44**	**20.56**
	2	12	(2,14×14,2)	**657.91**	**25.09**
	2	13	(2,15×15,2)	**664.74**	**18.26**
	2	14	(2,16×16,2)	**662.44**	**20.56**
	k	r	$(k,k+r)\times(k+r,k)$	***Traces (similarity)***	***Overlap***
$k=3,r=1,2,\dots,K-k$	3	1	(3,4×4,3)	**642.54**	**40.46**
	3	2	(3,5×5,3)	**642.01**	**40.99**
	3	3	(3,6×6,3)	**629.96**	**53.04**
	3	4	(3,7×7,3)	**620.32**	**62.68**
	3	5	(3,8×8,3)	**631.22**	**51.78**
	3	6	(3,9×9,3)	**627.66**	**55.34**
	3	7	(3,10×10,3)	**624.1**	**58.9**
	3	8	(3,11×11,3)	**622.84**	**60.16**
	3	9	(3,12×12,3)	**622.84**	**60.16**
	3	10	(3,13×13,3)	**619.28**	**63.72**
	3	11	(3,14×14,3)	**621.69**	**61.31**
	3	12	(3,15×15,3)	**632.37**	**50.63**
	3	13	(3,16×16,3)	**626.4**	**56.6**
	⋮	⋮	⋮	⋮	⋮
	k	r	$(k,k+r)\times(k+r,k)$	***Traces (similarity)***	***Overlap***
$k=15,r=1$	15	1	(15,16×16,15)	**501.99**	**181.01**

(a)

k	$(k,k+r)\times(k+r,k)$	***Traces (similarity)***	***Overlap***	
2	(2,3×3,2)	**664.74**	**18.26**	$k=2,3,..15,r=1$
3	(3,4×4,3)	**642.54**	**40.46**	
4	(4,5×5,4)	**656.47**	**26.53**	
5	(5,6×6,5)	**655.49**	**27.51**	
6	(6,7×7,6)	**646.49**	**36.51**	
7	(7,8×8,7)	**535.56**	**147.44**	
8	(8,9×9,8)	**660.13**	**22.87**	
9	(9,10×10,9)	**526.85**	**156.15**	
10	(10,11×11,10)	**655.75**	**27.25**	
11	(11,12×12,11)	**558.63**	**124.37**	
12	(12,13×13,12)	**656.51**	**26.49**	
13	(13,14×14,13)	**516.55**	**166.45**	
14	(14,15×15,14)	**582.1**	**100.9**	
15	(15,16×16,15)	**501.99**	**181.01**	
k	$(k,k+r)\times(k+r,k)$	***Traces (similarity)***	***Overlap***	
2	(2,4×4,2)	**664.74**	**18.26**	$k=2,3,..15,r=2$
3	(3,5×5,3)	**642.01**	**40.99**	
4	(4,6×6,4)	**642.03**	**40.97**	
5	(5,7×7,5)	**632.74**	**50.26**	
6	(6,8×8,6)	**524.53**	**158.47**	
7	(7,9×9,7)	**529.54**	**153.46**	
8	(8,10×10,8)	**511.22**	**171.78**	
9	(9,11×11,9)	**538.15**	**144.85**	
10	(10,12×12,10)	**542.05**	**140.95**	
11	(11,13×13,11)	**539.27**	**143.73**	
12	(12,14×14,12)	**521.34**	**161.66**	
13	(13,15×15,13)	**521.32**	**161.68**	
14	(14,16×16,14)	**560.42**	**122.58**	
⋮	⋮	⋮	⋮	
k	$(k,k+r)\times(k+r,k)$	***Traces (similarity)***	***Overlap***	
2	(2,16×16,2)	**662.44**	**20.56**	$k=2,r=14$

(b)

Similarity				***Overlap***			
K	***Max Trace***	***Min Trace***	***Average Traces***	***CV***	***Max Overlap***	***Min Overlap***	***Average Overlap***
2	673.87	657.91	**665.716**	**0.007**	25.09	9.13	**17.284**
3	642.54	619.28	**627.941**	**0.012**	63.72	40.46	**55.059**
4	656.47	605.2	**619.508**	**0.025**	77.8	26.53	**63.492**
5	655.49	496.77	**565.512**	**0.102**	186.23	27.51	**117.488**
6	646.49	491.14	**558.707**	**0.097**	191.86	36.51	**124.293**
7	610.85	490.91	**552.79**	**0.08**	192.09	72.15	**130.21**
8	660.13	488.96	**568.891**	**0.118**	194.04	22.87	**114.109**
9	639.97	499.18	**567.37**	**0.108**	183.82	43.03	**115.63**
10	655.75	537.34	**591.535**	**0.082**	145.66	27.25	**91.465**
11	622.38	539.27	**579.192**	**0.061**	143.73	60.62	**103.808**
12	656.51	516.11	**585.423**	**0.132**	166.89	26.49	**97.577**
13	627.43	516.55	**555.1**	**0.113**	166.45	55.57	**127.9**
14	582.1	560.42	**571.26**	**0.027**	122.58	100.9	**111.74**

(c)

Table 6.4: A collection of tables show the values calculated from Q matrices at different k with $k+r$ mapped distances.

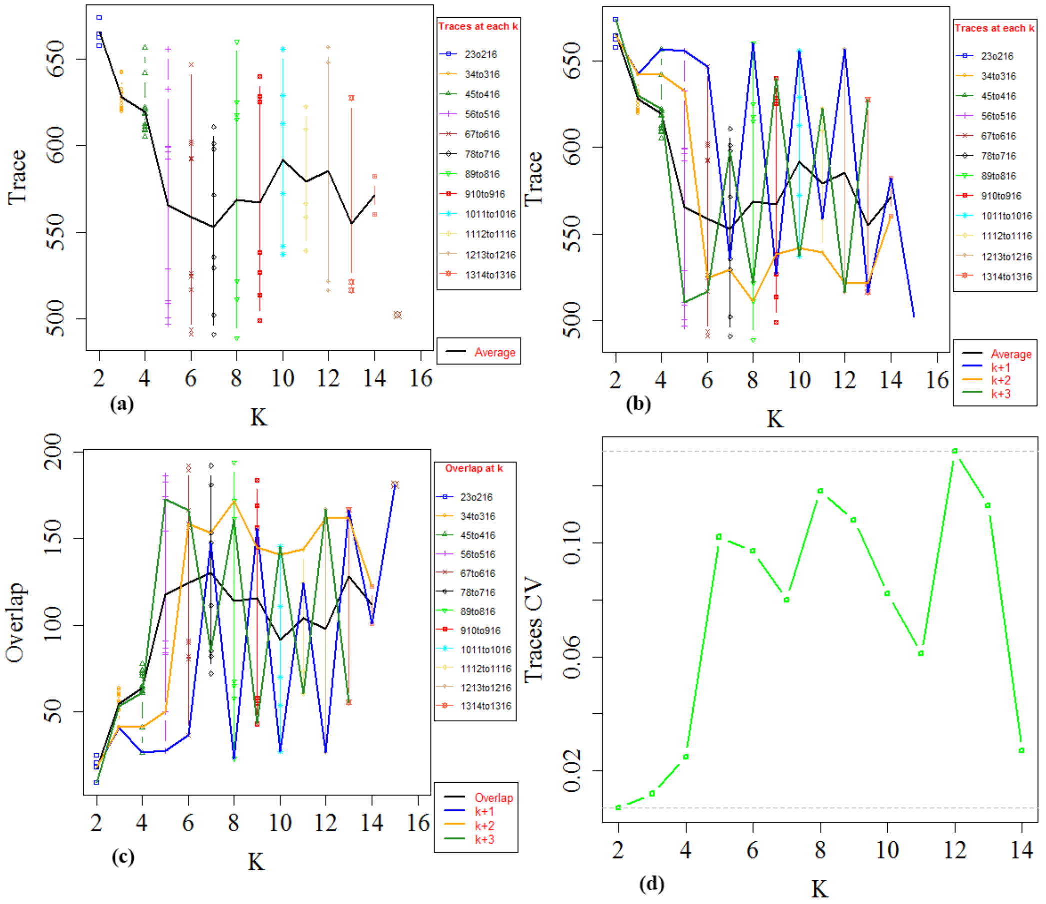


Figure 6.2: Plots (a)-(d) represent the values computed from different Q matrices at different k for $k + r$ mapped distances.

Plot (a) shows the similarity at different k with different $k + r$, and the solid black line represents the values for average similarity which is a maximum at $k = 2$ with minimum average overlap. Plots (b) and (c) show the differences as composite graphs for the behaviour of trace values of Q matrices (number of elements similarity) and overlap, average traces (average number of elements similarity) and average overlap at different k with coloured lines for $k + r$. Plot (b) indicates more than one peak at different k for $k + 1$ (the solid blue line) mapped distances, for which a small number of elements split. This increased the trace values (similarity) as the few elements form a cluster. This can happen when some variables have large spread or extreme values. Similarly, plot (c) shows overlaps and average overlaps at different k with $k + r$ in various colour lines (blue, orange, green and black). Plot (d) represents clusters are stable at $k = 2$ which has a minimum coefficient of variation.

1st run:					*WBC:* **Existing indexes values**					
K	***Dunn***	***DH***	***DH Critical***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***Gap Critical***	***CCC***
2	**0.155**	0.273	0.834	**1026.262**	**0.597**	**0.808**	**0.761**	**0.666**	0.017	**23**
3	0.144	**2.306**	0.794	568.966	0.523	1.474	0.933	0.671	-0.119	11.371
4	0.145	2.459	0.826	488.255	0.255	1.661	1.089	0.792	-0.004	14.735
5	0.056	1.397	0.742	429.646	0.149	1.79	1.163	0.754	-0.103	16.352
6	0.056	0.618	0.777	352.949	0.255	1.792	1.143	0.867	-0.012	17.597
7	0.054	4.258	0.823	316.552	0.19	1.742	1.647	0.838	-0.036	14.775
8	0.058	0.349	0.696	294.096	0.258	1.984	0.933	0.835	-0.049	16.304
9	0.058	1.006	0.768	275.668	0.184	1.942	1.711	0.944	0.014	17.478
10	0.061	4.008	0.647	243.233	0.184	1.944	1.515	0.854	-0.069	16.343

(a)

Number of runs		***Dunn***	***DH***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***CCC***
2nd	***K***	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
	Values	**0.155**	**2.922**	**1026.262**	**0.597**	**0.808**	**0.831**	**0.664**	**23**
3rd	***K***	**2**	**2**	**2**	**2**	**2**	**2**	**10**	**2**
	Values	**0.155**	**2.032**	**1026.262**	**0.597**	**0.808**	**0.791**	**0.948**	**23**
4th	***K***	**2**	**3**	**2**	**2**	**2**	**2**	**3**	**2**
	Values	**0.155**	**1.536**	**1026.262**	**0.597**	**0.808**	**0.776**	**0.671**	**23**
5th	***K***	**5**	**3**	**2**	**2**	**2**	**3**	**2**	**2**
	Values	**0.194**	**2.127**	**1026.262**	**0.597**	**0.808**	**0.686**	**0.67**	**23**

(b)

Table 6.5: Values computed from different exisitng indexes. The optimal number of clusters is highlighted in bold.

Table 6.5(a) shows the values of the indexes from $k = 2$ to $k = 10$ with optimal values highlighted in bold, while Table 6.5(b) shows only optimal values with the corresponding index. The result show *Dunn*, *DH*, *SD* and *Gap* indexes frequently determine 2 as the best number of clusters while *CH*, *DB*, *Sil* and *CCC* identify the correct number of clusters in each run. Although all the indexes work well *Dunn*, *DH*, *SD* and *Gap* were infrequently inconsistent and also sometimes overestimated the number of clusters. We observed using the results from Table 6.5 that the new approach performs well. It is a satisfactory solution based on the true number of clusters (prior information) that the data consist of two classes (Benign and Malignant).

6.3 Datasets with Unknown Clustering Structure

This section will present the performance of the new approach on two datasets, the Framingham Heart Study (FHS) [174] data and the Medical Expenditure Panel Surveys (MEPS) [175] data. Originally these datasets had no a priori clustering structure or pre-determined classes. They will be used to determine whether the new approach provides sensible clustering structure and compare the results with other existing validation indexes. The *k-means* algorithm is sensitive to noise or extreme values as discussed in Chapter 3. This can affect the clustering structure and so the outliers, elements which are distanced from the most of the elements in the dataset were removed (see [57]) prior to the analysis. These outliers are identified and removed using box plots, standard deviation and Inter Quartile Range (IQR).

6.3.1 Dataset: Framingham Heart Study (FHS)

The Framingham Heart Study was a longitudinal population based study of cardiovascular disease among initially healthy people in the community of Framingham Massachusetts. The study began in 1948 with 5209 healthy men and women aged between 28-62 years. Since that time many studies have been carried out using the FHS dataset such as [176, 177, 178]. The subjects participating in the Framingham study were regularly surveyed to check their cardiovascular condition. The clinical examination data contained cardiovascular risk factors and markers of disease including blood pressure, lung function, smoking history etc. The subset of the FHS dataset used here was for 4434 participants who had data collected during three examinations over the period from 1956 to 1968. The total number of observations in this dataset is 11627 with 39 variables and there is no information on the number of clusters. Due to missing values 1918 observations were removed. In the following only the first examination period will be used for analysis. This consists of 3884 observations. Interest is in finding the people having risk of heart disease among ages 57 to 70 inclusive. In order to identify reasonable clusters, the variables TOTCHOL (Total Cholesterol), GLUCOSE (Casual serum glucose) and BMI (body mass index) were examined. The *k-means* algorithm is sensitive to outlier (extreme values) [179], As mentioned above these outliers were identified and removed from the dataset using box plots and IQR. The remaining 1027 elements from FHS dataset are used for the analysis.

Table 6.7(a) below depicts the similarity values and overlaps at fixed k with different $k + r$ distances while Table 6.7(b) shows these values at different k with fixed $k + r$ mapping distances. Table 6.7(c) shows the maximum, minimum, traces (elements similarity) and overlap, average trace (similarity), overlap and the coefficient of variation (CV). The results indicate clusters are stable at $k = 2$ and $k = 3$ when average similarity between clusters is a maximum with minimum average overlap. The new approach gives $k = 3$ as the estimated number of clusters (using criteria 2 Chapter 4 section 4.3.3) for FHS dataset.

$k = 2, r = 1,2,\dots,K - k$

k	r	$(k,k+r) \times (k+r,k)$	*Traces (similarity)*	*Overlap*
2	1	(2,3×3,2)	**1006.46**	**20.54**
2	2	(2,4×4,2)	**769.91**	**257.09**
2	3	(2,5×5,2)	**971.2**	**55.8**
2	4	(2,6×6,2)	**966.75**	**60.25**
2	5	(2,7×7,2)	**925.67**	**101.33**
2	6	(2,8×8,2)	**900.68**	**126.32**
2	7	(2,9×9,2)	**931.49**	**95.51**
2	8	(2,10×10,2)	**900.68**	**126.32**
2	9	(2,11×11,2)	**900.68**	**126.32**
2	10	(2,12×12,2)	**931.49**	**95.51**
2	11	(2,13×13,2)	**931.49**	**95.51**
2	12	(2,14×14,2)	**941.76**	**85.24**
2	13	(2,15×15,2)	**956.48**	**70.52**
2	14	(2,16×16,2)	**941.76**	**85.24**

$k = 3, r = 1,2,\dots,K - k$

k	r	$(k,k+r) \times (k+r,k)$	*Traces (similarity)*	*Overlap*
3	1	(3,4×4,3)	**773.63**	**253.37**
3	2	(3,5×5,3)	**965.94**	**61.06**
3	3	(3,6×6,3)	**957.57**	**69.43**
3	4	(3,7×7,3)	**927.33**	**99.67**
3	5	(3,8×8,3)	**915.15**	**111.85**
3	6	(3,9×9,3)	**929.61**	**97.39**
3	7	(3,10×10,3)	**905.07**	**121.93**
3	8	(3,11×11,3)	**905.07**	**121.93**
3	9	(3,12×12,3)	**943.3**	**83.7**
3	10	(3,13×13,3)	**941.59**	**85.41**
3	11	(3,14×14,3)	**952.43**	**74.57**
3	12	(3,15×15,3)	**967.84**	**59.16**
3	13	(3,16×16,3)	**952.06**	**74.94**
⋮	⋮	⋮	⋮	⋮

$k = 15, r = 1$

k	r	$(k,k+r) \times (k+r,k)$	*Traces (similarity)*	*Overlap*
15	1	(15,16×16,15)	**745.04**	**281.96**

(a)

$k = 2,3,..15, r = 1$

k	$(k,k+r) \times (k+r,k)$	*Traces (similarity)*	*Overlap*
2	(2,3×3,2)	**1006.46**	**20.54**
3	(3,4×4,3)	**773.63**	**253.37**
4	(4,5×5,4)	**784.28**	**242.72**
5	(5,6×6,5)	**994.4**	**32.6**
6	(6,7×7,6)	**763.43**	**263.57**
7	(7,8×8,7)	**960.74**	**66.26**
8	(8,9×9,8)	**747.03**	**279.97**
9	(9,10×10,9)	**771.24**	**255.76**
10	(10,11×11,10)	**980.05**	**46.95**
11	(11,12×12,11)	**835.5**	**191.5**
12	(12,13×13,12)	**916.47**	**110.53**
13	(13,14×14,13)	**860.55**	**166.45**
14	(14,15×15,14)	**862.71**	**164.29**
15	(15,16×16,15)	**745.04**	**281.96**

$k = 2,3,..15, r = 2$

k	$(k,k+r) \times (k+r,k)$	*Traces (similarity)*	*Overlap*
2	(2,4×4,2)	**769.91**	**257.09**
3	(3,5×5,3)	**965.94**	**61.06**
4	(4,6×6,4)	**774.4**	**252.6**
5	(5,7×7,5)	**750.33**	**276.67**
6	(6,8×8,6)	**747.92**	**279.08**
7	(7,9×9,7)	**722.01**	**304.99**
8	(8,10×10,8)	**813.58**	**213.42**
9	(9,11×11,9)	**777.71**	**249.29**
10	(10,12×12,10)	**816.75**	**210.25**
11	(11,13×13,11)	**875.37**	**151.63**
12	(12,14×14,12)	**852.69**	**174.31**
13	(13,15×15,13)	**798.15**	**228.85**
14	(14,16×16,14)	**886.72**	**140.28**
⋮	⋮	⋮	⋮

$k = 2, r = 14$

k	$(k,k+r) \times (k+r,k)$	*Traces (similarity)*	*Overlap*
2	(2,16×16,2)	**941.76**	**85.24**

(b)

	Similarity			***Overlap***			
K	***Max Trace***	***Min Trace***	***Average Traces***	***CV***	***Max Overlap***	***Min Overlap***	***Average Overlap***
2	**1006.46**	**769.91**	**926.893**	**0.058**	**257.09**	**20.54**	**100.107**
3	**967.84**	**773.63**	**925.892**	**0.054**	**253.37**	**59.16**	**101.108**
4	**927.07**	**774.4**	**861.57**	**0.054**	**252.6**	**99.93**	**165.43**
5	**994.4**	**741.75**	**848.197**	**0.081**	**285.25**	**32.6**	**178.803**
6	**876.81**	**747.92**	**832.769**	**0.054**	**279.08**	**150.19**	**194.231**
7	**960.74**	**722.01**	**810.203**	**0.081**	**304.99**	**66.26**	**216.797**
8	**813.58**	**747.03**	**779.975**	**0.03**	**279.97**	**213.42**	**247.025**
9	**777.71**	**704.88**	**744.286**	**0.035**	**322.12**	**249.29**	**282.714**
10	**980.05**	**791.62**	**855.648**	**0.078**	**235.38**	**46.95**	**171.352**
11	**898.37**	**804.74**	**848.342**	**0.045**	**222.26**	**128.63**	**178.658**
12	**916.47**	**763.44**	**838.195**	**0.076**	**263.56**	**110.53**	**188.805**
13	**860.55**	**780.47**	**813.057**	**0.052**	**246.53**	**166.45**	**213.943**
14	886.72	862.71	**874.715**	**0.019**	164.29	140.28	**152.285**

(c)

Table 6.6: A collection of tables show the values calculated for Q matrices at different k with $k + r$ mapping distances.

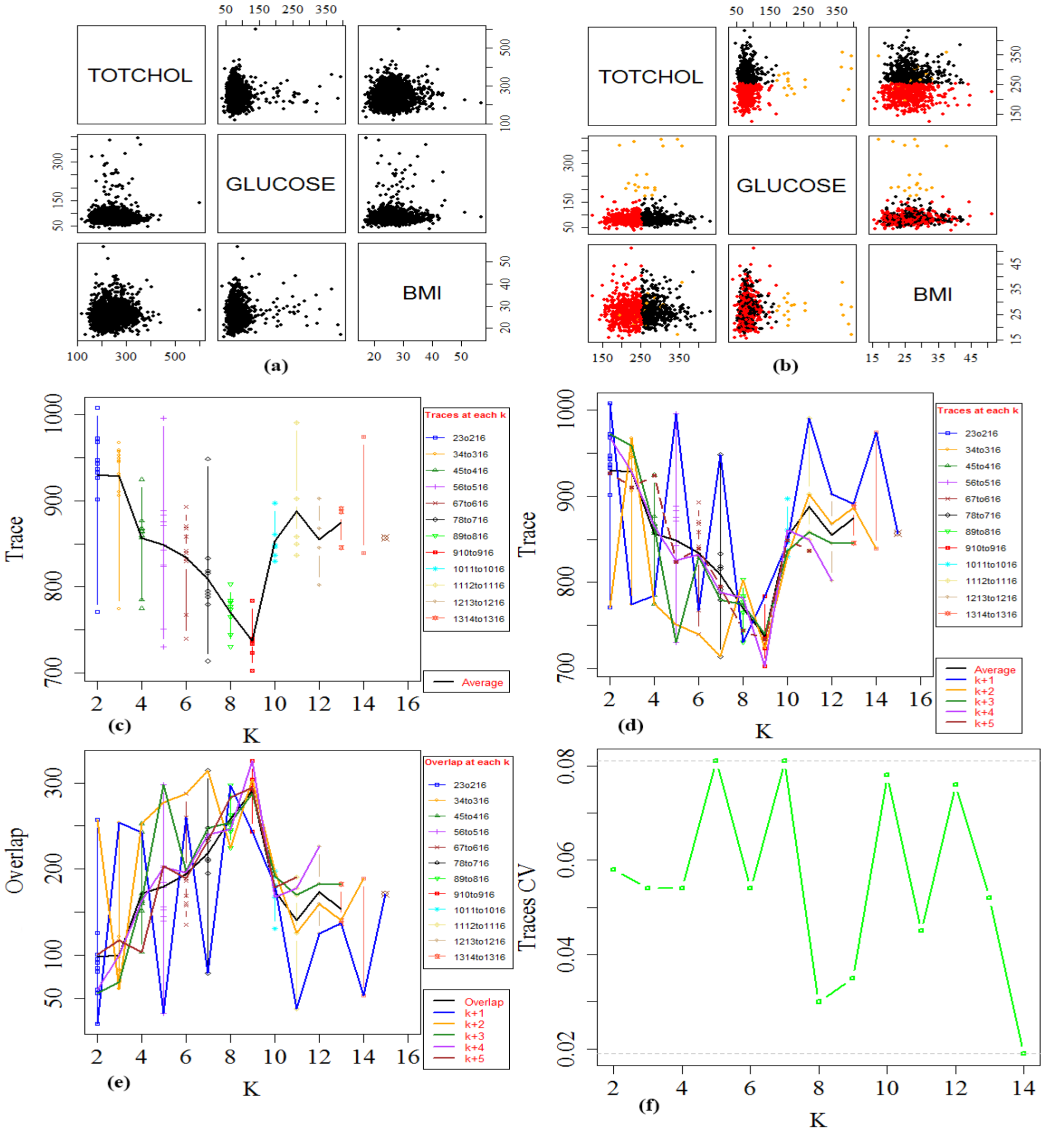


Figure 6.3: Part (a) shows FHS data scatter plot matrix, (b) *k-means* clusters and plots (c)-(f) are values from different Q matrices at different k for $k + r$ mapped distances.

Plot 6.3(b) shows the number of clusters obtained by the *k-means* algorithm using $k = 3$ with membership of elements labeled in different colours. Plot (c) shows the traces (number of elements similarity) at each k with different $k + r$ distances and the solid black line represents the values of average similarity at different k. Plots (d) and (e) show the differences as composite graphs for the behaviour of the trace values and overlap, average traces (similarity) and overlaps for $k + r$ distances shown by different coloured lines, while plot (f) represents the coefficient of variation for different k. These plots indicate the average similarity is a maximum till at $k = 3$ where there is minimum overlap and this set of clusters are settled as the minimum CV value is at $k = 3$. This confirms $k = 3$ is the optimum choice of k.

Table 6.7 shows the centroids of the clusters when $k = 3$ from the *k-means* algorithm.

Clusters	**TOTCHOL**	**GLUCOSE**	**BMI**	**Number of elements**
$C_{(3,1)}$	265.053	262.737	27.133	19
$C_{(3,2)}$	217.298	81.900	26.300	570
$C_{(3,3)}$	287.594	83.142	26.608	438

Table 6.7: Framingham Heart Study values of centroid for each cluster.

In Table 6.7 for clusters $C_{(3,2)}$ and $C_{(3,3)}$ the average values of cholesterol and glucose are smaller than the average value of the $C_{(3,1)}$ clusters. Therefore, an extra cluster with high average values of Cholesterol, Glucose and BMI is supported by the new approach. From Table 6.8 the *Dunn*, *CH* and *DB* indexes determined the number of clusters fluctuating between 2 and 9. The indexes *DH*, *Sil*, *SD*, *Gap* and *CCC* suggested 2 as the estimated number of clusters. It can be concluded that the FHS data consist of 3 clusters in which 2 clusters are large and 1 cluster is small. The small cluster may be ignored by the existing indexes. In Figure 6.3(b) and Table 6.7 (centroid of cluster values) it can be seen that 3 clusters and the respective elements assigned within these clusters have similar properties.

1^{st} run:				*FHS:* Existing indexes values						
K	***Dunn***	***DH***	***DH Critical***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***Gap Critical***	***CCC***
2	0.003	0.642	0.682	718.871	**0.43**	1.194	**0.095**	**1.796**	0.167	**77.548**
3	0.005	**1.467**	0.663	597.576	0.386	1.306	0.115	1.799	0.064	50.471
4	0.008	1.692	0.262	910.253	0.412	0.848	0.169	1.751	0	57.418
5	0.006	1.028	0.66	903.586	0.369	0.945	0.144	1.769	-0.055	55.919
6	0.007	1.536	0.648	**990.674**	0.382	**0.816**	0.147	1.694	-0.12	53.127
7	0.006	2.332	0.332	965.291	0.356	0.916	0.146	1.826	0.039	57.264
8	**0.01**	2.704	-0.123	962.34	0.316	0.957	0.133	1.811	0.006	56.512
9	0.009	3.56	0.399	986.368	0.344	0.959	0.163	1.869	0.084	56.999
10	0.006	2.024	-0.123	959.257	0.344	0.914	0.137	1.81	0.036	56.165

(a)

Number of runs		***Dunn***	***DH***	***CH***	***Sil***	***DB***	***SD***	***Gap***	***CCC***
2nd	***K***	**2**	**2**	**6**	**4**	**6**	**2**	**2**	**2**
	Values	**0.043**	**0.987**	**990.677**	**0.458**	**0.816**	**0.091**	**1.818**	**77.548**
3rd	***K***	**9**	**2**	**6**	**2**	**6**	**2**	**2**	**2**
	Values	**0.009**	**1.393**	**990.674**	**0.43**	**0.816**	**0.089**	**1.83**	**77.548**
4th	***K***	**7**	**2**	**9**	**2**	**6**	**2**	**2**	**2**
	Values	**0.01**	**1.393**	**987.382**	**0.43**	**0.816**	**0.106**	**1.83**	**77.548**
5th	***K***	**8**	**2**	**6**	**2**	**4**	**2**	**2**	**2**
	Values	**0.011**	**0.762**	**990.677**	**0.43**	**0.845**	**0.105**	**1.836**	**77.548**

(b)

Table 6.8: Shows the optimal numbers of clusters with their values highlighted in bold from eight different existing indexes with 5 multiple runs.

The new approach showed the estimated number of clusters to be $k = 3$ which is reasonable for this data. By examining Figure 6.3(b) and Table 6.7 there is strong evidence for an extra cluster with a high value of cholesterol and glucose while the other indexes ignore this $C_{(3,1)}$ small size cluster.

6.3.2 Dataset: Medical Expenditure Panel Survey (MEPS)

Recently in the United States of America (USA) there has been a need for determining the health care expenditure at the state level and for this reason the Medical Expenditure Panel Survey (MEPS) dataset has been designed. The MEPS began in 1996 and each year a new panel of sample households is collected for the USA civilian population. Besides health care expenditure information the survey also collects information on sources of payments and health insurance coverage. The MEPS also provides information on respondent health status, demographics and social-economic characteristics, employment, access to health care and satisfaction with health care. MEPS provides a large public dataset which contains a wealth of information and has been studied for different purposes [180, 181, 182]. Data includes on the following categories are available from MEPS.

- Unique person identifiers and survey administration variables
- Geographic variables
- Demographic variables
- Health status variables
- Disability days variables
- Access to care variables
- Employment variables
- Health insurance variables
- Utilization, expenditure, and source of payment variables
- Weight and variance estimation variables
- Income and tax filing variables
- Person-level priority condition variables

The dataset subset being used here for analysis is 2011 full-year consolidated H147 data from the MEPS HC which includes 2052 variables and 35313 number of observation (elements) in total and 11473 elements with missing information were removed. Here we are interested in the relation between health status and expenditure. Three relevant numerical variables TOTEXP (total expenditure), TOTSLF (self-expenditure) and BMI53 (body mass index) are used to define health expenditure in the cluster analysis following. These three variables include extreme values which are consider outliers (e.g. the values for $0 ≤ TOTEX ≤ $2226997 and $0 ≤ TOTSLF ≤ $93536). To determine sensible clustering structure, outliers were removed before applying the algorithms. Hence a dataset with ages inter quartile range between 30 and 58 inclusive as these are the most cases and \$10000 ≤ TOTEXP ≤ \$40000 was examined. This gave 859 total observations for analysis. These observations were derived from the year 2011 H147 data. The results were evaluated using the new approach and existing validation indexes.

Table 6.9(a) shows similarity and overlap at fixed k for different $k + r$ distances while Table 6.9(b) shows similarity at different k for fixed $k + r$ distances. Table 6.9(c) shows the minimum, maximum, similarity and overlap, average similarity and overlap and CV values for different k. The values in Table 6.9(c) shows average similarity is a maximum (801, 798 and 799) when $k = 2, k = 3$ and $k = 4$ with minimum average overlap (58, 61 and 60) and a quite small variation in CV value when $k = 4$ and this estimates the best number of clusters. Figure 6.4 shows a scatter plot matrix of the data, results from the *k-means* algorithm at $k = 4$ and line plots from Table 6.9.

$k = 2, r = 1,2,\ldots,K-k$

k	r	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***
2	1	(2,3×3,2)	**731.75**	**127.25**
2	2	(2,4×4,2)	**729.02**	**129.98**
2	3	(2,5×5,2)	**839.09**	**19.91**
2	4	(2,6×6,2)	**847.68**	**11.32**
2	5	(2,7×7,2)	**788.35**	**70.65**
2	6	(2,8×8,2)	**788.35**	**70.65**
2	7	(2,9×9,2)	**802.4**	**56.6**
2	8	(2,10×10,2)	**799.67**	**59.33**
2	9	(2,11×11,2)	**791.08**	**67.92**
2	10	(2,12×12,2)	**791.08**	**67.92**
2	11	(2,13×13,2)	**822.31**	**36.69**
2	12	(2,14×14,2)	**836.36**	**22.64**
2	13	(2,15×15,2)	**813.72**	**45.28**
2	14	(2,16×16,2)	**836.36**	**22.64**

$k = 3, r = 1,2,\ldots,K-k$

k	r	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***
3	1	(3,4×4,3)	**842.77**	**16.23**
3	2	(3,5×5,3)	**721.04**	**137.96**
3	3	(3,6×6,3)	**719.7**	**139.3**
3	4	(3,7×7,3)	**781.19**	**77.81**
3	5	(3,8×8,3)	**806.11**	**52.89**
3	6	(3,9×9,3)	**821.67**	**37.33**
3	7	(3,10×10,3)	**817.85**	**41.15**
3	8	(3,11×11,3)	**800.57**	**58.43**
3	9	(3,12×12,3)	**800.57**	**58.43**
3	10	(3,13×13,3)	**837.71**	**21.29**
3	11	(3,14×14,3)	**820.43**	**38.57**
3	12	(3,15×15,3)	**771.64**	**87.36**
3	13	(3,16×16,3)	**830.26**	**28.74**
⋮	⋮	⋮	⋮	⋮

$k = 15, r = 1$

k	r	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***
15	1	(15,16×16,15)	**683.02**	**175.98**

(a)

$k = 2,3,\ldots15, r = 1$

k	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***
2	(2,3×3,2)	**731.75**	**127.25**
3	(3,4×4,3)	**842.77**	**16.23**
4	(4,5×5,4)	**734.2**	**124.8**
5	(5,6×6,5)	**814.87**	**44.13**
6	(6,7×7,6)	**663.39**	**195.61**
7	(7,8×8,7)	**801.66**	**57.34**
8	(8,9×9,8)	**748.33**	**110.67**
9	(9,10×10,9)	**829.32**	**29.68**
10	(10,11×11,10)	**700.13**	**158.87**
11	(11,12×12,11)	**835.44**	**23.56**
12	(12,13×13,12)	**752.48**	**106.52**
13	(13,14×14,13)	**809.94**	**49.06**
14	(14,15×15,14)	**638.75**	**220.25**
15	(15,16×16,15)	**683.02**	**175.98**

$k = 2,3,\ldots15, r = 2$

k	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***
2	(2,4×4,2)	**729.02**	**129.98**
3	(3,5×5,3)	**721.04**	**137.96**
4	(4,6×6,4)	**708.82**	**150.18**
5	(5,7×7,5)	**635.49**	**223.51**
6	(6,8×8,6)	**662.64**	**196.36**
7	(7,9×9,7)	**702.04**	**156.96**
8	(8,10×10,8)	**734.41**	**124.59**
9	(9,11×11,9)	**703.16**	**155.84**
10	(10,12×12,10)	**722.62**	**136.38**
11	(11,13×13,11)	**727.99**	**131.01**
12	(12,14×14,12)	**722.79**	**136.21**
13	(13,15×15,13)	**607.74**	**251.26**
14	(14,16×16,14)	**812.6**	**46.4**
⋮	⋮	⋮	⋮

$k = 2, r = 14$

k	$(k, k+r) \times (k+r, k)$	***Traces (similarity)***	***Overlap***
2	(2,16×16,2)	**836.36**	**22.64**

(b)

	Similarity			***Overlap***			
K	***Max Trace***	***Min Trace***	***Average Traces***	***CV***	***Max Overlap***	***Min Overlap***	***Average Overlap***
2	847.68	729.02	**801.23**	**0.045**	129.98	11.32	**57.77**
3	842.77	719.7	**797.808**	**0.05**	139.3	16.23	**61.192**
4	840.26	708.82	**799.143**	**0.052**	150.18	18.74	**59.857**
5	814.87	635.49	**739.923**	**0.081**	223.51	44.13	**119.077**
6	771	662.64	**727.371**	**0.066**	196.36	88	**131.629**
7	801.66	666.81	**699.501**	**0.058**	192.19	57.34	**159.499**
8	748.33	676.95	**706.835**	**0.033**	182.05	110.67	**152.165**
9	829.32	666.69	**713.4**	**0.075**	192.31	29.68	**145.6**
10	737.62	668.46	**702.16**	**0.037**	190.54	121.38	**156.84**
11	835.44	690.73	**743.408**	**0.074**	168.27	23.56	**115.592**
12	752.48	691.38	**728.365**	**0.038**	167.62	106.52	**130.635**
13	809.94	607.74	**728.967**	**0.147**	251.26	49.06	**130.033**
14	812.6	638.75	**725.675**	**0.169**	220.25	46.4	**133.325**

(c)

Table 6.9: A collection of tables show the values calculated from Q matrices at different k for $k + r$ mapping distances.

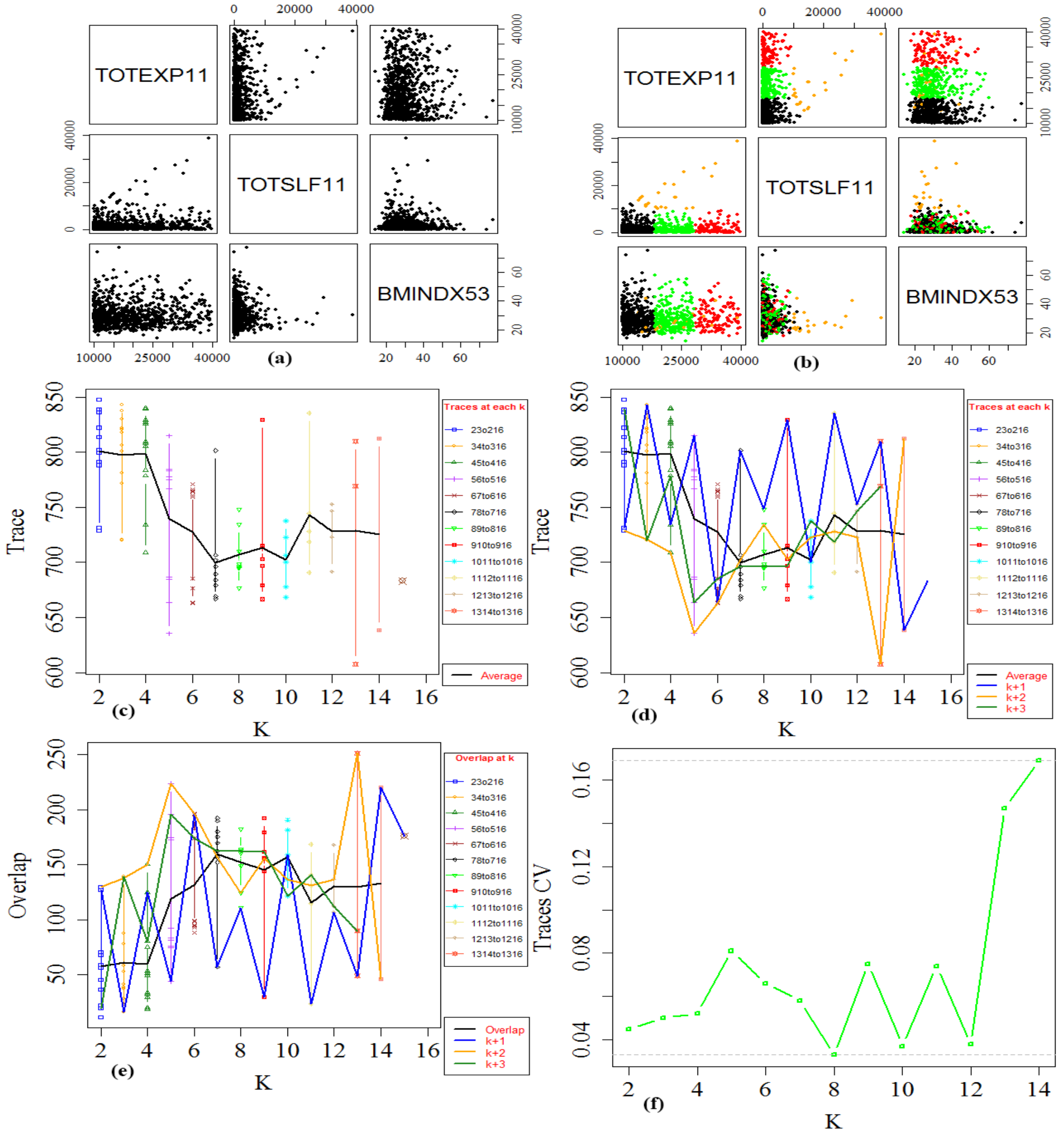


Figure 6.4: Part (a) shows a data scatter plot matrix, (b) *k-means* clusters when $k = 4$ and plots (c)-(f) represents results from Q matrices.

In Figure 6.4 plot (b) shows the number of clusters obtained from *k-means* at $k = 4$ with membership of elements labelled in different colours. Plot (c) shows the similarity at each k with different $k + r$ distances and the black solid line represents the average similarity values are a maximum till $k = 4$ and then decrease for higher k values. Plots (d) and (e) show the differences as composite graphs for the behaviour of trace values (number of elements similarity) and overlap, average traces (average number of elements similarity) and average overlap with coloured lines for $k + r$ distances, while (f) represents the coefficient of variation at different k. The plots above clearly show average similarity is a maximum (criteria 2 Chapter 4 section 4.3.3) with minimum average overlap at $k = 4$. This indicates the estimated number of clusters and the minimum CV value shows set of clusters are stable at the best $\boldsymbol{K}$. The table below represents the clusters centroids when $k = 4$ from the *k-means* algorithm.

Clusters	TOTEXP11	TOTSLF11	BMINDX53	Number of elements
$C_{(4,1)}$	13261.5	1512.119	30.390	472
$C_{(4,2)}$	33737.50	1782.050	30.971	119
$C_{(4,3)}$	22348.56	16767.957	27.365	23
$C_{(4,4)}$	22869.89	1413.833	30.753	245

Table 6.10: Data clusters centroid values at $k = 4$.

The results in the above table shows cluster $C_{(4,3)}$ is an extra small cluster identified by the new approach. In this cluster subjects have an average total and self-expenditure which is higher than the average expenditure for the other 3 clusters. The results in Table 6.11 show *DH*, *Sil*, *SD* and *CCC* indexes indicate 2 while *Dunn* and *Gap* indicate 3 as the estimated number of clusters. The *CH* and *DB* indexes overestimate the number of clusters even though *DB* sometimes indicates 2 but these indexes give inconsistent estimates. All indexes frequently identify 2 or 3 as the estimated number of clusters except the *CH* index which works poorly for the H147 data. The new approach determined the estimated number of clusters as 4 where the average similarity is a maximum with minimum average overlap. This choice of k is supported by Figure 6.4(c) where the segment of the black solid lines settled at $k = 4$.

1st run:	*MEPS (H147):* Existing indexes values									
K	*Dunn*	*DH*	*DH Critical*	*CH*	*Sil*	*DB*	*SD*	*Gap*	*Gap Critical*	*CCC*
2	0.005	**0.818**	0.653	1461.823	0.595	**0.759**	**0**	**1.379**	0.141	**45.007**
3	0.007	1.381	0.686	1013.783	**0.603**	0.932	0	1.26	0.057	31.679
4	0.008	0.39	0.577	1320.865	0.555	1.127	0.001	1.359	0.265	27.132
5	**0.011**	0.373	0.562	1321.585	0.459	0.839	0.001	1.39	0.097	31.864
6	0.003	1.026	0.472	1440.067	0.486	0.914	0.001	1.234	-0.018	30.24
7	0.003	2.165	-0.056	1477.988	0.357	0.781	0.001	1.503	-0.011	27.398
8	0.006	0.863	0.547	**1483.472**	0.394	0.855	0.001	1.502	0.018	34.208
9	0.005	1.086	0.581	1432.768	0.392	0.851	0.001	1.51	0.02	32.74
10	0.006	1.935	0.614	1373.39	0.404	0.844	0.001	1.502	0.025	34.278

(a)

Number of runs		*Dunn*	*DH*	*CH*	*Sil*	*DB*	*SD*	*Gap*	*CCC*
2nd	*K*	4	2	10	2	2	4	3	2
	Values	0.008	1.042	1501.266	0.595	0.759	0	1.262	45.007
3rd	*K*	3	2	7	2	6	2	2	2
	Values	0.007	1.005	1477.882	0.595	0.756	0	1.409	45.007
4th	*K*	3	2	10	2	2	4	3	2
	Values	0.006	1.005	1499.755	0.595	0.759	0	1.43	45.007
5th	*K*	4	2	9	2	2	2	3	2
	Values	0.008	1.004	1516.593	0.595	0.759	0	1.44	45.007

(b)

Table 6.11: Summarises the optimal numbers of clusters with the values highlighted in bold from eight different existing indexes with 5 multiple runs.

The new approach in comparison to the values computed from existing indexes performed better in finding $k = 4$. This choice of k is also visually acceptable by examining the plot in Figure 6.4(b) where elements in the clusters are labelled with different colours. The above discussion provides evidence for isolation of an extra distinct cluster as suggested by the new approach. Similarly, the illustration and discussion corresponding to the FHS data also provide strong indication of an extra cluster for the data. It is concluded based on these graphs it is highly recommended to examine the plotted lines to understand easily the clustering structure using the new approach.

6.4 Summary

In this chapter different real world datasets were used to compare the new approach with the use of various existing and well known validation indexes for determining the optimum number of clusters. The datasets were divided into two categories: one with clustering structure known in advance while the others had unknown clustering structure. The Physed and Wisconsin Breast Cancer datasets were used to validate finding the correct answer when the number of clusters was known a priori. The FHS and MEPS datasets had unknown clustering structure and the new approach was used and compared with other indexes to estimate the number of clusters and their stability.

The results showed for the FHS and MEPS datasets with no prior cluster information that the new approach was able to detect more clusters than the existing validation indexes. These extra clusters had completely distinct characteristics compared to the other clusters. It was also observed that in the case of adjacent or non-adjacent mapping, the plotted lines contain multiple peaks which signify some adjacent or non-adjacent mapped clusters contain only a small number of elements. This may increase the trace values (similarity) due to some extreme values from the variables. Thus more than one peak in the plots shows the data consist of noise, and is not suitable for adjacent mapping. Therefore, more sequential mapping is required to calculate and determine the estimated number of clusters. The usefulness of the new approach has been examined for real datasets that naturally have a complex structure. By demonstrating and comparing results with other indexes the new approach was shown to work well for selecting more sensible clusters in the FHS and MEPS data. The results showed there is no generic approach appropriate for every type of dataset. To find the best solution required using existing indexes to be run multiple times. The new approach avoids these multiple runs. However, the results show application of the new approach worked well for estimating the number of clusters when the set of clusters are stable and the data are from the health area. Therefore, it is expected that new approach will be successful and valuable in broader areas of real applications besides the health datasets.

Chapter 7

Conclusion

7.1 Discussion

The *k-means* algorithm provides a method to construct any potential number of clusters as specified by the user, but for determining a meaningful and objective quality for the clustering structure we need to evaluate the result using some criterion for estimating the best number of clusters. In this thesis, a study has been made of this issue and a new approach proposed for systematically estimating the number of clusters. The thesis considered the effects of different clustering techniques as well as clustering validation indexes, and proposed a new approach that is based on forward and backward mapping of common elements in sequence sets of clusters to determine the best number of clusters.

Although the *k-means* algorithm has been investigated by many researchers from various perspectives, it is not well configured for the purpose of estimating the best number of clusters, and does not provide strong evidence of cluster stability and quality. To decide what is the optimal number of clusters there is no comprehensive solution available so far in the literature. Although many existing approaches are available, none of them provide firm and satisfactory answers, due to the high complexity of datasets when some elements are sufficiently close that overlaps (no boundaries) or groups of various sizes and shapes of different variations (mixture shapes) may occur. In contrast, the new proposed approach is able to specify the best number of clusters systematically from the results obtained by the *k-means* algorithm. In addition, it indicates whether the clusters obtained are fully separated (no overlap between clusters) or partially separated (some elements belong to more than one cluster).

The new approach has been validated and compared with eight different existing validation indexes by applying it on 12 varieties of simulated datasets and 4 real datasets. The datasets consisted of different sizes and mixtures of clusters, with large and small variations between elements in different clusters, and various numbers of clusters greater than two. Subsequently, for the initial values of the k trace (similarity) or average trace (average similarity) values may decrease as k increases and so indicate overlaps. If the values of average similarity reach a maximum peak, or continue along a maximum plateau and then begin to decrease for higher k an optimum k is indicated. This optimum is confirmed if all the clusters in the dataset are settled or well separated with minimum overlap (for only maximum peak (see figures 5.7(d), 5.15(d) and 5.17(d)) or for maximum plateau (see Figures 5.11(d), 5.12(d), 5.14 (d))) for partial separation. For the situation of full separation with no overlap see in figures (5.5(d), 5.6(d), 5.10(d) and 5.16(d)) where average similarity has a maximum equal to number of elements and a 0 coefficient of variation. Finally, in the situation of severe noise clusters may be partially separated with minimum CV value at the best $\boldsymbol{K}$ (see figures 5.8(d) and 5.13(d)).

This study shows that none of the indexes discussed in Chapter 3 are appropriate for every type of data, since the application of these indexes behaves inconsistently for all datasets used in the study. From the results, even though the existing indexes work well in some situations, it was observed that the new approach performed well for all different types of datasets that included cluster shapes of circular, square, elliptical, mixtures of various and equal size clusters with large and small variations. In the above situation, the new approach is not only useful to find the best number of clusters but also to determine whether the set of clusters are fully separated or partially separated. In addition, it also examines the cluster stability and indicates how stable the clusters are at the best ***K*** as judged by a minimum coefficient of variation.

The new approach was applied to the Medical Expenditure Panel Survey (MEPS) and the Framingham Heart Study (FHS) real application datasets, which did not have any clusters or classes specified in advance. The robustness and efficiency of the new approach was checked also for the Rusipini, Physical activities and Breast Cancer real datasets where the clusters or classes were known in advance. The new approach identified the correct number of cluster for both these cases. It also worked effectively and successfully for the MEPS and FHS real application datasets to determine a sensible clustering structure. The main purpose of analysing these datasets is generally to find the groups of people who are relatively similar based on the variables selected like BMI, Cholesterol, and Glucose for FHS, or BMI, TOTEXP and TOTSLF for MEPS.

This investigation has showed that the new approach for evaluating clustering results worked well to detect clustering structures for certain types of complicated datasets such as those above. Typically, existing cluster validity indexes depend on the nature (structure of data) of the datasets (for example *CH*, *Sil* indexes work well for case1 and case2 of type3 datasets in Chapter 5 but were unable to find the correct answer for some other cases of type1 and type2 datasets). Although these existing indexes are useful, their results are not consistent, so it is an open issue to choose the right clustering algorithm and validation index to obtain the best number of clusters. The proposed new approach was shown to outperform these existing validation indexes by providing more reasonable estimates for the number of clusters across a wide range of different datasets.

It was expected that a limitation for the new approach would be when the dataset contains large and small variations in the components of the multidimensional set of values for each elements. The case2 of type3 dataset is an example, which is mixture of larger and smaller variations of various size clusters. Even though it is an adverse situation the performance of the new approach was still satisfactory. Based on the discussion and results from all the datasets, it is highly recommended that for better understanding of the data it is necessary to examine carefully the plots obtained from Q matrices, since these can provide better information visually for exploratory analysis.

7.2 Future Research Work

In *k-means* clustering validations and finding the optimal number of clusters are usually calculated and determined by multiple runs with different initial k numbers, and the best results determine the optimal cluster number, discussed in details in Chapter 3 sections. For small dataset this is not a significant problem, but for large and complex datasets this can be a serious issue. Clearly both multiple runs and randomly chosen of initial seeds are time consuming. The results showed that the new approach has less computation, relatively faster and does not require multiple runs for any dataset including the large and complex datasets. The implementation of the new approach is very convenient, scalable and time efficient. It requires only the mapping of the elements between the k number of clusters with simple associated computations. This was tested and experimented using variety of datasets and in different domains. Details of all these conclusions can be found in Chapters 5 and 6. The new approach has performed well with equal and different sizes of clusters for datasets with low, medium and high densities and variances. It was more robust for spherical and non-spherical (elliptical) dataset shapes with both low and high variances. One of the main motivations of this thesis was to provide an effective and purposeful guidance for determining the best and stable number of clusters. However, the results showed the existing validation indexes did not perform well with spherical, non-spherical (elliptical) data distribution and complex data types such as MEPS and FHS.

In clustering analysis, there might be also some confusion why cluster validation is necessary. Mostly, cluster analysis is conducted as a part of exploratory data analysis. Although abundant researches on *k-means* validation indexes are exist in the literature, but none of them is convincingly acceptable [183], especially for when the dataset complexity increase. The new approach has described and added number of critical steps in cluster analysis as an exploratory analysis to find not only the best and stable set of clusters but also to identify the fully or partially separated clusters and the number of overlapping elements between different clusters.

This new way for estimating the number of clusters based on inter cluster mapping of elements provides much opportunity for future research. First, the research can be carried out to include more high dimensional datasets from different fields of science like astronomy, business, finances and genomics to check the performance of the new approach on a much wide range of datasets than those used in the current study. Second, as the *k-means* algorithm is one of many possible partitioning algorithm and the proposed approach is computed from the *k-means* clustering results, an extension to this research can be its application to other clustering algorithms particularly to any appropriate partitioning (PAM, CLARA and CLARANS etc.) algorithms discussed in Chapter 2. Third, a number of existing indexes with special properties have been discussed (e.g. in [129, 133]) which can also be considered for comparison with the new approach. Fourth, it will be also valuable to study the behaviour of the approach with different cluster discrimination measures (Euclidean, City Block and Mahalanobis etc.) for the partitioning algorithms described in chapter 2. Fifth, it will be also worthwhile to examine each $k + r$ mapped distance at different k, which may indicate the presence of small or large variation in the data if there are more peaks (maximum similarity). Sixth, the new approach can also be considered for use on a variety of simulated datasets, such as generated dataset using univariate and bivariate normal distribution with a non zero correlation.

Finally, a future direction can be also to check the behaviour of forward and backward split at each $k + r$ mapped distance to identify the elements that may be similar within mapped clusters while dissimilar between clusters. In the case of fully separated clusters without overlap it is very simple, as average similarity at the best $\boldsymbol{K}$ is equal to N so mapped elements for $k + r$ will be the same elements (see tables 5.2, 5.4, 5.11, 5.23 and 5.21). In the circumstances when clusters are not fully separated, it is possible to find the number of elements by taking a subset of the source to all the target sets of clusters. This further work will provide those elements that will be similar within a cluster and belonging to a different cluster, which is beyond the scope of this thesis. Although this may not be easily carried out in the case of overlapped clusters and high dimension datasets, it will be valuable to investigate the behaviour of each element for which some variables dominate and due to which the cluster structure changes. This will also find the number of elements belonging to different clusters which can be used to find the variables or observations in the datasets that may affect the cluster structure or quality.

As mentioned above this approach is the beginning of the probability and statistical approach to determine the optimal or best cluster number (the words “optimal” and “best” are used commonly in the literature whenever the analysis and validation reached the best results). This area of research is new and open for more statistical research work that can be extended and investigated with more advanced techniques. This can also further verify and justify the optimal number of clusters especially for large and complex data with different shapes and dimensions in different domains particularly the medical domain.

Finally, a self dependent approach, without using *k-means*, would be an excellent advance in this area of research and will better define the optimal solutions.

7.3 Summary

In clustering, the *k-means* algorithm is one of the most popular algorithms for detecting and estimating the number of clusters in a dataset. In this study, we examined the performance of a new approach not only for some synthetic and UCI datasets where the best number of clusters is known in advance but also for the real-world FHS and MEPS datasets that have no classes or clusters structure available. In this case the approach was shown to be useful for finding groups of objects with similar appearance patterns across various medical conditions and health expenditure related problems. From the results, it was observed that there are no clustering evaluation indexes available that can be used to solve the problem of determining the estimated number of clusters and simultaneously can explore the clustering structure (e.g. fully separated and stable set of clusters) at the best $\boldsymbol{K}$**.**

Usually the existing approaches using validation indexes have been developed with certain criteria that may work well with specific types of datasets. Applying these indexes mainly depends on how well the specific datasets meet those criteria, and results may be quite different if this is not the case. This research aimed to maximize the mapping similarity based on the forward and backward inter cluster mapping. The results showed appropriate clustering classification outcomes even in unfavourable cases. Furthermore, a major problem in public health data is that there is no satisfactory approach for determining the best number of clusters, although an appropriate decision about the cluster number is critical, and this is often addressed by naively applying existing validation clustering indexes. The results showed for both simulated and real datasets that the new approach provides an acceptable solution for estimating the number of clusters for datasets containing complex cluster structure. The new approach has shown robustness in detecting complex cluster structure of data and to identify the best number of clusters, while also indicating where the set of cluster are fully isolated or have some degree of overlap. This was in contrast to the other approaches, which simply provide an estimated number of clusters with only the numerical values of indexes. This new approach showed to be more descriptive, informative, analytical and stable than the other approaches (discussed in Chapter 3), especially in terms of clusters contents, mapping, overlapping elements and clusters stability [149,150, 151].

Appendix A

Approach Implementation Using R

The proposed approach is implemented using R computing language and below is some of the main parts of the code. The original full source code is produced using Sweave (combination of latex and R code) in Rstudio (tool) with extension .Rnw while the codes below do not include latex syntax. The below code was used to analyse the dataset “physed” with three variables (flexibility, speed and strength). In case of using different dataset, replace the “physed” with the new dataset name and the variables required to analysis and compute the results for the new approach.

```
#load the packages
library(clusterSim)
library(psych)
library(matrixcalc)
library(xlsx)
library(foreign)
#load the Dataset
physed=read.dta("D:/data/physed.dta",convert.dates=TRUE,convert.factors=TRUE,missing.type=FALSE,c
onvert.underscore=FALSE, warn.missing.labels=TRUE)
# summary of datasets Physed
summaryphysed=summary(physed[,c(1:2)])
describephysed=describe(physed[,c(1:2)])
# Scatter plot of dataset Physed
plot(physed[,c(2,3,4)],pch=20,cex=1.5,main="Scatter plot for data", col.main="red", font.main=1,
family="serif")
# Number of observations
Numberofobjectphysed = nrow(physed)
#clusters using k-means from k=2 to k=16 with setting different parameters
clphysed= list()
for(i in 2:16)
  {
   clphysed[[i]]<- kmeans(physed[,c(2,3,4)], centers=i,iter.max=1000,nstart=25)
}
# k matrices represents elements only on the diagonal
physedObs22<- table(clphysed[[2]]$cluster,clphysed[[2]]$cluster);physedObs22
physedObs33<- table(clphysed[[3]]$cluster,clphysed[[3]]$cluster);physedObs33
physedObs44<- table(clphysed[[4]]$cluster,clphysed[[4]]$cluster);physedObs44
physedObs55<- table(clphysed[[5]]$cluster,clphysed[[5]]$cluster);physedObs55
```

```
physedObs66<- table(clphysed[[6]]$cluster,clphysed[[6]]$cluster);physedObs66
physedObs77<- table(clphysed[[7]]$cluster,clphysed[[7]]$cluster);physedObs77
physedObs88<- table(clphysed[[8]]$cluster,clphysed[[8]]$cluster);physedObs88
physedObs99<- table(clphysed[[9]]$cluster,clphysed[[9]]$cluster);physedObs99
physedObs1010<- table(clphysed[[10]]$cluster,clphysed[[10]]$cluster);physedObs1010
physedObs1111<- table(clphysed[[11]]$cluster,clphysed[[11]]$cluster);physedObs1111
physedObs1212<- table(clphysed[[12]]$cluster,clphysed[[12]]$cluster);physedObs1212
physedObs1313<- table(clphysed[[13]]$cluster,clphysed[[13]]$cluster);physedObs1313
physedObs1414<- table(clphysed[[14]]$cluster,clphysed[[14]]$cluster);physedObs1414
physedObs1515<- table(clphysed[[15]]$cluster,clphysed[[15]]$cluster);physedObs1515
physedObs1616<- table(clphysed[[16]]$cluster,clphysed[[16]]$cluster);physedObs1616
```

- ➢ Forward and backward mapping the elements when $k = 2$ for $k + r$ in a sequence of $r = 1,2,\ldots,K - k$

```
### Description of clusters for Physed data at K=3 Centers, Total with sum of squares, Within SS,
Between SS and Size of clusters
round(clphysed[[2]]$centers,digits=3);round(clphysed[[2]]$totss,digits=3);round(clphysed[[2]]$wit
hinss,digits=3);round(clphysed[[2]]$tot.withinss,digits=3);round(clphysed[[2]]$betweenss,digits=3
);clphysed[[2]]$size
#Plot at k=2
plot(physed[,c(2,3,4)],pch=20,cex=1,col=physed.color.cluster2,main="K=2 clusters);
points(clphysed[[2]]$centers, col = 2:3, pch = 8)
# At k=2 to 1,2,..,K-k forward and backward mapping of common elements and their proportions,
combined mapped proportions and combined mapped elements matrices
physedObs23     <- table(clphysed[[2]]$cluster,clphysed[[3]]$cluster);physedObs23
physedcomprop23 <- round(physedObs23/rowSums(physedObs23),digits=3);physedcomprop23
physedObs32     <- table(clphysed[[3]]$cluster,clphysed[[2]]$cluster);physedObs32
physedcomprop32 <- round(physedObs32/rowSums(physedObs32),digits=3);physedcomprop32
cpphysed.23x32  <- round((physedcomprop23)%*%(physedcomprop32),digits=2)%*%physedObs22

physedObs24     <- table(clphysed[[2]]$cluster,clphysed[[4]]$cluster);physedObs24
physedcomprop24 <- round(physedObs24/rowSums(physedObs24),digits=3);physedcomprop24
physedObs42     <- table(clphysed[[4]]$cluster,clphysed[[2]]$cluster);physedObs42
physedcomprop42 <- round(physedObs42/rowSums(physedObs42),digits=3);physedcomprop42
cpphysed.24x42  <- round((physedcomprop24)%*%(physedcomprop42),digits=2)%*%physedObs22

physedObs25<- table(clphysed[[2]]$cluster,clphysed[[5]]$cluster);physedObs25
physedcomprop25<- round(physedObs25/rowSums(physedObs25),digits=3);physedcomprop25
physedObs52 <- table(clphysed[[5]]$cluster,clphysed[[2]]$cluster);physedObs52
physedcomprop52 <- round(physedObs52/rowSums(physedObs52),digits=3);physedcomprop52
cpphysed.25x52<-round((physedcomprop25)%*%(physedcomprop52),digits=2)%*%physedObs22

physedObs26<- table(clphysed[[2]]$cluster,clphysed[[6]]$cluster);physedObs26
physedcomprop26<- round(physedObs26/rowSums(physedObs26),digits=3);physedcomprop26
physedObs62 <- table(clphysed[[6]]$cluster,clphysed[[2]]$cluster);physedObs62
physedcomprop62 <- round(physedObs62/rowSums(physedObs62),digits=3);physedcomprop62
```

```
cpphysed.26x62<-round((physedcomprop26)%*%(physedcomprop62),digits=2)%*%physedObs22

physedObs27<- table(clphysed[[2]]$cluster,clphysed[[7]]$cluster);physedObs27
physedcomprop27<- round(physedObs27/rowSums(physedObs27),digits=3);physedcomprop27
physedObs72 <- table(clphysed[[7]]$cluster,clphysed[[2]]$cluster);physedObs72
physedcomprop72 <- round(physedObs72/rowSums(physedObs72),digits=3);physedcomprop72
cpphysed.27x72<-round((physedcomprop27)%*%(physedcomprop72),digits=2)%*%physedObs22

physedObs28<- table(clphysed[[2]]$cluster,clphysed[[8]]$cluster);physedObs28
physedcomprop28<- round(physedObs28/rowSums(physedObs28),digits=3);physedcomprop28
physedObs82 <- table(clphysed[[8]]$cluster,clphysed[[2]]$cluster);physedObs82
physedcomprop82 <- round(physedObs82/rowSums(physedObs82),digits=3);physedcomprop82
cpphysed.28x82<-round((physedcomprop28)%*%(physedcomprop82),digits=2)%*%physedObs22

physedObs29<- table(clphysed[[2]]$cluster,clphysed[[9]]$cluster);physedObs29
physedcomprop29<- round(physedObs29/rowSums(physedObs29),digits=3);physedcomprop29
physedObs92 <- table(clphysed[[9]]$cluster,clphysed[[2]]$cluster);physedObs92
physedcomprop92 <- round(physedObs92/rowSums(physedObs92),digits=3);physedcomprop92
cpphysed.29x92<-round((physedcomprop29)%*%(physedcomprop92),digits=2)%*%physedObs22

physedObs210<- table(clphysed[[2]]$cluster,clphysed[[10]]$cluster);physedObs210
physedcomprop210<- round(physedObs210/rowSums(physedObs210),digits=3);physedcomprop210
physedObs102 <- table(clphysed[[10]]$cluster,clphysed[[2]]$cluster);physedObs102
physedcomprop102 <- round(physedObs102/rowSums(physedObs102),digits=3);physedcomprop102
cpphysed.210x102<-round((physedcomprop210)%*%(physedcomprop102),digits=2)%*%physedObs22

physedObs211<- table(clphysed[[2]]$cluster,clphysed[[11]]$cluster);physedObs211
physedcomprop211<- round(physedObs211/rowSums(physedObs211),digits=3);physedcomprop211
physedObs112 <- table(clphysed[[11]]$cluster,clphysed[[2]]$cluster);physedObs112
physedcomprop112 <- round(physedObs112/rowSums(physedObs112),digits=3);physedcomprop112
cpphysed.211x112<-round((physedcomprop211)%*%(physedcomprop112),digits=2)%*%physedObs22

physedObs212<- table(clphysed[[2]]$cluster,clphysed[[12]]$cluster);physedObs212
physedcomprop212<- round(physedObs212/rowSums(physedObs212),digits=3);physedcomprop212
physedObs122 <- table(clphysed[[12]]$cluster,clphysed[[2]]$cluster);physedObs122
physedcomprop122 <- round(physedObs122/rowSums(physedObs122),digits=3);physedcomprop122
cpphysed.212x122<-round((physedcomprop212)%*%(physedcomprop122),digits=2)%*%physedObs22

physedObs213<- table(clphysed[[2]]$cluster,clphysed[[13]]$cluster);physedObs213
physedcomprop213<- round(physedObs213/rowSums(physedObs213),digits=3);physedcomprop213
physedObs132 <- table(clphysed[[13]]$cluster,clphysed[[2]]$cluster);physedObs132
physedcomprop132 <- round(physedObs132/rowSums(physedObs132),digits=3);physedcomprop132
cpphysed.213x132<-round((physedcomprop213)%*%(physedcomprop132),digits=2)%*%physedObs22

physedObs214<- table(clphysed[[2]]$cluster,clphysed[[14]]$cluster);physedObs214
physedcomprop214<- round(physedObs214/rowSums(physedObs214),digits=3);physedcomprop214
physedObs142 <- table(clphysed[[14]]$cluster,clphysed[[2]]$cluster);physedObs142
physedcomprop142 <- round(physedObs142/rowSums(physedObs142),digits=3);physedcomprop142
cpphysed.214x142<-round((physedcomprop214)%*%(physedcomprop142),digits=2)%*%physedObs22
```

```
physedObs215<- table(clphysed[[2]]$cluster,clphysed[[15]]$cluster);physedObs215
physedcomprop215<- round(physedObs215/rowSums(physedObs215),digits=3);physedcomprop215
physedObs152 <- table(clphysed[[15]]$cluster,clphysed[[2]]$cluster);physedObs152
physedcomprop152 <- round(physedObs152/rowSums(physedObs152),digits=3);physedcomprop152
cpphysed.215x152<-round((physedcomprop215)%*%(physedcomprop152),digits=2)%*%physedObs22

physedObs216<- table(clphysed[[2]]$cluster,clphysed[[16]]$cluster);physedObs216
physedcomprop216<- round(physedObs216/rowSums(physedObs216),digits=3);physedcomprop216
physedObs162 <- table(clphysed[[16]]$cluster,clphysed[[2]]$cluster);physedObs162
physedcomprop162 <- round(physedObs162/rowSums(physedObs162),digits=3);physedcomprop162
cpphysed.216x162<-round((physedcomprop216)%*%(physedcomprop162),digits=2)%*%physedObs22
```

- ➢ Forward and backward mapping the elements when $k = 3$ for $k + r$ in a sequence of $r = 1,2,\ldots,K - k$

```
### Description of clusters for Physed data at K=3 Centers, Total with sum of squares, Within SS,
Between SS and Size of clusters
round(clphysed[[3]]$centers,digits=3);round(clphysed[[3]]$totss,digits=3);round(clphysed[[3]]$wit
hinss,digits=3);round(clphysed[[3]]$tot.withinss,digits=3);round(clphysed[[3]]$betweenss,digits=3
);clphysed[[3]]$size
# Plot at k=3
plot(physed[,c(2,3,4)],pch=20,cex=1,col =physed.color.cluster3,main=”K=3 clusters”)
points(clphysed[[3]]$centers, col = 2:4, pch = 8)
# at k=3 to 1,2,..,K-k forward and backward mapping of common elements and their proportions,
combined mapped proportions and combined mapped elements matrices
physedObs23<- table(clphysed[[2]]$cluster,clphysed[[3]]$cluster);physedObs23
physedcomprop23<- round(physedObs23/rowSums(physedObs23),digits=3);physedcomprop23
cpphysed.32x23<-round((physedcomprop32)%*%(physedcomprop23),digits=2)%*%physedObs33

physedObs34<-     table(clphysed[[3]]$cluster,clphysed[[4]]$cluster);physedObs34
physedcomprop34<- round(physedObs34/rowSums(physedObs34),digits=3);physedcomprop34
physedObs43 <-     table(clphysed[[4]]$cluster,clphysed[[3]]$cluster);physedObs43
physedcomprop43 <- round(physedObs43/rowSums(physedObs43),digits=3);physedcomprop43
cpphysed.34x43<-round((physedcomprop34)%*%(physedcomprop43),digits=2)%*%physedObs33

physedObs35<- table(clphysed[[3]]$cluster,clphysed[[5]]$cluster);physedObs35
physedcomprop35<- round(physedObs35/rowSums(physedObs35),digits=3);physedcomprop35
physedObs53 <- table(clphysed[[5]]$cluster,clphysed[[3]]$cluster);physedObs53
physedcomprop53 <- round(physedObs53/rowSums(physedObs53),digits=3);physedcomprop53
cpphysed.35x53<-round((physedcomprop35)%*%(physedcomprop53),digits=2)%*%physedObs33

physedObs36<- table(clphysed[[3]]$cluster,clphysed[[6]]$cluster);physedObs36
physedcomprop36<- round(physedObs36/rowSums(physedObs36),digits=3);physedcomprop36
physedObs63 <- table(clphysed[[6]]$cluster,clphysed[[3]]$cluster);physedObs63
physedcomprop63 <- round(physedObs63/rowSums(physedObs63),digits=3);physedcomprop63
cpphysed.36x63<-round((physedcomprop36)%*%(physedcomprop63),digits=2)%*%physedObs33

physedObs37<- table(clphysed[[3]]$cluster,clphysed[[7]]$cluster);physedObs37
```

```
physedcomprop37<- round(physedObs37/rowSums(physedObs37),digits=3);physedcomprop37
physedObs73 <- table(clphysed[[7]]$cluster,clphysed[[3]]$cluster);physedObs73
physedcomprop73 <- round(physedObs73/rowSums(physedObs73),digits=3);physedcomprop73
cpphysed.37x73<-round((physedcomprop37)%*%(physedcomprop73),digits=2)%*%physedObs33

physedObs38<- table(clphysed[[3]]$cluster,clphysed[[8]]$cluster);physedObs38
physedcomprop38<- round(physedObs38/rowSums(physedObs38),digits=3);physedcomprop38
physedObs83 <- table(clphysed[[8]]$cluster,clphysed[[3]]$cluster);physedObs83
physedcomprop83 <- round(physedObs83/rowSums(physedObs83),digits=3);physedcomprop83
cpphysed.38x83<-round((physedcomprop38)%*%(physedcomprop83),digits=2)%*%physedObs33

physedObs39<- table(clphysed[[3]]$cluster,clphysed[[9]]$cluster);physedObs39
physedcomprop39<- round(physedObs39/rowSums(physedObs39),digits=3);physedcomprop39
physedObs93<- table(clphysed[[9]]$cluster,clphysed[[3]]$cluster);physedObs93
physedcomprop93<- round(physedObs93/rowSums(physedObs93),digits=3);physedcomprop93
cpphysed.39x93<-round((physedcomprop39)%*%(physedcomprop93),digits=2)%*%physedObs33

physedObs310<- table(clphysed[[3]]$cluster,clphysed[[10]]$cluster);physedObs310
physedcomprop310<- round(physedObs310/rowSums(physedObs310),digits=3);physedcomprop310
physedObs103<- table(clphysed[[10]]$cluster,clphysed[[3]]$cluster);physedObs103
physedcomprop103<- round(physedObs103/rowSums(physedObs103),digits=3);physedcomprop103
cpphysed.310x103<-round((physedcomprop310)%*%(physedcomprop103),digits=2)%*%physedObs33

physedObs311<- table(clphysed[[3]]$cluster,clphysed[[11]]$cluster);physedObs311
physedcomprop311<- round(physedObs311/rowSums(physedObs311),digits=3);physedcomprop311
physedObs113<- table(clphysed[[11]]$cluster,clphysed[[3]]$cluster);physedObs113
physedcomprop113<- round(physedObs113/rowSums(physedObs113),digits=3);physedcomprop113
cpphysed.311x113<-round((physedcomprop311)%*%(physedcomprop113),digits=2)%*%physedObs33

physedObs312<- table(clphysed[[3]]$cluster,clphysed[[12]]$cluster);physedObs312
physedcomprop312<- round(physedObs312/rowSums(physedObs312),digits=3);physedcomprop312
physedObs123<- table(clphysed[[12]]$cluster,clphysed[[3]]$cluster);physedObs123
physedcomprop123<- round(physedObs123/rowSums(physedObs123),digits=3);physedcomprop123
cpphysed.312x123<-round((physedcomprop312)%*%(physedcomprop123),digits=2)%*%physedObs33

physedObs313<- table(clphysed[[3]]$cluster,clphysed[[13]]$cluster);physedObs313
physedcomprop313<- round(physedObs313/rowSums(physedObs313),digits=3);physedcomprop313
physedObs133<- table(clphysed[[13]]$cluster,clphysed[[3]]$cluster);physedObs133
physedcomprop133<- round(physedObs133/rowSums(physedObs133),digits=3);physedcomprop133
cpphysed.313x133<-round((physedcomprop313)%*%(physedcomprop133),digits=2)%*%physedObs33

physedObs314<- table(clphysed[[3]]$cluster,clphysed[[14]]$cluster);physedObs314
physedcomprop314<- round(physedObs314/rowSums(physedObs314),digits=3);physedcomprop314
physedObs143<- table(clphysed[[14]]$cluster,clphysed[[3]]$cluster);physedObs143
physedcomprop143<- round(physedObs143/rowSums(physedObs143),digits=3);physedcomprop143
cpphysed.314x143<-round((physedcomprop314)%*%(physedcomprop143),digits=2)%*%physedObs33

physedObs315<- table(clphysed[[3]]$cluster,clphysed[[15]]$cluster);physedObs315
physedcomprop315<- round(physedObs315/rowSums(physedObs315),digits=3);physedcomprop315
```

```
physedObs153<- table(clphysed[[15]]$cluster,clphysed[[3]]$cluster);physedObs153
physedcomprop153<- round(physedObs153/rowSums(physedObs153),digits=3);physedcomprop153
cpphysed.315x153<-round((physedcomprop315)%*%(physedcomprop153),digits=2)%*%physedObs33

physedObs316<- table(clphysed[[3]]$cluster,clphysed[[16]]$cluster);physedObs316
physedcomprop316<- round(physedObs316/rowSums(physedObs316),digits=3);physedcomprop316
physedObs163<- table(clphysed[[16]]$cluster,clphysed[[3]]$cluster);physedObs163
physedcomprop163<- round(physedObs163/rowSums(physedObs163),digits=3);physedcomprop163
cpphysed.316x163<-round((physedcomprop316)%*%(physedcomprop163),digits=2)%*%physedObs33
```

⋮ ⋮

- ➢ Forward and backward mapping the elements when $k = 15$ for $k + r$ in a sequence of $r = 1,2, \ldots, K - k$

```
### Description of clusters for Physed data at K=15 Centers, Total with sum of squares, Within
SS, Between SS and Size of clusters
round(clphysed[[15]]$centers,digits=3);round(clphysed[[15]]$totss,digits=3);round(clphysed[[15]]$
withinss,digits=3);round(clphysed[[15]]$tot.withinss,digits=3);round(clphysed[[15]]$betweenss,dig
its=3);clphysed[[15]]$size
 #Plot k=15
plot(physed[,c(2,3,4)],pch=20,cex=1,col =physed.color.cluster15,main="K=15 clusters)
points(clphysed[[15]]$centers, col = 2:15, pch = 8)
#at k=15 to 1,2,..,K-k forward and backward mapping of common elements and their proportions,
combined mapped proportions and combined mapped elements matrices
physedObs152<- table(clphysed[[15]]$cluster,clphysed[[2]]$cluster);physedObs152
physedcomprop152<- round(physedObs152/rowSums(physedObs152),digits=3);physedcomprop152
cpphysed.152x215<-round((physedcomprop152)%*%(physedcomprop215),digits=2)%*%physedObs1515

physedObs153<- table(clphysed[[15]]$cluster,clphysed[[3]]$cluster);physedObs153
physedcomprop153<- round(physedObs153/rowSums(physedObs153),digits=3);physedcomprop153
cpphysed.153x315<-round((physedcomprop153)%*%(physedcomprop315),digits=2)%*%physedObs1515

physedObs154<- table(clphysed[[15]]$cluster,clphysed[[4]]$cluster);physedObs154
physedcomprop154<- round(physedObs154/rowSums(physedObs154),digits=3);physedcomprop154
cpphysed.154x415<-round((physedcomprop154)%*%(physedcomprop415),digits=2)%*%physedObs1515

physedObs155<- table(clphysed[[15]]$cluster,clphysed[[5]]$cluster);physedObs155
physedcomprop155<- round(physedObs155/rowSums(physedObs155),digits=3);physedcomprop155
cpphysed.155x515<-round((physedcomprop155)%*%(physedcomprop515),digits=2)%*%physedObs1515

physedObs156<- table(clphysed[[15]]$cluster,clphysed[[6]]$cluster);physedObs156
physedcomprop156<- round(physedObs156/rowSums(physedObs156),digits=3);physedcomprop156
cpphysed.156x615<-round((physedcomprop156)%*%(physedcomprop615),digits=2)%*%physedObs1515

physedObs157<- table(clphysed[[15]]$cluster,clphysed[[7]]$cluster);physedObs157
physedcomprop157<- round(physedObs157/rowSums(physedObs157),digits=3);physedcomprop157
cpphysed.157x715<-round((physedcomprop157)%*%(physedcomprop715),digits=2)%*%physedObs1515
```

```
physedObs158<- table(clphysed[[15]]$cluster,clphysed[[8]]$cluster);physedObs158
physedcomprop158<- round(physedObs158/rowSums(physedObs158),digits=3);physedcomprop158
cpphysed.158x815<-round((physedcomprop158)%*%(physedcomprop815),digits=2)%*%physedObs1515

physedObs159<- table(clphysed[[15]]$cluster,clphysed[[9]]$cluster);physedObs159
physedcomprop159<- round(physedObs159/rowSums(physedObs159),digits=3);physedcomprop159
cpphysed.159x915<-round((physedcomprop159)%*%(physedcomprop915),digits=2)%*%physedObs1515

physedObs1510<- table(clphysed[[15]]$cluster,clphysed[[10]]$cluster);physedObs1510
physedcomprop1510<- round(physedObs1510/rowSums(physedObs1510),digits=3);physedcomprop1510
cpphysed.1510x1015<-round((physedcomprop1510)%*%(physedcomprop1015),digits=2)%*%physedObs1515

physedObs1511<- table(clphysed[[15]]$cluster,clphysed[[11]]$cluster);physedObs1511
physedcomprop1511<- round(physedObs1511/rowSums(physedObs1511),digits=3);physedcomprop1511
cpphysed.1511x1115<-round((physedcomprop1511)%*%(physedcomprop1115),digits=2)%*%physedObs1515

physedObs1512<- table(clphysed[[15]]$cluster,clphysed[[12]]$cluster);physedObs1512
physedcomprop1512<- round(physedObs1512/rowSums(physedObs1512),digits=3);physedcomprop1512
cpphysed.1512x1215<-round((physedcomprop1512)%*%(physedcomprop1215),digits=2)%*%physedObs1515

physedObs1513<- table(clphysed[[15]]$cluster,clphysed[[13]]$cluster);physedObs1513
physedcomprop1513<- round(physedObs1513/rowSums(physedObs1513),digits=3);physedcomprop1513
cpphysed.1513x1315<-round((physedcomprop1513)%*%(physedcomprop1315),digits=2)%*%physedObs1515

physedObs1514<- table(clphysed[[15]]$cluster,clphysed[[14]]$cluster);physedObs1514
physedcomprop1514<- round(physedObs1514/rowSums(physedObs1514),digits=3);physedcomprop1514
cpphysed.1514x1415<-round((physedcomprop1514)%*%(physedcomprop1415),digits=2)%*%physedObs1515

physedObs1516<- table(clphysed[[15]]$cluster,clphysed[[16]]$cluster);physedObs1516
physedcomprop1516<- round(physedObs1516/rowSums(physedObs1516),digits=3);physedcomprop1516
physedObs1615<- table(clphysed[[16]]$cluster,clphysed[[15]]$cluster);physedObs1615
physedcomprop1615<- round(physedObs1615/rowSums(physedObs1615),digits=3);physedcomprop1615
cpphysed.1516x1615<-round((physedcomprop1516)%*%(physedcomprop1615),digits=2)%*%physedObs1515
```

- Traces at $k = 2,3, \dots 15$.

```
#### Trace of matrices at k=2
trcphysed23=matrix.trace(cpphysed.23x32);trcphysed23
trcphysed24=matrix.trace(cpphysed.24x42);trcphysed24
trcphysed25=matrix.trace(cpphysed.25x52);trcphysed25
trcphysed26=matrix.trace(cpphysed.26x62);trcphysed26
trcphysed27=matrix.trace(cpphysed.27x72);trcphysed27
trcphysed28=matrix.trace(cpphysed.28x82);trcphysed28
trcphysed29=matrix.trace(cpphysed.29x92);trcphysed29
trcphysed210=matrix.trace(cpphysed.210x102);trcphysed210
trcphysed211=matrix.trace(cpphysed.211x112);trcphysed211
trcphysed212=matrix.trace(cpphysed.212x122);trcphysed212
trcphysed213=matrix.trace(cpphysed.213x132);trcphysed213
```

```
trcphysed214=matrix.trace(cpphysed.214x142);trcphysed214
trcphysed215=matrix.trace(cpphysed.215x152);trcphysed215
trcphysed216=matrix.trace(cpphysed.216x162);trcphysed216
```

```
# Trace of matrices at k=3
trcphysed34=matrix.trace(cpphysed.34x43);trcphysed34
trcphysed35=matrix.trace(cpphysed.35x53);trcphysed35
trcphysed36=matrix.trace(cpphysed.36x63);trcphysed36
trcphysed37=matrix.trace(cpphysed.37x73);trcphysed37
trcphysed38=matrix.trace(cpphysed.38x83);trcphysed38
trcphysed39=matrix.trace(cpphysed.39x93);trcphysed39
trcphysed310=matrix.trace(cpphysed.310x103);trcphysed310
trcphysed311=matrix.trace(cpphysed.311x113);trcphysed311
trcphysed312=matrix.trace(cpphysed.312x123);trcphysed312
trcphysed313=matrix.trace(cpphysed.313x133);trcphysed313
trcphysed314=matrix.trace(cpphysed.314x143);trcphysed314
trcphysed315=matrix.trace(cpphysed.315x153);trcphysed315
trcphysed316=matrix.trace(cpphysed.316x163);trcphysed316
```

```
  ⋮                                    ⋮
```

```
# Trace of matrices at k=15
trcphysed1516=matrix.trace(cpphysed.1516x1615);trcphysed1516
```

- Overlap at $k = 2,3, \ldots 15$.

```
##### Nnumber of objects overlaps at k=2
offdiagphysed23=numberofobjectphysed-trcphysed23;offdiagphysed23
offdiagphysed24=numberofobjectphysed-trcphysed24;offdiagphysed24
offdiagphysed25=numberofobjectphysed-trcphysed25;offdiagphysed25
offdiagphysed26=numberofobjectphysed-trcphysed26;offdiagphysed26
offdiagphysed27=numberofobjectphysed-trcphysed27;offdiagphysed27
offdiagphysed28=numberofobjectphysed-trcphysed28;offdiagphysed28
offdiagphysed29=numberofobjectphysed-trcphysed29;offdiagphysed29
offdiagphysed210=numberofobjectphysed-trcphysed210;offdiagphysed210
offdiagphysed211=numberofobjectphysed-trcphysed211;offdiagphysed211
offdiagphysed212=numberofobjectphysed-trcphysed212;offdiagphysed212
offdiagphysed213=numberofobjectphysed-trcphysed213;offdiagphysed213
offdiagphysed214=numberofobjectphysed-trcphysed214;offdiagphysed214
offdiagphysed215=numberofobjectphysed-trcphysed215;offdiagphysed215
offdiagphysed216=numberofobjectphysed-trcphysed216;offdiagphysed216
```

```
# Nnumber of objects overlaps at k=3
offdiagphysed34=numberofobjectphysed-trcphysed34;offdiagphysed34
offdiagphysed35=numberofobjectphysed-trcphysed35;offdiagphysed35
offdiagphysed36=numberofobjectphysed-trcphysed36;offdiagphysed36
offdiagphysed37=numberofobjectphysed-trcphysed37;offdiagphysed37
offdiagphysed38=numberofobjectphysed-trcphysed38;offdiagphysed38
offdiagphysed39=numberofobjectphysed-trcphysed39;offdiagphysed39
offdiagphysed310=numberofobjectphysed-trcphysed310;offdiagphysed310
offdiagphysed311=numberofobjectphysed-trcphysed311;offdiagphysed311
offdiagphysed312=numberofobjectphysed-trcphysed312;offdiagphysed312
offdiagphysed313=numberofobjectphysed-trcphysed313;offdiagphysed313
```

```
offdiagphysed314=numberofobjectphysed-trcphysed314;offdiagphysed314
offdiagphysed315=numberofobjectphysed-trcphysed315;offdiagphysed315
offdiagphysed316=numberofobjectphysed-trcphysed316;offdiagphysed316
```

⋮ ⋮

```
##### Number of objects overlaps at k=15
offdiagphysed1516=numberofobjectphysed-trcphysed1516;offdiagphysed1516
```

- ➢ Print Q matrices at $k = 2,3, \dots 15$.

```
##### Q combined mapped matrices at k=2 with different K+r
cpphysed.23x32;cpphysed.24x42;cpphysed.25x52;cpphysed.26x62;cpphysed.27x72;cpphysed.28x82
cpphysed.29x92;cpphysed.210x102;cpphysed.211x112;cpphysed.212x122;cpphysed.213x132;cpphysed.214x1
42;cpphysed.215x152;cpphysed.216x162
```

```
# Q combined mapped matrices at k=3 with different K+r
cpphysed.34x43;cpphysed.35x53;cpphysed.36x63;cpphysed.37x73;cpphysed.38x83;cpphysed.39x93
cpphysed.310x103;cpphysed.311x113;cpphysed.312x123;cpphysed.313x133;cpphysed.314x143;cpphysed.315
x153;cpphysed.316x163
```

⋮ ⋮

```
# Q combined mapped matrices at k=15 with different K+r
  cpphysed.1516x1615
```

- ➢ Create data frame for traces and overlap at different $k = 2,3, \dots 15$ for $k + 1, k + 2, \dots . k + 14$.

```
### Traces and overlaps at different k with k+1 mapped distances
mat23_1516 <- matrix(c(23,34,45,56,67,78,89,910,1011,1112,1213,1314,1415,1516))
###### Create data frame using cbind
trcphysedKplus1=cbind(c(2:15),mat23_1516,c(trcphysed23,trcphysed34,trcphysed45,trcphysed56,trcphy
sed67,trcphysed78,trcphysed89,trcphysed910,trcphysed1011,trcphysed1112,trcphysed1213,trcphysed131
4,trcphysed1415,trcphysed1516),c(offdiagphysed23,offdiagphysed34,offdiagphysed45,offdiagphysed56,
offdiagphysed67,offdiagphysed78,offdiagphysed89,offdiagphysed910,offdiagphysed1011,offdiagphysed1
112,offdiagphysed1213,offdiagphysed1314,offdiagphysed1415,offdiagphysed1516))
colnames(trcphysedKplus1) <- c("K","mat23_1516","Kplus1trace","Kplus1offdiag")
```

```
# Traces and overlaps at different k with k+2 mapped distances
mat24_1416 <- matrix(c(24,35,46,57,68,79,810,911,1012,1113,1214,1315,1416))
###### Create data frame using cbind
trcphysedKplus2=cbind(c(2:14),mat24_1416,c(trcphysed24,trcphysed35,trcphysed46,trcphysed57,trcphy
sed68,trcphysed79,trcphysed810,trcphysed911,trcphysed1012,trcphysed1113,trcphysed1214,trcphysed13
15,trcphysed1416),c(offdiagphysed24,offdiagphysed35,offdiagphysed46,offdiagphysed57,offdiagphysed
68,offdiagphysed79,offdiagphysed810,offdiagphysed911,offdiagphysed1012,offdiagphysed1113,offdiagp
hysed1214,offdiagphysed1315,offdiagphysed1416))
colnames(trcphysedKplus2) <- c("K","mat24_1416","Kplus2trace","Kplus2offdiag")
```

⋮ ⋮

```
# Traces and overlaps at k=2 with k+14 mapped distances
mat216_162 <- matrix(c(216))
###### Create data frame using cbind
trcphysedKplus14=cbind(c(2),mat216_162,c(trcphysed216),c(offdiagphysed216))
colnames(trcphysedKplus14) <- c("K","mat216_162","Kplus14trace","Kplus14offdiag")
```

- Create data frame for traces and overlap when fixed $k = 2,3, \ldots 15$ for different $k + 1, k + 2, \ldots . k + 14$.

```
### Traces and overlap at fixed k=2 with 1,2,...,K-k for k+r
mat23_216 <- matrix(c(23,24,25,26,27,28,29,210,211,212,213,214,215,216))
###### Create data frame using cbind
trcphysedK23to216=cbind(c(2:2),mat23_216,c(trcphysed23,trcphysed24,trcphysed25,trcphysed26,trcphy
sed27,trcphysed28,trcphysed29,trcphysed210,trcphysed211,trcphysed212,trcphysed213,trcphysed214,tr
cphysed215,trcphysed216),c(offdiagphysed23,offdiagphysed24,offdiagphysed25,offdiagphysed26,offdia
gphysed27,offdiagphysed28,offdiagphysed29,offdiagphysed210,offdiagphysed211,offdiagphysed212,offd
iagphysed213,offdiagphysed214,offdiagphysed215,offdiagphysed216))
colnames(trcphysedK23to216) <- c("FixedK","mat23_216","K23to216trace","K23to216offdig")
mat34_316 <- matrix(c(34,35,36,37,38,39,310,311,312,313,314,315,316))
###### Create data frame using cbind
trcphysedK34to316=cbind(c(3:3),mat34_316,c(trcphysed34,trcphysed35,trcphysed36,trcphysed37,trcphy
sed38,trcphysed39,trcphysed310,trcphysed311,trcphysed312,trcphysed313,trcphysed314,trcphysed315,t
rcphysed316),c(offdiagphysed34,offdiagphysed35,offdiagphysed36,offdiagphysed37,offdiagphysed38,of
fdiagphysed39,offdiagphysed310,offdiagphysed311,offdiagphysed312,offdiagphysed313,offdiagphysed31
4,offdiagphysed315,offdiagphysed316))
colnames(trcphysedK34to316) <- c("FixedK","mat34_316","K34to316trace","K34to316offdig")
```

⋮ ⋮

```
# Traces and overlap at fixed k=15 with K-k=1 for k+r
mat1516_1615 <- matrix(c(1516))
###### Create data frame using cbind
trcphysedK1516to1615=cbind(c(15),mat1516_1615,c(trcphysed1516),c(offdiagphysed1516))
colnames(trcphysedK1516to1615)<-c("FixedK","mat1516_1615","K1516to1516trace","K1516to1516offdig")
```

- Compute coefficient of variation (CV).

```
### Compute Coefficient of variation (CV) ###
cofvar <- function(q) {
    return(sd(q)/mean(q))
  }
```

- Traces and overlap range at different k.

```
### at each fixed k Trace Range
ateachKtraceRange=range(c(min(trcphysedK23to216[,3],trcphysedK34to316[,3],trcphysedK45to416[,3],t
rcphysedK56to516[,3],trcphysedK67to616[,3],trcphysedK78to716[,3],trcphysedK89to816[,3],trcphysedK
910to916[,3],trcphysedK1011to1016[,3],trcphysedK1112to1116[,3],trcphysedK1213to1216[,3],trcphysed
K1314to1316[,3],trcphysedK1415to1416[,3],trcphysedK1516to1615[,3])
,max(trcphysedK23to216[,3],trcphysedK34to316[,3],trcphysedK45to416[,3],trcphysedK56to516[,3],trcp
hysedK67to616[,3],trcphysedK78to716[,3],trcphysedK89to816[,3],trcphysedK910to916[,3],
trcphysedK1011to1016[,3],trcphysedK1112to1116[,3],trcphysedK1213to1216[,3],
trcphysedK1314to1316[,3],trcphysedK1415to1416[,3],trcphysedK1516to1615[,3])))
# at each fixed k Overlap Range
ateachKtraceoffdiagRange=range(c(min(trcphysedK23to216[,4],trcphysedK34to316[,4],trcphysedK45to41
6[,4],trcphysedK56to516[,4],trcphysedK67to616[,4],trcphysedK78to716[,4],trcphysedK89to816[,4],trc
physedK910to916[,4],trcphysedK1011to1016[,4],trcphysedK1112to1116[,4],trcphysedK1213to1216[,4],tr
cphysedK1314to1316[,4], trcphysedK1415to1416[,4],trcphysedK1516to1615[,4])
,max(trcphysedK23to216[,4],trcphysedK34to316[,4],trcphysedK45to416[,4],trcphysedK56to516[,4],trcp
hysedK67to616[,4],trcphysedK78to716[,4],trcphysedK89to816[,4],trcphysedK910to916[,4],trcphysedK10
11to1016[,4],trcphysedK1112to1116[,4],trcphysedK1213to1216[,4],trcphysedK1314to1316[,4],trcphysed
K1415to1416[,4],trcphysedK1516to1615[,4]))
```

- Maximum, minimum and average of traces at different k.

```
### Max trace at fixed k for 1,2,...,K-k for k+r ###
Maxtracephysed=c(max(trcphysedK23to216[,3]),max(trcphysedK34to316[,3]),max(trcphysedK45to416[,3])
,max(trcphysedK56to516[,3]),max(trcphysedK67to616[,3]),max(trcphysedK78to716[,3]),max(trcphysedK8
9to816[,3]),max(trcphysedK910to916[,3]),max(trcphysedK1011to1016[,3]),max(trcphysedK1112to1116[,3
]),max(trcphysedK1213to1216[,3]),max(trcphysedK1314to1316[,3]),max(trcphysedK1415to1416[,3]),max(
trcphysedK1516to1615[,3]))
# Min trace at fixed k for 1,2,...,K-k for k+r
Mintracephysed=c(min(trcphysedK23to216[,3]),min(trcphysedK34to316[,3]),min(trcphysedK45to416[,3])
,min(trcphysedK56to516[,3]),min(trcphysedK67to616[,3]),min(trcphysedK78to716[,3]),min(trcphysedK8
9to816[,3]),min(trcphysedK910to916[,3]),min(trcphysedK1011to1016[,3]),min(trcphysedK1112to1116[,3
]),min(trcphysedK1213to1216[,3]),min(trcphysedK1314to1316[,3]),min(trcphysedK1415to1416[,3]),min(
trcphysedK1516to1615[,3]))
# Traces Range for different k+r
Kplusrtrace = range(Maxtracephysed, Mintracephysed)
# Traces Average at different k
Averagetracephysed=c(mean(trcphysedK23to216[,3]),mean(trcphysedK34to316[,3]),mean(trcphysedK45to4
16[,3]),mean(trcphysedK56to516[,3]),mean(trcphysedK67to616[,3]),mean(trcphysedK78to716[,3]),mean(
trcphysedK89to816[,3]),mean(trcphysedK910to916[,3]),mean(trcphysedK1011to1016[,3]),mean(trcphysed
K1112to1116[,3]),mean(trcphysedK1213to1216[,3]),mean(trcphysedK1314to1316[,3]),mean(trcphysedK141
5to1416[,3]),mean(trcphysedK1516to1615[,3]))
# Traces Coefficient of variation (CV)
CVtracephysed=c(cofvar(trcphysedK23to216[,3]),cofvar(trcphysedK34to316[,3]),cofvar(trcphysedK45to
416[,3]),cofvar(trcphysedK56to516[,3]),cofvar(trcphysedK67to616[,3]),cofvar(trcphysedK78to716[,3]
),cofvar(trcphysedK89to816[,3]),cofvar(trcphysedK910to916[,3]),cofvar(trcphysedK1011to1016[,3]),c
ofvar(trcphysedK1112to1116[,3]),cofvar(trcphysedK1213to1216[,3]),cofvar(trcphysedK1314to1316[,3])
,cofvar(trcphysedK1415to1416[,3]),cofvar(trcphysedK1516to1615[,3]))
```

- Maximum, minimum and average overlap range at different k.

```
### Max overlap at fixed k for 1,2,...,K-k for k+r ###
Maxoffdiagphysed=c(max(trcphysedK23to216[,4]),max(trcphysedK34to316[,4]),max(trcphysedK45to416[,4
]),max(trcphysedK56to516[,4]),max(trcphysedK67to616[,4]),max(trcphysedK78to716[,4]),max(trcphysed
K89to816[,4]),max(trcphysedK910to916[,4]),max(trcphysedK1011to1016[,4]),max(trcphysedK1112to1116[
,4]),max(trcphysedK1213to1216[,4]),max(trcphysedK1314to1316[,4]),max(trcphysedK1415to1416[,4]),ma
x(trcphysedK1516to1615[,4]))
# Min overlap at fixed k for 1,2,...,K-k for k+r
Minoffdiagphysed=c(min(trcphysedK23to216[,4]),min(trcphysedK34to316[,4]),min(trcphysedK45to416[,4
]),min(trcphysedK56to516[,4]),min(trcphysedK67to616[,4]),min(trcphysedK78to716[,4]),min(trcphysed
K89to816[,4]),min(trcphysedK910to916[,4]),min(trcphysedK1011to1016[,4]),min(trcphysedK1112to1116[
,4]),min(trcphysedK1213to1216[,4]),min(trcphysedK1314to1316[,4]),min(trcphysedK1415to1416[,4]),mi
n(trcphysedK1516to1615[,4]))
# Overlap Range for different k+r
Kplusrtraceoffdiag = range(Maxoffdiagphysed, Minoffdiagphysed)
### Average overlap at different k
Avegaroffdiagphysed=c(mean(trcphysedK23to216[,4]),mean(trcphysedK34to316[,4]),mean(trcphysedK45to
416[,4]),mean(trcphysedK56to516[,4]),mean(trcphysedK67to616[,4]),mean(trcphysedK78to716[,4]),mean
(trcphysedK89to816[,4]),mean(trcphysedK910to916[,4]),mean(trcphysedK1011to1016[,4]),mean(trcphyse
dK1112to1116[,4]),mean(trcphysedK1213to1216[,4]),mean(trcphysedK1314to1316[,4]),mean(trcphysedK14
15to1416[,4]),mean(trcphysedK1516to1615[,4]))
```

➢ Create a data frame at different k.

```
###### Create data frame using cbind
ExtremvaluesphysedRange=cbind(c(2:15),Maxtracephysed,Mintracephysed,Averagetracephysed,CVtracephy
sed,Maxoffdiagphysed,Minoffdiagphysed,Avegaroffdiagphysed)
# Name the columns
colnames(ExtremvaluesphysedRange)=cbind("K","Maxtrace","Mintrace","AverageTrace","CVTrace","Maxof
fdiag","Minoffdiag","Averageoffdiag")
# Remove single value at k=15 values
ExtremvaluesphysedRange=subset(ExtremvaluesphysedRange, ExtremvaluesphysedRange[ , 1] <= 14)
### Define colors to be used for graphs
plot_colors1<- c("blue","orange","forestgreen","darkorchid1","brown","black","green","red",
“cyan","khaki","tan","tomato","salmon","sienna","black")
```

➢ Plot the traces and average traces at different k.

```
# set the graph parameter
par(xpd = NA, mar = c(5, 4, 4, 4.6)+0.1)
plot(trcphysedK23to216[,c(1,3)],xlab="K",ylab="Trace",xlim=c(2,16),ylim=ateachKtraceRange,type =
"b",lty=1, pch=0 ,cex=0.5,col="blue")                                        # at k=2
lines(trcphysedK34to316[,c(1,3)],type="b", lty=1, pch=1 ,cex=0.5, col="orange")      # at k=3
lines(trcphysedK45to416[,c(1,3)],type="b", lty=1,pch=2 ,cex=0.5, col="forestgreen") # at k=4
lines(trcphysedK56to516[,c(1,3)],type="b", lty=1,pch=3 ,cex=0.5, col="darkorchid1") # at k=5
lines(trcphysedK67to616[,c(1,3)],type="b", lty=1, pch=4 ,cex=0.5,col="brown")       # at k=6
lines(trcphysedK78to716[,c(1,3)],type="b", lty=1,pch=5 ,cex=0.5, col="blacK")       # at k=7
lines(trcphysedK89to816[,c(1,3)],type="b", lty=1,pch=6 ,cex=0.5, col="green")       # at k=8
lines(trcphysedK910to916[,c(1,3)],type="b", lty=1,pch=7 ,cex=0.5, col="red")        # at k=9
lines(trcphysedK1011to1016[,c(1,3)],type="b", lty=1,pch=8 ,cex=0.5, col="cyan")     # at k=10
lines(trcphysedK1112to1116[,c(1,3)],type="b", lty=1,pch=9 ,cex=0.5, col="khaki")    # at k=11
lines(trcphysedK1213to1216[,c(1,3)],type="b", lty=1,pch=10 ,cex=0.5, col="tan")     # at k=12
lines(trcphysedK1314to1316[,c(1,3)],type="b", lty=1, pch=11 ,cex=0.5,col="tomato")  # at k=13
lines(trcphysedK1415to1416[,c(1,3)],type="b", lty=1,pch=12 ,cex=0.5, col="salmon")  # at k=14
lines(trcphysedK1516to1615[,1],trcphysedK1516to1615[,3],pch=13,type="p",col="sienna")#at k=15
# Average trace
lines(ExtremvaluesphysedRange[,c(1,4)],type="l",pch=0,cex=0.5,lty=1,col="black",lwd=1.2)
# Plot legend
legend(c("23o216","34to316","45to416","56to516","67to616","78to716","89to816","910to916","1011to1
016","1112to1116","1213to1216","1314to1316"),x=16.7,y=max(ateachKtraceRange),lty=1,cex=0.6,ncol=1
, col=plot_colors1,pt.cex = 0.5, pch=0:13,title=expression(bold("Traces at each k")),title.col=
"red",y.intersp=1.7)
legend(x=16.7,y=min(ateachKtraceRange),c("Average"),col=c("black"),cex=0.67,lty=1,text.col="red",
ncol = 1,lwd=1.5)
### plot with CV
plot((ExtremvaluesphysedRange[,c(1,5)]),type="b",  xlab="K",  ylab="Traces  CV",  xlim=c(2,14),
ylim=c(min(ExtremvaluesphysedRange[,5],na.rm=TRUE),max(ExtremvaluesphysedRange[,5],na.rm=TRUE)),p
ch=0,cex=0.5,lty=1, col="green",main="Coefficient of variation (CV) from Q \n matrices trace at
different k",col.main="red", font.main=1,family="serif",cex.lab=1.5, cex.axis=1.5, cex.main=1.3,
cex.sub=1.5,lwd=1.5)
abline(h = c(range(ExtremvaluesphysedRange[,5],na.rm=TRUE)), lty=2, col = "gray")
```

- Composite plot for the overlap, average overlap and overlap for $k+1, k+2$ & $k+3$ at different k.

```
### Graph for overlaps and average overlap


# Set the graph parameters
par(xpd = NA, mar = c(5, 4, 4, 5.5)+0.1)
plot(trcphysedK23to216[,c(1,4)],xlab="K",ylab="Overlap",xlim=c(2,16),ylim=ateachKtraceoffdiagRang
e,type = "b",lty=1,pch=0,cex=0.5,col="blue)                                 # at k=2
lines(trcphysedK34to316[,c(1,4)],type="b", lty=1, pch=1 ,cex=0.5, col="orange")      # at k=3
lines(trcphysedK45to416[,c(1,4)],type="b", lty=1,pch=2 ,cex=0.5, col="forestgreen") # at k=4
lines(trcphysedK56to516[,c(1,4)],type="b", lty=1,pch=3 ,cex=0.5, col="darkorchid1") # at k=5
lines(trcphysedK67to616[,c(1,4)],type="b", lty=1, pch=4 ,cex=0.5,col="brown")        # at k=6
lines(trcphysedK78to716[,c(1,4)],type="b", lty=1,pch=5 ,cex=0.5, col="blacK")        # at k=7
lines(trcphysedK89to816[,c(1,4)],type="b", lty=1,pch=6 ,cex=0.5, col="green")        # at k=8
lines(trcphysedK910to916[,c(1,4)],type="b", lty=1,pch=7 ,cex=0.5, col="red")         # at k=9
lines(trcphysedK1011to1016[,c(1,4)],type="b", lty=1,pch=8 ,cex=0.5, col="cyan")      # at k=10
lines(trcphysedK1112to1116[,c(1,4)],type="b", lty=1,pch=9 ,cex=0.5, col="khaki")     # at k=11
lines(trcphysedK1213to1216[,c(1,4)],type="b", lty=1,pch=10 ,cex=0.5, col="tan")      # at k=12
lines(trcphysedK1314to1316[,c(1,4)],type="b", lty=1, pch=11 ,cex=0.5,col="tomato")   # at k=13
lines(trcphysedK1415to1416[,c(1,4)],type="b", lty=1,pch=12 ,cex=0.5, col="salmon")   # at k=14
lines(trcphysedK1516to1615[,1],trcphysedK1516to1615[,4],pch=13,type="p", col="sienna")#at k=15
# Average overlap
lines(ExtremvaluesphysedRange[,c(1,8)],type="l",pch=0,cex=0.5,lty=1, col="black",lwd=1.2)
# Plot legend
legend(c("23o216","34to316","45to416","56to516","67to616","78to716","89to816","910to916","1011to1
016","1112to1116","1213to1216","1314to1316"),x=16.7,y=max(ateachKtraceoffdiagRange)
,lty=1,cex=0.57,ncol=1, col=plot_colors1,pt.cex = 0.5, pch=0:13,title=expression(bold("Overlap at
each k")),title.col= "red",y.intersp=1.7)
legend(x=16.7,y=min(ateachKtraceoffdiagRange),c("Overlap"),col=c("black"),cex=0.71,lty=1,text.col
="red",lwd=1.2,ncol=1)
```

References

1. Runciman, W.B., Roughead, E.E., Semple, S.J. and Adams, R.J. (2003). *Adverse drug events and medication errors in Australia*. International Journal for Quality in Health Care, vol 15, supplement 1, pp. i49-i59.

2. Wang, J., Zhou, B. and Yan, R. (2012). *Benefits and barriers in mining the healthcare industry data*. International Journal of Strategic Decision Sciences (IJSDS), vol 3, no 4, pp.51-67.

3. Haughton, D., Deichmann, J., Eshghi, A., Sayek, S., Teebagy, N. and Topi, H. (2003). *A review of software packages for data mining.* The American Statistician, vol 57, no 4, pp. 290-309.

4. Tukey, J. (1977). *Exploratory data analysis.* Reading, Mass. Addison-Wesley.

5. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy, R. (1996). *Advances in Knowledge Discovery and Data Mining*. The MIT Press.

6. Verma, G. and Verma, V. (2012). *Role and Applications of Genetic Algorithm in Data Mining.* International Journal of Computer Applications (0975-888), vol 48, no 17, pp. 1-8.

7. Mishra, R. (2007). *Implications of Data Mining in Digital Library Environment.* Inflibnet center.

8. Pal, J. K. (2011). *Usefulness and applications of data mining in extracting information from different perspectives.* Annals of library and information studies, vol 58, no 1, pp. 7-16.

9. Leon, A. (2008). *ERP demystified*. Tata McGraw-Hill Education.

10. Folorunso, O. and Ogunde, A.O. (2004). *Data Mining as a technique for knowledge management in business process redesign*. The Electronic Journal of Knowledge Management, vol 2, no 1, pp. 33-44.

11. Fayyad, U.M., Piatetsky-Shapiro, G. and Smyth, P. (1996). *From data mining to knowledge discovery: an overview*. Al magazine, vol 17, no 3, pp. 37-54.

12. Berger, A. M. and Berger, C. R. (2004). *Data Mining as a Tool for Research and Knowledge Development in Nursing.* Computers Informatics Nursing, vol 22, no 3, pp. 123-131.

13. Myatt, G. J. (2007). *Making sense of data : a practical guide to exploratory data analysis and data mining*. John Wiley & Sons.

14. Myatt, G. J. and Johnson, W .P. (2008). *Making sense of data II : a practical guide to data visualization, advanced data mining methods, and applications*. Hoboken, N.J., John Wiley & Sons.

15. Cunningham, P., Cord, M. and Delany, S. J. (2008). *Supervised learning*. In *Machine Learning Techniques for Multimedia*. Springer Berlin Heidelberg, pp. 21-49.

16. Larose, D.T. (2006). *Data mining methods and models*. John Wiley & Sons.

17. Cheng, C. H., Chen, Y. H. and Liu, J. W. (2009). *Classifying Cinnamomums using rough sets classifier based on interval-discretization.* Plant Systematics and Evolution, vol 280, no 1, pp. 89-97.

18. Tan, P. N., Kumar, V. and Steinbach, M. (2005). *Introduction to data mining* (1st ed.). Boston ,Pearson Addison Wesley.

19. Camdeviren, H. A., Yazici, A. C., Akkus, Z., Bugdayci, R. and Sungur, M. A. (2007). *Comparison of logistic regression model and classification tree: An*

application to postpartum depression data. Expert Systems Application, vol 32, no 4, pp. 987-994.

20. Han, J. and Kamber, M. (2006). *Data mining : concepts and techniques*. Morgan Kaufmann series in data management systems. San Francisco Morgan Kaufmann Publishers.

21. Wang, F. and Rudin, C. (2014). *Falling rule lists.* arXiv preprint arXiv: 1411.5899.

22. Berry, M. W. and Browne, M. (2006). *Lecture notes in data mining*. Hackensack, N.J. , World Scientific.

23. Larose, D. T. (2005). *Discovering knowledge in data : an introduction to data mining*. Hoboken, N.J., Wiley-Interscience.

24. Sumathi, S., Sivanandam, S .N. (2006). *Introduction to Data Mining and its Applications*. Springer Berlin Heidelberg.

25. Hand, D., Mannila, H. and Smyth, P. (2001). *Principles of data mining (adaptive computation and machine learning).* Bradford Book Cambridge.

26. Refaat, M. (2007). *Data preparation for data mining using SAS*. San Francisco Morgan Kaufmann Publishers.

27. Hair, J. F. (1998). *Multivariate data analysis*. Upper Saddle River, N. J., Prentice Hall.

28. Timm, N. H. (2002). *Applied multivariate analysis*. NetLibrary, Springer New York.

29. Harber, P., Lew, M., Tashkin, D. P. and Simmons, M. (1987). *Factor analysis of clinical data from asbestos workers: Implications for diagnosing and screening.* British Journal of Industrial Medicine, vol 44, no 11, pp. 780-784.

30. Jolliffe, I. T. (2002). *Principal component analysis*. Springer series in statistics, New York Springer.

31. Tan, P. N. and Kumar, V. (2005). *Chapter 6 Association Analysis: Basic Concepts and Algorithms.* Introduction to Data Mining, Addison-Wesley.

32. Agrawal, R., Imieliński, T. and Swami. A. (1993). *Mining association rules between sets of items in large databases*. In ACM SIGMOD Record, vol. 22, no 2, pp. 207-216.

33. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H. and Verkamo, A. I. (1996). *Fast Discovery of Association Rules.* Advances in knowledge discovery and data mining, vol 12, no 1, pp. 307-328.

34. Concaro, S., Sacchi, L., Cerra, C., Fratino, P. and Bellazzi, R. (2009). *Mining healthcare data with temporal association rules: Improvements and assessment for a practical use*. In Artificial Intelligence in Medicine, Springer Berlin Heidelberg, pp. 16-25.

35. Reps, J. M., Aickelin, U., Ma, J. and Zhang, Y. (2014). *Refining Adverse Drug Reactions using Association Rule Mining for Electronic Healthcare Data*. Data Mining Workshop (ICDMW) IEEE International Conference, pp. 763-770.

36. Dasseni, E., Verykios, V. S., Elmagarmid, A. K., Bertino E. (2001). *Hiding association rules by using confidence and support*. Information Hiding, Springer Berlin Heidelberg, (pp. 369-383).

37. Srikant, R., Vu, Q. and Agrawal. R., (1997). *Mining Association Rules with Item Constraints*. In *KDD*, vol 97, pp. 67-73.

38. Srikant, R. and Agrawal, R. (1996). *Mining quantitative association rules in large relational tables.* In SIGMOD Record, vol 25, no 2, pp. 1-12.

39. Zheng, Z., Kohavi, R. and Mason, L. (2001). *Real world performance of association rule algorithms*. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, ACM San Francisco California, pp. 401-406.

40. Hill, T., Lewicki, P. and Lewicki, P. (2006). *Statistics: methods and applications: a comprehensive reference for science, industry, and data mining*. StatSoft, Inc..

41. Koch, I. (2013). *Analysis of Multivariate and High-Dimensional Data*. Cambridge University Press, vol 32.

42. Mwasiagi, J., Wang, X. H. and Huang, X. B. (2009). *The use of k-means and artificial neural network to classify cotton lint.* Fibers and Polymers, vol 10, no 3, pp. 379-383.

43. MacQueen, J. (1967). *Some methods for classification and analysis of multivariate observations*. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Oakland CA, USA, vol 1, no 14, pp. 281-297.

44. Kaufman, L., Rousseeuw, P.J. (1990). *Finding groups in data : an introduction to cluster analysis*. New York , Wiley.

45. Park, H. S., Lee, J. S. and Jun, C. H.(2006). *A K-means-like Algorithm for K-medoids Clustering and Its Performance.* Proceedings of ICCIE, pp. 102-117.

46. Han, J., Kamber, M., Tung, A., Miller, H. and Han, J. (2001). *Spatial clustering methods in data mining: A survey.* Geographic Data Mining and Knowledge Discovery, Taylor and Francis.

47. Huang, Z. (1998). *Extensions to the k-means algorithm for clustering large data sets with categorical values.* Data mining and knowledge discovery, vol 2, no 3, pp. 283-304.

48. Ng, R. T. and Han, J. (1994). *Efficient and Eective Clustering Methods for Spatial Data Mining*. Proceedings of 20th International Conference on Very Large Data Bases (VLDB), pp. 144-155.

49. Ng, R.T. and Jiawei, H. (2002). *CLARANS: a method for clustering objects for spatial data mining.* Knowledge and Data Engineering, IEEE Transactions, vol 14, no 5, pp. 1003-1016.

50. Ray, A. K. and Acharya, T. (2004). *Information Technology: Principles and Applications*. PHI Learning Pvt. Ltd..

51. Carreira-Perpinán, M. A. and Wang, W. (2013). *The K-modes algorithm for clustering.* arXiv preprint arXiv:1304.6478.

52. Huang, Z. (1997) *Clustering large data sets with mixed numeric and categorical values*. In Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining,(PAKDD). Singapore, pp. 21-34.

53. Murtagh, F. (1983). *A survey of recent advances in hierarchical clustering algorithms.* The Computer Journal, vol26, no 4, pp. 354-359.

54. Han, J. and Kamber, M. (2001). *Data mining: concepts and techniques*. Morgan Kaufmann San Francisco, Calif, USA.

55. Crowley, J. and Ankerst, D. (2006). *Handbook of statistics in clinical oncology* (2nd ed.). Boca Raton, Chapman & Hall/CRC.

56. Zhang, T., Ramakrishnan, R., and Livny, M. (1996). *BIRCH: An Efficient Method for Very Varge Databases*. In ACM SIGMOD, vol 25, no 2, pp. 103-114.

57. Guha, S., Rastogi, R. and Shim, K. (1998). *CURE: an efficient clustering algorithm for large databases*. In ACM SIGMOD Record, vol. 27, no 2, pp. 73-84.

58. Guha, S., Rastogi, R., and Shim, K . (1999). *ROCK: A robust clustering algorithm for categorical attributes*. In Data Engineering Proceedings 15th International Conference*i*, IEEE, pp. 512-521.

59. Bandyopadhyay, S. and Coyle, E. J. (2003). *An energy efficient hierarchical clustering algorithm for wireless sensor networks*. In INFOCOM. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, vol 3, pp. 1713-1723

60. Zhao, Y., Karypis, G. and Fayyad, U. (2005). *Hierarchical clustering algorithms for document datasets.* Data mining and knowledge discovery, vol 10, no 2, pp. 141-168.

61. Jung, Y., Park, H., Du, D.Z. and Drake, B.L. (2003). *A decision criterion for the optimal number of clusters in hierarchical clustering.* Journal of Global Optimization, vol 25, no 1, pp. 91-111.

62. Matignon, R. (2007). *Data mining using SAS enterprise miner*. John Wiley & Sons, Wiley Series in Computational Statistics, vol 638.

63. Sander, J., Ester, M., Kriegel, H.P. and Xu, X. (1998). *Density-based clustering in spatial databases: The algorithm gdbscan and its applications.* Data mining and knowledge discovery, vol 2, no 2, pp. 169-194.

64. Han, J., Kamber, M. and Pei, J. (2011). *Data mining : concepts and techniques* (3rd ed.). Morgan Kaufmann , Burlington Elsevier Science.

65. Wang, W., Yang, J. and Muntz, R. (1997). *STING: A statistical information grid approach to spatial data mining*. Proceedings of 23rd International Conference on Very Large Data Bases (VLDB), vol 97, pp. 186-195.

66. Agrawal, R., Gehrke, J., Gunopulos, D. and Raghavan, P. (1998). *Automatic subspace clustering of high dimensional data for data mining applications*. SIGMOD Rec., vol 27, no 2, pp. 94-105.

67. Sheikholeslami, G., Chatterjee, S. and Zhang, A. (1998). *Wavecluster: A multi-resolution clustering approach for very large spatial databases*. Proceedings of 24th International Conference on Very Large Data Bases (VLDB), vol 98, pp. 428-439.

68. Jiawei, H. and Kamber, M. (2001). *Data mining: concepts and techniques.* San Francisco, CA, itd: Morgan Kaufmann.

69. Fraley, C. and Raftery, A. E. (2005). *Bayesian regularization for normal mixture estimation and model-based clustering*. Washington Univ Seattle Dept of Statistics.

70. Fraley, C. and Raftery, A. E. (2007). *Bayesian regularization for normal mixture estimation and model-based clustering.* Journal of Classification, vol 24, no 2, pp. 155-181.

71. King, R. S. (2015). *Cluster analysis and data mining : an introduction*. Dulles, Virginia Boston, Massachusetts New Delhi, Mercury Learning and Information.

72. Oh, M. S. and Raftery, A. E. (2007). *Model-based clustering with dissimilarities: A Bayesian approach.* Journal of Computational and Graphical Statistics, vol 16, no 3.

73. Martinez, W. L., Martinez, A. and Solka, J. (2011). *Exploratory data analysis with MATLAB.* CRC Press.

74. Schwarz, G. (1978*). Estimating the dimension of a model. The annals of statistics, vol* 6, no 2, pp. 461-464.

75. Gan, G., Ma, C., and Wu, J. (2007). *Data clustering : theory, algorithms, and applications*. Philadelphia, Pa., Alexandria, Va., SIAM, Society for Industrial and Applied Mathematics. American Statistical Association.

76. Zhang, S., Zhang, C. and Yang, Q. (2003). *Data preparation for data mining.* Applied Artificial Intelligence, vol 17, no 5-6, pp. 375-381.

77. Pal, N. R. and Jain, L. C. (2005). *Advanced techniques in knowledge discovery and data mining*. Advanced information and knowledge processing, New York, Springer-Verlag.

78. Card, S. K., Mackinlay, J. D. and Shneiderman, B. (1999). *Readings in information visualization: using vision to think*. Morgan Kaufmann.

79. Nocke, T., Schumann, H. and Böhm, U. (2004). *Methods for the visualization of clustered climate data.* Computational Statistics, vol 19, no 1, pp. 75-94.

80. Keim, D. A. (2001). *Visual exploration of large data sets.* Communications of the ACM, vol 44, no 8, pp. 38-44.

81. Keim, D.A. (2002). *Information visualization and visual data mining.* Visualization and Computer Graphics, IEEE Transactions, vol 8, no 1, pp. 1-8.

82. Liu, P., El-Darzi, E., Lei, L., Vasilakis, C., Chountas, P. and Huang, W. (2005). *An Analysis of Missing Data Treatment Methods and Their Application to Health Care Dataset*. In Advanced Data Mining and Applications, Springer Berlin Heidelberg, pp. 583-590.

83. Fowlkes, E.B., Gnanadesikan, R. and Kettenring, J.R. (1988). *Variable selection in clustering.* Journal of Classification, vol 5, no 2, pp. 205-228.

84. Guyon, I. and Elisseeff, A. (2003). *An introduction to variable and feature selection.* The Journal of Machine Learning Research, vol 3, pp. 1157-1182.

85. O'Hara, R. B. and Sillanpaa, M. J. (2009). *A review of Bayesian variable selection methods: what, how and which.* Bayesian analysis, vol 4, no. 1, pp. 85-117.

86. Hall, P. and Miller, H. (2009). *Using generalized correlation to effect variable selection in very high dimensional problems.* Journal of Computational and Graphical Statistics, vol 18, no 3.

87. Schaffer, C. M. and Green, P. E. (1996). *An empirical comparison of variable standardization methods in cluster analysis.* Multivariate Behavioral Research, vol 31, no 2, pp. 149-167.

88. Jain, A. K., Murty, M. N. and Flynn, P. J. (1999). *Data clustering: a review.* ACM computing surveys (CSUR), vol 31 ,no 3, pp. 264-323.

89. Doreswamy, D. (2012). *A novel design specification distance (DSD) based K-mean clustering performance evaluation on engineering materials' database.* International Journal of Computer Applications, vol 55, no 15, pp. 26-33.

90. Vimal, A., Valluri, S. R. and Karlapalem, K. (2008). *An Experiment with Distance Measures for Clustering*. In COMAD, pp. 241-244.

91. Aggarwal, C. C., Hinneburg, A. and Keim, D. A. (2001). *On the surprising behavior of distance metrics in high dimensional space.*, Springer, pp. 420-434.

92. Xing, E. P., Jordan, M. I., Russell, S. and Ng, A. Y. (2002). *Distance metric learning with application to clustering with side-information*. In Advances in neural information processing systems, pp. 505-512.

93. Pandit, S. and Gupta, S. (2011). *A comparative study on distance measuring approaches for clustering.* International Journal of Research in Computer Science, vol 2, no 1, pp. 29-31.

94. Yang, L. and Jin, R. (2006). *Distance metric learning: A comprehensive survey.* Michigan State Universiy, 2.

95. Lebeda, A. and Jendrulek, T. (1987). *Cluster analysis as a method for evaluation of genetic similarity in specific host — parasite interaction (Lactuca sativa — Bremia lactucae).* Theoretical and Applied Genetics, vol 75, no1, pp. 194-199.

96. Xiang, S., Nie, F. and Zhang, C. (2008). *Learning a Mahalanobis distance metric for data clustering and classification.* Pattern Recognition, vol 41, no 12, pp. 3600-3612.

97. Holliday, J. D., Hu, C. and Willett, P. (2002). *Grouping of coefficients for the calculation of inter-molecular similarity and dissimilarity using 2D fragment bit-strings.* Combinatorial chemistry & high throughput screening, vol 5, no 2, pp. 155-166.

98. Kaufman, L. and Rousseeuw, P.J. (2009). *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, Vol. 344.

99. Gan, G. (2011). *Data clustering in C++ : an object-oriented approach*. Data clustering in C plus plus. Boca Raton, Fla, CRC Press.

100. Salkind, N. J. (2006). *Encyclopedia of Measurement and Statistics*. Thousand Oaks, SAGE Publications.

101. Mao, J. and Jain, A.K. (1996). *A self-organizing network for hyperellipsoidal clustering (HEC).* Neural Networks, IEEE Transactions on, vol 7, no 1, pp. 16-29.

102. Picard, N. and Bar-Hen, A. (2012). *A Criterion Based on the Mahalanobis Distance for Cluster Analysis with Subsampling.* Journal of Classification, vol 29, no 1, pp. 23-49.

103. Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Englewood Cliffs, N.J., Prentice Hall .

104. De Maesschalck, R., Jouan-Rimbaud, D. and Massart, D.L. (2000). *The Mahalanobis distance.* Chemometrics and Intelligent Laboratory Systems, vol 50, no 1, pp. 1-18.

105. Legendre, L. and Legendre, P. (1983). *Numerical ecology*. Amsterdam, Elsevier.

106. Hovenga, E. J. S., Kidd, M. R. and Cesnik, B. (1996). *Health informatics : an overview*. Melbourne, Churchill Livingstone.

107. McAullay, D., Williams, G., Chen, J., Jin, H., He, H., Sparks, R. and Kelman, C. (2005). *A delivery framework for health data mining and analytics*. In Proceedings of the Twenty-eighth Australasian conference on Computer Science. Australian Computer Society, vol 38, pp. 381-387.

108. Bereznicki, B. J., Peterson, G. M., Jackson, S. L., Walters, E. H., Fitzmaurice, K. D. and Gee, P. R. (2008). *Data-mining of medication records to improve asthma management.* Medical Journal of Australia, vol 189, no 1, pp. 21-25.

109. Goodwin, L. K., Iannacchione, M. A., Hammond, W. E., Crockett, P., Maher, S. and Schlitz, K. (2001). *Data Mining Methods Find Demographic Predictors of Preterm Birth.* Nursing Research, vol 50, no 6, pp. 340-345.

110. Goodwin, L.K. and Iannacchione, M. A. (2002). *Data Mining Methods for Improving Birth Outcomes Prediction.* Outcomes Management, vol 6, no 2, pp. 80-85.

111. Muller, R. and Möckel, M. (2008). *Logistic regression and CART in the analysis of multimarker studies.* Clinica Chimica Acta, vol 394, no 1-2, pp. 1-6.

112. Correa-Velez, I., Sundararajan, V., Brown, K. and Gifford, S.M. (2007). *Hospital utilisation among people born in refugee-source countries: An analysis of hospital admissions, Victoria, 1998-2004.* Medical Journal of Australia, vol 186, no 11, pp. 577-580.

113. Krasnik, A., Norredam, M., Sorensen, T. M., Michaelsen, J. J., Nielsen, A. S. and Keiding, N. (2002). *Effect of ethnic background on Danish hospital utilisation patterns.* Social Science & Medicine, vol 55, no 7, pp. 1207-1211.

114. Rué, M., Cabré, X., Soler-González, J., Bosch, A., Almirall, M. and Serna, M. C. (2008). *Emergency hospital services utilization in Lleida (Spain): A cross-sectional study of immigrant and Spanish-born populations.* BMC Health Services Research, vol 8, no 1.

115. Dias, S. F., Severo, M. and Barros, H. (2008). *Determinants of health care utilization by immigrants in Portugal.* BMC Health Services Research, vol 8, no 1.

116. Brunero, S., Fairbrother, G., Lee, S. and Davis, M. (2007). *Clinical characteristics of people with mental health problems who frequently attend an Australian emergency department.* Australian Health Review, vol 31, no 3, pp. 462-470.

117. Pascual, J. C., Malagón, A., Arcega, J. M., Gines, J. M., Navinés, R., Gurrea, A., Garcia-Ribera, C. and Bulbena, A. (2007). *Utilization of psychiatric emergency services by homeless persons in Spain.* General Hospital Psychiatry, vol 30, no 1, pp. 14-19.

118. Wu, S. and Chow, T. W. (2004). *Clustering of the self-organizing map using a clustering validity index based on inter-cluster and intra-cluster density*. Pattern Recognition, vol 37, no 2, pp. 175-188.

119. Sun, Y., Zhu, Q. and Chen, Z. (2002). *An iterative initial-points refinement algorithm for categorical data clustering.* Pattern Recognition Letters, vol 23, no 7, pp. 875-884.

120. Arthur, D. and Vassilvitskii, S. (2007). *k-means++: The advantages of careful seeding*. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, pp. 1027-1035.

121. Bahmani, B., Moseley, B., Vattani, A., Kumar, R. and Vassilvitskii,S.(2012). *Scalable k-means++*. Proceedings of the VLDB Endowment, vol 5, no 7, pp. 622-633.

122. Tan, P., Kumar, V. and Steinbach, M. (2005). *Introduction to data mining*. Boston, Pearson Addison Wesley.

123. Handl, J., Knowles, J. and Kell, D. B. (2005). *Computational cluster validation in post-genomic data analysis.* Bioinformatics, vol 21, no 15, pp. 3201-3212.

124. Halkidi, M., Batistakis, Y. and Vazirgiannis, M. (2001). *On clustering validation techniques.* Journal of Intelligent Information Systems, vol 17, no 2, pp. 107-145.

125. Halkidi, M., Batistakis, Y. and Vazirgiannis, M. (2002). *Cluster validity methods: Part I.* SIGMOD Record, vol 31, no 2, pp. 40-45.

126. Halkidi, M., Batistakis, Y. and Vazirgiannis, M. (2002). *Clustering validity checking methods: Part II.* SIGMOD Record, vol 31, no 3, pp. 19-27.

127. Rendón, E., Abundez, I. M., Gutierrez, C., Zagal, S.D., Arizmendi, A., Quiroz, E.M. and Arzate, H.E. (2011). *A comparison of internal and external cluster validation indexes*. Proceedings of the 2011 American Conference on Applied Mathematics and the 5th World Scientific and Engineering Academy and Society (WSEAS) International conference, San Francisco, CA, USA, vol 29, pp. 158-163.

128. Duds, R. O. and Hart, P. E. (1973). *Pattern classification and scene analysis.* John Wiley and Sons, Inc..

129. Milligan, G. W. and Cooper, M. C. (1985). *An examination of procedures for determining the number of clusters in a data set.* Psychometrika, vol 50, no 2, pp. 159-179.

130. Sarle, W. S. (1983). *Cubic clustering criterion*. SAS Institute.

131. Ray, S. and Turi, R. H. (1999). *Determination of number of clusters in k-means clustering and application in colour image segmentation*. In Proceedings of the 4th international conference on advances in pattern recognition and digital techniques, pp. 137-143.

132. Chou, C. H., Su, M. C. and Lai, E. (2003). *A New Cluster Validity Measure for Clusters with Different Densities*. IASTED International Conference on Intelligent Systems & Control, pp. 276-281.

133. Dimitriadou, E., Dolničar, S. and Weingessel, A. (2002). *An examination of indexes for determining the number of clusters in binary data sets.* Psychometrika, vol 67, no 1, pp. 137-159.

134. Liu, Y., Li, Z., Xiong, H., Gao, X. and Wu, J. (2010). *Understanding of internal clustering validation measures*. In Data Mining (ICDM), IEEE 10th International Conference, pp. 911-916.

135. Kim, M. and Ramakrishna, R. S. (2005). *New indices for cluster validity assessment.* Pattern Recognition Letters, vol 26, no 15, pp. 2353-2363.

136. Caliński, T. and Harabasz, J. (1974). *A dendrite method for cluster analysis.* Communications in Statistics-theory and Methods, vol 3 no 1, pp. 1-27.

137. Dunn, J.C. (1974). *Well-separated clusters and optimal fuzzy partitions.* Journal of cybernetics, vol 4, no 1, pp. 95-104.

138. Subhash, S. (1996). *Applied multivariate techniques.* John Wily & Sons Inc., Canada.

139. Saitta, S., Raphael, B. and Smith, I. F. (2007). *A bounded index for cluster validity,* in *Machine learning and data mining in pattern recognition*. Springer Berlin Heidelberg, pp. 174-187.

140. Günter, S. and Bunke, H. (2003).*Validation indices for graph clustering.* Pattern Recognition Letters, vol 24 no 8, pp. 1107-1113.

141. Kaufman, L. and Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis.* New York, Wiley.

142. Rousseeuw, P.J. (1987). *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis.* Journal of computational and applied mathematics, vol 20, pp. 53-65.

143. Davies, D. L. and Bouldin, D. W. (1979). *A cluster separation measure.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol 2, pp. 224-227.

144. Halkidi, M., Vazirgiannis, M. and Batistakis, Y. (2000). *Quality scheme assessment in the clustering process*, In Principles of Data Mining and Knowledge Discovery, Springer Berlin Heidelberg, pp. 265-276.

145. Tibshirani, R., Walther, G. and Hastie, T. (2001). *Estimating the number of clusters in a data set via the gap statistic.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol 63, no 2, pp. 411-423.

146. Jain, A. K. (2010). *Data clustering: 50 years beyond K-means.* Pattern recognition letters, vol 31, no 8, pp. 651-666.

147. Babu, G. P. and Murty, M. N. (1993). *A near-optimal initial seed value selection in k-means means algorithm using a genetic algorithm.* Pattern recognition letters, vol 14, no 10, pp. 763-769.

148. Khan, S. S. and Ahmad, A. (2004). *Cluster center initialization algorithm for K-means clustering.* Pattern recognition letters, vol 25, no 11, pp. 1293-1302.

149. Mehar, A. M., Matawie, K. and Maeder, A. (2013). *Determining an optimal value of K in K-means clustering*. IEEE International Conference on Bioinformatics and Biomedicine, Shanghai, China, pp. 51-55.

150. Matawie, K., Mehar Muhammad, A. and Maeder, A. (2015). *An approach to determine clusters overlap for k-means clustering.* International Workshop on Statistical Modelling, Linz, Austria, vol 2, pp. 163-166.

151. Mehar, A., Maeder, A., Matawie, K. and Ginige, A. (2010). *Blended Clustering for Health Data Mining. E- Health,* Springer Berlin Heidelberg, pp. 130-137.

152. Johnson, M., Paulusma, D. and van Leeuwen, E. J. (2013). *Algorithms to measure diversity and clustering in social networks through dot product graphs*. Algorithms and Computation, Springer Berlin Heidelberg, pp. 130-140.

153. Egecioglu, O., Ferhatosmanoglu, H. and Ogras, U. (2004). *Dimensionality reduction and similarity computation by inner-product approximations.* Knowledge and Data Engineering, IEEE Transactions, vol 16 no 6 pp. 714-726.

154. Kaski, S. (1998). *Dimensionality reduction by random mapping: fast similarity computation for clustering*. Neural Networks Proceedings. IEEE World Congress on Computational Intelligence, vol 1, pp. 413-418.

155. Campbell, W. M., Sturim, D. E. and Reynolds, D. A. (2006). *Support vector machines using GMM supervectors for speaker verification.* Signal Processing Letters, IEEE, vol 13, no 5, pp. 308-311.

156. Kuwata, T. and Sato-Ilic, M. (2010). *Learning Based Self-organized Additive Fuzzy Clustering Method*. Advances in Intelligent Decision Technologies, Springer Berlin Heidelberg, pp. 589-596.

157. Elsayed, T., Lin, J. and Oard, D. W. (2008). *Pairwise document similarity in large collections with MapReduce*. Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers, Association for Computational Linguistics , pp. 265-268 .

158. Ram, P. and Gray, A. G. (2012). *Maximum inner-product search using cone trees*. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 931-939

159. Cheng, D., Kannan, R., Vempala, S. and Wang, G. (2006). *A divide-and-merge methodology for clustering.* ACM Transactions on Database Systems (TODS), vol 31, no 4, pp. 1499-1525.

160. Auvolat, A. and Vincent, P. (2015). *Clustering is Efficient for Approximate Maximum Inner Product Search.* arXiv preprint arXiv, 1507.05910.

161. Friedman, H. P. and Rubin, J. (1967). *On some invariant criteria for grouping data.* Journal of the American Statistical Association, vol 62, no 320, pp. 1159-1178.

162. Johnson, C.R. (1981). *Row stochastic matrices similar to doubly stochastic matrices.* Linear and Multilinear Algebra, vol 10, no 2, pp. 113-130.

163. Ross, S. M. (1996). *Stochastic processes*. John Wiley & Sons, New York.

164. Ruspini, E . H. (1970). *Numerical methods for fuzzy clustering.* Information Sciences, vol 2, no 3, pp. 319-350.

165. Su, M. C. and Chou, C . H. (2000). *A k-means algorithm with a novel non-metric distance*. Proceedings of Joint Conference of Information Science, Association for Intelligent Machinery, Atlantic City, US, pp. 417-420

166. Gabriella, V. R (2011). *Stability Selection of the Number of Clusters*. ScholarWorks@ Georgia State University.

167. Pakhira, M. K., Bandyopadhyay, S. and Maulik, U. (2004). *Validity index for crisp and fuzzy clusters.* Pattern recognition, vol 37, no 3, pp. 487-501.

168. Kothari, R. and Pitts, D. (1999). *On finding the number of clusters.* Pattern recognition letters, vol 20, no 4, pp. 405-416.

169. Makles, A. (2012). *Stata tip 110: How to get the optimal k-means cluster solution.* Stata Journal, vol 12, no 2, pp. 347-351.

170. William, H. W. (1992). *Breast Cancer Wisconsin (origina) dataset.* Retrived 10 May 2015 from UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29.

171. Safavi, A. A., Parandeh, N. M. and Salehi, M. (2010). *Predicting breast cancer survivability using data mining techniques*. Software Technology and Engineering (ICSTE) 2nd International Conference, IEEE, vol 2, pp.V2-227.

172. Camastra, F. and Verri, A. (2005). *A novel kernel method for clustering.* Pattern Analysis and Machine Intelligence, IEEE Transactions, vol 27,no 5, pp. 801-805.

173. Salojärvi, J., Kaski, S. and Sinkkonen, J. (2003). *Discriminative clustering in Fisher metrics*. Artificial neural networks and neural information processing, pp. 161-164.

174. BioLINCC, (1956-1968). *Framingham Heart Study (FHS) Longitudinal Data.* Retrived: 12 August 2015, from Biologic Specimen and Data Repository Information Coordinating Center, https://biolincc.nhlbi.nih.gov/requests/teaching-dataset-request /3246 / comments/.

175. Survey, M. E. P. (2013). *MPES HC 137 2011 fullyear consolidated data file.* Retrived: 20 March 2015, from Meps, https://meps.ahrq.gov/data_stats/download_data_files_%20detail.jsp%20?cboPufNumber=HC-147.

176. Fowler, J. H. and Christakis, N. A. (2008). *Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the Framingham Heart Study*. British Madical Journal (BMJ), vol 337.

177. Pencina, M. J., D'Agostino, R. B., Larson, M. G., Massaro, J. M. and Vasan, R .S. (2009). Predicting the 30-year risk of cardiovascular disease The Framingham Heart Study, Circulation, vol 119, no 24, pp. 3078-3084.

178. Higgins, M., Province, M., Heiss, G., Eckfeldt, J., Ellison, R. C., Folsom, A. R., Rao, D. C., Sprafka, J. M. and Williams, R. (1996). *NHLBI Family Heart Study: objectives and design.* American journal of epidemiology, vol 143, no 12, pp. 1219-1228.

179. Hautamäki, V., Cherednichenko, S., Kärkkäinen, I., Kinnunen, T. and Fränti, P. (2005). *Improving k-means by outlier removal.* Image Analysis, Springer Berlin Heidelberg, pp. 978-987.

180. Bishu, K. G., Gebregziabher, M., Dismuke, C. E. and Egede, L. E. (2015). *Quantifying the Incremental and Aggregate Cost of Missed Workdays in Adults with Diabetes.* Journal of general internal medicine, pp. 1-7.

181. Zuvekas, S. H. and Olin, G. L. (2009). *Validating household reports of health care use in the medical expenditure panel survey.* Health Services Research, vol 44, no 5, pp. 1679-1700.

182. Sommers, J. P. (2005). *Producing State Estimates with the Medical Expenditure Panel Survey, Household Component*. US Department of Health & Human Services, Agency for Healthcare Research and Quality.