UNIVERSITY COLLEGE LONDON

DEPARTMENT OF STATISTICAL SCIENCE

DOCTORAL THESIS

Distance construction and clustering of football player performance data

Author:
Serhat Emre AKHANLI

Supervisor:
Dr. Christian HENNIG

January 21, 2019

DECLARATION

I, Serhat Emre AKHANLI confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

ABSTRACT

I present a new idea to map football players information by using multidimensional scaling, and to cluster football players. The actual goal is to define a proper distance measure between players. The data was assembled from whoscored.com. Variables are of mixed type, containing nominal, ordinal, count and continuous information. In the data pre-processing stage, four different steps are followed through for continuous and count variables: 1) representation (i.e., considerations regarding how the relevant information is most appropriately represented, e.g., relative to minutes played), 2) transformation (football knowledge as well as the skewness of the distribution of some count variables indicates that transformation should be used to decrease the effective distance between higher values compared to the distances between lower values), 3) standardisation (in order to make within-variable variations comparable), and 4) variable weighting including variable selection. In a final phase, all the different types of distance measures are combined by using the principle of the Gower dissimilarity (Gower, 1971).

As the second part of this thesis, the aim was to choose a suitable clustering technique and to estimate the best number of clusters for the dissimilarity measurement obtained from football players data set. For this aim, different clustering quality indexes have been introduced, and as first proposed by Hennig (2017), a new concept to calibrate the clustering quality indexes has been presented. In this respect, Hennig (2017) proposed two random clustering algorithms, which generates random clustering points from which standardised clustering quality index values can be calculated and aggregated in an appropriate way. In this thesis, two new additional random clustering algorithms have been proposed and the aggregation of clustering quality indexes has been examined with different types of simulated and real data sets. At the end, this new concept has been applied on the dissimilarity measurement of football players.

IMPACT STATEMENT

In recent years, the industry of professional football (soccer) has grown and developed, due to an expanding interest in statistical analysis of this sport especially with the existence of a vast amount of footballing data. More features and more ideas have been contributed by professional management using the most advanced tools. Discovering such footballing talents is another big challenge for the industry, and team' scouts and footballing agents generally track such players by using expert's knowledge, watching football matches, or looking at the descriptive statistics of football players. In this research, exploring such talented players are further improved by advanced statistical techniques with wise interpretation of the big amount of football players data. The idea here is to combine all different football players features and present in one single setting, so that users can have the opportunity of finding their players of interest with one click rather than setting the features manually, but at the same time they can have the flexibility of playing with the weights of features if they are interested in some features more than the others. This design is only depending on statistical data sets, and my claim here is that using directly this design for the recruitment of new footballing talents is not a sufficient way, but I simply provide a short-cut to football scouts or managers for finding their players of interest from a statistical and scientific point of view.

The benefits inside academia for this thesis is quite different than the footballing idea above. The idea of aggregation of clustering validation indexes is an intelligent contribution to the statistical literature, especially for cluster analysis. With this new concept, researchers may have not only the flexibility of the usage of various clustering validation indexes together in one setting, but also have the chance of finding an optimum clustering algorithm and determining the best number of cluster.

ACKNOWLEDGEMENTS

First, I would like to sincerely thank to my supervisor Dr. Christian Hennig for his guidance and his feedback during my PhD journey at UCL. Thank you for reading and correcting my manuscripts and for all the interesting conversations. All his comments and advices were very valuable not only for my thesis, but also for my further research. I also give my sincere thanks to my secondary supervisor Dr. Ioannis Kosmidis, who gave me the opportunity to view my research from different angles.

I am also grateful to Turkish Ministry of Education for financially supporting my PhD studies.

My special thanks to my friend, Peter Kenny, who has contributed in this thesis with their sincere advice and discussion. He also gave me lots of supports during my difficult times of my PhD. I am also grateful to my friend, Isil Akpinar for her support in overcoming numerous obstacles, especially for the thesis submission. I would like to thank my friend Anil Duman for his valuable help and time regarding the design of the football survey and all the discussion that we had, and also thanks to him for being a good friend to me. I am also very thankful to my good friend Dr.Daglar Tanrikulu for his hospitality during my visit in Iceland and for all his advice in my stressful time. I also would like to thanks to my friend Yesim Durmaz for being always such a good friend and especially her help and support during the last year of my PhD. In the end, I would like to thank my girlfriend Oya Kalaycioglu for her love and support throughout my PhD career.

I am also very thankful to Istanbul Basaksehir Football Club to give the opportunity to work as a football analyst for them in the last year of my PhD. The football club also provided me a network with other football analytic companies that later gives me the chance to work in one of companies, Sentio Sports Analytics as a data analyst.

Finally, I am very grateful to members of my family, particularly my dear mother Cumhure Akhanli for their big support, concern and care. Also, big thanks to my brother, Cem Akhanli, he is always my best friend and listens to me all the time and gives me always advice since our childhood. My family were the source of my inspiration, strength and encouragement. It was impossible to do the PhD without their support and love.

PUBLICATIONS

[1] Akhanli, Serhat Emre and Hennig, C.M. Some Issues in Distance Construction for Football Players Performance Data. In Archives of Data Science, Volume 2.1. ISSN:2363-9881. https://publikationen.bibliothek.kit.edu/1000066924, 2017.
The GfKl (German Classification Society) Best Paper Award

[2] Hennig, C., and Akhanli, S. Football and the Dark Side of Cluster. In Mucha, H. J., editors, *Big data clustering: Data pre-processing, variable selection, and dimension reduction*, WIAS-Report, Berlin, GER. ISBN 0946-8838. https://www.wias-berlin.de/publications/wias-publ/run.jsp?template=abstract&type=Report&year=2017&number=29, 2015.

Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Aims and Objectives

Professional football (soccer) clubs invest a lot of resources in the recruitment of new footballing talent. I propose that traditional methods of scouting by direct observation of players can be enhanced further by intelligent interpretation of the vast amount of footballing data now available on the performance of players. My research is to design a methodology to map football players' performance data in order to explore their similarity structure. This type of information can be very useful for football scouts and managers when assessing players, and also journalists and football fans will be interested. For instance, football scouts and managers try to find talented players that have certain characteristics to complement their team's strategy, and my design could support them in the recruitment of such players. On the other hand, some managers want to retain their teams as stable as possible. When a player leaves from a team, the manager will probably be willing to find another player with similar characteristics to the departed player in order to retain their system. In general, the strategy can be used to analyse team structure. Therefore, the similarity design for players can be very informative and practicable for football squads and managers.

The project will be guided from a statistical and scientific point of view through Statistical Learning algorithms, in particular cluster analysis, which can be presented with a simple interpretation for use in industry. In this sense, I intend to design of a dissimilarity measure, used for MDS (*Multidimensional scaling*) and dissimilarity – based clustering. I pursue two stages in this regard: 1) define a proper distance measure between players based on their performance and characteristic information, 2) cluster and visualise the data by using this distance. Because the first goal is to define a proper distance (dissimilarity) measure between objects, I work on the variables in terms of how they reflect players' characteristics.

In this thesis, the first stage, designing a proper distance measure is presented. In this regard, data is pre-processed in such a way that player information is well characterized. Four steps are followed through in this respect:

1) **Representation:** This is about how to represent the relevant information in the variables, by defining new variables, summarising or framing information in better ways.

2) **Transformation:** This should be applied in order to match the distances that are interpretatively meaningful with the effective differences on the transformed variables. This is an issue involving subject-matter knowledge that cannot be decided by the data alone.

3) **Standardisation:** Variables should be standardised in such a way that a difference in one variable can be traded off against the same difference in another variable when aggregating variables for computing distances; in other words, making the variables comparable in size.

4) **Variable weighting:** Some variables may be more important and relevant than others. Weighting is about appropriately matching the importance of variables.

The data are quite complex with many types of variables that need individual treatment. In this sense, several dissimilarity measures are discussed, and new types of dissimilarities have been designed in terms of how well they match the interpretation of dissimilarity and similarity in the application of interest. In the final phase, all different types of dissimilarity measures are aggregated, and presented in one single dissimilarity matrix for overviewing.

In the second stage, my focus is how to choose an appropriate clustering method and how to determine the best number of clusters. To decide about appropriate cluster analysis methodology and the number of clusters, researchers should consider what data analytic characteristics the clusters they are aiming at are supposed to have. For this aim, different clustering validation index values (e.g., low within-cluster distances or high between-cluster separation) can be evaluated, which is crucially dependent on the aim of clustering. Hennig (2017) introduced several validation criteria that refer to different desirable characteristics of a clustering and stated that the user can be interested in several of these criteria rather than just one of them. In this respect, he proposed two random clustering algorithms that are meant to generate clusters with standardised characteristics so that users can aggregate them in a suitable way, specifying weights for the various criteria that are relevant to the clustering application at hand. As a continuation of Hennig (2017)'s paper, some new additional random clustering algorithms are introduced, and the calibration of indexes are further scrutinized with simulation studies and analysed with some famous real data sets. In a final phase, this new concept is performed on the dissimilarity matrix obtained from football players data set as explained in the first part.

Prior to reviewing literatures and analysing all these concepts, I provide some brief background information regarding the game of football for readers who are unfamiliar with this sport. Additionally, the literatures of football statistics are briefly reviewed, especially in terms of cluster analysis.

1.2 History of Football

> "*Some people think football is a matter of life and death. I assure you, it is much more serious than that"* - Bill Shankly, best known as the manager of Liverpool from 1959 to 1974.

Football is one of the biggest (perhaps the biggest) global sport all over the world. Millions of people regularly go to football stadiums, whereas billions more watch the game on television. The world's most popular sport has a long and interesting history. Historical evidence and sources suggest that football has been played in Egypt, Ancient China, Greece and Rome. Approximately 2,500 BC, Egyptians played a football-like game during feasts of fertility. Around 400 BC, a different form of football-like game, called *'Cuju'* (translated "kick the ball with foot") was popularly played in China. The game was used by military leaders as a competitive sport to keep soldiers physically fit. In Ancient Rome, the game became so popular that it was included in the early Olympics. In fact, it is believed that football has been originated from England in the twelfth century. The kings of that time actually banned football, because interest for the traditional sports, such as fencing and archery was being reduced.

The contemporary history of football was first codified in 1863 in London, England. Twelve London clubs created more strict football rules, and then formed The Football Association, the same FA that holds today's popular FA Cup. The British have also been considered instrumental to spreading the game to other European countries, such as Spain, France, Netherlands, and Sweden, and across the world. Eventually, a governing body of football was formed by these countries, and the FIFA was founded. In 1930, the FIFA held football's first World Cup tournament in Uruguay with 13 teams. From this time, the tournament has been awarded every four years except in 1942 and 1946 when it was not held because of the Second World War.

Aside from the World Cup, several international football competitions between national teams exist, such as the Euro Championships, Copa America and the African Cup of Nations. Domestically the strongest leagues can be regarded as England (*English Premier League*), Spain (*La Liga*), Germany (*Bundesliga*) and Italy (*Serie A*).

Although football is particularly known as men's sport, the most prominent team sport has been played by women since the time of the first recorded women's games is the late 1960's and early 1970's. The FIFA Women's World Cup has been organised every four years since 1991.

In some countries, such as in the United States, Philippines, and Korea, football is referred to as soccer. The word soccer was first invented in England to distinguish between rugby football and association football. People referred to association football "assoccer", while they called rugby football "rugger" in order to avoid confusion, and later "assoccer" became "soccer". In this project, I use the word "football" in the European sense.

For sources, see *History of Football - The Origins* (http://www.fifa.com/about-fifa/who-we-are/the-game/), *History of Football* (http://www.history.co.uk/study-topics/history-of-football-tennis/history-of-football), *Football* (https://en.wikipedia.org/wiki/Football), *A Brief History of the Game* (http://www.hornetfootball.org/documents/football-history.htm) and *Who Invented Football* (http://www.football-bible.com/soccer-info/who-invented-football.html).

1.3 The Game of Football

Football is played between two teams of eleven players each with a spherical ball, and the main objective is to score by getting the ball into the opposing goal with any part of the body except the arms and hands. The side that scores the most goals wins. If both teams have the same number of goals or neither of them scored a goal, it is considered a draw. The rules of football are officially referred to as the "Laws of the Game" (17 Laws), which are described on the FIFA website in detail (see FIFA - Law of The Game in **'Reference'** section), but I will briefly summarise here.

Field of Play: The game is played on either natural or artificial surfaces, which must be green and rectangular in shape. Figure 1.1 illustrates areas of football field with its dimensions and marks.

Ball: It must be spherical with a circumference of 68-70 cm and a weight of 400 to 450 grams and made of leather (or similar) and of a certain pressure.

Number of Players: A football match is played by two teams of no more than eleven players each, one of which is the goalkeeper. A game cannot be played if either team has less than seven players. In official football competitions, the maximum number of substitutions is three. The number of substitutes in pitch is seven in general, but differs depending on the competitions. The goalkeepers are the only players allowed to handle the ball in penalty area (including goal area). Each team will have a designated captain.

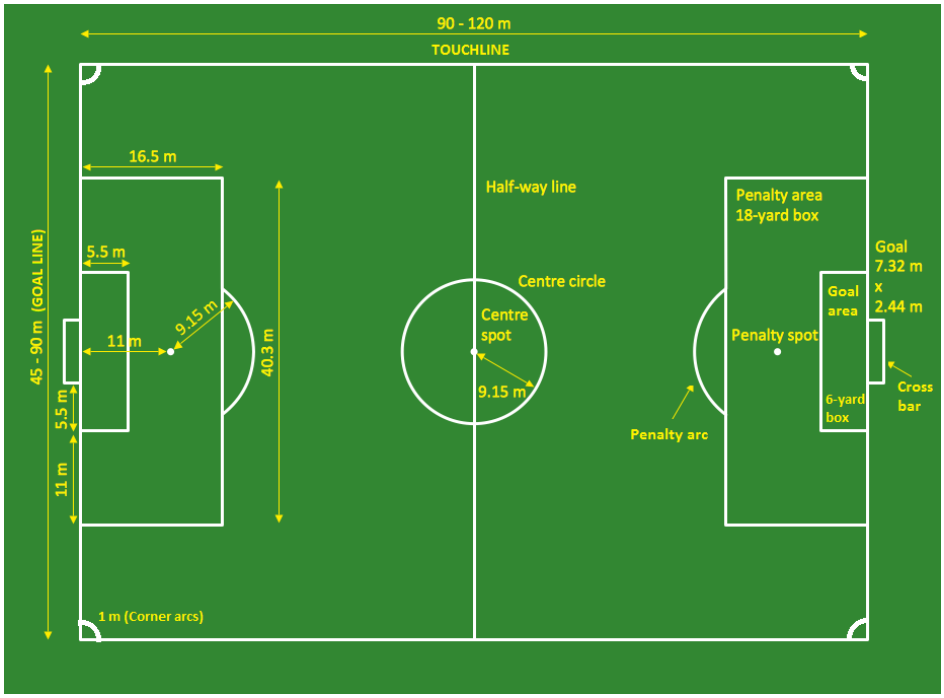


Figure 1.1: Field of football pitch

Player's equipment: Players must wear a jersey, short, socks, shin guards, and football boots. Goalkeepers will additionally wear padded gloves and should wear a kit will distinguish them from the outfield players and referees.

Head Referee: The referee should enforce the Laws of the Game during the course of a match.

Assistant Referees: The role of assistant referees is primarily to assist the head referee, and they also should enforce the Laws of the Game during a match. There should be two assistant referees, located at each touchline. In these days, games are being played with two more assistant referees at each goal lines in some competitions.

Match duration: The length of a football match is 90 minutes, played in two halves consisting of 45 minutes each with a 15 minute half-time break between halves. Additional time (also known as stoppage time or injury time) is often played at the end of each half to compensate for time lost through substitutions, injured players requiring attention, or other stoppages.

Start and restart of play: A kick-off starts play at the beginning of the first and the second half of the match or after a goal is scored. The team which starts the game is determined by a coin toss at the beginning of the match. During the kick-off, only two players from the team which starts game are allowed to be inside of the centre circle: the one kicking and the one receiving the ball.

Ball in and out of play: The ball is out of play when it has entirely crossed a goal line or touchline whether on the ground or in air. The ball remains in play at all other times, except play is stopped by the referee under any legitimate circumstances.

Method of scoring: A goal is scored if the ball entirely crosses the goal line whether on the ground or in air between the two goalposts and under the crossbar, as long as no violation of the rules has occurred.

Offside: A player is in offside position when the pass is played through to him/her, if there are fewer than two players (including the goalkeepers) between him/her and the goal line. A player cannot be caught offside in their own half. A free kick is awarded to the opposition if a player is caught offside. Figure 1.2 gives better illustration to understand the rule.

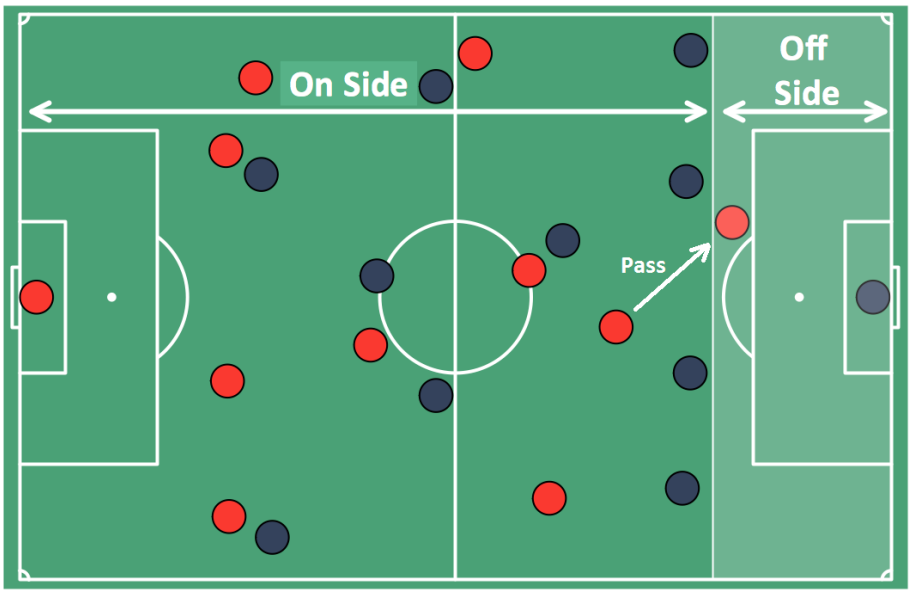


Figure 1.2: Offside

Fouls and misconducts are called if a player uses excessive force against an opponent whilst playing the game either deliberately or undeliberately or to handle the ball (except the goalkeepers in penalty area). When this occurs, the referee may show the yellow card for caution and the red card for dismissal. Two yellow cards are equivalent to one red card.

Free kicks are given by the referee after a foul or a rule infringement is committed. A free kick can either be "direct" in which a kicker may score directly, or "indirect", in which another player must touch the ball before a goal can be scored. The opposing team must be a minimum of 9.15 meter from the ball when the free kick is taken. A player is penalised for offside when receiving the ball directly from a free kick.

Penalty kicks are given if a player from the opposing team commits a foul inside his/her own penalty area. The kick must be direct and taken from the penalty spot. All the players (except the kicker from the awarded team and the goalkeepers from the opposing team) must be outside of the penalty area and penalty arc, until the penalty is taken.

A throw-in is awarded to a team if the ball has entirely crossed the touchline whether on the ground or in air. It is given to the team opposing the side that touched the ball last. The thrower must use both hands, have each foot either on the touchline or on the ground outside the touchline, and deliver the ball from behind and over their head from the point where the ball left the field of play. A goal cannot be scored directly from throw-in. A player is not penalised for an offside, when receiving the ball directly from throw-in.

A goal kick is awarded to the defending team if the opposing team causes the ball to entirely cross the goal line whether on the ground or in air without a goal being scored. Any player from the defending team is allowed to take the goal kick. It must be taken anywhere on the goal area and must exceed the penalty area. A goal can be scored directly from a goal kick against the opposing team. A player is not penalised for offside when receiving the ball directly from a goal kick.

A corner kick is given to the attacking team if the ball has entirely crossed the goal line whether on the ground or in air (without a goal being scored), and was last touched by a player from the defending team. A corner kick is taken from inside the corner arc closest to the point where the ball crosses the goal line. All defending players must be at least 9.15 meters from the corner arc until the kick is taken. A goal can be scored directly from a corner kick, and an attacking player who directly receives the ball from a corner kick cannot be penalised for offside.

1.4 Statistics in Football

> "*The ball is round, the game lasts 90 minutes, and everything else is pure theory*" - Josef Herberger, best known as the manager of West Germany from 1950 to 1964.

Statistics can be regarded as one of the most commonly applied scientific areas in sports. In football, statistical science is particularly used in prediction of match results in early years. Publications about statistical models for football predictions started from the 1950's. Moroney (1954) proposed the first model regarding the analysis of football results. He showed that football games can be fitted adequately by adopting both Poisson and Negative Binomial distribution. Reep and Benjamin (1968) analysed the series of ball passing between players during football games by using the negative binomial distribution. Hill (1974) indicated that football match results can be

predictable, and are not pure chance. Maher (1982) proposed the first model predicting outcomes of football games between teams by using the Poisson distribution.

In more recent years, studies have typically been conducted to predict players' abilities, to rate players' performances, to enhance their physical performance, etc. in many applications. Mohr et al. (2003) assessed physical fitness, development of fatigue during games, and match performances of high standard soccer players. The results showed that: (1) players performed high-intensity running during a game; (2) fatigue occurred independently of competitive standard and of team position; (3) defenders covered a shorter distance than players in other playing positions; (4) defenders and attackers had a poorer performance than midfielders and full-backs based on the presented recovery test; (5) player's physical performance changed in different seasons. Impellizzeri et al. (2006) compared the effect of specific (small-sided games) versus generic (running) aerobic interval training in junior football players, and measures match performance of them. The results of this study showed that specific and generic are equally effective modes of aerobic interval training. Rampinini et al. (2007) examined the construct validity of selected field tests as indicators of match-related physical performance in top-level professional soccer players, and showed in their empirical study that repeated-sprint ability and incremental running tests can be interpreted as measures of match-related physical performance.

Knorr-Held (2000) analysed time-dependency of team strengths by using recursive Bayesian estimation to rate football teams on the basis of categorical outcomes of paired comparisons, such as win, draw and loss. Results indicated that recent activities have more influence on estimating current abilities than outcomes in the past. On the other hand, it is well known that team skills change during the season, making model parameters time-dependent. Rue and Salvesen (2000) applied a Bayesian dynamic generalized linear model to estimate the time dependency between the teams, and to predict the next week of matches for betting purposes and retrospective analysis of the final ranking. Di Salvo et al. (2007) observed football players' performance characteristics based on their positions. According to their findings, distances covered at different categories of intensities are significantly different in the different playing positions. Although the concept here is slightly similar to the application of players' performance characteristics in different positions, the methodologies that I proposed in my report are different than the approach in this paper. Karlis and Ntzoufras (2009) designed the Time-Independent Skellam distribution model, which fits the difference between home and away scores, as an alternative to the Possion model that fits the distribution of scores.

McHale et al. (2012) described an index system and its construction of a football player's performance rating system based on the English Premier League, and clarified that the index provides a rating system of a player's best match performance rather than exploring the best play-

ers. McHale and Szczepański (2014) presents a mixed effects model for identifying goal scoring ability of football players, and found that their model performed well for partitioning players' abilities that may have influenced their goal scoring statistics in the previous season. Furthermore, some public attempts have been made to rate players, such as *Castrol Performance Index* (http://www.fifa.com/castrolindex/), and *EA SPORTS Player Performance Index* (http://www.premierleague.com/en-gb/players/ea-sports-player-performance-index.html)

1.5 Cluster Analysis in Sports

Sports have embraced statistics in assisting player recruitment and playing strategies. Different statistical methodologies have been applied to various types of sports data. Cluster analysis is one of the most powerful methodologies that has been used for aggregating similar types of players in several applications. Ogles and Masters (2003) suggested that by using cluster analysis (Ward's method) based on the increase in error sum of squares and the interpretability of the solutions, marathon runners can be categorised in five definable groups in terms of their motives for running. Gaudreau and Blondin (2004) examined if there is any relationship between several coping strategies used by groups of athletes. A hierarchical cluster analysis was conducted using Ward's method with a squared Euclidean distance measure, and the number of clusters was determined using a dendrogram [1], the agglomeration schedule coefficients [2], and the interpretability of the cluster solution. Wang et al. (2009) observed perceived coaching behaviour among basketball players, and showed that three distinct groups, which were found by using an agglomerative hierarchical clustering method, could be identified in terms of coaching behaviour. Dendrograms and agglomeration schedules were generated to provide a basis for determining the number of clusters. Yingying et al. (2010) applied different clustering techniques on athlete physiological data, and analysed a group of athletes with their performance, and proposed a new hierarchical clustering approach, which is the combination of Dynamic Time Warping [3] and hierarchical clustering, and adopted Rand Index [4] evaluation to analyse the cluster similarity.

[1] A dendrogram is a tree diagram frequently used to illustrate the arrangement of the clusters produced by hierarchical clustering (Everitt et al., 2011, chap. 4).

[2] The agglomeration schedule shows the amount of error created at each clustering stage when two different objects - cases in the first instance and then clusters of cases - are brought together to create a new cluster (Norušis, 2012, chap. 16).

[3] Dynamic Time Warping is a technique that aligns time series in such a way that the ups and downs are more synchronised.

[4] The Rand Index is a measure of the similarity between two data clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusterings (Rand, 1971).

Similarly to my idea, Lutz (2012) used several different clustering algorithms to group NBA players based on several statistics, such as goals, assists, rebounds, blocks, etc. and analysed how each cluster affects winning in the regular season. The goal was to determine which type of players are most relevant to winning. However, constructing a distance matrix and data pre – processing are inspected in greater detail in my project, and I pay much more attention to players' similarities rather than how players affect the match results.

More recently, Kosmidis and Karlis (2015) introduced the framework of copula-based finite mixture models for cluster applications. In one of the applications in the paper, they used NBA players' data to form groups of players in terms of their performance. The aim in particular is to analyse how the introduced model is performed.

In the literature, cluster analysis has not been explicitly applied to football player's performance data, specifically to form a similarity (or dissimilarity) structure of player's performance in detail. As specified in Section 1.1, this project will consider every aspect of football player performance in order to obtain a proper distance matrix between players based on their performances, and group them by using distance-based clustering algorithms.

1.6 Thesis Structure

This report is structured as follows. This chapter, Chapter 1, contains an introduction and describes the purpose of my research project. A brief history and some fundamental and standard rules to play football were described. In addition, some literature regarding football statistics, specifically in terms of clustering were reviewed. In the next chapter, I introduce the football player performance data set with descriptions of the variables that will be used in the application part of the thesis.

In Chapter 3, I discuss the methodologies and review the literature that are relevant to data pre-processing and the dissimilarity construction for the sake of the application part of the next chapter, Chapter 4. The strategy of cluster analysis is introduced, then data pre-processing, which is one of the fundamental procedures of constructing a distance matrix, is scrutinized. I refer to some literature for different types of dissimilarity measures. Because the data set to be used for this project contains compositional data, I deliver a theoretical background behind this topic, and review the literature and the concept of distance measures in compositional data. Finally, I provide some information about how to aggregate mixed-type variables, missing values and distances, and summarise Chapter 3.

In Chapter 4, the main goal is to analyse the football data set, which was collected from

the website, www.whoscored.com, by using the information from the third chapter. Data preprocessing steps are used for the aim of designing a dissimilarity matrix. Afterwards, aggregating different types of variables as well as the combination of different dissimilarity matrices are discussed and analysed. At the end of the analysis, I compare the final constructed dissimilarity measurement with the plain standardised Euclidean distance, and show them in a mapping form.

In Chapter 5, more literature about clustering algorithms have been reviewed and two famous dimension reduction techniques have been described in terms of visual exploration for the data set of interest. In the final section, I consider some external validation criteria, specifically the most commonly used one, the adjusted Rand index, to be used for further analysis. In Chapter 6 clustering validation indexes for estimating the number of clusters are described and a new approach, aggregation of clustering validation indexes, is introduced for the sake of finding an optimum choice of clustering algorithm and the number of clusters.

Examination of clustering validation indexes for different types of data sets have been scrutinized in Chapter 7. First, data sets are simulated from different scenarios, where some of them are based on statistical distributions, and some others are generated based on different shaped clusters. Second, some famous data sets that have been used in real applications are analysed for the sake of estimating the number of clusters by using the idea of calibration of clustering validation indexes. In Chapter 8 the dissimilarity matrix obtained from football players data set has been studied, and the final result regarding the choice of clustering algorithm and the determination of number of cluster for this dissimilarity matrix has been discussed and finalised from a statistical and subjective-matter point of view.

In Chapter 9, some concluding remarks are made regarding all discussions and findings of this thesis, and some recommendations are given for future work.

CHAPTER 2

FOOTBALL PLAYERS PERFORMANCE DATA SET

To motivate the reader, the football players performance data set, which will be used in later chapters, is introduced in this short chapter.

2.1 Data Description

The data set, which contains 3152 football players characterized by 102 variables, was obtained from the website, www.whoscored.com [1]. The collection of this data set is based on the 2014-2015 football season with 8 major leagues (England, Spain, Italy, Germany, France, Russia, Netherlands, Turkey). The data set consists of the players who have appeared in at least one game during the season. Goalkeepers have completely different characteristics from outfield players and were therefore excluded from the analysis. Variables are of mixed type, containing binary, count and continuous information. The variables can be grouped as follows:

2.1.1 Profile variables

- **League:** This variable gives the league to which the player belongs. Leagues are ranked according to their perceived standard of football and the ranking scores are based on the information on the official website for European football (UEFA) will be used, see http://www.uefa.com/memberassociations/uefarankings/country/index.html. More information with regards to the computation of the scores can be found on the same website. Table 2.1 shows the ranking score for each league.

[1] I was granted permission by the company to use the data available on their website for my research.

Table 2.1: League ranking scores

Country	League Name	Ranking scores
Spain	La Liga	99.427
England	Premier League	80.391
Germany	Bundesliga	79.415
Italy	Serie A	70.510
France	Ligue 1	52.416
Russia	Premier League	50.498
Netherlands	Eredivisie	40.979
Turkey	Super Lig	32.600

- **Team:** Two variables will be used for the analysis: 1) Team points from the ranking table of national league based on the 2014-2015 football season, 2) team ranking scores based on the information on the UEFA website, see http://www.uefa.com/memberassociations/uefarankings/club/index.html, which provides performance rankings for those teams participating in international tournaments.

  Team and league variables are represented by ranking scores, so that they could be treated as ordinal variables, but some information may be lost by assigning an ordinal number for each category. For example, although Spain is far ahead of England and Germany based on their league ranking scores, see Table 2.1, Spain becomes much closer to them if league variables is treated as ordinal. Thus, team and the league variable is treated as continuous variables. For the analysis which is to follow, I denote $x_{(l)}$, $x_{(tp)}$ and $x_{(tc)}$ as the league scores, the team points and the team coefficients variables, respectively.

- **Name:** Player's name. This information will not be used in the analysis, but will be used for visualisation in order to identify which points belong to which players.

- **Age, Weight, Height:** Player's age, weight and height. All these variables are treated as continuous variables.

2.1.2 Position variables

Two types of position variables are defined here: 1) 15 variables indicating how many times a player played in a given position during the 2014-2015 football season (count variables), 2) 11 variables which are based on previously recorded information of a player for different positions in the previous seasons (binary variables). $Y_{(15)}$ and $Y_{(11)}$ denote the first and second type of position

variables, respectively. Figure 2.1 gives an idea of the two different types of positions.

(a) Position-1

(b) Position-2

Figure 2.1: Position variables

D: Defender, M: Midfielder, A: Attacking, FW: Forward, C: Center, L: Left, R: Right

Since a player can play in several positions, multiple binary variables will be applied for $Y_{(11)}$ variables. For instance, a player is in the position of AM(CLR), which means that this player has (or had) played in the attacking midfielder (AM) position, and he can also play in either centre, left or right (CLR) side of the field in attacking position. Thus, I will apply '1' for AMC, AML, and AMR variables, and the other variables will remain '0'. Consider another example that a player can play in AM(CL) or MC or FW, so in this case I apply '1' for AMC, AML, MC, FW, and the other variables are '0'. Table 2.2 shows how this assignment has been made for the binary information.

2.1.3 Performance variables

These types of variables are all count variables and can be grouped into five categories in terms of their meanings. Table 2.3 summarises all the variables, and the categories of performance variables with their sub-categories. Some performance variables are divided into sub-variables that represent

Table 2.2: Assignments for $Y_{(11)}$

Main Positions	**Players Positions**	**Assignment of variables**
Defender	D, DC, DCL, DCR, DCLR	DC
	D, DL, DCL, DLR, DCLR	DL
	D, DR, DCR, DLR, DCLR	DR
Defensive Midfielder	DMC	DMC
Midfielder	M, MC, MCL, MCR, MCLR	MC
	M, ML, MCL, MLR, MCLR	ML
	M, MR, MCR, MLR, MCLR	MR
Attacking Midfielder	AMC, AMCL, AMCR, AMCLR	AMC
	AML, AMCL, AMLR, AMCLR	AML
	AMR, AMCR, AMLR, AMCLR	AMR
Forward	FW	FW

players' information in some distinct sub-groups, such as tackles, blocks, etc. On the other hand, some performance variables are partitioned in different sub-parts in terms of their meanings, then those parts are also divided into sub-variables, such as shots, goals, passes, key passes. The total attempts from a sub-part of one action have to be the same as total attempts from another sub-part of the same action. For example, a player has made 10 total shots of which 5 were from the penalty area, 3 from the six yard box and 2 from the out of box, whereas 6 in open play, 1 in counter attack, 3 in set piece and 0 in penalty taken. Note that this is not applied for the *'Type'* category in Pass action, because passes can also be another type different than cross, corner and free-kick. For example, the *'Other'* sub-variables in the *'Type'* categories of Key pass and Assist actions are partitioned as the unspecified parts, see Table 2.3, but this has not been specified for Pass action in the data set.

The descriptions of the performance variables are summarised in Table 2.4. The meanings of these variables are fundamentally important for the further analysis. For instance, four types of 'Other' sub-variables are available in the data. The 'Other' sub-variables from body parts of shots and goals reflect the information of any other body parts, excluding right or left foot or head, whereas the 'Other' sub-variables from types of key passes and assists are defined as unclassified sub-variables, which will be explained in detail in Section $4.1.1$. Some of variables' definitions, which do not exist in Table 2.4, have been defined in Section 1.3.

Table 2.3: Variables on the football player performance data

PROFILE	**POSITION**	**PERFORMANCE**				
		Time	***Subjective***	***Defensive***	***Offensive***	***Pass***
League	**Position-1**	Apps	MotM	**Tackles**	**Shots**	**Passes**
Team-1	• DC	Mins	Ratings	• Dribble past	*1. Zones*	*1. Length*
Team-2	• DL			• Tackle	• Out of box	• AccLP
Name	• DR			Offsides	• Six yard box	• InAccLP
Age	• DMC			Interceptions	• Penalty area	• AccSP
Height	• DML			Fouls	*2. Situations*	• InAccSP
Weight	• DMR			Fouled	• Open play	*2. Type*
	• MC			Clearances	• Counter	• AccCr
	• ML			**Blocks**	• Set piece	• InAccCr
	• MR			• Shots blocked	• Penalty taken	• AccCrn
	• AMC			• Crosses blocked	*3. Body parts*	• InAccCrn
	• AML			• Passes blocked	• Right foot	• AccFrk
	• AMR				• Left foot	• InAccFrk
	• FW				• Head	**Key passes**
	• FWL				• Other	*1. Length*
	• FWR				*4. Accuracy*	• Long
	Position-2				• Off target	• Short
	• DC				• On target	*2. Type*
	• DL				• Blocked	• Cross
	• DR				**Goals**	• Corner
	• DMC				*1. Zones*	• Thrball
	• MC				• Out of box	• Free-kick
	• ML				• Six yard box	• Throw-in
	• MR				• Penalty area	• Other
	• AMC				*2. Situations*	**Assists**
	• AML				• Open play	• Cross
	• AMR				• Counter	• Corner
	• FW				• Set piece	• Thrball
					• Penalty taken	• Free-kick
					3. Body parts	• Throw-in
					• Right foot	• Other
					• Left foot	
					• Head	
					• Other	
					UnsTchs	
					Dispossesses	
					Aerials	
					• Won	
					• Lost	
					Dribbles	
					• Successful	
					• Unsuccessful	

D: Defender, M: Midfielder, A: Attacking, FW: Forward, C: Center, L: Left, R: Right
Apps: Appearances, MotM: Man of the match, UnsTchs: Unsuccessful touches, Thrball: Through-ball
Acc: Accurate, InAcc: Inaccurate, LP: Long pass, SP: Short pass, Cr: Cross, Crn: Corner, Frk: Free-kick

Table 2.4: Description of performance variables

Category	Variable	Description
Time	Appearances	Number of games that a player played during the 2014-2015 football season.
	Minutes	Number of minutes that a player played during the 2014-2015 football season.
Subjective	Ratings	The rate information of players based on a system which is calculated live during the game based on a unique, comprehensive statistical algorithm. The algorithm has been explained on the website, https://www.whoscored.com/Explanations.
	Man of the match	Number of games that a player has the highest rating in a match.
Defensive	Dribble past	Being dribbled past by an opponent without winning a tackle (failure in the action of tackling).
	Tackle	Dispossessing an opponent, whether the tackling player comes away with the ball or not (success in the action of tackling).
	Offside	Being caught in an offside position resulting in a free kick to the opposing team.
	Interception	Preventing an opponent's pass from reaching their team-mates.
	Foul	An illegal manoeuvre by a player that results in a free kick for the opposing team (does not include offsides).
	Fouled	Being illegally impeded by an opponent, resulting in a free kick.
	Clearance	Action by a defending player that temporarily removes the attacking threat on their goal or that effectively alleviates pressure on their goal.
	Block	Prevention by an outfield player of an opponent shot from reaching the goal, cross or pass.
Offensive	Shot & Goal	An attempt to score a goal or to score a goal from any areas of the field (see Figure 1.1), in any situations, made with any (legal) part of the body, or either on/off target or blocked by an opponent (the category of *'Accuracy'* is not a sub-group of the 'Goal' action).
	Open play	An attempt to score a goal which has not stemmed directly from a dead ball situation.
	Counter	An attempt to score from a counter attack move.
	Set piece	An attempt to score that has been scored via a set piece situation (corner kick, free-kick or throw in).
	Shot on target	An attempt to score which required intervention to stop it going in or resulted in a goal/shot which would go in without being diverted.
	Unsuccessful touch	Bad control.
	Dispossessed	Being tackled by an opponent without attempting to dribble past them.
	Aerial	Winning or losing a header in a direct contest with an opponent.
	Dribble	Taking on an opponent and successfully making it past them whilst retaining the ball or not making it past them.
Pass	Long pass	An attempted/accurate pass of 25 yards or more, otherwise it is a short pass.
	Cross	An attempted/accurate pass from a wide position to a central attacking area.
	Key pass	The final pass leading to a shot at goal from a team-mate.
	Through ball	An attempted/accurate pass between opposition players in their defensive line to find an onrushing teammate (running through on goal).
	Assist	A pass that directly leads to a chance scored (i.e., to a goal) by a team-mate.

CHAPTER 3

DATA PRE-PROCESSING AND DISTANCE CONSTRUCTION FOR CLUSTER ANALYSIS

In this chapter, the objective is to review the literature and resources in detail, and to demonstrate the techniques to be used in the next chapter. Because one of the applications of this thesis (Chapter 4) covers the design of a dissimilarity measure prior to clustering, I mainly concentrate on this subject in this chapter.

3.1 Introduction

Cluster analysis is an unsupervised learning technique, which examines multivariate data by a variety of numerical methods, that can be used to group a set of objects into partitions according to some measure of closeness. In brief, cluster analysis is the art of finding groups in data (Kaufman and Rousseeuw, 1990).

A vast amount of clustering methods has been developed in several different fields with very diverse applications over the last four decades. Milligan (1996) listed seven steps to reach a proper clustering result. By considering those steps, the strategy of cluster analysis can be summarised in the following order:

1. Data collection for clustering
2. Choosing the measurements/variables
3. Data pre-processing, including transformation, standardisation, etc.
4. Design of a similarity or dissimilarity measure

5. Choosing a clustering method

6. Determining the number of clusters

7. Interpretation, testing, replication, cluster validation

In general, the application of clustering has been performed by the order above. Different methods can be used for different aims of clustering, but there is no such thing as a universally "best clustering method" (Hennig, 2015a).

Data collection for clustering has been provided in the previous chapter and *choosing the measurements/variables* will be discussed more in Chapter 4, because these topics are mainly related to the application of interest, and such decisions will be made based on subject-matter knowledge.

3.2 Data Pre-processing

Clustering and mapping multivariate results are strongly affected by data pre-processing decisions, including how to choose, transform, standardise variables, and to design a dissimilarity measure. Data should be processed in such a way that the resulting distance between observations matches how distance is interpreted in the application of interest (Hennig and Hausdorf, 2006). Jiang et al. (2004) expressed that data pre-processing is indispensable prior to performing any cluster analysis. Hu (2003) pointed out that data pre-processing has a huge impact on the data mining process. Templ et al. (2008) stated that cluster analysis results often strongly depend on the preparation of the data. The variety of options is huge, but guidance is scant. Different ways of data pre-processing are not objectively "right" or "wrong"; they implicitly construct different interpretations of the data.

Data pre-processing can be categorised under the following sub-topics:

1. Variable Representation

2. Variable Transformation

3. Variable Standardisation

4. Weighting variables

5. Variable selection and dimension reduction

In general, data pre-processing steps are performed in the order above, but this can change based on interpretation of data. Although some steps have the same meaning or are referred to different names in some resources, the rationales might be quite different. For example, Gan et al. (2007) interpreted that PCA (Principal Component Analysis) and SVD (Singular Value Decomposition) can be defined as data transformation techniques, and added that data standardisation can be viewed as a special case of data transformation. Furthermore, representation can be the generalisation of all these steps, or variable selection and dimension reduction can be defined as special cases of weighting and so on. The meaning of the terms will be described in the next sections in detail.

3.2.1 Variable representation

Variable representation can be defined as making decisions about how to reflect the relevant information in the variables, potentially excluding variables, defining new variables, summarising or framing information in better ways. In this respect, the second step of Milligan's list, "choosing the measurements/variables" can also be interpreted as one of data representation methods. Constructing the original data in more revealing form is typically a critical step for data scientists (Donoho, 2015, sect. 1). For instance, in order to better represent football player's information, counts of actions such as shots, goals etc. can be used relative to the period of time the player played during a football season. In addition, relevant variables can be interpreted in proportional form, which is complementary information into their totals (Aitchison, 1986). On the other hand, a collection of time series can be represented as time points, whereas they can also be represented in time-frequency domain by means of a Fourier Transform [1] or Wavelet Transform [2] or some other multi-scale forms.

In principle, the determination of representation is fundamentally related to interpreting the relevant information in better form, but statistical consideration may play a role here (Hennig, 2015a).

3.2.2 Variable transformation

The application of variable transformation is a mathematical modification of the values of a variable. The modification can be either linear or non-linear, but in general and here this technique is expressed as non-linear transformation of variables. To transform data, a transforming function

[1] The Fourier transform decomposes a function of time into the frequencies

[2] Wavelet Transform is similar to the Fourier transform with a completely different merit function

must be selected and used. Most commonly used transformations are power transformation (e.g., square root), logarithmic transformation, inverse transformation, and trigonometric transformation such as sine wave transformations. Statistically speaking, reasons for transforming data can typically be the general assumptions of statistics, such as normality, stabilising variance, reducing the effect of outliers, and so on.

Two main questions can be asked about data transformation: 1) Is transformation necessary prior to any statistical analysis? 2) If so, which transformations are most suitable for such data? In cluster analysis, there is no explicit rule to determine whether data transformation is required or which transformation should be chosen, unless statistical assumptions are required for some clustering techniques (e.g., model-based clustering). Gelman and Hennig (2015) emphasised that statistical assumptions for transformation are often not relevant for cluster analysis, where such assumptions only apply to model-based clustering, and only within the clusters, which are not known prior to transformation. Romesburg (1984) discussed that transformation can be preferable in case of existing outliers in the data because outliers are so deviant, so that their influence are much stronger than the influence of non-outliers in terms of determining the parameters of a standardising function and ultimately for determining the similarity among objects. Templ et al. (2008) advised that heavily-skewed data are first transformed to a more symmetric distribution, even though the data is not required to be normally distributed in cluster analysis. That is because if a good cluster structure exists for a variable, distribution can be expected to have two or more modes. A transformation to more symmetry will preserve the modes but remove large skewness. Kaufman and Rousseeuw (1990) recommended that logarithmic transformation can be applied to ratio scale variables in order to treat those variables as interval-scaled, but they also pointed out that this procedure could be very sensitive if zero values exist. As an example of applications, Witten (2011) applied a power transformation to count data for adopting a Possion dissimilarity matrix in order to remove over-dispersion and showed that clustering based on a Possion mixture model performs well on the transformed data.

Nevertheless, the rationale for transformation can be viewed in a different manner when designing a dissimilarity measure for clustering. The idea behind this argument is "*interpretative distance*" (or interpretative dissimilarity), which can be defined as

Definition 3.2.1. (Interpretative distance) is the distance between the data objects which are designed (e.g., the transformation, etc.) in a way that the resulting differences should match the appropriate distances between objects based on subject matter-reasons in terms of the application of interest. (Hennig and Liao, 2013).

The issue of interpretative distance is actually related to subject-matter knowledge that cannot be decided by the data alone; in other words, interpretative distance should be adapted to the meaning of variables and the specific application (Gelman and Hennig, 2015). For instance, Hennig and Liao (2013) argued that the interpretative dissimilarity between different saving amounts can be better represented by ratios rather than differences, so that the logarithmic transformation can be applied in this regard. Therefore, when designing a distance matrix, the decision of making transformation should be informed first and foremost by the context, namely the concept of interpretative dissimilarity, and if necessary depends on the clustering algorithms, secondly the consideration of outliers' influence, skewness of variable distribution, or some other statistical assumptions.

3.2.3 Variable standardisation

Variable standardisation is a linear transformation, which governs the relative weight of variables against each other when aggregating them; in other words, it makes units of variables comparable. Location or scale information may be lost after standardisation. Standardisation can also be viewed as a special case of weighting, because mathematically both standardisation and weighting are multiplications by a constant. Weights in standardisation are determined by statistical parameters of variables, whereas weights can also be chosen based on subject-matter importance of variables, which will be discussed in Section 3.2.4.

The standardisation formula is shown in Equation (3.1).

$$x_{ij}^{*} = \frac{x_{ij} - l(x_j)}{s(x_j)}, \quad i = 1, \ldots, n \quad and \quad j = 1, \ldots, p, \tag{3.1}$$

where x_{ij} is the j^{th} variable on the i^{th} object in the data, $l(x_j)$ and $s(x_j)$ are location and scale functions, respectively. Some of these methods have typical location estimators, such as mean or median, but only the scale parameters are relevant because cluster analysis to be used here is invariant against location changes. In such cases, Equation (3.1) is characterised as $x_{ij}^{*} = x_{ij}/s(x_j)$. Various types of standardisation methods with their scale parameters, which are obtained from different sources (e.g., Gan et al. (2007, chap. 4) and Milligan and Cooper (1988)), are listed in Table 3.1. Note that some studies, see Milligan and Cooper (1988), considered the 'Rank' transformation as one of the standardisation approaches, but this could be discussed as transformation due to its non-linear impact on variables.

Table 3.1: Some data standardisation methods

Name	**Scale** $\langle s(x_j) \rangle$
Unit-variance (Z-score)	$\sigma_j = \left[\frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2\right]^{1/2}$
Average absolute deviation	$AAD(x_j) = \frac{1}{n}\sum_{i=1}^{n}\lvert x_{ij} - m(x_j)\rvert$
Median absolute deviation	$MAD(x_j) = m(\lvert x_{ij} - m(x_j)\rvert)$
Range	$R_j = \max\limits_{1\leq j\leq n}(x_{ij}) - \min\limits_{1\leq j\leq n}(x_{ij})$
IQR	$IQR(x_j) = Q_3 - Q_1$

$m(\cdot)$ is the median function, and Q_3 and Q_1 is the upper and lower quartiles, respectively

In cluster analysis, this technique is a required step in order to remove the effects of the scaling of the variables when constructing a dissimilarity matrix, unless methodology, such as the Mahalanobis distance, which standardises variables internally, is used (Hennig, 2015a). The (squared) Mahalanobis distance is shown in Equation (3.2).

$$d_M(x_i, x_j)^2 = (x_i - x_j)^T S^{-1}(x_i - x_j), \tag{3.2}$$

where S is a scatter matrix such as the sample covariance matrix.

Several studies have been conducted to examine the performance of different types of standardisation techniques. Milligan and Cooper (1988) studied eight standardisation techniques (z-score, range, sum, maximum, and rank, etc.) for generated artificial continuous data sets by using a Monte Carlo algorithm and concluded that range standardisation is more effective than the other ones in terms of superior recovery of the cluster structure. Three design factors were introduced in the study by modifying the data generation algorithm: Cluster Separation, Maximum Variances, and Error Conditions which can be considered between-subject factors. Standardization Procedures, Clustering Methods, and Coverage Level are within-subject factors, since these are measured repeatedly on each data set. The correct cluster structure was known beforehand. Gnanadesikan et al. (1995) analysed eight simulated and real continuous data sets and found that range standardisation was preferable to standardisation to unit variance, although both approaches were generally effective. They also specified that standardising based on estimates of within-cluster variability worked well overall. Everitt et al. (2011) suggested that if all the variables are continuous with different units of measurement, standardising each variable to unit variance can be a reasonable choice prior to any analysis.

Art et al. (1982) discussed a method in which clusters are first guessed based on smallest dissimilarities between objects, then estimated a within-cluster covariance matrix in Mahalanobis distance by using an iterative algorithm. Gnanadesikan et al. (1995) extended their approach, and used the combination of between and within-cluster covariance matrix, instead of adopting only a within-cluster covariance matrix. Alternatively, De Soete (1986) suggested to re-weight variables, so that weighted Euclidean distances approximately optimise *ultrametricity* [3].

On the other hand, Hennig and Liao (2013) approached the concept slight differently in their analysis of mixed-type variables, and argued that range standardisation is not preferable in case of existing extreme outliers on a certain variable. That is because the distance between the non-outlier points can be approximately 0 and only the outliers are considerably far away from the rest. They suggested that using a robust statistic, such as the IQR standardisation can be a proper choice in this case; however, the distance between these outliers can be much larger than the distance between the rest of the observations on this variable, so that outliers may dominate distance on other variables. Their conclusion was to adopt standardisation to unit variance, which is not only a compromise between the two approaches, but is also better calibrated to mixed-type of variables. Gan et al. (2007) also remarked that the choice of proper standardisation method depends on the original data and the subject of the application.

3.2.4 Weighting variables

The framework here is to use weights as a linear transformation of the variables based on subject-matter reasons. Statistical aspects can be adopted in order to find appropriate weights, but Section 3.2.3 has already covered this as a special case of weighting strategies.

The subjective weight assignment was employed to the variables in some applications of cluster analysis, see Sokal and Rohlf (1981), Gordon (1990), Hennig and Liao (2013). On the other hand, Sneath et al. (1973) advised to reduce the subjective importance of judgements on the variables, because this may reflect an existing classification task in the data, so that previously unnoticed groups may emerge. However, Gelman and Hennig (2015) clarified that the use of the terms objectivity and subjectivity in statistics is often counter-productive, and decisions are valuable in statistics that can often not be made in an objective manner; therefore, researchers may have the opportunity to recognize the information from different perspectives. Following to Gelman and Hennig (2015)'s argument, in Section 4.1.4 the decision of weight assignment is made based on subject-matter reasons when constructing a dissimilarity matrix in Chapter 4.

[3]The ultrametric property, which was first introduced by (Hartigan, 1967), (Jardine et al., 1967) and (Johnson, 1967), is that for any three objects, the two largest distances between objects are equal.

3.2.5 Variable selection and dimension reduction

Variable selection and dimension reduction are special cases of weighting and representation, which might be required for performing reasonable cluster analysis in terms of better predictability and interpretability, or can be legitimate to reduce the computational cost in high dimensional data. The difference between these two techniques can be explained as; variable selection reduces the data space by removing unimportant variables (assigning zero weights to those variables), whereas dimension reduction transforms (or represents) the data into lower dimensional space by using a combination of the variables. Hennig and Meila (2015) summarised feature selection techniques in different forms, such as applying prior to clustering, integrating with clustering, selecting a subset of the available variables (e.g., variable selection), defining new variables (e.g., using linear combinations). On the other hand, Alelyani et al. (2013) reviewed feature selection methods by categorising them as different models, such as filter model, wrapper model and hybrid model. According to certain criteria, filter model methods evaluate the score of each feature, but they do not utilise any clustering algorithm to test the quality of the features, whereas wrapper models find a subset of features, then evaluate the clustering quality for a different subset(s), and finally continue these two steps until the desired quality is found. The Hybrid model can be considered as a combination of the two models above.

For carrying out dimension reduction before clustering, PCA (*Principal Component Analysis*) and MDS can be viewed as popular classical methods. PCA, which was invented by Karl Pearson, see Pearson (1901), reduces dimension by preserving most of the covariance of the data, whereas MDS reduces dimension by using distances between data points. Classical MDS (Kruskal, 1964a) and PCA are the same when dissimilarities are given by Euclidean distance. Other traditional dimension reduction methods for statistical learning algorithms can be seen in the review of Fodor (2002), and some projection pursuit-type methods aiming at finding low-dimensional representations of the data particularly suitable for clustering are shown in the articles of Hennig (2004), Hennig (2005), Tyler et al. (2009) and Bolton and Krzanowski (2012). Alternatively, Vichi et al. (2007) discussed factorial dimensionality reduction of variables with simultaneous K-means clustering on a three-mode data set. Rocci et al. (2011) proposed a new dimension reduction method as the combination of linear discriminant analysis and K-means clustering for two way data and examine its performances with a simulation study. On the other hand, Hennig and Meila (2015) clarified that although PCA and MDS are occasionally used for forming informative variables, they are not directly related to clustering, due to their objective functions (variance, stress); therefore, some information that is important for clustering might be lost.

Variable selection is essentially a further method of constructing weights from a data matrix.

Fowlkes et al. (1988) proposed a forward selection method to identify the subset of the variables for the context of complete linkage hierarchical clustering. Witten and Tibshirani (2012) proposed a framework of sparse clustering, particularly K-means and sparse hierarchical clustering, in which a Lasso-type penalty is used for selecting the features. Variable selection is also used in model based clustering. Raftery and Dean (2006) recast the variable selection problem as a model selection problem and used modified versions of the BIC, and then Maugis et al. (2009) proposed a generalized version of their model. Ritter (2014) analysed a variable selection strategy behind robust cluster analysis.

The challenge of clustering high-dimensional data is a big topic in the new century. Many clustering methods become computationally expensive in high dimensional space, but distance-based clustering techniques are not, unless n (number of observations) is large. That is because the size of computed distance matrix does not rely on the dimension of the data space. However, as the number of dimensions grows, the relative Euclidean distance between a point in a set and its closest neighbour and between that point and its furthest neighbour changes in some non-obvious ways. Aggarwal et al. (2001) examined the behaviour of different power (p) values, of Minkowski distance in different high dimensional data sets, and showed that the fractional distance metric is a $L_p - norm$, where $p \in (0, 1)$ that provides more meaningful results than the $L_p - norm$, where $p > 1$ both from the theoretical and empirical perspective in cluster analysis. Alternatively, Friedman and Meulman (2004) proposed a new procedure for computing a weight for each variable by using an iterative approach, instead of assigning a weight to each variable for the entire data set. Their approach is to find clusters on different subsets of variables by combining conventional distance-based clustering methods with a particular distance measure.

So far, I reviewed some publications regarding the topics of variable selection and dimension reduction from different perspectives. The idea of adopting these methods is often related to the performance of clustering, specifically in case of the presence of homogeneous "noise" variables. However, the decision of selecting variables is fundamentally important for the meaning of the resulting clustering in real application. Modifying or excluding variables implies changing the meaning of clusters, see Hennig (2015b), Gelman and Hennig (2015) and Hennig and Meila (2015). For instance; Hennig and Liao (2013) analysed a socio-economic stratification data set, in which the variables *income*, *savings*, *education* and *housing* were essential. The variable of income does not show any clear grouping structure; however, it should be retained, because it reflects some meaningful information in the sense of socio-economic stratification. Hennig and Meila (2015) remarked that the information is shared by two variables that in terms of their meaning are essential for the clustering aim is additional information that should not be lost. Therefore, it is advisable not to operate any of these procedures automatically in any application, unless the clustering aim

is not directly related to the application of interest.

3.3 Aggregation of Variables for Constructing Dissimilarity/Similarity Measures

Clustering is often defined as grouping similar objects together. The process of constructing a design, which determines how the objects are close to each other or how far away they are is one of the central stages prior to performing cluster analysis. *Similarity* (or *proximity*), *dissimilarity*, (or *distance*) can be referred to as the terms of this process. Everitt et al. (2011) specified that a similarity coefficient indicates the strength of the relationship between two data points. Many clustering investigations start with the distance or proximity measures between points ($n \times n$ one-mode matrix). In mathematical form, dissimilarity measures can be defined as;

$$d(\mathbf{x}, \mathbf{y}) = d(x_1, x_2, \ldots, x_d;\ y_1, y_2, \ldots, y_d) \tag{3.3}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_d)$ are two d-dimensional data points. A distance function d is a metric in a set E if and only if it satisfies the following conditions (Anderberg, 1973):

1. Reflexivity: $d(\mathbf{x}, \mathbf{y}) = 0 \Longleftrightarrow \mathbf{x} = \mathbf{y}$
2. Nonnegativity: $d(\mathbf{x}, \mathbf{y}) \geq 0 \quad \forall$ **x** and **y** in E
3. Symmetry: $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$
4. Triangle inequality: $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{y}, \mathbf{z})$

Here **x**, **y** and **z** are arbitrary data points. Dissimilarity is defined by the first three, whereas metric is defined by all these four conditions. In some situations, the metric condition is not preferable, and the triangle inequality is usually connected to Euclidean intuition. For example, Hennig and Hausdorf (2006) argued in their application on presence-absence data of species on regions. They stated that if two species A and B present on two small disjoint areas, they are very dissimilar, but both should be treated as similar to a species C covering a larger area that includes both A and B if clusters are to be interpreted as species grouped together. Thus, researchers need to consider these conditions, unless the conditions are not connected with their analysis in terms of interpretation of the application's context.

Kaufman and Rousseeuw (1990) explained that instead of using a dissimilarity coefficient function, a similarity coefficient function, $s(\mathbf{x}, \mathbf{y})$ can also be applied for finding groups in data. $s(\mathbf{x}, \mathbf{y})$ is typically a complementary form of the distance function, which can be defined as follows;

$$d(\mathbf{x}, \mathbf{y}) = 1 - s(\mathbf{x}, \mathbf{y}) \tag{3.4}$$

Values in between 0 and 1 indicate various degree of resemblance, and in general it is assumed that the following properties hold:

1. $0 \leq s(\mathbf{x}, \mathbf{y}) \leq 1$
2. $s(\mathbf{x}, \mathbf{x}) = 1$
3. $s(\mathbf{x}, \mathbf{y}) = s(\mathbf{y}, \mathbf{x})$

for two arbitrary data points, **x** and **y** in the set. Given a data set $D = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$, each object of which is described by a d-dimensional feature vector, the dissimilarity matrix and the similarity matrix for D are defined in Equation (3.5) and Equation (3.6)

$$M_d(D) = \begin{bmatrix} 0 & d_{12} & \ldots & d_{1n} \\ d_{21} & 0 & \ldots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \ldots & 0 \end{bmatrix} \tag{3.5}$$

$$M_s(D) = \begin{bmatrix} 1 & s_{12} & \ldots & s_{1n} \\ s_{21} & 1 & \ldots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \ldots & 1 \end{bmatrix} \tag{3.6}$$

where $d_{i,j} = d(\mathbf{x}_i, \mathbf{y}_j)$ and $s_{i,j} = s(\mathbf{x}_i, \mathbf{y}_j)$ are dissimilarities and similarities between i^{th} and j^{th} objects, respectively. More information about theoretical properties of dissimilarity/similarity coefficients can be found in Gower (1966), Gower (1971), Gower (1982), Gower (1985) and Gower and Legendre (1986).

The choice of dissimilarity is fundamentally important for analysis, and the decision of which type of distance measures are more appropriate relies on a combination of experience, knowledge and interpretation. Gower and Legendre (1986) pointed out, "a dissimilarity or similarity coefficient has to be considered in the context of the descriptive statistical study of which it is a part,

including the nature of the data, and the intended type of analysis". Various dissimilarities are available for different types of variables, which fall into one of two groups: numerical or categorical.

3.3.1 Dissimilarity measures for categorical variables

Categorical variables have values that describe a 'quality' or 'characteristic' of a data unit, such as 'what type' or 'which category'. They fall into mutually exclusive (in one category or in another) and exhaustive (include all possible options) categories. Therefore, categorical variables are qualitative variables and tend to be represented by a non-numeric value. They can be classified in two groups: ordinal and nominal.

Binary variables

Binary variables have only two possible outcomes; in other words, if two objects x_1 and x_2 are represented by p binary variables, let a_{ij} be the number of variables $k = 1, \ldots, p$ on which $x_{1k} = i$, $x_{2k} = j$, $i, j \in \{0, 1\}$. Before selecting an appropriate proximity measure for binary variables, researchers need to consider the character of binary information in terms of the meaning for 0. In this sense, binary variables can be categorised in two kinds in terms of their meanings.

1) **Symmetric binary variable:** if both of its states are equally valuable (e.g., gender variable).
2) **Asymmetric binary variable:** if the outcome of the states are not equally valuable (e.g., positive or negative outcomes of a disease test, or presence-absence variable).

The similarity coefficients for binary variables are defined in terms of the entries in a cross-classification set for counts of binary outcomes between two objects (See Table 3.2). Some of the most commonly used proximity measures are listed in Table 3.3, and various similarity measures for such data have been proposed, see Shi (1993) and Choi et al. (2010). In addition, the characteristics of similarity measures for binary data and their relationship between each other have been discussed, see Gower and Legendre (1986).

Categorical variables with more than two levels

In the previous section, I stated that binary variables can only take two values, but here I examine categorical variables, which may take on more than two levels. (e.g., eye colour). The most

Table 3.2: 2×2 *Contingency table for binary data*

		Object j		
	Outcome	1	0	Total
Object i	1	a_{11}	a_{10}	$a_{11} + a_{10}$
	0	a_{01}	a_{00}	$a_{01} + a_{00}$
	Total	$a_{11} + a_{01}$	$a_{10} + a_{00}$	$p = a_{11} + a_{01} + a_{10} + a_{00}$

Table 3.3: Similarity measures for binary data

Code	**Coefficients**	**Formula** $(s(\mathbf{x}, \mathbf{y}))$	**Type**
S1	Matching coefficient	$\frac{a_{11}+a_{00}}{p}$	Symmetric
S2	Jaccard coefficient	$\frac{a_{11}}{a_{11}+a_{01}+a_{10}}$	Asymmetric
S3	Kulczynski (1927b)	$\frac{a_{11}}{a_{01}+a_{10}}$	Asymmetric
S4	Rogers and Tanimoto (1960)	$\frac{a_{11}+a_{00}}{a_{11}+2(a_{01}+a_{10})+a_{00}}$	Symmetric
S5	Sneath et al. (1973)	$\frac{a_{11}}{a_{11}+2(a_{01}+a_{10})}$	Asymmetric
S6	Gower and Legendre (1986)	$\frac{a_{11}+a_{00}}{a_{11}+\frac{1}{2}(a_{01}+a_{10})+a_{00}}$	Symmetric
S7	Gower and Legendre (1986)	$\frac{a_{11}}{a_{11}+\frac{1}{2}(a_{01}+a_{10})}$	Asymmetric

common way to construct the similarity or dissimilarity between some objects i and j is to use the *simple matching* approach, see Kaufman and Rousseeuw (1990).

$$s(\mathbf{x}_i, \mathbf{x}_j) = \frac{m}{p} \quad and \quad d(\mathbf{x}_i, \mathbf{x}_j) = \frac{p-m}{p}, \tag{3.7}$$

where m is the number of matches and p is the total number of nominal variables. An alternative method is to use many binary variables for a variable including multiple categories; in other words, dummy variables will be generated for those categories, then simply use an appropriate similarity measure for binary variables. However, Everitt et al. (2011) pointed out that this approach is not satisfactory for symmetric binary variables, because large number of 'negative' matches will inevitably be involved.

Ordinal variables

A discrete ordinal variable can be treated as a categorical variable, the only difference is that the M categories are ordered in a meaningful sequence (e.g., ranking). In distance analysis, this can be treated as continuous variables by using plain Likert codes [4], which can be achieved by scoring the categories in ordinal manner. Researchers have used different approaches for ordinal variables; for instance, Conover (1980) suggested straightforward scores which are obtained by ranking, or Ostini and Nering (2006) applied a more sophisticated method based on item response theory [5].

Kaufman and Rousseeuw (1990) advised that the usual distance formulas for continuous variables (e.g., Euclidean) can be applied to ranking scores of ordinal variables, due to their interval-scaled structure; however, they claimed that in case of different values of M categories in multiple ordinal variables, it is useful to convert all variables to the $0 - 1$ range for assigning equal weight to each variable. The formula is given by

$$z_{ik} = \frac{r_{ik} - 1}{M_k - 1} \tag{3.8}$$

where r_{ik} is the rank of i^{th} object and the k^{th} variable, and M_k is the highest rank for k^{th} variable. In this way, all z_{ik} will lie between 0 and 1.

In some cases, variables can have a more complex structure which may need to be treated in a customised way. For example, Hennig and Liao (2013) dealt with a problem for one variable which contains several relevant levels, but cannot be ordered. They managed the issue in such a way that the distances between those levels interpretatively make sense in terms of the application of interest.

3.3.2 Dissimilarity measures for numerical variables

A numerical variable (quantitative data) can be classified as a collection of numbers, which can be either a discrete or a continuous. In terms of distance measurement, Stevens (1946) discussed more about the level of measurement, and claimed that all measurement in science was conducted using four different types of scales that he called "nominal" and "ordinal" classified as *qualitative*,

[4] Likert (1932) introduced Likert scale codes as a technique for the measurement of attitudes, which emerges from collective responses to a set of items. Respondents are asked to indicate their level of agreement with a given statement by way of an ordinal scale (e.g., strongly disagree, disagree, neither, agree, or strongly agree), and the numerical codes can be assigned as $1, 2, \ldots, O_j$, where O_j, $j = 1, \ldots, q$ are ordered finite sets.

[5] Item Response Theory is a body of theory describing the application of mathematical models to data from questionnaires and tests as a basis for measuring abilities, attitudes, or other variables (Embretson and Reise, 2013)

and "interval" and "ratio" classified as *quantitative*. Stevens aimed at defining which arithmetic operations could be carried out on measurements in a meaningful way. By this he meant that "meaningful" operations only should make use of those features of a measurement which carry information.

Definition 3.3.1. The **scale type** of a measurement operation is defined by the family Φ of transformations ϕ by which the measurements can be transformed without losing their original information content. Statistical statements about sets of measurements $\{x_1, \ldots, x_n\}$ are **meaningful** if they are invariant, i.e., for all $\phi \in \Phi$, $S(x_1, \ldots, x_n)$ holds if and only if $S(\phi(x_1), \ldots, \phi(x_n))$ holds.

There has been a controversy in the literature about Stevens's concept of "meaningfulness" and his scale type classification. The most problematic aspect is that Stevens blames a lot of applications statistical methodology as "meaningless" that have been used often and in some situations have proven useful. More discussions about the theory of measurement can be found in Hand (1996), and in Andrew Gelman's post (http://andrewgelman.com/2015/04/28/whats-important-thing-statistics-thats-not-textbooks/).

The nature of numerical variables can be divided into two categories in terms of their scales (Kaufman and Rousseeuw, 1990):

1) **Interval-scale** is a measurement where the difference between two values is meaningful. For instance, the difference between a temperature of $100\,^{\circ}\mathrm{C}$ degrees and $90\,^{\circ}\mathrm{C}$ degrees is the same difference as between $90\,^{\circ}\mathrm{C}$ degrees and $80\,^{\circ}\mathrm{C}$ degrees. Note that all meaningful statements from ordinal scales are still meaningful and all ordinal information is still valid.

2) **Ratio-scale:** Ratio-scaled variables are always positive measurements, in which the proportional difference between two values is meaningful; for example, the distinction between 3 and 30 has the same meaning as the distinction between 1 and 10. Note that all meaningful statements from interval scales are still meaningful and all interval information is still valid.

The determination of which two categories above is related to the variable of interest depends on the interpretation of the data. For example, one can discuss whether the distance between proportional variables are better represented in ratio-scaled or interval-scaled basis (e.g., should the distance between 0.05% and 0.10% be the same as the distance between 50.05% and 50.10%?), see the discussion in Section 4.2.2 for the selection of these two categories in terms of the application of interest.

In the next sections, count and continuous types of variables will be discussed, and I pay more attention to proportional (compositional) variables later, see Section 3.4, which is relevant to the

application part of this report. Other types of data, such as time series and repeated measurement are not the concern of this application.

Continuous variables

If all variables are continuous, the use of dissimilarity or distance measures can be made by quantifying proximities between objects. A variety of distance measures have been proposed, and several of them are summarized in Table 3.4. The most commonly used distances are the Euclidean and the Manhattan distances, for which L_2 and L_1 norms are used, respectively, in the formulas. Hennig (2015a) pointed out another difference between these two distances that the Euclidean distance is invariant to rotations of the data, but in the Manhattan distance the role of original variables is more important than axes obtained by potential rotation. Figure 3.1 illustrates the difference between the Euclidean (L_2 norm) and the Manhattan (L_1 norm) distances for two variables. The Minkowski distance (L_q norm) is the general form of those types of distances. Note that for $q \geq 1$, the Minkowski distance is a metric as a result, whereas for $q < 1$ it is not a metric, because this violates the triangle inequality (Gower and Legendre, 1986).

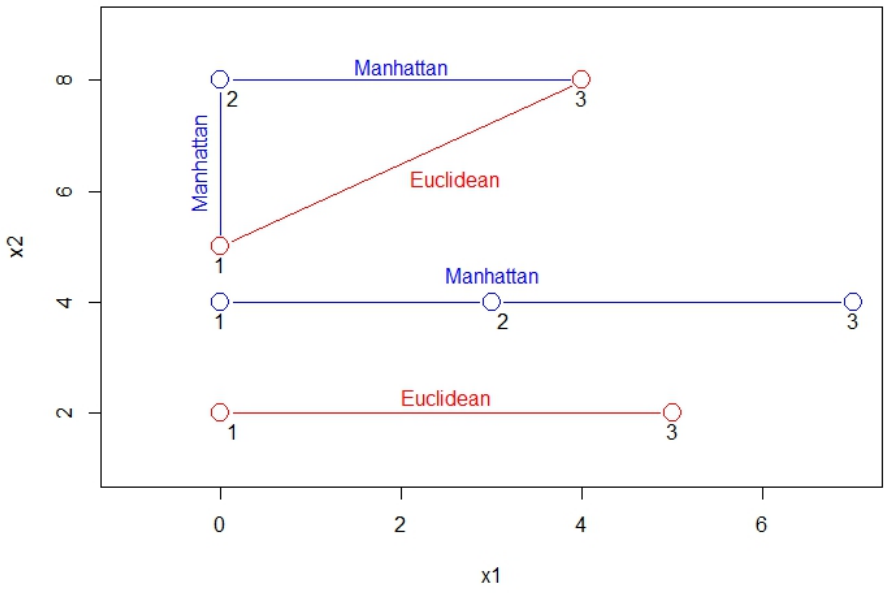


Figure 3.1: Euclidean (red) and Manhattan/city block distances (blue).

Mahalanobis distance typically aggregates strongly dependent variables down, so that joint information is only used once. Hennig (2015a) pointed out that adopting the Mahalanobis distance

is reasonable if clusters can come in all kinds of elliptical shapes; however, the weights of the variables are determined by the covariance matrix, **S**, not by their meaning. Once more, researchers may need to look at the specific application for having a clear idea about how to choose variable weights, distance measures, variable transformation and so on.

Everitt et al. (2011) provided some clarifications and references for the other distance measures, such as the Canberra distance, which is very sensitive to small differences, and the Pearson correlation, which can used for clustering variables rather than clustering objects.

Table 3.4: Dissimilarity measures for continuous data

Code	**Measures**	**Formula** $(d(\mathbf{x}_i, \mathbf{x}_j))$
D1	Euclidean distance	$(\sum_{k=1}^{p}(x_{ik} - x_{jk})^2)^{1/2}$
D2	Manhattan (City Block) distance	$\sum_{k=1}^{p} \lvert x_{ik} - x_{jk} \rvert$
D3	Minkowski distance	$(\sum_{k=1}^{p}(x_{ik} - x_{jk})^q)^{1/q}$
D4	Canberra distance	$\sum_{k=1}^{p} \frac{\lvert x_{ik} - x_{jk} \rvert)}{(\lvert x_{ik} \rvert + \lvert x_{jk} \rvert)\rvert)}$
D5	Pearson correlation	$(1 - \rho(\mathbf{x}_i, \mathbf{x}_j))/2$
D6	(Squared) Mahalanobis distance	$(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{S}^{-1} (\mathbf{x}_i - \mathbf{x}_j)$

x_{ik} and x_{jk} are, respectively, the k^{th} variable value of the p-dimensional observations for objects i and j, $\rho(\mathbf{x}_i, \mathbf{x}_j)$ is the Pearson correlation coefficient, **S** is a scatter matrix such as the sample covariance matrix

Count variables

Count data, which can only take non-negative integer values, can be as a special case of numerical variables. Dissimilarity analysis for count data often emerges in applications of ecological and biological science. Distance measures for continuous variables, especially City-Block (Manhattan) distance in Table 3.4, are also applicable for count data. On the other hand, some other dissimilarities, which can be applied for count data, are available in the literature, such as Bray-Curtis dissimilarity (Bray and Curtis, 1957). This dissimilarity coefficient (Equation (3.9)) is not a true metric, because it does not satisfy the triangle inequality axiom, despite the fact that it has been used or discussed in some applications, specifically in ecological studies, see Faith et al. (1987), Clarke et al. (2006), Warton et al. (2012) and Greenacre and Primicerio (2014):

$$d(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{k=1}^{p} |x_{ik} - x_{jk}|}{\sum_{k=1}^{p} (x_{ik} + x_{jk})} \tag{3.9}$$

The Poisson distribution can form the basis for some analyses of count data. Over-dispersion, which occurs when the variance is larger than the expected value, is often encountered when fitting very simple parametric models, such as those based on the Poisson distribution. Over-dispersion is a very common feature in applied data analysis because in practice, populations are frequently heterogeneous contrary to the assumptions implicit within widely used simple parametric models. However, it does not play a role in distance-based clustering, because no such model assumptions are essential. On the other hand, analysts could argue the impact of over-dispersion if the aim of their applications is to apply model-based clustering for count data.

Count data has also been used for model-based clustering in different applications. Karlis and Meligkosidou (2003) analysed model-based clustering for count data based on the multivariate Poisson distribution. Witten (2011) proposed an approach for clustering RNA sequencing data, which involve non-negative counts, using a new dissimilarity measure that is based upon the Poisson model. For the over dispersion problem, Robinson et al. (2010) and Anders and Huber (2010) have used negative binomial model as an alternative and extension of Poisson model, but Witten (2011) concluded that the Poisson model after the transformation performs better than the more complicated negative binomial model with moderate over-dispersion, so that the clustering proposals based on the Poisson model perform well on the transformed data. Alternatively, Cai et al. (2004) used the chi – squared statistics as a measure of the deviation of observed tag counts from expected counts, and employ it within a K-means clustering procedure.

3.4 Compositional Data

Pearson (1896) argued that if X, Y and Z are uncorrelated, then X/Z and Y/Z will not be uncorrelated. *Spurious correlation* is the term to describe the correlation between ratios of variables. Chayes (1960) later showed that some of the correlations between components of the compositions, which are the act of combining parts or elements to form a whole, must be negative because of the unit sum constraint.

Compositional data, which describe parts of some whole, was first identified by the *spurious correlation* (Bacon-Shone, 2011). They are commonly presented as vectors of *proportions*, *percentages*, *concentrations*, or *frequencies*. These types of data are an essential feature of many disciplines, such as in geology, economics, medicine, ecology and psychology. The introduced football data set in Chapter 2 involves many different sub-categories under some of the count vari-

ables, see Table 2.3, and in Section 4.1.1 these sub-variables are represented in proportional forms so that compositional data needs to be discussed for this sake. Therefore, the aim of this section is to deliver a theoretical background behind this topic, and to review the literature and the concept of distance measures in compositional data.

3.4.1 Theory of compositional data

Compositional data was more scrutinized by John Aitchison, who set up an axiomatic theory for the analysis of compositional data (Aitchison, 1986). The definition and theoretical properties of compositional data are summarised below.

Definition 3.4.1. (Compositions) A (row) vector, $\mathbf{x} = [x_1, x_2, \ldots, x_D]$, is a $D-part$ composition when all its components are non-negative real numbers and carry only relative information.

$$x_1 \geq 0, x_2 \geq 0, \ldots, x_D \geq 0 \quad and \quad x_1 + x_2 + \ldots + x_D = \kappa. \tag{3.10}$$

The fixed meaning of *relative information* refers to that where the only information is contained in the ratios between the components of the composition and the numerical value of each component by itself is irrelevant (Pawlowsky-Glahn et al., 2015).

Definition 3.4.2. (Compositions as equivalence classes) Two vectors of D positive real components $\mathbf{x}, \mathbf{y} \in \Re_+^D$ $(x_i, y_i > 0, \forall i = 1, 2, \ldots, D)$ are compositionally equivalent if there exists a positive constant $\lambda \in \Re_+$ such that $\mathbf{x} = \lambda \mathbf{y}$.

Definition 3.4.3. (Closure) For any vector of D strictly positive real components $\mathbf{z} = [z_1, z_2, \ldots, z_D] \in \Re_+^D$, and $z_i > 0 \; \forall i = 1, 2, \ldots, D$, the closure of $\mathbf{z}$ to $\kappa > 0$ is defined as

$$C(\mathbf{z}) = \left[\frac{\kappa z_1}{\sum_{i=1}^{D} z_i}, \frac{\kappa z_2}{\sum_{i=1}^{D} z_i}, \ldots, \frac{\kappa z_D}{\sum_{i=1}^{D} z_i} \right]. \tag{3.11}$$

Definition 3.4.4. The d-dimensional *simplex* is the set defined by

$$S^d = \left\{ \mathbf{x} = [x_1, x_2, \ldots, x_d] \, | x_i > 0, i = 1, 2, \ldots, d; \sum_{i=1}^{d} x_i < \kappa \right\}. \tag{3.12}$$

Definition 3.4.5. (Sample space) The sample space of compositional data, which is the d-dimensional simplex embedded in D-dimensional real space, is the set defined by

$$S^D = \left\{ \mathbf{x} = [x_1, x_2, \ldots, x_D] \,|x_i > 0, i = 1, 2, \ldots, D; \sum_{i=1}^{D} x_i = \kappa \right\}. \tag{3.13}$$

The composition is completely specified by the components of a d-part sub-vector such as $(x_1, x_2, \ldots, x_d)$, where $d = D - 1$ (Aitchison, 1986).

The difference between the compositional sample space and other real spaces can be seen in Figure 3.2, where $d = 2$ and $\kappa = 1$, and the relationship can be shown in Equation (3.14),

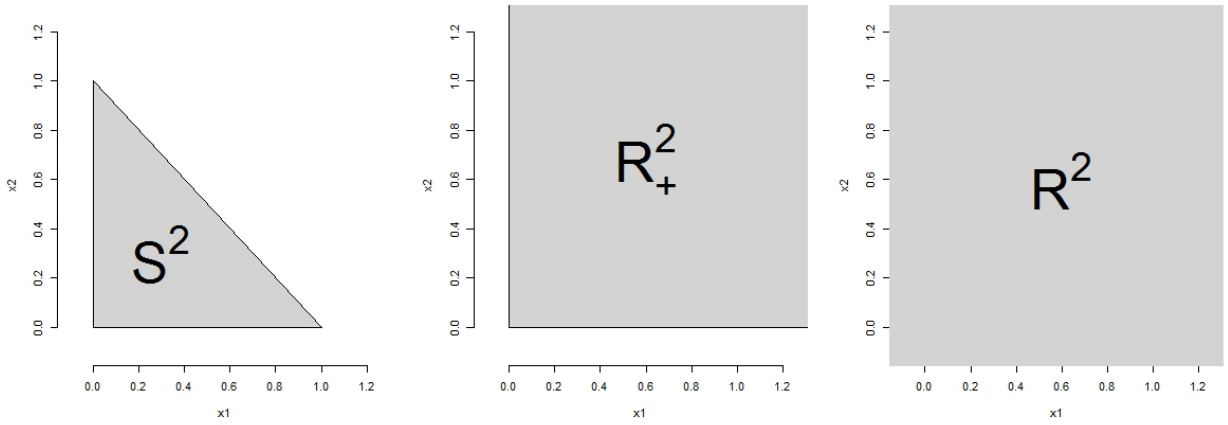


Figure 3.2: The relationship of S^2, $\Re^2_+$ and $\Re^2$

$$S^d \subset \Re^d_+ \subset \Re^d. \tag{3.14}$$

Definition 3.4.6. (Sub–compositions) Given a composition **x** and a selection of indices $S = \{i_1, i_2, \ldots, i_s\}$, a sub- composition $\mathbf{x}_s$, with s parts, is obtained by applying the closure operation to the sub-vector $[x_{i_1}, x_{i_2}, \ldots, x_{i_s}]$ of **x**. The set of sub–scripts S indicate which parts are selected in the sub–compositions, not necessarily the first s ones.

3.4.2 Log-ratio transformation

A data set which contains multiple count variables is considered as "closed" when all count variables are dependent. To get rid off all spurious correlations, a proper transformation should be considered for "opening" the data prior to performing cluster analysis, (Aitchison, 1986). Three different types of transformation have been proposed in this respect.

- **Additive log-ratio** (*alr*), in which one variable, x_D, must be selected to open the data, and that variable is subsequently lost for further analysis. The following form shows the *alr* transformation;

$$alr(\mathbf{x}) = \left[\ln \frac{x_1}{x_D}, \ln \frac{x_2}{x_D}, \ldots, \ln \frac{x_{D-1}}{x_D}\right]. \tag{3.15}$$

  The problem with the *alr* is that distances between points in the transformed space are not the same for different divisors, x_D.

- **Centred log-ratio** (*clr*), which uses the geometric average of all components in each composition, so that it does not depend on the results of one single other variable. The mathematical form is as follows;

$$clr(\mathbf{x}) = \left[\ln \frac{x_1}{g_m(\mathbf{x})}, \ln \frac{x_2}{g_m(\mathbf{x})}, \ldots, \ln \frac{x_D}{g_m(\mathbf{x})}\right], \quad g_m(\mathbf{x}) = \left(\prod_{i=1}^{D} x_i\right)^{1/D}. \tag{3.16}$$

  The disadvantage of this transformation is that the covariance matrix of *clr* is singular, so that it can be problematic in some standard statistical analysis (Aitchison, 1986, section 4.6).

- **Isometric log-ratio** (*ilr*) provides a suitable orthonormal basis in compositional space. Egozcue et al. (2003) suggested this type of transformation for avoiding collinearity in compositional form, whereas the *clr* transformation results in collinear data. Thus, the *ilr* avoids not only the arbitrariness of *alr*, but also the singularity of the *clr*.

  Definition 3.4.7. For any composition $\mathbf{x} \in S^D$, the *ilr* transformation associated to an orthonormal basis, $\mathbf{e}_i$, $i = 1, 2, \ldots, D-1$, of the simplex S^D, is the transformation: $S^D \to \Re^{D-1}$ given by

$$\mathbf{z} = ilr(\mathbf{x}) = \left[\langle \mathbf{x}, \mathbf{e}_1 \rangle_a, \langle \mathbf{x}, \mathbf{e}_2 \rangle_a, \ldots, \langle \mathbf{x}, \mathbf{e}_{D-1} \rangle_a\right], \tag{3.17}$$

  where

$$\langle \mathbf{x}, \mathbf{y} \rangle_a = \frac{1}{2D} \sum_{i=1}^{D} \sum_{j=1}^{D} \ln \frac{x_i}{x_j} \ln \frac{y_i}{y_j} \tag{3.18}$$

  is the Aitchison inner product, $\mathbf{x}, \mathbf{y} \in S^D$. Alternatively, the *ilr* transformation can be expressed in the *clr* form as

$$ilr_V(\mathbf{x}) = clr(\mathbf{x}) \cdot \mathbf{V} = \ln(\mathbf{x}) \cdot \mathbf{V}, \tag{3.19}$$

where $V = (\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{D-1})$ is orthogonal such that $\mathbf{V} \cdot \mathbf{V}^T = I_{D-1}$ (identity matrix of $D-1$ elements).

3.4.3 Dissimilarity measures for compositional data

The concept of a distance between two compositions is essential in the statistical analysis of compositional data, especially in applications such as cluster analysis and multidimensional scaling (Aitchison et al., 2000), so that Aitchison (1986) introduced the *Aitchison distance*, which includes a logarithmic transformation and a ratio-scaled type of difference, see Equation (3.20)

Definition 3.4.8. (Aitchison distance): The distance between $\mathbf{x}, \mathbf{y} \in S^D$ is expressed as

$$\begin{aligned}
d_a(\mathbf{x}, \mathbf{y}) &= \sqrt{\frac{1}{2D} \sum_{i=1}^{D} \sum_{j=1}^{D} \left\{ \ln \frac{x_i}{x_j} - \ln \frac{y_i}{y_j} \right\}^2} \\
&= \sqrt{\frac{1}{D} \sum_{i=1}^{D-1} \sum_{j=i+1}^{D} \left\{ \ln \frac{x_i}{x_j} - \ln \frac{y_i}{y_j} \right\}^2} \\
&= \sqrt{\frac{1}{D} \sum_{i<j} \left\{ \ln \frac{x_i}{x_j} - \ln \frac{y_i}{y_j} \right\}^2} \\
&= \sqrt{\sum_{i=1}^{D} \left\{ \ln(x_i) - \ln(y_i) - \frac{1}{D} \left(\sum_{j=1}^{D} \ln(x_j) \right) - \frac{1}{D} \left(\sum_{k=1}^{D} \ln(y_k) \right) \right\}^2} \\
&= \sqrt{\sum_{i=1}^{D} \left\{ \ln \frac{x_i}{g_m(\mathbf{x})} - \ln \frac{y_i}{g_m(\mathbf{y})} \right\}^2},
\end{aligned} \tag{3.20}$$

where $g_m(\cdot) = \left(\prod_{i=1}^{D} x_i \right)^{1/D}$ (Geometric mean of the compositions). It is obvious that the Aitchison distance between compositions $\mathbf{x}$ and $\mathbf{y}$ is computed by using the *clr* transformation, which is the analogue to Euclidean distance in the compositional space (Lovell et al., 2011). Since the ilr has the feature of isometricity, the distance is invariant under permutation of the parts of a composition (Egozcue et al., 2003). The following equation summarises all these arguments.

$$d_a(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{D} \{clr(x_i) - clr(y_i)\}^2} = d_e(clr(\mathbf{x}), clr(\mathbf{y})) \propto d_e(ilr(\mathbf{x}), ilr(\mathbf{y})), \tag{3.21}$$

where d_e is the Euclidean distance between two data points. On the other hand, distances between the *alr* vectors do not have such a straightforward representation, see Equation (3.21), because they do not provide an isometry between S^D and $\Re^{D-1}$.

Aitchison (1992) then proposed that any scalar measure of difference between two compositions should verify four specific requirements: *scale invariance*, *permutation invariance*, *perturbation invariance* and *sub-compositional coherence*, and the Aitchison distance (Equation (3.20)) is one of few distance measures to fulfil the following axioms.

i Scale invariance: For any positive real value $\lambda \in \Re_+$,

$$d(\lambda\mathbf{x}, \lambda\mathbf{y}) = d(\mathbf{x}, \mathbf{y}). \tag{3.22}$$

Martín-Fernández et al. (1998) stated that this condition is not essential if it is implicitly assumed that any scalar measure is always applied to compositional observations in which their sum is equal to one.

ii Permutation invariance: A function is permutation invariant if it yields equivalent results when the ordering of the parts in the compositions is permuted.

iii Perturbation invariance: Let **x, y** $\in S^D$ and $q = (q_1, q_2, \ldots q_d),\ q \in \Re_+^D$. Then,

$$d(q \oplus \mathbf{x}, q \oplus \mathbf{y}) = d(\mathbf{x}, \mathbf{y})\ for\ every\ perturbation\ q, \tag{3.23}$$

where "$\oplus$" stands for componentwise multiplication.

$$\mathbf{x} \oplus \mathbf{y} = \boldsymbol{C}[x_1 y_1, x_2 y_2, \ldots, x_D y_D] \in S^D \tag{3.24}$$

iv Sub-compositional coherence: For a sub-composition $\mathbf{x_s}, \mathbf{y_s}$ of $\mathbf{x}, \mathbf{y} \in S^D$, namely a subset of the components of **x**, **y**:

$$d(\mathbf{x, y}) \geq d(\mathbf{x}_s, \mathbf{y}_s). \tag{3.25}$$

Martín-Fernández et al. (1998) reviewed these four properties for the dissimilarity measures in Table 3.5, and showed that scale invariance, perturbation invariance and sub-compositional coherence are not satisfied by Euclidean, Manhattan, Minkowski and some other distances. For example, let $\mathbf{X}$ be the compositional data set formed by the four observations in $\mathbf{S}^3$:

$$\mathbf{x}_1 = (0.1, 0.2, 0.7), \quad \mathbf{x}_2 = (0.2, 0.1, 0.7), \quad \mathbf{x}_3 = (0.3, 0.4, 0.3), \quad \mathbf{x}_4 = (0.4, 0.3, 0.3).$$

The distance between $\mathbf{x}_1$ and $\mathbf{x}_2$ is the same as the distance between $\mathbf{x}_3$ and $\mathbf{x}_4$ on Minkowski distance for any power (e.g., Euclidean or Manhattan). That is because these measures of differences are translation invariant. Martín-Fernández et al. (1998) then claimed that the difference between $\mathbf{x}_1$ and $\mathbf{x}_2$ must be greater than the difference between $\mathbf{x}_3$ and $\mathbf{x}_4$ from a compositional point of view, because such distance measures should fulfil the four requirements above. On the other hand, Martín (1998) examined different dissimilarity coefficients, see Table 3.5 and Table 3.4, for six different clustering methods applied to three compositional data sets. The results based on cluster validity coefficients, such as Agglomerative coefficients, Divisive coefficients, Silhouette coefficients and Dunn's Partition coefficients (Kaufman and Rousseeuw, 1990) indicated that J-divergence, which is one of the distances that does not satisfy three properties (scale invariance, perturbation invariance and sub-compositional coherence), is a reasonable measure of difference between two compositions.

Furthermore, Lovell et al. (2011) examined the relationship between the Euclidean and the Aitchison's distance, they then concluded that the Euclidean distance does not accurately reflect relative changes in components, e.g., RNA sequence count data, that are conventionally dealt with on a logarithmic scale, whereas the Aitchison's distance with its focus on the ratio of corresponding components, emphasises these differences in relative abundance much more effectively. However, the interpretation of how to represent the relevant information can be different in some other applications.

3.4.4 Dealing with zeros

One of the major issues in log-ratio analysis of compositional data is zero components, as logarithms of zero values are undefined, and zero values occur quite often. To circumvent this problem, Aitchison (1986) suggested replacing each zero value with a small numerical value. However, differences in the value selected may lead to different results in applications, such as cluster analysis.

[6] J-divergence, which is based on the Kullback–Leibler divergence (Kullback and Leibler, 1951), is a popular method of measuring the similarity between two probability distributions

Table 3.5: Some measures of differences between two compositions

Code	Measures	Formula ($d(\mathbf{x}_i, \mathbf{x}_j)$)
D1	Angular distance	$\arccos\left(\sum_{k=1}^{D}\sqrt{\frac{x_{ik}^2}{\sum x_{ik}^2}}\sqrt{\frac{x_{jk}^2}{\sum x_{jk}^2}}\right)$
D2	Bhattacharyya (arccos) (Bhattachayya, 1943)	$\arccos\left(\sum_{k=1}^{D}\sqrt{x_{ik}}\sqrt{x_{jk}}\right)$
D3	Bhattacharyya (log) (Bhattachayya, 1943)	$-\ln\left(\sum_{k=1}^{D}\sqrt{x_{ik}}\sqrt{x_{jk}}\right)$
D4	J-divergence [6]	$\left[\sum_{k=1}^{D}\ln\left(\frac{x_{ik}}{x_{jk}}\right)(x_{ik}-x_{jk})\right]^{\frac{1}{2}}$
D5	Jeffries-Matusita distance	$\left[\sum_{k=1}^{D}\left(\sqrt{x_{ik}}-\sqrt{x_{jk}}\right)\right]^{\frac{1}{2}}$

Distances for continuous variables, see Table 3.4, can also be applied to two compositions

Martín-Fernández et al. (2011) reviewed related papers on this problem from different perspectives, and concluded that there is no general methodology for the 'zero problem'. Their final remark was to propose some recent techniques to deal with some kinds of zeros: rounded, count, and essential.

i Rounding zeros: This kind of zero is mostly recorded when very small observed percentages are rounded to zero, so that the correct value is indeed not zero. Martín-Fernández et al. (2011) suggested two types of techniques in this sense:

- Non-parametric replacement, which is simply consists of replacing each rounded zero in the composition by an appropriate small value, δ_{ij}, then modifying the non-zero values in a multiplicative way, see more details in Martín-Fernández et al. (2003).
- A parametric modified EM algorithm, which is a modification of the common EM algorithm (Palarea-Albaladejo and Martín-Fernández, 2008) that replaces unobserved values by small values. The imputation of those small quantities depends conditionally on the the information included in the observed data.

ii Count zeros: This can be defined as non-occurrence of event in component(s), which is precisely recorded as zero. In this respect, Daunis-i Estadella et al. (2008) introduced the *Bayesian-multiplicative* approach which is the combination of two methodologies, of which the idea comes from Walley (1996) and Martín-Fernández et al. (2003). Martín-Fernández et al. (2003) described the process as follows:

Let $\mathbf{c}_i$ be a counts vector with D categories in a data set $\mathbf{C}$. Let T_i be the total count in $\mathbf{c}_i$ and θ_i its associated parameter vector of probabilities from a multinomial distribution. The prior distribution for θ_i is the conjugate distribution of the multinomial: a Dirichlet distribution with parameter vector α_i, where $\alpha_{ij} = s_i p_{ij}$, $j = 1, \ldots, D$. The vector $\mathbf{p}_i$ is the a priori expectation for θ_i and the scalar s_i is known as the strength of that prior.

From Bayes theorem, after one sample vector of counts $\mathbf{c}_i$ is collected, the posterior Dirichlet distribution for θ_i takes a new parameter vector α_i^*, where $\alpha_{ij}^* = c_{ij} + s_i p_{ij} = c_{ij} + \alpha_{ij}$ and the posterior estimation for θ_{ij} is

$$\widehat{\theta}_{ij} = \frac{c_{ij} + s_i p_{ij}}{\sum_{k=1}^{D}(c_{ik} + s_i p_{ik})} = \frac{c_{ij} + \alpha_{ij}}{T_i + s_i}. \tag{3.26}$$

Some common priors have been proposed for the corresponding posterior estimation of $\widehat{\theta}_{ij}$, see Table 3.6. A Bayesian multiplicative strategy is implemented by replacing each rounded zero in the composition by an appropriate small value $\widehat{\theta}_{ij}$ and then modifying the non-zero values in a multiplicative way. The following expression is suggested by (Martín-Fernández et al., 2003) to achieve this aim.

$$xr_{ij} = \begin{cases} \frac{\alpha_{ij}}{T_i + s_i} & \text{if } x_{ij} = 0, \\ x_{ij}(1 - \sum_{k|x_{ik}=0} \frac{\alpha_{ik}}{T_i + s_i}) & \text{if } x_{ij} > 0. \end{cases} \tag{3.27}$$

This approach replaces zeros with very small values, so that the results can be applicable in distances (e.g., Aitchison distance) in which a logarithmic transformation is involved. For instance, let $\mathbf{c} = (9, 0, 5)$ be a vector, and Perks prior will be selected for this example ($s_i = 1$ and $\alpha_{ij} = 1/3$), see Table 3.6, so the posterior estimate of $\mathbf{xr}$ can be seen as follows:

$$\mathbf{xr} = \left(\frac{9}{14}(1 - \frac{1/3}{15}), \frac{1/3}{15}, \frac{5}{14}(1 - \frac{1/3}{15})\right) = \left(\frac{22}{35}, \frac{1}{45}, \frac{22}{63}\right),$$

where the ratio between the first and the third components is preserved: $\frac{22}{35}/\frac{22}{63} = \frac{9}{5}$.

iii Essential zeros: The other potential and more considerable problem is the case of the total count being equal to zero; in other words, all components in one composition are truly zero. Martín-Fernández et al. (2011) specified that there is recently no general methodology for dealing with essential zeros, in spite of the fact that they provide some suggestions for different types of data, see Aitchison et al. (2003), Bacon Shone (2003), Fry et al. (2005).

Table 3.6: Proposed Dirichlet priors and corresponding posterior estimation, $\widehat{\theta}_{ij}$

Prior	s_i	α_{ij}	$\widehat{\theta}_{ij}$
Haldane	0	0	$\frac{c_{ij}}{T_i}$
Perks	1	$1/D$	$\frac{c_{ij}+(1/D)}{T_i+1}$
Jeffreys	$D/2$	$1/2$	$\frac{c_{ij}+(1/2)}{T_i+(D/2)}$
Bayes-Laplace	D	1	$\frac{c_{ij}+1}{T_i+D}$

3.4.5 Final remarks on compositional data

Aitchison et al. (2000) recommended that log-ratio analysis should be applied for compositional data, specifically in such activities as cluster analysis and MDS, but some substantial issues might be present. Jackson (1997) examined the covariance and correlation structure of log-ratio analysis and specified that the method has some problems: (1) Variables in compositional form show strong negative correlations due to closure and the implicit dependency of the variables on one another, (2) the log-ratio values are undefined in case of presence of zero values in the data. Moreover, Zier and Rehder (1998) criticised that

The distance structure is destroyed, even the ranks of the distances are not equal in S^2 and $\Re^2$, so that such activities as cluster analysis and MDS do not work properly, and the result of distances are strange due to strong dependence on the denominator.

(See the whole discussion between "*Aitchison and others*" and "*Zier and Rehder*" in Aitchison et al. (2000), Rehder and Zier (2001) and Aitchison et al. (2001)).

One of the arguments against the log-ratio transformation is that the Aitchison distance can be problematic for small proportions. For instance, let us consider two compositions with three components: $\mathbf{x}_1 = (a, s, s)$ and $\mathbf{x}_2 = (s, a, s)$, where $a \approx 1$ and $s \approx 0$, then the difference between Aitchison and Manhattan distance can be shown as follows:

$$\begin{aligned} d_a(\mathbf{x}_1, \mathbf{x}_2) &= \sqrt{\sum_{i=1}^{D} \left\{ \ln \frac{x_{1i}}{g_m(\mathbf{x}_1)} - \ln \frac{x_{2i}}{g_m(\mathbf{x}_2)} \right\}^2} \propto \ln \frac{a}{s} \qquad when \quad \lim_{s \to 0} \ln \frac{a}{s} = \infty, \\ d_m(\mathbf{x}_1, \mathbf{x}_2) &= \sum_{i=1}^{D} |x_{1i} - x_{2i}| \propto |a - s| \qquad\qquad\quad when \quad \lim_{s \to 0} |a - s| = a. \end{aligned} \tag{3.28}$$

In fact, Aitchison et al. (2000) pointed out the problem in Expression (3.28) that if one of the components of a composition tends toward zero, then the distance of that composition from others will tend toward infinity, but they claimed that this feature is not inappropriate, because in some applications, a composition with one of the parts absent may be completely different from compositions with all components positive. However, this feature may not be satisfactory in some other applications, since the Aitchison distance is dominated by differences between small percentages in an inappropriate manner, so that the resulting differences between objects may not match an appropriate "interpretative distance", see the whole discussions in Section 4.2.2.

3.5 Aggregating Mixed-Type Variables, Missing Values and Distances

The framework here is the construction of a dissimilarity measure by aggregating variables or variable-wise distances. Aggregating the same kind of variables has been discussed in the previous sections, but here I discuss how to aggregate mixed types of variables. Two different forms of aggregation are proposed here to construct a final distance measure to be used in cluster analysis: 1) Computing different distance matrices variable by variable, then aggregating those distance matrices into a single distance matrix, 2) Computing different distance matrices, in which the same type of variables are first aggregated together, and then combining the resulting distances into a single distance matrix.

A standard way of the first distance design is the Gower dissimilarity (Gower, 1971):

$$d_G(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{k=1}^{p} w_k \delta_{ijk} d_k(x_{ik}, x_{jk})}{\sum_{k=1}^{p} w_k \delta_{ijk}}, \tag{3.29}$$

where w_k is a variable weight and $\delta_{ijk} = 1$ except if x_{ik} or x_{jk} are missing, in which case $\delta_{ijk} = 0$. As a special case of Gower, the L_1 distance can be adopted for each variable, and the weights are chosen based on standardisation to $[0, 1]$-range. Mathematically speaking, d_k and w_k are given by

$$d_k(x_{ik}, x_{jk}) = |x_{ik} - x_{jk}|, \tag{3.30}$$

and $w_k = 1/R_k$, and R_k is the range of the variables, see Table 3.1. This holds when variables are interval-scaled. Equation (3.30) can be applied for ordinal variables after replacing by their ranks (Kaufman and Rousseeuw, 1990). If variables are either binary or nominal, then $w_k = 1$ unless x_{ik} is missing, and d_k is defined as

$$d_k = \begin{cases} 1 & \text{if } x_{ik} \neq x_{jk}, \\ 0 & \text{if } x_{ik} = x_{jk}. \end{cases} \tag{3.31}$$

The Gower dissimilarity is very general and can be applied to most applications of distance-based clustering for mixed types variables. However, regarding the influence on the clustering, Hennig and Liao (2013) argued that nominal variables should be weighted down against continuous variables, since many clustering methods tend to identify gaps in variable clustering distributions with cluster borders. Euclidean aggregation was used in their application, because they suggest that the differences in some variables within the social class should not be extreme, and the weights were chosen based on standardisation to unit variance of each variable, except categorical variables. The definition of Euclidean aggregation is given by

$$d_E(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{p} w_k d_k(x_{ik}, x_{jk})^2}. \tag{3.32}$$

In case of presence of missing values, the coefficients, δ_{ijk} can also be integrated in the formula. Hennig (2015a) discussed that missing values for nominal variables can also be treated as an own category, and for continuous variables one could give missing values a constant distance to every other value. Missing values are not of big concern in this report, since the data set does not include any missing values, except the variables of *"Team coefficients"* (See Section 4.1.1). More references for missing values can be found in Everitt et al. (2011).

As stated above, the second approach is to combine the distances, in which the same kind of variables are aggregated with Gower or Euclidean dissimilarity or some other distance design beforehand. This method can be employed if researchers are interested in utilising different types of distance measures for different kind of variables. The general formula is given by

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^{h} w_l d_l(x_{il}, x_{jl}), \tag{3.33}$$

where $d_l(x_{il}, x_{jl})$ is the l^{th} aggregated distance, and w_l is the l^{th} aggregated distance weight based on the selected standardisation technique. This is a version of Gower aggregation, and one could also use the Euclidean aggregation here. Then, the argument is how to choose appropriate weights for different distance measures. Gower suggested standardisation to $[0, 1]$-range, whereas Hennig and Liao (2013) applied standardisation to unit variance for w_l. Here, one could argue from the distribution shape of distances in order to find appropriate weights. For example, if the distances

are pretty much uniformly distributed, then range standardisation can be applied to the distances because of the non-existence of the extreme values in distance measures, whereas standardisation to unit variance can be chosen for normal-shaped distributions of the distances, since the statistic that characterizes the spread of normal distribution is the variance (or the standard deviation).

The idea can be discussed from another perspective. For instance, researchers construct similarity coefficients for binary categorical variables. Standard deviation is not inherently meaningful in this case, because only certain values can occur. However, if the data contains a categorical variable of few categories and continuous variables, and the distances in categorical variable can take only few values, then range standardisation may construct the biggest possible gap between the different categories. When aggregating them with continuous variables, cluster analysis methods are very prone to define the clusters that coincide with certain values of categorical variables. Therefore, it is advisable for investigators to consider all these arguments when deciding of how to combine the distances from different types.

The weights can also be chosen in a subjective way by formalising variable importance, see Kaufman and Rousseeuw (1990) and Hennig (2015a), but the weights for standardisation should be taken into account before assigning appropriate subjective weights, because the distances or the variables should be comparable. One of the aspects of choosing subjective weights is whether the scale of weights will change the clustering structure. The idea behind this argument is *equivariance* [7] and *invariance* [8] characteristics of statistical methods. To be more precise, the question is "Are the cluster analysis and MDS invariant against multiplying all distance matrices by a constant, c?". For example, when running hierarchical clustering, two conditions can be specified: 1) Number of clusters, 2) Height of dendrogram. If we fix the height of the dendrogram that we cut, the clustering will not be invariant against multiplying all distance matrices by a constant, because both the height and the cutting position will change; hence the algorithm is invariant, otherwise it is equivariant. On the other hand, when running K-means, multiplying all the data by the same constant is not going to change the clustering result, so that K-means is invariant under scaling by the same constant, whereas means are equivariant. By and large, researchers who are interested in assigning some subjective weights for the variables or the distances should consider these two contexts for the decision of how multiplying every input (e.g., data or distance matrices) by the same constant changes the results.

[7] A function f is said to be equivariant under the transformation group G with domain X if for all $x \in X$ and $g \in G$, then $f(g(x)) = g(f(x))$.

[8] A function f is said to be invariant under the transformation group G with domain X if for all $x \in X$ and $g \in G$, then $f(g(x)) = f(x)$

3.6 Summary

Different data pre-processing steps are discussed in terms of their influences on clustering. Then, the theory of dissimilarity measure is introduced, and different measures of similarity or dissimilarity are examined for different kinds of variables. Compositional data was reviewed, since the data to be used in this report contains compositional information. In the final section, I discussed how to aggregate mixed-type variables, missing values and distances.

In Section 3.1, seven steps are summarised for the strategy of cluster analysis, and the first four steps are explained in detail here, and the remaining steps will be reviewed in Chapter 5.

CHAPTER 4

DISTANCE CONSTRUCTION OF FOOTBALL PLAYER PERFORMANCE DATA

I have provided some background information on football, football in statistics and sports in cluster analysis in Chapter 1. Data description was presented in Chapter 2 in order to understand the data properly, while Chapter 3 has covered the relevant methodologies and literature to be used in this chapter, where all such analysis related to distance construction of football players performance data set will be presented in detail.

4.1 Variable Pre-processing

In this section, four different steps are followed through for different types of variables 1) representation: i.e., considerations regarding how the relevant information is most appropriately represented, 2) transformation: Football knowledge as well as the skewness of the distribution of some count variables indicates that transformation should be considered decreasing the effective distance between higher values compared to the distances between lower values, 3) standardisation: In order to make within-variable variations comparable between variables), and 4) variable weighting.

4.1.1 Representation

Profile variables

Seven profile variables are introduced in Section 2.1.1. League variable, $x_{(l)}$ will be represented as it is, whereas two team variables, $x_{(tp)}$ and $x_{(tc)}$ have quite different representations. $x_{(tp)}$ variable is the information of team points from the ranking table of the 2014-2015 football season. Here points are represented in terms of per game performance of teams by standardising by the total number of games, because there have been different number of teams from each season, hence each team plays different number of games. For example, *Atletico Madrid* from *La Liga* collected 78 points, which is 2.05 ($= 78/38$) in per game representation, while *CSKA Moscow* from *Russian Premier League* had 60 points, which is 2.00 ($= 60/30$) in per game representation in the 2014-2015 football season. Therefore, per game representation is applied for $x_{(tp)}$ for the sake of reflecting players' team information in a better way.

The $x_{(tc)}$ variable is the information of team coefficients based on the results of clubs competing in the five previous seasons of the UEFA Champions League and UEFA Europa League, see more information in https://www.uefa.com/memberassociations/uefarankings/club/index.html. This information can be interpreted as a combination of $x_{(l)}$ and $x_{(tp)}$ variables based on the games which took place in the competitions between European clubs. $x_{(tc)}$ has some missing values since some clubs have not qualified for any competitions in Europe. For the solution, I propose that $x_{(l)}$ and $x_{(tp)}$ variables should be up-weighted by the assigned weight of $x_{(tc)}$ variable in case of existence of missing values in the $x_{(tc)}$ variable. For instance, *Getafe* from *La Liga* has no information in $x_{(tc)}$ variable, so that no distance information exists between a player from any teams and a player from *Getafe* for $x_{(tc)}$ variable. In this case, the relevant distance information will be depending only on $x_{(l)}$ and $x_{(tp)}$ variables. Mathematical definition of the weight assignment for the team and league variables will be provided in Section 4.2.3.

Another key issue is that some players have played in different teams in the 2014-2015 football season (e.g., Player X transferred from Team A from Team B in the same season). Here my suggestion is to average $x_{(l)}$, $x_{(tp)}$ and $x_{(tc)}$ variables based on weights with respect to number of minutes that players played for the different teams. Table 4.1 is a demonstration of how this calculation is made.

After carrying out the representation step as explained above, the summary of profile variables can be seen in Figure 4.1. The histogram of the $x_{(l)}$ variable contains multiple peaks which represent league information of players from eight different leagues, and very short bars represent the players who played in different leagues. The team variables are distributed more evenly than the

Table 4.1: An example of league and team scores representation for a player who played in multiple teams, where the i subscript represents the i^{th} player, and the j subscript represents the j^{th} team information.

Player	Juan Cuadrado		**Representation**		
			League score	*Team point*	*Team coef.*
League	England	Italy	$\frac{\sum_j x_{ij}m_{ij}}{\sum_j m_{ij}}$	$\frac{\sum_j y_{ij}m_{ij}}{\sum_j m_{ij}}$	$\frac{\sum_j z_{ij}m_{ij}}{\sum_j m_{ij}}$
Team	Chelsea	Fiorentina			
League score (x_{ij})	80.391	70.510	71.538	1.742	58.775
Team point (y_{ij})	2.28	1.68			
Team coef. (z_{ij})	142.078	49.102			
Minutes (m_{ij})	198	1705			

league variable, because each league has different number of teams; hence, variations are expected to be larger. As clarified above, the $x_{(tc)}$ variable includes some missing values, and Figure 4.1c shows that approximately one-third of the observations are missing. To put it in another way, over the last five years one-third of the clubs have not qualified for any European competition. For the other profile variables (Age, Height, Weight), they will be taken as they are. The summary of these variables is shown in Figure 4.1.

Position variables

Position variables of two kinds have been introduced in Section 2.1.2. Here all binary variables ($Y_{(11)}$) will be represented as they are, but the 15 count variables ($Y_{(15)}$) will be represented quite differently. In Section 3.4, I clarified that relevant count variables can be represented in compositional forms. The information of how many times a player played in different positions can be interpreted as relative information, since all these counts reflect the characterisation of where players are located on the field, hence the $Y_{(15)}$ variables are represented in proportional forms, see the general definition below for the calculation of $Y_{(15)}$, where $\mathbf{z} = Y_{(15)}$.

Definition 4.1.1. Let $\mathbf{z}$ be a (row) vector $\mathbf{z} = [z_1, z_2, \ldots, z_D]$, and $z_i \geq 0 \ \forall i = 1, 2, \ldots, D$. Proportional forms of $\mathbf{z}$ can be then shown as

$$C(\mathbf{z}) = \left[\frac{z_1}{\sum_{i=1}^{D} z_i}, \frac{z_2}{\sum_{i=1}^{D} z_i}, \ldots, \frac{z_D}{\sum_{i=1}^{D} z_i} \right]. \tag{4.1}$$

As an example, a player appeared in 50 games during the season, and his appearances are 20 games in centre back (DC), 20 games in left back (DL), 10 games in right back (DR), so the proportions will be 0.4, 0.4, 0.2 for DC, DL, DR, respectively.

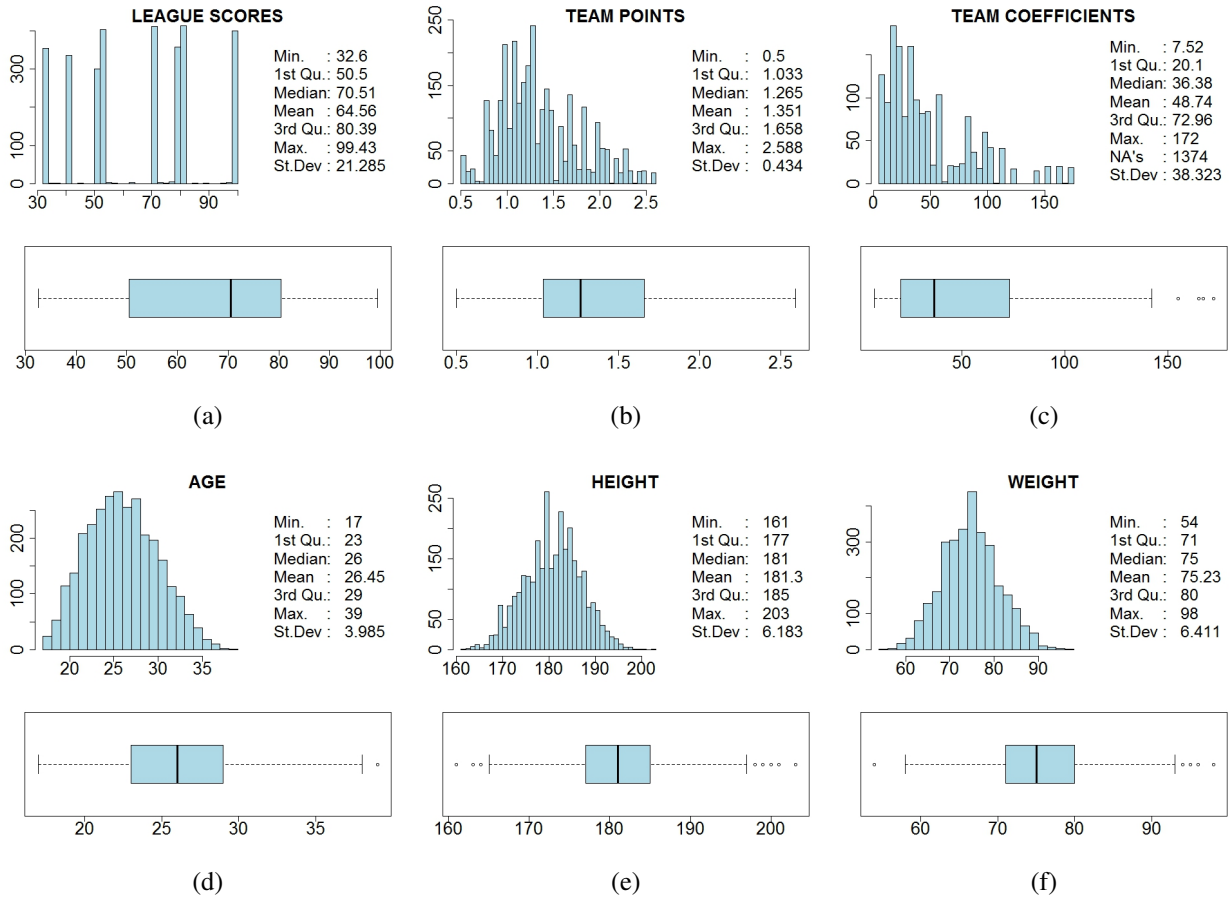


Figure 4.1: Summary of profile variables

One of the issues for $Y_{(15)}$ variables is that there is no information available in the data set for a player who has not started in the first eleven in any game during the season. As a solution of this problem, I propose a similar idea to that which is $x_{(tc)}$ variable in Section 4.1.1, namely that the value pertaining to $Y_{(11)}$ will be used for $Y_{(15)}$ when the latter has zero counts, see Equation (4.2).

$$C(Y_{(15)}) = \begin{cases} C(Y_{(15)}) & \text{if} \quad \sum_{i=1;z_i \in Y_{(15)}}^{D} z_i = 0, \\ C(Y_{(11)}) & \text{Otherwise.} \end{cases} \tag{4.2}$$

For instance, a player has not played from the beginning of any matches during the 2014-2015 football season, but he has actually played in DC, M(CLR), FW positions based on the information of $Y_{(11)}$ variables. Then, the assignment for the player in $Y_{(15)}$ variables will be 0.2 each for the positions, DC, MC, ML, MR, FW, see Table 2.2. This is the issue of essential zeros which was discussed in Section 3.4.4.

The bar plot for two types of positions are shown in Figure 4.2. The first graph reflects the

information about the sum of the components from each variable of $Y_{(15)}$, where decimal numbers are rounded down to integer numbers, whereas the second graph is based on the sum of binary values from each variable in $Y_{(11)}$. Apparently, the total number in the second graph are larger than the first one in spite of the existence of more variables, because $Y_{(15)}$ variables are composition variables, so that the position-wise per player sum should be 1, while $Y_{(11)}$ variables are all binary, and players can play in multiple positions, so that the summation of the rows can be either 1 or more.

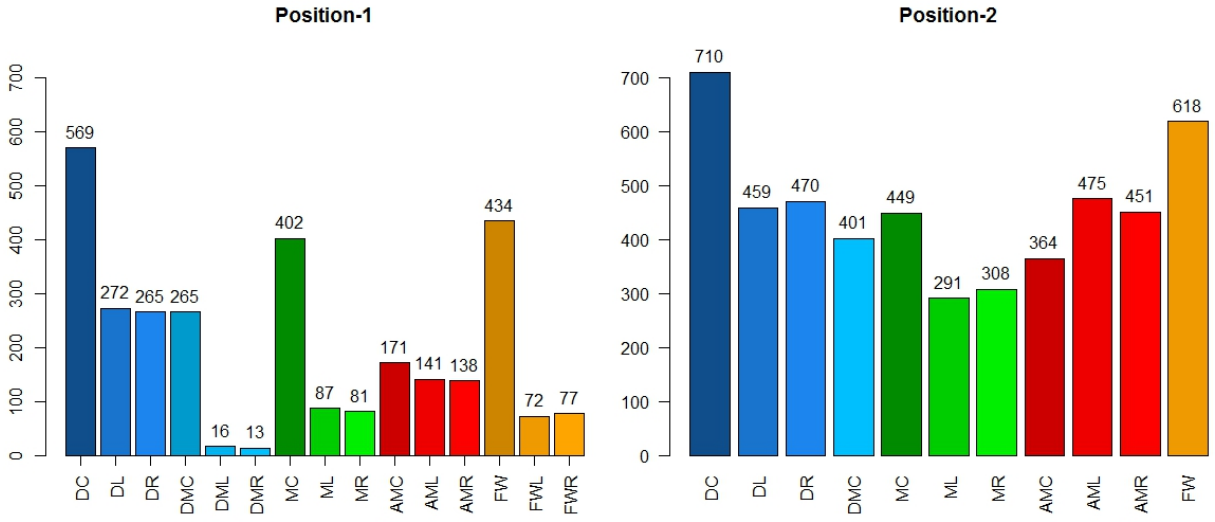
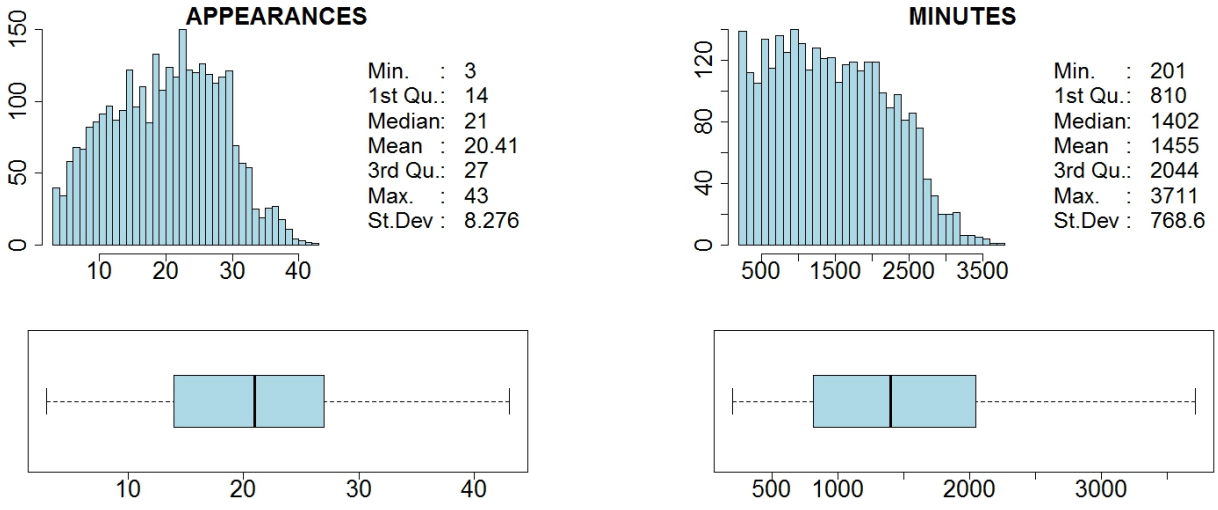


Figure 4.2: Frequencies of position variables, $Y_{(15)}$ and $Y_{(11)}$, respectively.

Performance variables

Performance variables are classified into five categories, see Section 2.1.3. Time variables (Appearances, Minutes) are used as they are, and the summary of these variables is shown in Table 4.3. I decided not to use the *Subjective variables*, see Table 2.3, because *Rating* is designed as a combination of all the performance variables in the data set, see its definition in Table 4.3, and *Man of the match* is chosen based on the maximum rating score that a player had in one match. Thus, it would not be legitimate to use them, and this is a decision about what meaning the results should have.

The other types of performance variables, which are all counts, can be classified into two different categories: a) counts of actions (upper level), b) the compositions of actions (lower level).



(a) (b)

Figure 4.3: Summary of time variables

Upper level count variables

A representation issue here is that counts of actions such as shots, blocks etc., should be used relative to the period that the player played in order to characterise players in an appropriate way. Here the counts of actions which contain sub-variables, such as shots, passes, etc., are created by summing up the sub-variables of the relevant actions (e.g., $SHOT = Shot_{RF} + Shot_{LF} + Shot_{Head} + Shot_{Other}$).

Two representation options can be examined by standardising the upper level count variables: a) Per 90 minutes, which is the representation of the upper level count variables standardised by the period of time that the player played, b) per game, which is the representation of the upper level count variables standardised by the number of games that the player played. The question is which one better represents the player's characteristic. For example, two players played 60 and 90 minutes in each game on average, and assume that they play the same number of games. If per game representation is applied, then their values will be divided by the same number, but this will be inappropriate because they played for different time periods. Hence, since a game of football lasts for 90 minutes, I represent the counts as "per 90 minutes":

$$y_{ij} = \frac{x_{ij}}{m_i/90} = 90 \times \frac{x_{ij}}{m_i}, \tag{4.3}$$

where x_{ij} is the j^{th} count variable of player i, m_i is the number of minutes played by player i. Figure 4.4 shows the summary of upper level count variables represented as per 90 minutes. The

summary of the variables shows the distributional shape of count data, such as right skewed distributions. Some of these variables (e.g., goals, assists, offsides) have lots of zero values, whereas some others (e.g., tackles, passes, etc.) are distributed more evenly than the zero-dominated ones.

Nonetheless, another problem arises in terms of representation to 90 minutes for the players who played in a short period of time during the season. For example, a player only played 7 minutes and scored one goal during the season. Such a player exists in the data. In accordance with representation to 90 minutes, he scores approximately 12.5 goals in each game, which is not a realistic representation based on our football knowledge. Thus, I only aim to analyse players who have played a minimum 200 minutes during the season, which is chosen by my intuitive sense in terms of reflecting a proper representation of players. In this respect, 149 players are removed from the data set, hence 3003 players will be used for the further analysis.

Lower level count variables

Suppose that a player has 2.0 shots per 90 minutes, and the shots per zone are out of box: 0.4, penalty area: 1.3, six yard box: 0.3. When computing the distance between this player and another player, two different aspects of the players' characteristics are captured in these data, namely how often each player shoots, and how the shots distribute over the zones. If the data were used in the raw form given above, players with a big difference in the upper level variable "shots" would also differ strongly regarding the lower level variable "shot zone", and the overall distance would be dominated by the upper level variable with the information on the zonal distribution being largely lost. In order to separate the different aspects of interest, the lower level count variables are transformed to percentages, i.e., 0.2, 0.65 and 0.15 for out of box, penalty area, six yard box above, whereas the upper level count is taken as per 90 minutes count as defined above.

Percentage variables can be represented as proportion of total and/or success rates. For example, shot and goal are upper level count variables that contain common sub-categories (zone, situation, body part). Goal is essentially the successful completion of a shot, so that the sub-variables of goal can be treated as success rate of shot in the respective category as well as composition of total goals. Both are of interest for characterising the players in different ways, and therefore I will use both representations in some cases. Table 4.2 shows where this was applied.

In Section $2.1.3$, the discussion of the *'Other'* sub-variables was with respect to their meaning, and I stated that the '*other*' categories from body parts of shots and goals are specified, whereas the '*other*' categories from key passes and assists are unspecified. In fact, the statistical summary of these sub-variables, see Figure 4.5, specifically in terms of their frequencies, give us an idea of why these variables should be considered in different ways. Figure 4.5a and 4.5b shows a peak around zero percentages, because the other body parts hardly occur when shooting in football. However,

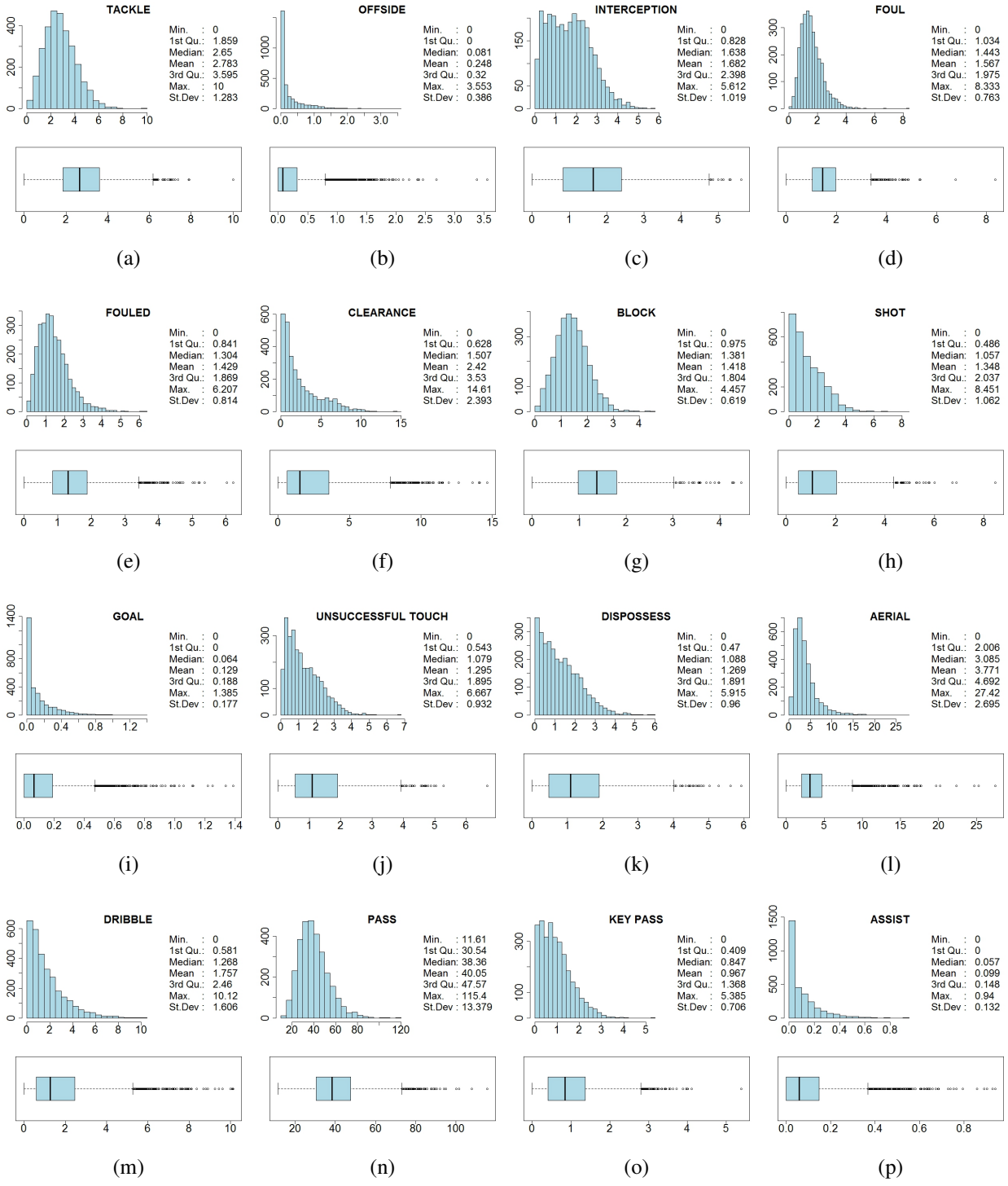
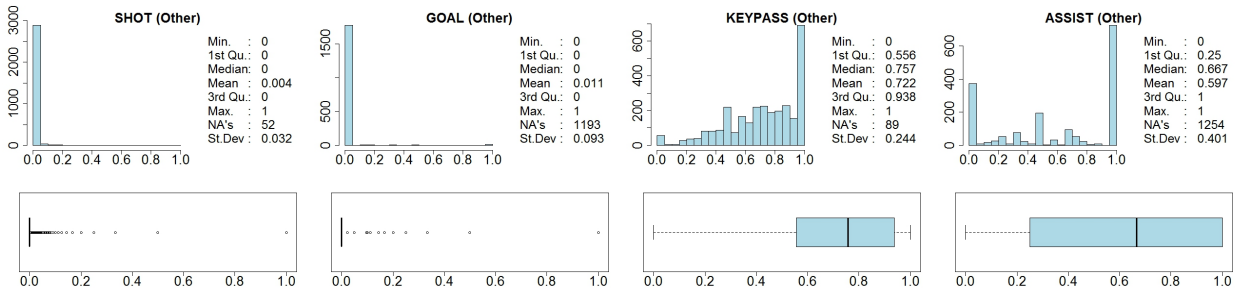


Figure 4.4: Summary of upper level count variables represented as per 90 *minutes*



(a) (b) (c) (d)

Figure 4.5: Summary of the 'Other' percentage variables

Figure 4.5c and 4.5d shows more or less left skewed shaped distributions, since a large number of key passes and assists are uncategorised. The information of all the other categories exists in the percentages of the overall of non-other categories regardless of their meaning, because of the argument of linear dependence in compositional data. As a part of the representation step, I consider excluding only the unspecified other categories, because they do not have any meaning in terms of reflecting player's characteristic. On the other hand, even if the unspecified other categories are excluded, it is very important to use them to compute the percentages, because otherwise the information of the actual percentages is lost. Therefore, the specified other categories will be used both for computing the percentages and for the distances, whereas the unspecified other categories will be used only for computing the percentages, not for computing the distances.

Table 4.2: Representation of lower level count variables

Variables (Include sub-categories)	**Proportion of total (standardised by)**	**Success rate (standardised by)**
Block	Total blocks	✗
Tackle, Aerial, Dribble	✗	Total tackles, total aerials, and total dribbles
Shot (4 sub-categories)	Total shots	✗
Goal (4 sub-categories)	Total goals	Shot count in different sub-categories, and total shots and shots on target for overall success rate
Pass (2 sub-categories)	Total passes	Pass count in different sub-categories, and total passes for overall accurate pass rate
Key pass (2 sub-categories)	Total key passes	✗
Assist	Total assists	Key pass count in different sub-categories, and total key passes for overall success rate

Dealing with zero values

In the previous section, my conclusion was to represent the lower level count variables in compositional form. For log-ratio analysis of compositional data, one of the major issues is zero components, as logarithms of zero values are undefined, see Section 3.4.4. To discuss this problem, any distance measures associated with logarithmic transformation should be reviewed in case of the presence of zero values, but I have not explained yet which distance measure is to be used for compositional data. However, as a reminder, the ultimate rationale behind what it to be here is that variables will be represented in such a way that the results should match how the variables are interpreted in the application of interest; for this reason, zero components can also be problematic in respect to characterising football players information. For example, consider two players who make different total shots per 90 minutes, where the first one has 10 total shots with 0 accuracy, while the second one has 1 total shot with 0 accuracy. If there is no change, the accuracy rate for both players will be 0%, but if the second player had had more shot opportunities, his accuracy rate could have been larger.

For managing the ratio preservation for non-zero values, Equation (3.27) was introduced as a *Bayesian-multiplicative* approach in Section 3.4.4, and some common priors had been proposed for corresponding posterior estimation, $\widehat{\theta}_{ij}$, see Table 3.6. Here, the Bayesian approach can be a tool to represent proportional variables based on subject matter reasons in terms of the application of interest. It is important to note that the Bayesian approach is only used for motivation of how to adjust these values, not for the sake of logarithmic transformation. I suggest a different prior to those in Table 3.6, because those priors for the estimation of zero components do not interpretatively meet my expectations. Table 4.3 provides an example to compare those priors and my suggested prior, which is based on the success rate of how many goals have been scored out of total shot attempts.

As introduced in Section 3.4.4, s_i is the strength of the prior, which I choose as 1, and p_{ij} is the prior expectation, which is actually more important than the strength parameter in my cases, because it determines the expectation of the estimated proportion. One of the preferred expected value is the mean, and the mean of the proportional variables can be used for finding a proper p_{ij}, so that the posterior estimation, $\widehat{\theta}_{ij}$, which will be the representation of the proportional variables by using my prior selection, should make more sense football-wise than adopting the other proposed priors. For instance, the 3^{rd} player on Table 4.3 brings very high success rate by using other methods with different priors. This is not properly accurate, since goals rarely occurs (e.g., see the histogram in Figure 4.4). Likewise, the success rate of the 3^{rd} player is 0.04 by using my priors, and the 1^{st} and the 2^{nd} estimated percentages are 0.0008 and 0.007, respectively, which are very different than the 3^{rd} one. This is essentially a reasonable prediction, because the number of shots

are very different. Even if the other priors also provide a big difference between those players, they do not accurately reflect the percentage information for the 3^{rd} player. The 4^{th} and the 5^{th} players are provided in order not only to see the comparison with the first three players, but also to show that the estimation is only used for zero components, otherwise it remains as constant, see Equation (3.27), unless one of the other components are non-zero.

Table 4.3: Comparison between the priors in Table 3.6 and my suggested priors, where c_{ij} is the j^{th} count variable of player i, and T_i is the total count of the j^{th} variable, see Equation (3.26)

Players	**Shots**	**Goals**	**Goal success rate ($\widehat{\theta}_{ij}$)**				
			None ($s_i = 0$, $\alpha_{ij} = 0$)	***Jeffreys*** ($s_i = D/2$, $\alpha_{ij} = 1/2$)	***Perks*** ($s_i = 1$, $\alpha_{ij} = 1/D$)	***Laplace*** ($s_i = D$, $\alpha_{ij} = 1$)	***My prior selection*** ($s_i = 1$, $\alpha_{ij} = \frac{1}{n}\sum_{i=1}^{n} c_{ij}/T_i$)
1	100	0	0.0000	0.0050	0.0005	0.0090	0.0008
2	10	0	0.0000	0.0450	0.0450	0.0830	0.0070
3	1	0	0.0000	0.2500	0.2500	0.5000	0.0400
4	100	1	0.0100	0.0100	0.0100	0.0100	0.0100
5	10	1	0.1000	0.1000	0.1000	0.1000	0.1000

The other potential and more considerable problem is the case of total count being equal to zero, which was introduced as *essential zeros* in Section 3.4.4. In this situation, there is no such evidence of how players perform in the relevant action; hence, no such prediction can be adopted for the relevant composition. Assigning zero values for each proportion may lead to incorrect interpretation, since I do not know what performances these types of players have displayed. Thus, when computing a distance measure, I consider only using the total count and ignoring compositional information for the players who have essential zeros in the related action; in other words, a composition in which all components are zero will be weighted as zero, and total count of the relevant action will be up-weighted by the total weights of the relevant composition. The following equation demonstrates the computation of distance measure for the players whose total upper level counts are zero:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} d(x_{iT}, x_{jT}) \sum_{t=1}^{D} w_t & \text{if } \sum_{t=1}^{D} x_{it} = 0 \quad \forall x_{it}, \\ d(x_{it} w_t, x_{jt} w_t) & \text{otherwise,} \end{cases} \tag{4.4}$$

where x_{iT} is the T^{th} upper level count variable of player i, x_{it} is the t^{th} lower level percentage variable of player i, and w_k is the subjective weight of the k^{th} lower level percentage variable. The choice of subjective weights for each variable will be discussed in Section 4.1.4.

4.1.2 Transformation

Variables are not always related to "interpretative distance" in a linear way, and a transformation should be applied in order to match interpretative distances with the effective differences of the transformed variables.

The upper level count variables have more or less skewed distributions, see Figure 4.4; for example, many players, particularly defenders, shoot very rarely during a game, and a few forward players may be responsible for the majority of shots. On the other hand, most blocks come from a few defenders, whereas most players block rarely. This means that there may be large absolute differences between players that shoot or block often, whereas differences at the low end will be low; but the interpretative distance between two players with large but fairly different numbers of blocks and shots is not that large, compared with the difference between, for example, a player who never shoots and one who occasionally but rarely shoots.

This suggests a non-linear concave transformation such as logarithm or square root for these variables, which effectively shrinks the difference between large values relative to the difference between smaller values, see Figure 4.6. For instance, the difference between players who shoot 25 and 20, and the difference between players who shoot 5 and 3 per 90 minutes might be more or less the same, after a choice of transformations is applied.

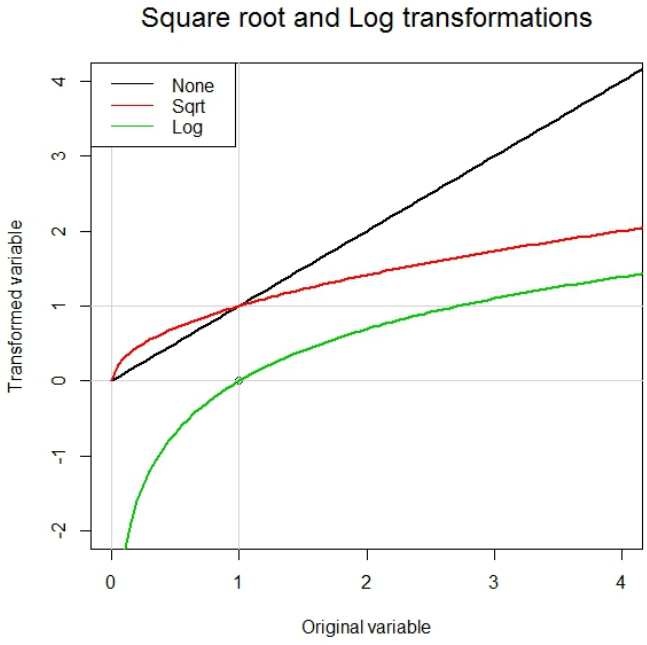


Figure 4.6: Comparison between non-linear concave transformations and no transformation

Prior to selecting an appropriate transformation technique, I consider adding a constant value, 'c', since the upper level count variables contain zero values, and the logarithmic transformation can only be applicable for non-zero positive values. For square root transformation, an additional constant value is not essential, since $\sqrt{0}$ is defined. However, I consider adding a constant value in order to examine different variations of square root transformation. The question is then which 'c' should be used for the selected transformation?

Hennig and Liao (2013) discussed how the resulting differences for logarithmic transformation are affected by different constants based on the subject of social stratification. The argument of the choice of 'c' depends on the relationship between *income* and *saving amounts* variables in terms of sociological interpretation. Here, since each count variable has different statistical information, the choice of 'c' should vary for different upper level count variables. For instance, Figure 4.7 shows how the locations of values vary by taking square root or logarithmic transformation with different constants. For this example, shot variables based on per 90 minutes representation, after adopting standardisation to unit variance, are selected. Standardisation was applied in order to make different transformation techniques comparable. 9 popular players based on the information of team from the year-2015 on the UEFA website (http://en.toty.uefa.com/) are selected for this experiment. Here the aim is actually to show how the distance measures between players are affected after taking logarithmic or square root transformation with different constant values. It seems that the difference between larger values are more affected than the difference between smaller values, which essentially justifies my belief. In addition, the logarithmic transformation has stronger influence on larger values than the square root transformation.

Although Figure 4.7 gives us a reference for the choice of 'c', I still want to make some kind of formal decision, which can be made by looking at the shape of the distributions. One could use variance-stabilizing transformation, which typically transform a Poisson distributed variable (e.g., upper level count variable) into one with an approximately standard Gaussian distribution. One of the famous variance-stabilizing transformations is the Anscombe transform (Anscombe, 1948) that is usually used to pre-process the data in order to make the standard deviation approximately constant. However, it is not clear what this has to do with football and the meaning of the data. The data can play a role for such decisions, but there is no guarantee that any decisions based on the data alone match with interpretative distance, unless interpretative distance is governed by the data. Therefore, I suggest that such decisions can be made by using some external data, which can be player's information from the previous year. In this sense, I can check whether the variation for one variable within the player one year to another is increasing with the result that the same player has on this variable either over both years.

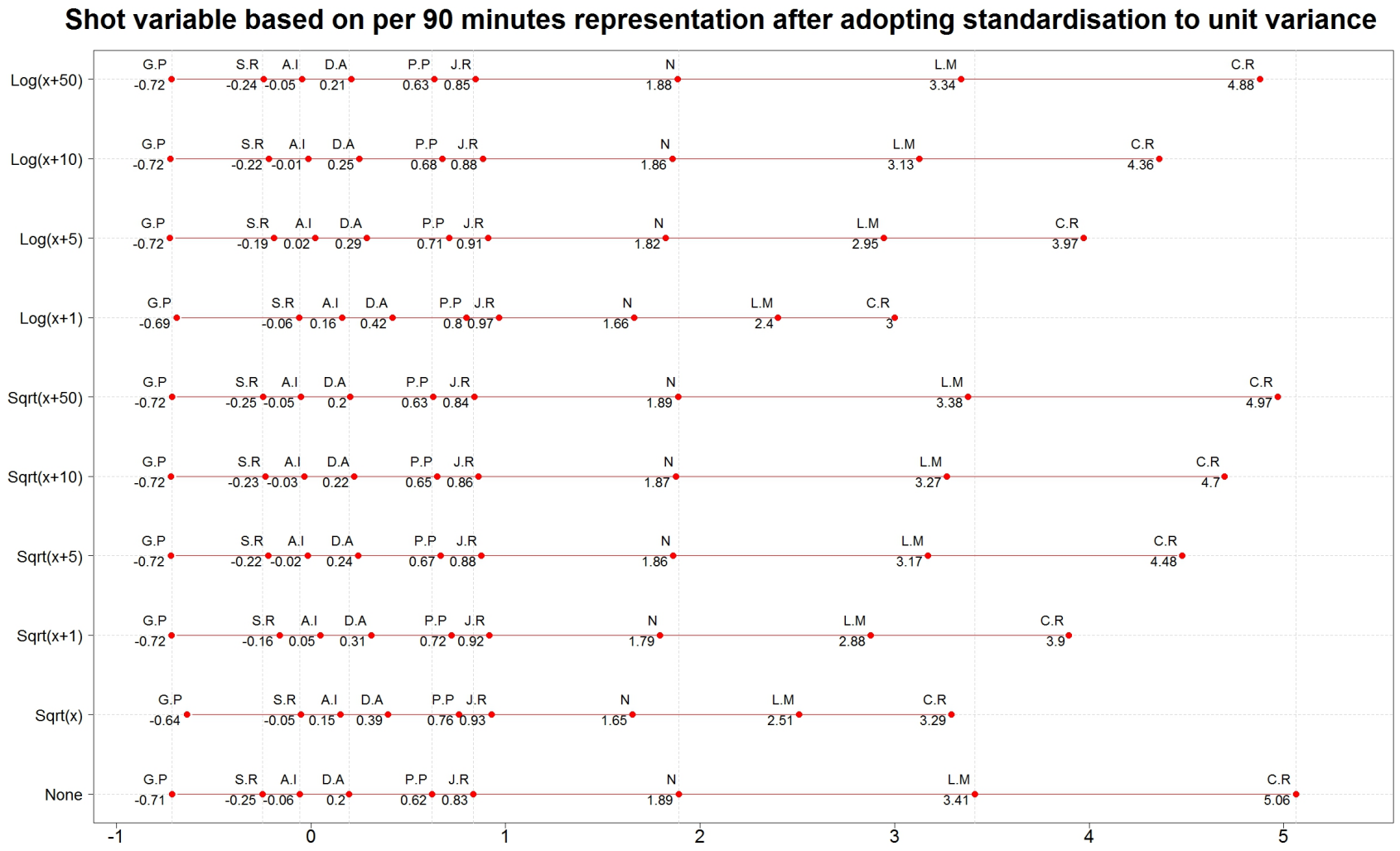


Figure 4.7: Comparison of transformations with different constants

D.A.: David Alaba, S.R.: Sergio Ramos , G.P.:Gerard Pique, A.I.: Andres Iniesta, P.P.: Paul Pogba, J.R.: James Rodriguez, L.M.: Lionel Messi, C.R.:Cristiano Ronaldo, N.: Neymar

The idea of the comparison between two year values is that the differences are approximated for the variation, because the only way of estimating variation from these two values is to look at the difference. Here, such transformation can be made in such a way that the variation does not depend on any of these values, so that I want to keep the variation constant in every place on the scale. Mathematically speaking, the slope of the regression, where a dependent variable (y) is defined as the variation of two year values and an independent variable (x) is characterised by a weighted average of two year values, is as close to zero as I can have. The theory of measurements suggests that the the distance computation for count variables are usually treated as absolute scale, so that the dependent variable, which represents the variation between the values in two years, can be characterised by the absolute difference. The weights for computing the independent variable are determined by number of minutes that a player played in two seasons, since the players information are better reflected by representation to per 90 minutes, see Section 4.1.1.

Two sample data sets from the season 2013-2014 and 2014-2015 have been used for this analysis. Same players are considered (paired data set), so that the sample sizes are equal for the two data sets ($50 \leq n \leq 100$). Note that I only used a sub-sample of the two data sets, because I was confronted with some computational difficulties when collecting all the players information from the 2013-2014 football season. Players are randomly chosen for each variable from the data set to be used in this application, and the sample sizes differ for different count variables, since some players' information is not available in the 2013-2014 football season. The following expressions are the demonstration of how the variables are assigned.

$$\begin{aligned} y_{ij} &= |x_{ij}^{(34)} - x_{ij}^{(45)}| \\ x_{ij} &= (m_{ij}^{(34)} x_{ij}^{(34)} + m_{ij}^{(45)} x_{ij}^{(45)})/(m_{ij}^{(34)} + m_{ij}^{(45)}) \\ x_{ij} &\geq 0, y_{ij} \geq 0 \quad \forall i\text{'s and } j\text{'s} \\ x_j &= (x_{1j}, \ldots, x_{nj}) \text{ and } y_j = (y_{1j}, \ldots, y_{nj}) \ (i = 1, \ldots, n \ and \ j = 1, \ldots, p), \end{aligned} \tag{4.5}$$

where $x_{ij}^{(34)}$ and $x_{ij}^{(45)}$ are the j^{th} variable on the i^{th} object of the season 2013-2014 and 2014-2015, respectively. $m_{ij}^{(34)}$ and $m_{ij}^{(45)}$ are the number of minutes from the j^{th} variable on the i^{th} object on the season 2013-2014 and 2014-2015, respectively.

I now check how the slope of linear regression of the absolute difference (y_j) regressed on the weighted average of the two data sets (x_j), where a monotonic transformation with an additional constant, 'c', is applied on each variable. If one of the p-values for the slope without applying any transformations is less than the critical value, this provides evidence that the variable should not be transformed; otherwise, I consider transforming the variable in a monotone way, such as loga-

rithmic or square root transformation. The selected transformation will be applied to each variable ($x_{ij}^{(34)}$ or $x_{ij}^{(45)}$), because the variation should be checked after transforming variables, whereas the weighted average of the two year values should be transformed as whole, not separately, because this new variable is considered as independent from $x_{ij}^{(34)}$ and $x_{ij}^{(45)}$. The following expressions show the calculation of the transformations.

$$\begin{aligned} y_{ij}^{f} &= |f(x_{ij}^{(34)}) - f(x_{ij}^{(45)})| \\ x_{ij}^{f} &= f((m_{ij}^{(34)} x_{ij}^{(34)} + m_{ij}^{(45)} x_{ij}^{(45)})/(m_{ij}^{(34)} + m_{ij}^{(45)})) \end{aligned} \tag{4.6}$$

$$f(x) = \begin{cases} \log(x + c) & \text{if logarithmic transformation applied,} \\ \sqrt{x + c} & \text{if square root transformation applied.} \end{cases} \tag{4.7}$$

The aim is to make the slopes approximately equal to zero. Finding an optimal constant value by minimising the slope function can be the solution of this problem, but I could not find any simple form of this approach, even if I use approximation methods, such as Taylor approximation. In this sense, one of the root-finding techniques, *The Bisection Method (interval halving method)*, see Algorithm 1, will be used for finding an optimal constant value for each variable. The method was first introduced as *The Intermediate Value Theorem* by Bernard Bolzano in 1817, see Edwards (2012). The design as shown in Algorithm 1 was then introduced by Burden and Faires (1985).

This method was applied on each upper level count variable, except *'Goal'* and *'Assist'*, because these variables are linearly related with football. The linear relationship can be explained as follows. *'Goal'* and *'Assist'* variables have a direct impact on match results, so that the differences of players on these variables should explicitly be reflected based on the idea of the interpretative distance. To explain the idea in terms of *'Goal'* and *'Assist'* variables, I can interpret that football is governed by goals, and assist is a type of pass that directly leads to a goal, therefore these variables should not be transformed because of their direct impact on match scores. Thus, no such transformations are applied to *'Goal'* and *'Assist'* variables.

The rest of the upper level count variables are transformed based on the result of The Bisection algorithm. Figure 4.8 displays 3 graphs for each variable. The graphs on top provide scatter plots of the untransformed variables, where y-axis are y_{ij}'s and x-axis are x_{ij}'s, see Equations 4.5. The graphs on bottom left and right indicate the best fit constant values for logarithmic and square root transformations, after the algorithm is applied. The y-axis are the slopes for different constant values, whereas x-axis are the constant values. Apparently, the roots for constant values were mostly found for logarithmic transformations, while square root transformation failed in many

Algorithm 1: The Bisection Algorithm

Data: Function f, endpoint values a, b, tolerance t, maximum iterations N_{max}
Result: Value which differs from a root of $f(x) = 0$ by less than t
Condition: $a < b$, either $f(a) < 0$ and $f(b) > 0$ or $f(a) > 0$ and $f(b) < 0$

$N \leftarrow 1$
while $N \leq N_{max}$ **do**
    # limit iterations to prevent infinite loop
    $c \leftarrow (a + b)/2$ *# new midpoint*
    if $f(c) = 0$ *or* $(b–a)/2 < t$ **then**
        # solution found
        `Output(c)`
        Stop
    $N \leftarrow N + 1$ *#increment step counter*
    if $sign(f(c)) = sign(f(a))$ **then**
        $a \leftarrow c$
    else
        $b \leftarrow c$ *# new interval*
`Output("Method failed")` *# max number of steps exceeded*

cases. However, the root finding for *'Pass'* and *'Offside'* variables was not successful, but the minimum absolute values of slopes on these variables can be assumed as optimal constants, because the aim is to make the slopes to be approximately equal to zero, so that the variation can also be minimised in this respect. For the *'Pass'* variable, the line is an increasing continuous function, see Figure 4.8n and the slope can only be minimum when the constant is zero, but constants cannot be zero in logarithmic transformation; hence, the smallest possible value is chosen in this respect, which is $\epsilon = 0.0001$. For the *'Offside'* variable, Figure 4.8m shows that the estimated slope can be found when the constant is around 0.006, because the p-value for the slope between y_{ij}^{log} versus $x_{ij}^{log}$is the maximum in this case.

Table 4.4 provides the p-values for the null hypothesis: "No slope", $\mathbf{p_1}$, prior to transformation and the optimal constant values, $\mathbf{c_{log}}$ and $\mathbf{c_{sqrt}}$, and the p-values, $\mathbf{p_2}$, after an appropriate transformation with the final constant to be used, which should be greater than the critical values. Note that the $\mathbf{p_2}$ values are no longer valid p-values, because they are based on optimisation results. As stated above, if one of the p-values of the slope between y_j and x_j is greater than the critical value, no such transformation is necessary. $\mathbf{p_1}$'s for *Foul* and *Key Pass* are greater than 0.1, whereas the p-

value for *Block* is greater than 0.05, which suggest that these variables should not be transformed, but since the p-values are very close to the critical values and the variation range of the values are better unified after applying logarithmic transformation with the constants, $\mathbf{c_{fin}}$, in Table 4.4, my decision is to transform these variables as well. As a result, to be consistent I choose logarithmic transformation for all upper level count variables with the constant values, $\mathbf{c_{fin}}$, shown in Table 4.4. Figure 4.9 provides a summary of the transformed upper level count variables.

Table 4.4: Optimal constants and the p-values of the slopes before and after transformations

Variable	$\mathbf{p_1}$	$\mathbf{c_{log}}$	$\mathbf{c_{sqrt}}$	$\mathbf{c_{fin}}$	$\mathbf{p_2}$
Tackle	0.0130	1.9816	0.1601	1.9816	0.999
Offside	0.0000	✗	✗	0.0060	0.213
Interception	0.0000	0.3184	✗	0.3184	0.998
Foul	0.1040	1.3315	0.2157	1.3315	0.999
Fouled	0.0000	0.7735	✗	0.7735	0.999
Clearance	0.0000	0.5505	✗	0.5505	0.998
Block	0.0910	0.7182	✗	0.7182	0.993
Shot	0.0000	0.3811	✗	0.3811	0.999
Unsuccessful touch	0.0150	1.1567	0.1181	1.1567	0.999
Dispossessed	0.0000	0.3101	✗	0.3101	0.999
Aerial	0.0000	1.3739	✗	1.3739	0.999
Dribble	0.0010	0.4557	✗	0.4557	0.998
Pass	0.0000	✗	✗	0.0001	0.731
Key Pass	0.1280	0.8093	0.0588	0.8093	0.999

$\mathbf{c_{fin}}$'s are the final constants to be used in the logartimic transformation

p_1's are the p-values of the slope for y_j and x_j

p_2's are the p-values of the slope for $\log(y_j + \mathbf{c_{fin}})$ and $\log(x_j + \mathbf{c_{fin}})$

$\mathbf{c_{log}}$'s and $\mathbf{c_{sqrt}}$'s are the optimal constants for $\log(x)$ and $\sqrt{x}$ transformations, respectively.

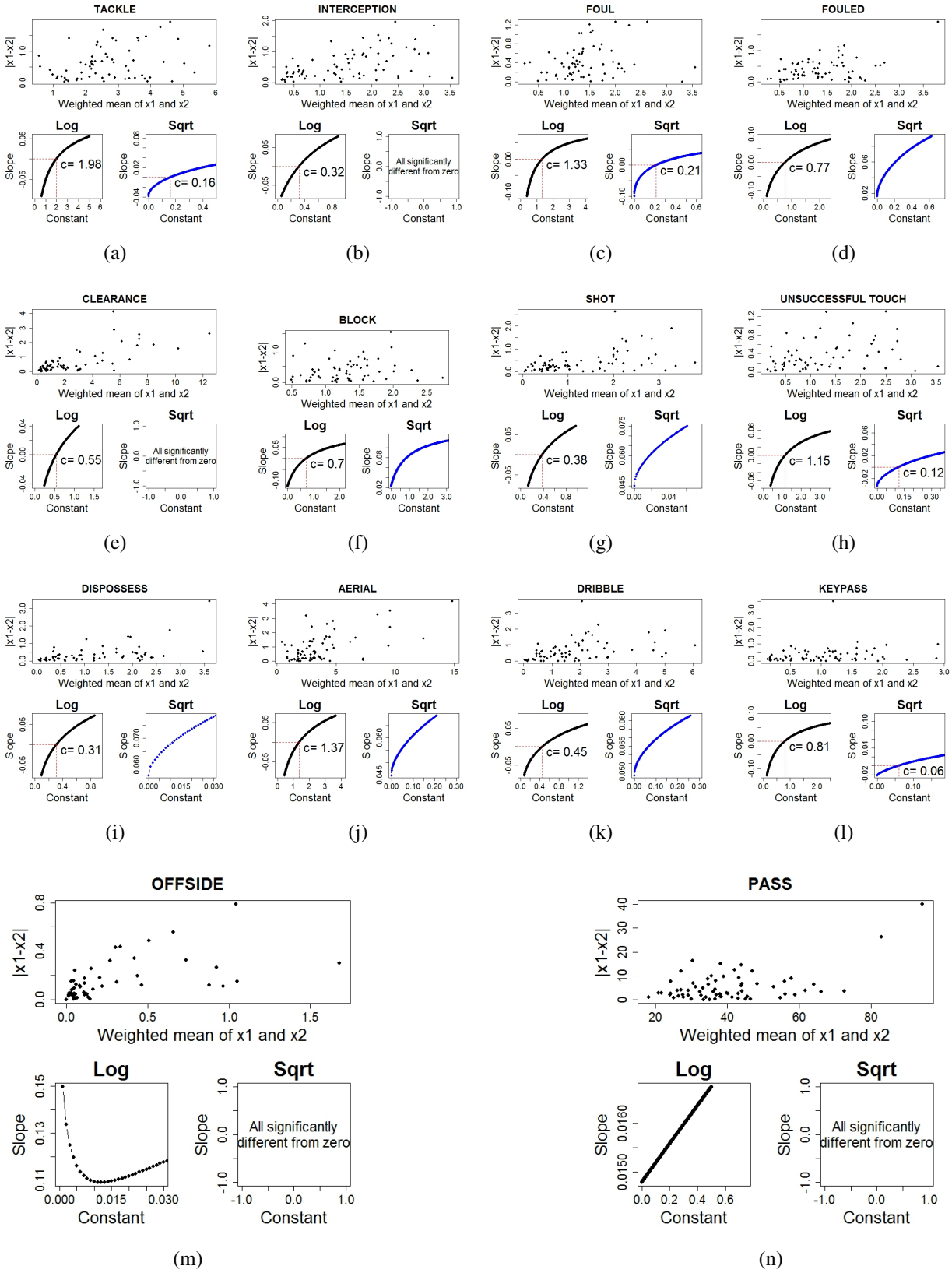


Figure 4.8: Analysis for optimal constant values on log and square root transformations

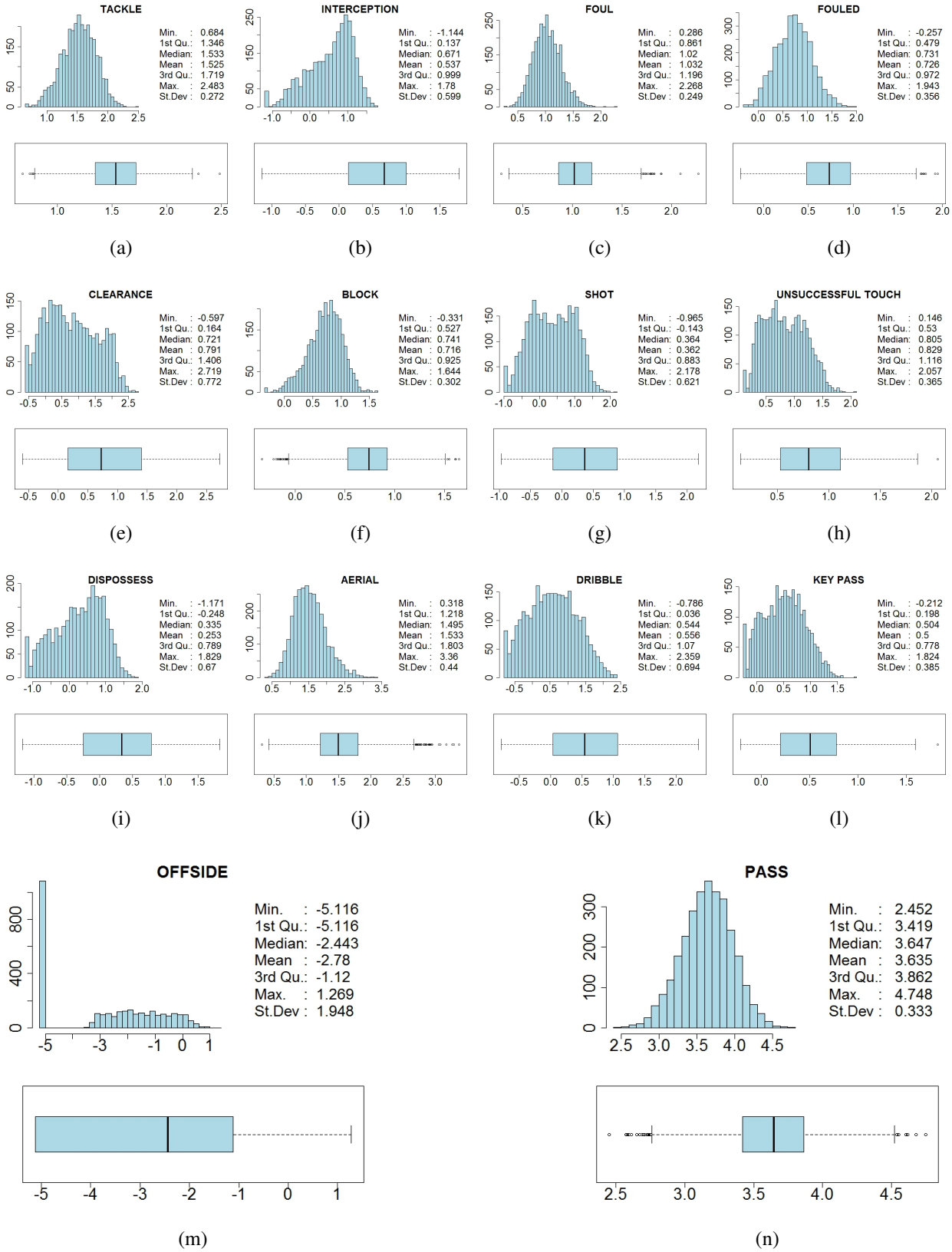


Figure 4.9: Summary of upper level count variables after $\log(x + c)$ *is applied*

4.1.3 Standardisation

Whereas transformation deals with relative differences between players on the same variable, standardisation and weighting are about calibrating the impact of the different variables against each other. Usually, weighting and standardisation both involve multiplying a variable with a constant, but they have different meanings. Standardisation is about making the measurements of the different variables comparable in size, whereas weighting is about giving the variables an impact on the overall distance that corresponds to their subject-matter importance.

I standardise transformed and untransformed (e.g, Goal and Assist) upper level count variables to average absolute deviation, see Table 3.1. The reason can be explained by the following argument. First, range standardisation heavily depends on most extreme values, and many of these variables do not have a natural maximum. For example, I observe maximum value of 5 blocks, but next year player can have 8 blocks based on per 90 minutes representation, which will change the variable range. Therefore, I would prefer a standardisation method that essentially assesses the overall variation (e.g., unit variance, median absolute deviation or average absolute deviation), not just the two most extreme values.

Second, The Manhattan (L_1) distance, which is governed by absolute values, will be used for upper level count variables, see Section 4.2.1, and within the variable I standardise in such a way that the same absolute value has always the same meaning. In particular, observations are only outliers when the performance of players in a certain respect is indeed outlying in which case they should still be located in such a place that the values have the direct interpretation, see more discussion about the effect of outliers on standardisation in Hennig and Liao (2013) and in Section 3.2.3. All these points to the fact that if I want every value to have the same impact, the estimation of variation should be based on the absolute values of differences (L_1 distance) between the values and the location parameter (e.g., median). Next, for computing the absolute values, the centre should be the median, which minimises the absolute deviation. Note that median and mean are approximately the same if the distribution of a variable is approximately symmetric, which is more or less the case for most of the transformed upper level count variables, see Table 4.9, but this is not the case for the untransformed ones, see Table 4.4i and Table 4.4p. Thus, it is determined that the average absolute deviation will be used for standardising upper level count variables.

For the lower level percentages, I standardise by dividing by the pooled average absolute deviation from all categories belonging to the same composition of lower level variables, regardless of their individual relative deviation. The pooled average absolute deviation is defined as the average of the average absolute deviation

$$s_{pooled} = \frac{\sum_{j=1}^{D} s_j}{D}, \tag{4.8}$$

where s_j is the average absolute deviation of j^{th} variable. The reason for this is that a certain difference in percentages between two players has the same meaning in each category, which does not depend on the individual deviation of the category variable. I want to have the resulting distances between percentages to count in the same way regardless of which part of the composition they are from. For example, consider compositional data with three components, and their deviations are 14, 18 and 20, respectively. If I take 0.1 away from the first one and give it to the second one, the change will be different in some of the distance measures (e.g., Aitchison distance) than if I take 0.1 away from the first one and give it to the third one. This argument can be mathematically proved by the following theory. Note that this is not directly in favour of the pooled variance, it is about standardising the compositions with the same number.

Definition 4.1.2. Let $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ be a D-part composition from the data set X, $i = 1, 2, \ldots, n$, with the following assumptions

i $\sum_{k=1}^{D} x_{ik} = 1$,

ii $0 \leq x_{ik} \leq 1$,

iii $D > 2$,

and let s_k, $k = 1, 2, \ldots, D$, be a standardised constant which may or may not depend on k for all $s_k > 0$. Then, consider the following distances

1. Standardised Euclidean distance:

$$d_E(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{D} \left\{ \frac{x_{ik}}{s_k} - \frac{x_{jk}}{s_k} \right\}^2}, \tag{4.9}$$

2. Standardised Manhattan distance:

$$d_M(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{D} \left| \frac{x_{ik}}{s_k} - \frac{x_{jk}}{s_k} \right|, \tag{4.10}$$

3. Aitchison distance:

$$d_A(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{D} \left\{ \log\left(\frac{x_{ik}}{g(x_i)} \right) - \log\left(\frac{x_{jk}}{g(x_j)} \right) \right\}^2}, \tag{4.11}$$

where $g(\mathbf{x}_i) = \left(\prod_{k=1}^{D} x_{ik} \right)^{1/D}$.

Axiom 4.1.1. Let $\mathbf{x}_1 = (x_{11},\, x_{12},\, \ldots,\, x_{1D})$, $\mathbf{x}_1^{(1)} = (x_{11}+\epsilon,\, x_{12}-\frac{\epsilon}{D-1},\, \ldots,\, x_{1D}-\frac{\epsilon}{D-1})$, $\mathbf{x}_1^{(q)} = (x_{11}-\frac{\epsilon}{D-1},\, \ldots,\, x_{1q}+\epsilon,\, \ldots,\, x_{1D}-\frac{\epsilon}{D-1})$ be D-part compositions. Assume that $\mathbf{x}_1, \mathbf{x}_1^{(1)}, \mathbf{x}_1^{(q)} \in X$, $0 < \epsilon \leq 1$, and $0 \leq x_{1k}^{(t)} \leq 1$ for all t. Then, the general distance should satisfy the following equation;

$$d(\mathbf{x}_1, \mathbf{x}_1^{(1)}) = d(\mathbf{x}_1, \mathbf{x}_1^{(q)}), \tag{4.12}$$

Theorem 4.1.1. *Equation* (4.12) *does hold for d_E and d_M if $s_k = s$, $\forall k$ and does not hold for d_A ($x_{1k}^{(t)} \neq 0$) in general.*

Proof:

- (Standardised Euclidean distance)

$$\begin{aligned}
d_E(\mathbf{x}_1, \mathbf{x}_1^{(1)})^2 - d_E(\mathbf{x}_1, \mathbf{x}_1^{(q)})^2 &= \sum_{k=1}^{D} \left\{ \frac{x_{1k}}{s_k} - \frac{x_{1k}^{(1)}}{s_k} \right\}^2 - \sum_{k=1}^{D} \left\{ \frac{x_{1k}}{s_k} - \frac{x_{1k}^{(q)}}{s_k} \right\}^2 \\
&= \left(\frac{\epsilon^2}{s_1^2} + \frac{\epsilon^2}{(D-1)^2} \sum_{k=2}^{D} \frac{1}{s_k^2} \right) - \left(\frac{\epsilon^2}{s_q^2} + \frac{\epsilon^2}{(D-1)^2} \sum_{\substack{k=1 \\ k \neq q}}^{D} \frac{1}{s_k^2} \right) \\
&= \epsilon^2 \left[\frac{1}{s_1^2} - \frac{1}{s_q^2} + \frac{1}{(D-1)^2} \left(\frac{1}{s_1^2} - \frac{1}{s_q^2} \right) \right] \\
&= \epsilon^2 \left(\frac{1}{s_1^2} - \frac{1}{s_q^2} \right) \left(1 - \frac{1}{(D-1)^2} \right) \\
&= 0 \quad \Longleftrightarrow \quad s_1 = s_q.
\end{aligned}$$

  If this is satisfied $\forall q$, then $s = s_k\ \forall k$

- (Standardised Manhattan distance)

$$\begin{aligned}
d_M(\mathbf{x}_1, \mathbf{x}_1^{(1)}) - d_M(\mathbf{x}_1, \mathbf{x}_1^{(q)}) &= \sum_{k=1}^{D} \left| \frac{x_{1k}}{s_k} - \frac{x_{1k}^{(1)}}{s_k} \right| - \sum_{k=1}^{D} \left| \frac{x_{1k}}{s_k} - \frac{x_{1k}^{(q)}}{s_k} \right| \\
&= \left(\frac{\epsilon}{s_1} + \frac{\epsilon}{D-1} \sum_{k=2}^{D} \frac{1}{s_k} \right) - \left(\frac{\epsilon}{s_q} + \frac{\epsilon}{D-1} \sum_{\substack{k=1 \\ k \neq q}}^{D} \frac{1}{s_k} \right) \\
&= \epsilon \left[\frac{1}{s_1} - \frac{1}{s_q} + \frac{1}{D-1} \left(\frac{1}{s_1} - \frac{1}{s_q} \right) \right] \\
&= \epsilon \left(\frac{1}{s_1} - \frac{1}{s_q} \right) \left(1 - \frac{1}{D-1} \right) \\
&= 0 \quad \Longleftrightarrow \quad s_1 = s_q
\end{aligned}$$

If this is satisfied $\forall q$, then $s = s_k \ \forall k$

- (Aitchison distance) The proof will be provided by a counter example. Table 4.5 shows that the example disproves the general axiom for Aitchison distance, which does not hold Equation (4.12) in general.

Table 4.5: Counter examples for the Aitchison distance

Compositions, where $\epsilon = 0.15$ **and** $q = 3$	**Percentages**
$\mathbf{x}_1$	$(0.40, 0.30, 0.20, 0.10)$
$\mathbf{x}_1^{(1)}$	$(0.55, 0.25, 0.15, 0.05)$
$\mathbf{x}_1^{(3)}$	$(0.35, 0.25, 0.35, 0.05)$
$d_A(\mathbf{x}_1, \mathbf{x}_1^{(1)}) - d_A(\mathbf{x}_1, \mathbf{x}_1^{(3)})$	-0.0368

Remark. For $D = 2$, all reasonable standardisations are the same for both variables, since $x_{i1} = 1 - x_{iD}$.

Remark. In Correspondence Analysis (Greenacre, 2017), another central axiom is the "principle of distributional equivalence", which states that if two columns (resp., two rows) of a contingency table have the same relative values, then merging them does not affect the dissimilarities between rows (resp., columns). I am here only concerned with dissimilarities between players, not with dissimilarities between variables. For dissimilarities between players, distributional equivalence holds when using a standardised Manhattan distance with s_k chosen independently of k pooling average L_1-variable-wise distances from the median, because when merging two variables $\mathbf{x}$ and $\mathbf{y} = \mathbf{cx}$, these simply sum up.

For all other variables (e.g., Age, Mins, Weight, etc.), average absolute deviation will be used for standardisation in order to make them comparable with upper level count and lower level compositions. However, variable standardisation will not be applied for position variables and league and team variables, because their distance designs are different, which will be discussed in Section 4.2.

As I argued in Section 4.1.1, for the unspecified *'other'* categories the distance computation is based on the non-other categories. For standardisation, if I include the unspecified *'other'* categories for computing the pooled average absolute deviation, then the biggest weights from those categories make the variation higher, and the impact of non-other categories will be lower when computing the distance. The summary statistics of the *Key pass (Other)* and *Assist (Other)* variables in Figure 4.5 can be a guidance to identify how those biggest weights affect the computation of the pooled average absolute deviation. Thus, the pooled deviation will be computed based on the non-other categories.

4.1.4 Weighting

Weighting is the concept of multiplying variables with different constants. Some variables may be more important than other variables, and weighting can take this into account. Assigning a different weight to each variable may lead to some variables being more dominant than the others for distance computation, which can influence the meaning of the results. Time variables and the upper level count variables which do not contain any sub-categories are weighted by one unit [1]. This can be interpreted as the characterization of players can be better reflected by utilizing their information equally than assigning different weights for each upper level count variable.

One aspect of variable weighting here is that in case that there are one or more lower level compositions of an upper level variable, the upper level variable is transformed and standardised individually whereas the categories of the composition are standardised together. This reflects the fact that the upper level count and the lower level distribution represent distinct aspects of a player's characteristics, and on this basis I assign the same weight to the upper level variable as to the whole vector of compositional variables, except the ones which contain more than one sub-category (e.g., Shot has four sub-categories: Zone, situation, accuracy, body parts). For example, a weight of one for transformed block counts is matched by a weight of $1/3$ for each of the sub-variables "shots blocked", "crosses blocked", "passes blocked". The weight assignment of these kinds is summarised in Table 4.6.

Table 4.6: Weight assignment for the upper level variables, which only contains one sub-category.

Variables	**Weights for each percentage**	**Weights for upper level count**	**Total weights for the variable**
Tackles	$1/2$	1	2
Blocks	$1/3$	1	2
Aerials	$1/2$	1	2
Dribbles	$1/2$	1	2

The percentage variables of the same composition are linearly dependent and are therefore correlated with each other; k percentage variables do not represent k independent parts of information. Variable selection and dimension reduction are very popular to deal with this. However, in distance construction, the problem is appropriately dealt with using weighting. There is no advantage in using, for example, only two variables out of "out of the box", "six yard box", "penalty area" and weight them by $1/2$ each; using all three means that they are treated symmetrically for

[1] Unit of measurement is defined as any specified amount of quantity (e.g., length, time, volume, etc.) by comparison with which any other quantity of the same kind is measured or estimated.

the construction of the distance, as is appropriate. At the same time down-weighting prevents redundant information dominating the overall distance. Therefore, it is not a problem that the same information is used twice, because the percentage variables are proportionally weighted down.

Next, weights for the variables that contain more than one sub-category, will be discussed. In Section 4.1.1, I stated that variables that contain common sub-categories can be represented in different forms, such as proportional and success rate. In this sense, the computation of these percentages as well as their weight assignments can be considered together. Table 4.7, 4.8 and 4.9 provide how the weight assignment as well as the representation for *Shots&Goals*, *Passes* and *Key passes&Assists* are made (Table 2.3 is the reference for the abbreviations on these tables). The question for the weight assignment of the upper level variables that contain more than one category is whether I should assign either the total weight of the whole vector of compositional variables (the first selection), or one unit to each variable as for the upper level variables that do not contain any sub-categories (the second selection). Here these arguments will be discussed with an example. I choose 3 players with their upper level count and lower level compositions of Shot action, see Table 4.10. The distance between the first and the second players is small if the first selection is used, but we can see a big difference between compositions. On the other hand, if I use the second selection, the distance between the second and the third players will be small, but the difference between their total shots is quite large. In this respect, the combination of these two arguments, which can be done by taking the average of the weights (e.g., the first selection, $w_1 = 4$ and the second selection, $w_2 = 1$, hence the average, $w_a = 2.5$), see Table 4.11, can be a reasonable choice for the weight assignment of the upper level variables that contain more than one category, because I do not want to make either upper or lower level variables dominate the resulting differences.

As discussed in Section 4.1.1, in case that an upper level count variable is zero for a player, the percentage variables are missing. In this situation, for overall distance computation between such a player and another player, the composition variables can be assigned weight zero and the weight that is normally on an upper level variable and its low level variables combined can be assigned to the upper level variable.

All these weight assignments are determined by some kind of analytical thinking of how the different variables are connected with my football knowledge. However, a manager or a scout from a team can be interested in some of the variables more than the others. For example, a manager can demand similar to a defender from his team and therefore consider the variables related to defending (e.g., Tackles, Blocks, etc.) to be more important than the variables related to attacking (e.g., Shots, Goals, etc.). Thus, weights can be modified by users in terms of which variables they are interested in more. For this aim, I designed an application with `R Shiny` for the users who

wish to have flexibility to play with the weights of the variables, see Figure 4.14 , 4.15 and 4.16 in Section 4.4 for different examples.

Table 4.7: Weight assignment for lower level compositions on Shot (x_1) and Goal (x_2) variables. 'Shot pro.' and 'Goal pro.' stand for the sub-categories of Shot and Goal variables in percentage representation, 'Goal suc.' represents the success rates in each sub-category of Goal variable, 'Goal suc1.' is the representation of the overall success rate for the Goal variable standardised by total shots, and 'Goal suc2.' is the representation of the overall success rate for the Goal variable standardised by total shots on target.

Categories	**Representation**	**Weights for lower level compositions**				**Total weights**
		OOB ($x_{i(1)}$)	SYB ($x_{i(2)}$)	PA ($x_{i(3)}$)		
	Shot pro. ($x_{1(j)}/x_1$)	$1/3$	$1/3$	$1/3$		1
Zones	Goal pro. ($x_{2(j)}/x_2$)	$1/6$	$1/6$	$1/6$		$1/2$ $\Big\}=\frac{7}{4}$
	Goal suc. ($x_{2(j)}/x_{1(j)}$)	$1/12$	$1/12$	$1/12$		$1/4$
		OP ($x_{i(4)}$)	C ($x_{i(5)}$)	SP ($x_{i(6)}$)	PT ($x_{i(7)}$)	
	Shot pro. ($x_{1(j)}/x_1$)	$1/4$	$1/4$	$1/4$	$1/4$	1
Situations	Goal pro. ($x_{2(j)}/x_2$)	$1/8$	$1/8$	$1/8$	$1/8$	$1/2$ $\Big\}=\frac{7}{4}$
	Goal suc. ($x_{2(j)}/x_{1(j)}$)	$1/16$	$1/16$	$1/16$	$1/16$	$1/4$
		RF ($x_{i(8)}$)	LF ($x_{i(9)}$)	H ($x_{i(10)}$)	O ($x_{i(11)}$)	
	Shot pro. ($x_{1(j)}/x_1$)	$1/4$	$1/4$	$1/4$	$1/4$	1
Body parts	Goal pro. ($x_{2(j)}/x_2$)	$1/8$	$1/8$	$1/8$	$1/8$	$1/2$ $\Big\}=\frac{7}{4}$
	Goal suc. ($x_{2(j)}/x_{1(j)}$)	$1/16$	$1/16$	$1/16$	$1/16$	$1/4$
		OnT ($x_{i(12)}$)	OffT ($x_{i(13)}$)	B ($x_{i(14)}$)		
	Shot pro. ($x_{1(j)}/x_1$)	$1/3$	$1/3$	$1/3$		1
Accuracy	Goal suc1. (x_2/x_1)		$3/8$			$3/8$ $\Big\}=\frac{7}{4}$
	Goal suc2. ($x_2/x_{2(12)}$)		$3/8$			$3/8$
					TOTAL	**7**

OOB: Out of box, SYB: Six yard box, PA: Penalty area, OP: Open play, C: Counter, SP: Set piece, PT: Penalty taken

RF: Right foot, LF: Left foot, H: Head, O: Other, OnT: On target, OffT: Off target, B: Blocked

Table 4.8: Weight assignment for lower level compositions on Pass (y) variables (AccP (y_1) and InAccP (y_2), where $y = y_1 + y_2$). 'Pass pro.' stands for the sub-categories of Pass variable in percentage representation, 'Accuracy' represents the success rates in each sub-category of Pass variable.

Categories	**Representation**	**Weights for lower level compositions**			**Total weights**
		Long ($y_{(1)}$)	Short ($y_{(2)}$)		
Length	Pass pro. ($y_{(j)}/y$)	$1/2$	$1/2$		$\left.\begin{matrix}1\\1/2\end{matrix}\right\} = \frac{3}{2}$
	Accuracy ($y_{i(j)}/y_{(j)}$)	$1/4$	$1/4$		
		Cross ($y_{(3)}$)	Corner ($y_{(4)}$)	FK ($y_{(5)}$)	
Type	Pass pro. ($y_{(j)}/y$)	$1/3$	$1/3$	$1/3$	$\left.\begin{matrix}1\\1/2\end{matrix}\right\} = \frac{3}{2}$
	Accuracy ($y_{i(j)}/y_{(j)}$)	$1/6$	$1/6$	$1/6$	
Overall success rate (y_1/y)		1			$= 1$
				TOTAL	**4**

Table 4.9: Weight assignment for lower level compositions Key pass (z_1) and Assist (z_2) variables. 'KP pro.' and 'Ass. pro.' stand for the sub-categories of Key pass and Assist variables in percentage representation, 'Ass. suc.' represents the success rates in each sub-category of Assist variable.

Cat.	**Representation**	**Weights for lower level compositions**					**Total weights**
		Long ($z_{1(1)}$)		Short ($z_{1(2)}$)			
Length	KP pro. ($z_{1(j)}/z_1$)	$1/2$		$1/2$			$= 1$
		Cr ($z_{i(3)}$)	Co ($z_{i(4)}$)	FK ($z_{i(5)}$)	T-b ($z_{i(6)}$)	T-in ($z_{i(7)}$)	
Type	KP pro. ($z_{1(j)}/z_1$)	$1/5$	$1/5$	$1/5$	$1/5$	$1/5$	$\left.\begin{matrix}1\\1/2\\1/4\end{matrix}\right\} = \frac{7}{4}$
	Ass. pro. ($z_{2(j)}/z_2$)	$1/10$	$1/10$	$1/10$	$1/10$	$1/10$	
	Ass. suc. ($z_{2(j)}/z_{1(j)}$)	$1/20$	$1/20$	$1/20$	$1/20$	$1/20$	
Overall success rate (z_2/z_1)		$1/4$					$= \frac{1}{4}$
						TOTAL	**3**

Cr: Cross, Co: Corner, FK: Free-kick, T-b: Through-ball, T-in: Throw-in

Table 4.10: An example for one of the selected upper level variables, Shot which contain more than one category. The number of total shots represents the upper level count variables, whereas the other numbers are the lower level percentages in the category of Shot variable

Player	Total	*Zones*			*Situations*				*Body parts*				*Accuracy*		
	shots	OOB	SYB	PA	OP	C	SP	PT	RF	LF	H	O	OffT	OnT	B
1	10	0.1	0.4	0.5	0.1	0.4	0.1	0.4	0.1	0.2	0.7	0.0	0.1	0.3	0.6
2	10	0.5	0.2	0.3	0.5	0.1	0.3	0.1	0.3	0.3	0.3	0.1	0.4	0.5	0.1
3	100	0.5	0.2	0.3	0.5	0.1	0.3	0.1	0.3	0.3	0.3	0.1	0.4	0.5	0.1

4.1.5 Summary of data pre-processing

The summary of all the data pre-processing steps is presented at the following list. The variables, for which data pre-processing steps are used, are summarised in Table 4.11

1. Representation:

    - I analyse players who have played a minimum 200 minutes during the season.
    - League and team scores are assigned to each player based on per game representation. Players who played in multiple teams have a weighted average of league and team scores. Weights are assigned based on representation to 90 minutes. Missing values in $x_{(tc)}$ are filled by using $x_{(l)}$ and $x_{(tp)}$.
    - For the position variables, $Y_{(11)}$ is represented as binary information, whereas $Y_{(15)}$ is proportionally represented.
    - Representation is not changed for Age, Weight, Height, Apps and Mins variables.
    - '*Other*' categories from Key pass and Assist variables will be excluded from the analysis, except for computing the percentages for the other categories.
    - Upper level variables are represented as per 90 minutes, whereas lower level variables are represented as percentages.
    - For zero percentages, a Bayesian-multiplicative approach is used as a motivation of how to adjust these values.
    - For essential zeros, a composition in which all components are zero will be weighted as zero, and total count of the relevant action will be up-weighted by the total weights of the relevant composition.

2. Transformation:

    - Upper level count variables are transformed by logarithmic function with appropriate constants based on the last year information for different variables.
    - Goal and Assist variables are not transformed, since they have a direct impact on match results.

3. Standardisation:

    - Upper level count variables are standardised by average absolute deviation, whereas lower level compositions are standardised by the pooled average absolute deviation from all categories belonging to the same composition of lower level variables.
    - For the variables which contain an unspecified *other* category, the pooled average absolute deviation will be computed based on the non-other categories.

4. Weighting:

    - Time variables and the upper level variables that do not contain any sub-categories are weighted by one.
    - I assign the same weights to the upper level variables, which only contain one sub-category, as to the whole vector of compositional variables.
    - The upper level variables that contain more than one category are weighted approximately by the average of the number of categories in the relevant variable.

Table 4.11: Summary of data pre-processing steps (UL: Upper level, LL: Lower level)

Variables	**Representation**		**Transformation**	**Standardisation**			**Weights**			**Total weights**
	TL	*LL*	**(**$\log(x+c)$**)**	*TL*	*LL*	*None*	*TL*	*LL*	*None*	
Age	-	-	✗	-	-	✓	-	-	1	1
Height	-	-	✗	-	-	✓	-	-	1	1
Weight	-	-	✗	-	-	✓	-	-	1	1
Apps	-	-	✗	-	-	✓	-	-	1	1
Mins	-	-	✗	-	-	✓	-	-	1	1
Offsides	✓	-	✓	✓	-	-	1	-	-	1
Interceptions	✓	-	✓	✓	-	-	1	-	-	1
Fouls	✓	-	✓	✓	-	-	1	-	-	1
Fouled	✓	-	✓	✓	-	-	1	-	-	1
Clearances	✓	-	✓	✓	-	-	1	-	-	1
Unsuc. Touches	✓	-	✓	✓	-	-	1	-	-	1
Dispossesses	✓	-	✓	✓	-	-	1	-	-	1
Tackles	✓	✓	✓	✓	✓	-	1	1	-	2
Blocks	✓	✓	✓	✓	✓	-	1	1	-	2
Aerials	✓	✓	✓	✓	✓	-	1	1	-	2
Dribbles	✓	✓	✓	✓	✓	-	1	1	-	2
Shots	✓	✓	✓	✓	✓	-	2.5	4	-	6.5
Goals	✓	✓	✗	✓	✓	-	2	3	-	5
Passes	✓	✓	✓	✓	✓	-	2.5	4	-	6.5
Key passes	✓	✓	✓	✓	✓	-	1.5	2	-	3.5
Assists	✓	✓	✗	✓	✓	-	1	1	-	2

4.2 Aggregation of Variables in Distance Design

There are different well-known ways of aggregating variables in order to define a distance measure. The same principle as before, "matching interpretative distance", applies here as well. Distance aggregation for the upper level count variables, the lower level compositions, the position variables, the team and league variables and the others will be discussed.

4.2.1 Upper level count variables

There are different types of variables in this data set (the position variables are treated in a non-Euclidean way, which will be explained later), and therefore I decided against using Euclidean aggregation, which implicitly treats the variables as if they are in a joint Euclidean space, and which weights larger differences on individual variables up when comparing two players. Instead, I aggregate variables by summing up the individual distances, i.e., following the principle for the Manhattan distance, as also used by Gower (1971). This means that distances on all variables are treated in the same way regardless of the size of the difference. Note that the Manhattan distance will also be adopted for *Age, Weight, Height, Mins, Apps* variables.

4.2.2 Lower level compositions

Percentage variables in the data set are compositional data in the sense of Aitchison (1986), who set up an axiomatic theory for the analysis of compositional data, see Section 3.4. I will argue here that for the compositional data in this application the simple Manhattan distance is more appropriate than what Aitchison proposed specifically for compositional data, which means that the principle of matching interpretative distance in distance construction can be in conflict, depending on the application, with a pure mathematical axiomatic approach.

Manhattan distance vs Aitchison distance

In order to compare the Manhattan and the Aitchison distance, I first discuss the Manhattan distance regarding the four axioms, see Section 3.4.3.

- The Manhattan distance does not fulfil **scale invariance**; if both compositions are multiplied by λ, the Manhattan distance is multiplied by λ. This, however, is irrelevant here, because I am interested in percentages only that sum up to 100.

- The Manhattan distance is **permutation invariant**.
- The Manhattan distance is not **perturbation invariant**, but as was the case for scale invariance, this is irrelevant here, because the percentages are relative counts and the operation of multiplying different categories in the same composition with different constants is not meaningful in this application.
- The Manhattan distance is not **sub-compositional coherent**, but once more this is not relevant, because in this application it is not meaningful to compare the values of the compositional distance with those from sub-compositions.

Aitchison's axioms were proposed as general principles for compositional data, but in fact the axioms were motivated by specific applications with specific characteristics, which mostly do not apply here. Furthermore, the Aitchison distance can be problematic for small percentages, which is mathematically shown in Expression (3.28). For football players, the Aitchison distance does not seem suitable for matching "interpretative distance". I also demonstrate this using three popular players from the data set, and the "Block" action, James Rodriguez (*JR*), Alexis Sanchez (*AS*) and Cesc Fabregas (*CF*).

Table 4.12: Percentage variables in block action for the three selected players

Players	**Shot blocked**	**Cross blocked**	**Pass blocked**
James Rodriguez (*JR*)	0.03	0.03	0.94
Alexis Sanchez (*AS*)	0.00 (≈ 0)	0.04	0.96
Cesc Fabregas (*CF*)	0.09	0.05	0.86

Table 4.13: Distances of block percentages for the three selected players

Distance	*JR-AS*	*JR-CF*	*AS-CF*
Manhattan	0.06	0.16	0.20
Aitchison (clr)	26.69	0.84	27.42
Aitchison (ilr)	26.69	0.84	27.42
Aitchison (alr)	32.56	1.33	33.74

Percentages and distances are presented in Table 4.12 and Table 4.13. *AS* has a very small proportion (≈ 0 but nonzero) in the sub-variable of "Shot blocked". The Aitchison distances in any log-ratio transformations between *JR* and *AS* as well as between *CF* and *AS* are quite large, whereas it is not very large between *JR* and *CF*. But *JR* and *AS* are quite similar players according

to the data; both block almost exclusively passes and hardly any shots or crosses. *CF* blocks substantially more shots and some more crosses than both others. Therefore, the two distances between *CF* and both *JR* and *AS* should be bigger than that between *JR* and *AS*, which is what the Manhattan distance delivers.

In addition, as explained in Theorem 4.1.1, the general principle is that differences within different variables should be treated the same. The Manhattan distance treats absolute differences between percentages in the same way regardless of the size of the percentages between which these differences occur, whereas the Aitchison distances is dominated by differences between small percentages in an inappropriate manner.

4.2.3 Team and league variables

The league ($x_{(l)}$) and team variables ($x_{(tp)}$ and $x_{(tc)}$) are introduced in Section 2.1.1 as well as their representations in Section 4.1.1. As a reminder, the descriptions of these variables are once again introduced as follows:

- $x_{(l)}$: League ranking scores based on the information on the UEFA website,
- $x_{(tp)}$: Team points from the ranking table of each league based on the 2014-2015 football season,
- $x_{(tc)}$: Team ranking scores based on the information on the UEFA website.

It seems to be appropriate to adopt the Manhattan distance for these variables, but $x_{(l)}$ and $x_{(tp)}$ variables are connected to each other, because league scores are governed by the success of the teams in their league. In terms of team performances, considering $x_{(l)}$ and $x_{(tp)}$ separately makes two similar teams far away from each other. The idea can be explained by the following example. I consider three players from Barcelona (Spain), Malaga (Spain) and Galatasaray (Turkey). *Malaga* finished the season somewhere in the middle and *Barcelona* finished in first place in the Spanish League, whereas *Galatasaray* was in first position in the Turkish League based on the data set. The idea is to reflect the two variables in one distance measure together. Table 4.14 is the illustration of the three selected players, and Table 4.15 shows the comparison between the Manhattan distance and the distance that I propose, see Equation (4.13), after applying standardization to average absolute deviation for each variable in order to make them comparable. The "interpretative distance" between *Malaga* and *Galatasaray* should be less than the Manhattan distance that I computed for the other pairs (BAR-MAL and BAR-GS) in Table 4.15; in other words, *Malaga* and *Galatasaray*

should be more similar to each other than their differences with *Barcelona*. That is because *Malaga* (which has the highest league score, but moderate team points) and *Galatasaray* (which has high team points, but lowest league score) should be closer to each other than *Barcelona* in terms of their performances in the 2014-2015 football season. Therefore, they should be incorporated in one distance that should match interpretative distance. The proposed distance measure is given as follows

$$d_1(i,j) = \left| \frac{x_{i(l)} - x_{j(l)}}{s_{(l)}} + \frac{x_{i(tp)} - x_{j(tp)}}{s_{(tp)}} \right|, \tag{4.13}$$

where $s_{(l)}$ and $s_{(tp)}$ are the average absolute deviations for $x_{(l)}$ and $x_{(tp)}$, respectively. $x_{(tc)}$ is not related to the argument above, since it is the combination of these two variables, see Section 2.1.1. As I already discussed in Section 4.1.1, $x_{(l)}$ and $x_{(tp)}$ should be up-weighted in case of existence of missing values in $x_{(tc)}$. Equation (4.14) shows how the idea of up-weighting deals with the missing values:

$$d_{lt}(i,j) = \begin{cases} d_1(i,j) & \text{if } x_i^{(tc)} \text{ or } x_j^{(tc)} \text{ are missing,} \\ \frac{2}{3}d_1(i,j) + \frac{1}{3}d_2(i,j) & \text{otherwise,} \end{cases} \tag{4.14}$$

where $d_2(i,j) = (|x_{i(tc)} - x_{j(tc)}|)/s_{(tc)}$ and $s_{(tc)}$ is the average absolute deviation of $x_{i(tc)}$.

Table 4.14: $x_{(l)}$ and $x_{(tp)}$ variables for the three selected players

Players	**Leagues**	**Teams**	**League scores** ($x_{(l)}$)	**Team points** ($x_{(tp)}$)
1	Spain	Barcelona (BAR)	99.427	2.47
2	Spain	Malaga (MAL)	99.427	1.31
3	Turkey	Galatasaray (GS)	32.600	2.26

Table 4.15: Distances of $x_{(l)}$ and $x_{(tp)}$ variables for the three selected players

Distances	**BAR-MAL**	**BAR-GS**	**MAL-GS**
Manhattan distance	2.68	3.62	5.33
League-Team distance	2.68	3.62	0.94

4.2.4 Position variables

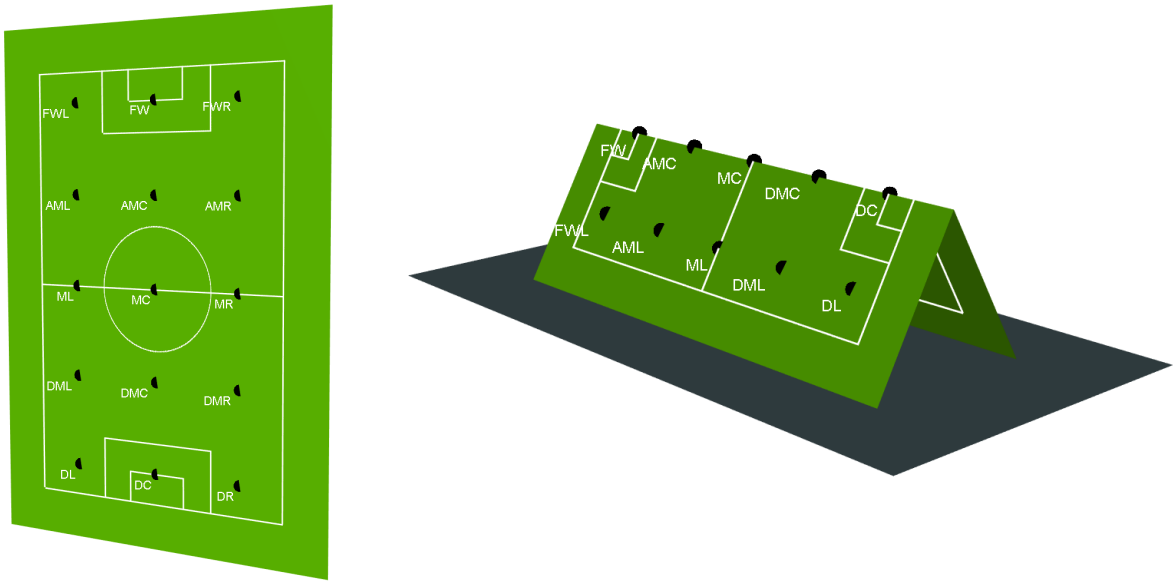
The data contains variables that reflect information on where players are located on the field. In Section 4.1.1, I described two types of position information. The first one ($C(Y_{(15)})$), which involves 15 sub – variables, provides the information of how players are proportionally distributed in different positions based on the number of appearances during the 2014-2015 football season, and the second one ($Y_{(11)}$), which involves 11 sub – variables, are the binary variables indicating where players played in the past seasons, which is historical background information of a player's position. Now I will present how I construct an appropriate distance measure for these two types of information.

Position variables-1

The distance measure for lower level compositions was previously discussed. Here the difference is that $C(Y_{(15)})$ variables also include information for which the locations of players are meaningful; hence, this information should also be embedded into the distance structure. For instance, the distance between a forward and a midfielder cannot be the same as the distance between a forward and a defender, since forward players should be location-wise further away from defenders than midfielders, see Figure 2.1a. Therefore, I need to design a distance measure in which geographical information of players is involved based on a player's position.

As a part of representing $C(Y_{(15)})$ variables, I build a three dimensional coordinate system in which a player's location is represented with one point on the field, and then compute the distance measures between these points. The football field can be viewed in two dimensional space, see Figure 2.1a, but I believe a player's location is more accurately represented in three dimensions rather than two dimensions. For example, consider a distance measure between defensive position in two dimensions. The distance between left and right side of the positions is two units, whereas the distance between centre and left or right is one unit. In terms of football knowledge, left or right side players should be more similar than centre players in the sense that they both have the characteristic of playing as a winger, not centre, but they play in different sides (left and right), which brings the idea that they should be less similar. On the other hand, central players can have different characteristics from the players who play on the left or right side of the field. In terms of retaining all the ideas above, I design an equilateral triangle in which centre, left and right sides should be equal to each other as one unit. This will be accomplished by upgrading the coordinate system from two dimensions to three dimensions, see Figure 4.10.

The next step is to construct an appropriate coordinate system. In the three dimensional coor-



(a) 2 dimensions

(b) 3 dimensions

Figure 4.10: Comparison between two and three dimensional fields

dinate system, the sides of positions (left, right or centre) are presented as the first two coordinates, say x and y, whereas the positions (D, DM, M, AM, FW) are presented as the third coordinate, say z. x and y will be defined by a weighted mean location of the sides. Formally, the computation of the weighted mean for a coordinate is given as follows:

$$c(q_{ik}) = \left(\frac{\sum_{j=1}^{3} w_{ijk} q_{ijk}}{\sum_{j=1}^{3} w_{ijk}} \right), \tag{4.15}$$

where q_{ijk} is the coordinate (x, y, or z) of the j^{th} side of positions (left, right or centre) for the k^{th} position (D, DM, M, AM, FW) of player i, respectively, and w_{ijk} is the weight of the j^{th} side of positions for the k^{th} position of player i. x and y coordinates are fixed numbers, which are determined as shown in Table 4.16, in order to satisfy one of the properties of an equilateral triangle that all three sides are equal. Weights are assigned by proportions of the positions in each side. For example, if a player who only plays in the sides of defence position, and has a composition as 0.4 in left, 0.4 in right and 0.2 in centre, then the weighted mean of player i is computed in Equation (4.15) with the weights (0.4, 0.4, 0.2).

Although the idea above seems to be satisfactory, another problem arises when computing a weighted mean, which can be explained by an example. Table 4.17 presents 3 different players with their compositions in defence position, and Figure 4.11a illustrates the first principle that was

Table 4.16: x and y coordinates for the side of positions (left, right and centre)

Coordinates	**Centre**	**Left**	**Right**
x	$1/2$	0	1
y	$\sqrt{3}/2$	0	0

previously explained, and Figure 4.11b gives an idea about the second principle, which will be described now. In contrast to the first idea, I assume that $d(P_1, P_2)$ and $d(P_1, P_3)$ (Numbers refer to Table 4.17) should be the same; in other words, the weighted mean of two different sides (left and right), which is the case for the 2^{nd} player, should not be closer to the centre position (1^{st} player) than the other right-side position (3^{rd} player). That is because based on the argument that the distances between all three sides in one position should be equal to each other, players who played in the same position, but different sides regardless of their composition values, should have the same distance as one unit (which is 0.5 in this example), unless they have common sides.

Table 4.17: Percentage variables in $C(Y_{(15)})$ for three players

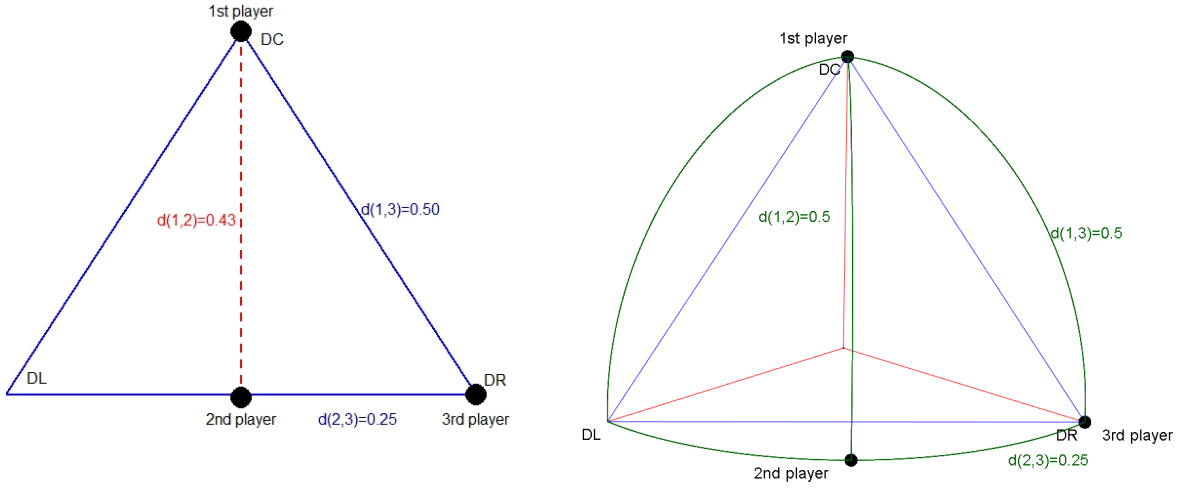
Players	**Centre**	**Left**	**Right**
P_1	1	0	0
P_2	0	0.5	0.5
P_3	0	0	1

This can be settled by designing a three dimensional space for each position (D, DM, M, AM, FW), and I obtain a weighted mean by using the *great-circle distance* [2] between two points – that is, the shortest distance over the Earth's surface. The great-circle distance is calculated by using the **'Haversine'** formula, which is defined as:

$$\begin{aligned} a_{ij} &= \sin^2\left(\frac{\varphi_i - \varphi_j}{2}\right) + \cos(\varphi_i) \cdot \cos(\varphi_j) \cdot \sin^2\left(\frac{\lambda_i - \lambda_j}{2}\right), \\ c_{ij} &= 2 \cdot atan2(\sqrt{a_{ij}}, \sqrt{1 - a_{ij}}), \\ d_{gc}(i,j) &= R \cdot c_{ij}, \end{aligned} \tag{4.16}$$

where φ_i and φ_j are the i^{th}, j^{th} latitudes, λ_i and λ_j are the i^{th}, j^{th} longitudes, and in terms of the standard $arctan$ function, whose range is $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$, it can be expressed as follows:

[2] Cajori (1928) credits an earlier use by de Mendoza et al. (1795). The term *haversine* was coined in Inman (1849)



(a) First principle

(b) Second principle

Figure 4.11: Comparison of the distance measures between three players in two and three dimensional cases for one position

$$atan2(y,x) = \begin{cases} arctan(\frac{y}{x}) & \text{if } x > 0, \\ arctan(\frac{y}{x}) + \pi & \text{if } x < 0 \text{ and } y \geq 0, \\ arctan(\frac{y}{x}) - \pi & \text{if } x < 0 \text{ and } y < 0, \\ +\frac{\pi}{2} & \text{if } x = 0 \text{ and } y > 0, \\ -\frac{\pi}{2} & \text{if } x = 0 \text{ and } y < 0, \\ \text{undefined} & \text{if } x = 0 \text{ and } y = 0. \end{cases} \tag{4.17}$$

In our situation, $R = 1$, and x and y coordinates are latitudes and longitudes, respectively. The great-circle distance is only used for computing the coordinates of the weighted mean, not for the distance between players; in other words, the coordinates of the weighted mean are computed based on the great circle plane, see Definition 4.2.1.

Definition 4.2.1. Let $\mathbf{G}^2$ be a great circle plane. A coordinate system on $\mathbf{G}^2$ is a bijection $\mathbf{G}^2 \leftrightarrow \mathbf{R}^2$. The point $P(x, y)$ is a notation representing a point $P \in \mathbf{G}^2$ corresponding to the element $(x, y) \in \mathbf{R}^2$.

Equation (4.18) gives how the coordinates of weighted mean are calculated for one side of position.

$$c(q^*_{ik}) = \left(\frac{\sum_{j=1}^{3} w_{ijk} q^*_{ijk}}{\sum_{j=1}^{3} w_{ijk}} \right), \tag{4.18}$$

where $q^*_{ijk} = P_k(x, y)$, is the coordinate (x, y, or z) of the j^{th} side of positions (left, right or centre) for the k^{th} position (D, DM, M, AM, FW) of player i, respectively, and w_{ijk} is the weight of the j^{th} side of positions for the k^{th} position of player i. Note that if * sign is given in $c(\cdot)$, then the coordinate(s) are calculated based on the great circle plane; otherwise, they are calculated based on the Euclidean plane. Figure 4.11 better illustrates the concept above. The first picture is an equilateral triangle, whereas the second one is the $1/8$ of the sphere with the equilateral triangle projection.

So far, I have only considered one position (e.g., defence). The question is how to compute a distance measure if a player plays in multiple positions. Figure 4.12 shows two different cases with the combination of five pieces of the $1/8$ of spheres. The figure at the bottom, the distances between the adjacent position (D, DM, MC, AMC, FW) in the same side (left, right or centre) are determined to be one unit, e.g., the distance between MR and AMR should be one unit. Note that the distance measures between both different positions and sides (e.g., the distance between DC and ML) are shown in Table 4.22, which is based on the Euclidean space, where for example the distance between DC and ML should be $\sqrt{5}$, since $d(P_{DC}, P_{ML}) = \sqrt{d(P_{DC}, P_{DL})^2 + d(P_{DL}, P_{ML})^2}$, see Table 4.22.

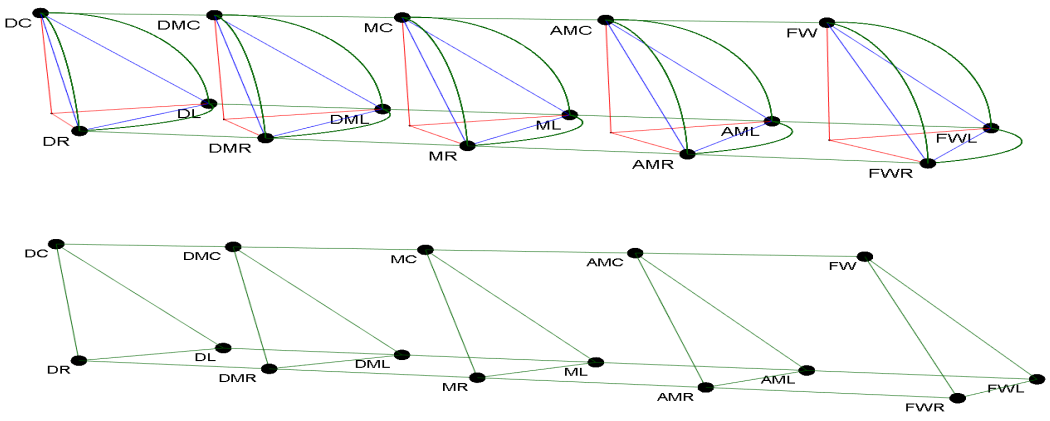


Figure 4.12: Comparison of two principles (which were illustrated in Figure 4.11) for multiple positions. The figure at the top is the illustration of how to obtain x and y coordinates (weighted mean) for each player in each position by using the great-circle distance, and the figure at the bottom gives a different drawing, in which curved lines are removed from the figure at the top.

The summary of all these steps can be explained with the following steps:

- The weighted mean of each position (D, DM, M, AM, FW) is independently computed on each $1/8$ sphere by using Equation (4.16), then each $1/8$ sphere will be projected on an equilateral triangle, see Figure 4.12.
- The weighted means of the positions will be computed by using x and y coordinates from each equilateral triangle, where the z coordinate represents each position (D, DM, M, AM, FW), in which the difference is one unit, see Figure 4.12. Table 4.18 mathematically summarises the idea above.
- Finally, the Manhattan distance will be used for computing the distance between points based on the weighted mean of x, y, z coordinates, so that the overall distance is defined as follows:

$$d_{pos1}(i_1, i_2) = d_M(c(q^*_{i_1}), c(q^*_{i_2})) + d_M(c(z_{i_1}), c(z_{i_2})), \tag{4.19}$$

  where

$$c(q^*_{i_1}) = \sum_{k=1|w_{ik}\neq 0}^{5} c(q^*_{i_1 k}) w_{ik}, \qquad c(z_{i_1}) = \sum_{k=1|w_{ik}\neq 0}^{5} c(z_{i_1 k}) w_{ik}. \tag{4.20}$$

  Here, $c(q^*_{i_1})$ are the weighted means that are computed by using the great-circle distance, see Equation (4.18), and $c(z_{i_1 j})$ is the weighted mean that is computed by using Equation (4.15). Note that if a player does not appear in one position, then the computation of a weighted centroid for that position will not be computed in the overall distance, see Equation (4.20).

In conclusion, Table 4.19, which shows the $C(Y_{(15)})$ compositions, and Table 4.20, which gives the distances (d_{pos1}) between the compositions for the players based on the team for the year 2015 are a demonstration of how the distance for the first types of positions are obtained. Here are the abbreviations for the players' names to be used in the following tables: P.P.: Paul Pogba, G.P.: Gerard Pique, D.A.: Dani Alves, L.M.: Lionel Messi, N.: Neymar, A.I.: Andres Iniesta, S.R.: Sergio Ramos, C.R.: Cristiano Ronaldo, J.R.: James Rodriguez, D.A.: David Alaba. The dissimilarity matrix results in Table 4.20 indicate that two centre defenders (Gerard Pique and Sergio Ramos) are position-wise the most similar players, whereas Dani Alves (right-defender) and Cristiano Ronaldo (forward) are the least similar ones.

Table 4.18: Summary of the distance measure for $C(Y_{(15)})$. w_{ijk} is the weight that is determined by proportions of the positions (k) in each side (j) for player i, $c(q^*_{ik})$ is the weighted mean of x_{ik} and y_{ik} coordinates, see Table 4.16, for the k^{th} position on the i^{th} player by using the great-circle distance to obtain a weighted mean, and z_{ik} is the z coordinate number for the k^{th} position on the i^{th} player.

Position	**Position-side**			**Computation**		z_{ik}
	Centre	*Left*	*Right*	$w_{ik} = \sum_{j=1}^{3} w_{ijk}$	Weighted means	
Defence	w_{i11}	w_{i21}	w_{i31}	$w_{i11} + w_{i21} + w_{i31}$	$c(q^*_{i1})$	$z_{i1} = 0$
Defensive Midfielder	w_{i12}	w_{i22}	w_{i32}	$w_{i12} + w_{i22} + w_{i32}$	$c(q^*_{i2})$	$z_{i2} = 1$
Midfielder	w_{i13}	w_{i23}	w_{i33}	$w_{i13} + w_{i23} + w_{i33}$	$c(q^*_{i3})$	$z_{i3} = 2$
Attacking Midfielder	w_{i14}	w_{i24}	w_{i34}	$w_{i14} + w_{i24} + w_{i34}$	$c(q^*_{i4})$	$z_{i4} = 3$
Forward	w_{i15}	w_{i25}	w_{i35}	$w_{i15} + w_{i25} + w_{i35}$	$c(q^*_{i5})$	$z_{i5} = 4$

Table 4.19: $C(Y_{(15)})$ compositions for the players based on the team of the year 2015.

Players	**DC**	**DL**	**DR**	**DMC**	**DML**	**DMR**	**MC**	**ML**	**MR**	**AMC**	**AML**	**AMR**	**FW**	**FWL**	**FWR**
P. P.	0.00	0.00	0.00	0.00	0.00	0.00	0.79	0.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00
G. P.	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D. A.	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
L. M.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.46	0.00	0.51
N.	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.97	0.00
A. I.	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
S. R.	0.97	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
C. R.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.33	0.62	0.00
J. R.	0.00	0.00	0.00	0.00	0.00	0.00	0.48	0.10	0.28	0.07	0.00	0.00	0.00	0.00	0.07
D. A.	0.36	0.20	0.00	0.12	0.00	0.00	0.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 4.20: Distances (d_{pos1}) between $C(Y_{(15)})$ compositions for the selected players.

Players	P.P.	G.P.	D.A.	L.M.	N.	A.I.	S.R.	C.R.	J.R.	D.A.
P. P.	0.00	1.08	1.50	1.20	1.37	0.08	1.05	1.24	0.25	0.63
G. P.	1.08	0.00	0.50	2.25	2.45	1.00	0.03	2.32	1.32	0.47
D. A.	1.50	0.50	0.00	2.31	2.45	1.50	0.53	2.47	1.51	0.88
L. M.	1.20	2.25	2.31	0.00	0.37	1.25	2.21	0.24	0.97	1.81
N.	1.37	2.45	2.45	0.37	0.00	1.45	2.42	0.17	1.16	1.98
A. I.	0.08	1.00	1.50	1.25	1.45	0.00	0.97	1.32	0.32	0.71
S. R.	1.05	0.03	0.53	2.21	2.42	0.97	0.00	2.29	1.29	0.44
C. R.	1.24	2.32	2.47	0.24	0.17	1.32	2.29	0.00	1.05	1.85
J. R.	0.25	1.32	1.51	0.97	1.16	0.32	1.29	1.05	0.00	0.87
D. A.	0.63	0.47	0.88	1.81	1.98	0.71	0.44	1.85	0.87	0.00

Position variables-2

In the previous distance structure, I design a distance measure which includes geographical information on the players on the field. In this section, the information is binary, which means that weighting computation is not necessary, but geographical information should again be incorporated. The proposed distance structure here is designed in another system in which binary and geographic information are combined. The idea will not only contribute another perspective from a positional point of view, but also represent binary information in a more effective way.

Hennig and Hausdorf (2006) proposed a new dissimilarity measure between species distribution areas, and argued in their application on presence-absence data of species in regions. They stated that if two species A and B are present on two small disjoint areas, they are very dissimilar, but both should be treated as similar to a species C covering a larger area that includes both A and B if clusters are to be interpreted as species grouped together. In this article, species distribution data is presented in a certain geographic region, and the *'geco coefficient'* (the name comes from "geographic distance and congruence"), which is the geographical distance between units, is also introduced as a new dissimilarity measure.

In fact, the idea comes from the Kulczynski coefficient (Kulczynski, 1927a), see Equation (4.21).

$$d_k(A_1, A_2) = 1 - \frac{1}{2}\left(\frac{|A_1 \cap A_2|}{|A_1|} + \frac{|A_1 \cap A_2|}{|A_2|}\right), \tag{4.21}$$

where A_i is the geographical region of i^{th} object, and $|A_i|$ denotes as the number of elements in the geographical region of i^{th} object. Then, Hennig and Hausdorf (2006) designed the geco coefficient in which the Kulczynski coefficient and the geographical information are incorporated. The general definition is given by

$$d_G(A_1, A_2) = \frac{1}{2}\left(\frac{\sum_{a\in A_1} \min\limits_{b\in A_2} u\left(d_R(a,b)\right)}{|A_1|} + \frac{\sum_{b\in A_2} \min\limits_{a\in A_1} u\left(d_R(a,b)\right)}{|A_2|}\right), \tag{4.22}$$

where u is a monotonically increasing transformation with $u(0) = 0$, and $d_R(a, b)$ is the distance between the objects a and b. In Equation (4.22), when computing the overall distance, the idea is not to incorporate the number of common absences. Here I define the coefficient in terms of $Y_{(11)}$, so that species are replaced by players, and geographical locations are replaced by positions. The concept is very similar; that is, the common absences will be ignored, because it will make such players more similar, even though they do not appear in those positions. For instance, Table 4.21

presents two players who do not have any historical information in the position of 'DC', 'DL', 'DR', 'DM', and 'FW', so that $d_R(a,b)$ for these positions will not be computed. If the players have played in common positions, then $d_R(a,b) = 0$, where both a and b represent presences in the relevant position(s). For the case of one absence in the first player and one presence in the second player for the same position, $d_R(a,b)$ will be computed by using the distance measures in Table 4.22, e.g., 'ML' position in Table 4.21. The distance between each position is determined by the similar idea that I previously described for $Y_{(15)}$; that is, the difference between each side in one position (centre, left, right) as well as the difference between the adjacent positions (D, DM, M, AM, FW) will be one unit, see Figure 4.12.

Table 4.21: An example for explaining the distance structure of $X_{pos}^{(11)}$ variables

Players	**Position**										
	DC	*DL*	*DR*	*DMC*	*MC*	*ML*	*MR*	*AMC*	*AML*	*AMR*	*FW*
1	0	0	0	0	0	1	1	0	1	1	0
2	0	0	0	0	1	0	0	1	1	1	0

I use a very similar distance measure to that of Equation (4.22) for computing the difference between players in $Y_{(11)}$ variables, but the transformation u is not adopted in the formula, since it is not relevant to my argument. By excluding the transformation function, the following distance measure will be adopted for $Y_{(11)}$ variables.

$$d_{pos2}(A_1, A_2) = \frac{1}{2}\left(\frac{\sum_{a\in A_1}\min_{b\in A_2} d_R(a,b)}{|A_1|} + \frac{\sum_{b\in A_2}\min_{a\in A_1} d_R(a,b)}{|A_2|}\right), \tag{4.23}$$

In conclusion, Table 4.23, which shows the $Y_{(11)}$ compositions, and Table 4.24, which gives the distances (d_{pos2}) between the compositions for the players based on the team for the year 2015 are demonstration of how the distance for the second types of positions are obtained. The d_{pos2} results indicate that three defenders (Gerard Pique, Sergio Ramos and Dani Alves) are the similar ones, while the dissimilarity between the forward players (Cristiano Ronaldo, Lionel Messi and Neymar) and these three defenders give the highest values, which makes sense in terms of the differences of these players' positions.

Table 4.22: Distances between each position for $Y_{(11)}$ variables. Here the values are obtained by using Euclidean geometry based on Figure 4.12 at the bottom. The positions with $(*)$ are only for use in d_{pos1}, see Equation (4.19).

$\mathbf{d_R(a,b)}$	DC	DL	DR	DMC	DML*	DMR*	MC	ML	MR	AMC	AML	AMR	FW	FWL*	FWR*
DC	0	1	1	1	$\sqrt{2}$	$\sqrt{2}$	2	$\sqrt{5}$	$\sqrt{5}$	3	$\sqrt{10}$	$\sqrt{10}$	4	$\sqrt{17}$	$\sqrt{17}$
DL	1	0	1	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{10}$	3	$\sqrt{10}$	$\sqrt{17}$	4	$\sqrt{17}$
DR	1	1	0	$\sqrt{2}$	$\sqrt{2}$	1	$\sqrt{5}$	$\sqrt{5}$	2	$\sqrt{10}$	$\sqrt{10}$	3	$\sqrt{17}$	$\sqrt{17}$	4
DMC	1	$\sqrt{2}$	$\sqrt{2}$	0	1	1	1	$\sqrt{2}$	$\sqrt{2}$	2	$\sqrt{5}$	$\sqrt{5}$	3	$\sqrt{10}$	$\sqrt{10}$
DML*	$\sqrt{2}$	1	$\sqrt{2}$	1	0	1	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{10}$	3	$\sqrt{10}$
DMR*	$\sqrt{2}$	$\sqrt{2}$	1	1	1	0	$\sqrt{2}$	$\sqrt{2}$	1	$\sqrt{5}$	$\sqrt{5}$	2	$\sqrt{10}$	$\sqrt{10}$	3
MC	2	$\sqrt{5}$	$\sqrt{5}$	1	$\sqrt{2}$	$\sqrt{2}$	0	1	1	1	$\sqrt{2}$	$\sqrt{2}$	2	$\sqrt{5}$	$\sqrt{5}$
ML	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	1	0	1	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	2	$\sqrt{5}$
MR	$\sqrt{5}$	$\sqrt{5}$	2	$\sqrt{2}$	$\sqrt{2}$	1	1	1	0	$\sqrt{2}$	$\sqrt{2}$	1	$\sqrt{5}$	$\sqrt{5}$	2
AMC	3	$\sqrt{10}$	$\sqrt{10}$	2	$\sqrt{5}$	$\sqrt{5}$	1	$\sqrt{2}$	$\sqrt{2}$	0	1	1	1	$\sqrt{2}$	$\sqrt{2}$
AML	$\sqrt{10}$	3	$\sqrt{10}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	1	0	1	$\sqrt{2}$	1	$\sqrt{2}$
AMR	$\sqrt{10}$	$\sqrt{10}$	3	$\sqrt{5}$	$\sqrt{5}$	2	$\sqrt{2}$	$\sqrt{2}$	1	1	1	0	$\sqrt{2}$	$\sqrt{2}$	1
FW	4	$\sqrt{17}$	$\sqrt{17}$	3	$\sqrt{10}$	$\sqrt{10}$	2	$\sqrt{5}$	$\sqrt{5}$	1	$\sqrt{2}$	$\sqrt{2}$	0	1	1
FWL*	$\sqrt{17}$	4	$\sqrt{17}$	$\sqrt{10}$	3	$\sqrt{10}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	1	0	1
FWR*	$\sqrt{17}$	$\sqrt{17}$	4	$\sqrt{10}$	$\sqrt{10}$	3	$\sqrt{5}$	$\sqrt{5}$	2	$\sqrt{2}$	$\sqrt{2}$	1	1	1	0

Table 4.23: $Y_{(11)}$ binary variables for the players based on the team of the year 2015.

Players	**DC**	**DL**	**DR**	**DMC**	**MC**	**ML**	**MR**	**AMC**	**AML**	**AMR**	**FW**
Paul Pogba	0	0	0	1	0	1	0	0	0	0	0
Gerard Pique	1	0	0	0	0	0	0	0	0	0	0
Dani Alves	1	0	1	0	0	0	0	0	0	0	0
Lionel Messi	0	0	0	0	0	0	0	1	0	0	0
Neymar	0	0	0	0	0	0	0	1	1	1	1
Andres Iniesta	0	0	0	0	0	0	0	1	1	1	0
Sergio Ramos	1	0	1	0	0	0	0	0	0	0	0
Cristiano Ronaldo	0	0	0	0	0	0	0	0	1	1	1
James Rodriguez	0	0	0	0	0	0	0	1	1	1	0
David alaba	1	1	0	0	1	1	0	0	0	0	0

Table 4.24: Distances (d_{pos2}) between $Y_{(11)}$ binary variables for the players based on the team of the year 2015.

Players	P.P.	G.P.	D.A.	L.M.	N.	A.I.	S.R.	C.R.	J.R.	D.A.
Paul Pogba	0.00	1.62	1.72	2.05	1.94	1.72	1.72	2.02	1.72	0.84
Gerard Pique	1.62	0.00	0.25	3.39	3.33	3.11	0.25	3.44	3.11	0.65
Dani Alves	1.72	0.25	0.00	3.41	3.35	3.11	0.00	3.43	3.11	1.07
Lionel Messi	2.05	3.39	3.41	0.00	0.14	0.40	3.41	0.36	0.40	2.51
Neymar	1.94	3.33	3.35	0.14	0.00	0.16	3.35	0.12	0.16	2.42
Andres Iniesta	1.72	3.11	3.11	0.40	0.16	0.00	3.11	0.38	0.00	2.19
Sergio Ramos	1.72	0.25	0.00	3.41	3.35	3.11	0.00	3.43	3.11	1.07
Cristiano Ronaldo	2.02	3.44	3.43	0.36	0.12	0.38	3.43	0.00	0.38	2.51
James Rodriguez	1.72	3.11	3.11	0.40	0.16	0.00	3.11	0.38	0.00	2.19
David Alaba	0.84	0.65	1.07	2.51	2.42	2.19	1.07	2.51	2.19	0.00

4.3 Aggregation of Distances

In Section 4.2, I discussed different distance measures for the variables of different kinds. The Manhattan distance was selected for aggregation of upper level count variables, lower level compositions, and some other variables. All these performance variables are represented in one distance measure, which I called as Performance distance. For team and league variables, the Manhattan distance was adopted in the sense that $x_{(l)}$ and $x_{(tp)}$ are combined prior to taking the absolute value differences. For two position variables, I construct one new dissimilarity measure between players positions, and the dissimilarity measure for the second positional variables is designed based on Hennig and Hausdorf (2006)'s *'geco coefficient'*, where species are replaced by players.

In order to find a single distance matrix to be used for clustering and visualisation, all these different types of distance matrices should be aggregated. The Gower dissimilarity, see Equation (3.30), is not applicable here, because it aggregates variable-wise distances d_k (See Section 3.5). The Euclidean aggregation, see Equation (3.32), is not adopted because the distances of the variables considered in this research are non-Euclidean. If researchers are interested in utilising different types of distance measures, aggregation of them can be obtained by Equation (3.33). Two aspects should be discussed: 1) how to choose an appropriate standardisation technique to make different distances comparable, 2) what subjective weights should be assigned to those distances after the selected standardisation method is applied.

In Section 3.5, I discussed that the distributional shape of dissimilarities can be helpful to find a proper standardisation approach. Two dissimilarity measures from the position variables seem to

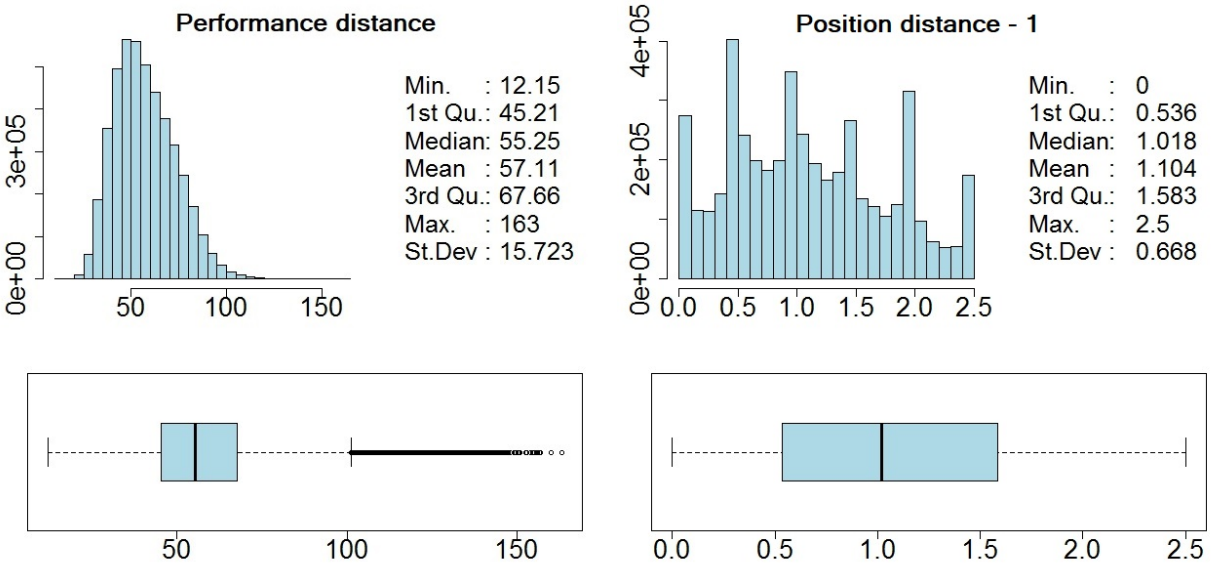
be approximately uniformly distributed, see Figure 4.13b and 4.13c. Performance distance, which is obtained by the upper level count variables, lower level compositions and some other variables, shows a more or less symmetrical shape, see Figure 4.13a. The distance from the league and the team variables, see Figure 4.13d, is right-skewed distributed. Based on the argument in Section 3.5, the average absolute deviation can be chosen as the proper standardisation for combining these dissimilarities, because first, I used the L_1 distance for aggregating variable-wise distances, so that it would be consistent to adopt the same standardisation technique for aggregating different types of distances; second, if I want every distance values to have the same impact, the variation of different distances should be computed by their absolute values of difference from the median, see Section 4.1.3.

However, I also want to investigate what choice of standardisation technique for dissimilarity aggregation is convenient by performing some kind of sensitivity analysis. The term "Sensitivity analysis" here means to observe whether the correlation between the vector of dissimilarities for one year and the vector of dissimilarities for the next year is high enough when applying a suitable standardisation technique. Different standardisation techniques will be analysed for this aim. In this sense, the standardisation technique with the highest correlation can be chosen as the proper standardisation for aggregation of the dissimilarities.

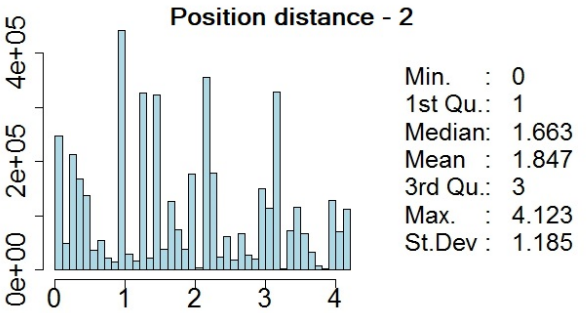
This correlation analysis is similar in concept to the Pearson Gamma index (Hubert and Schultz, 1976), which is simply the correlation of the vector of dissimilarities with the vector of "clustering induced dissimilarity", see Section 6.1.3 for more information. The only difference is that we inspect the correlation of two vectors of dissimilarities over two consecutive years. Furthermore, Hausdorf and Hennig (2005) carried out a very similar analysis by checking the correlation of two vectors of dissimilarities from different choices of dissimilarity matrices.

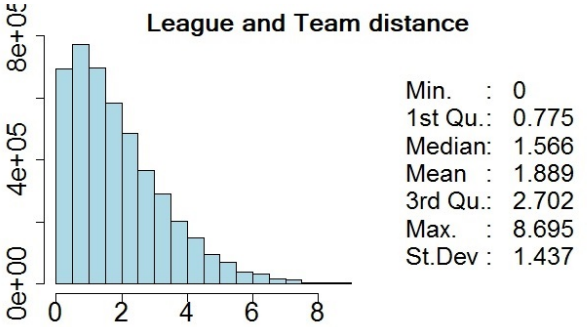
For this analysis, I collected all the available historical information of the players on different football seasons. For practical reason, [3] the part of the data set (players who played more than 1000 minutes, because the players information with more minutes is more consistent in terms of reflecting a proper representation of their information) was collected instead of using the whole data set. The collection of players historical data set is based on the seasons from 2009-2010 to 2016-2017. Each seasonal year contains a different data set, and the size of each data set varies, since some players are retired or do not play in certain seasons. Then, the correlation between the

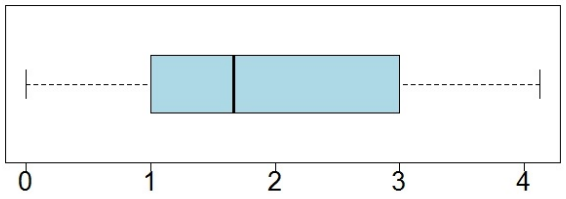
[3] At the beginning of this project, the 2014-2015 football season data set is simply collected from the website, www.whoscored.com by applying the concept of web-scraping, which is a computer software technique of automatically extracting information from websites. The idea of sensitivity analysis is considered at the end of the project, and at that time the website is designed to prevent web scraping, so that the collection of all the available historical information of the players is considered extracting manually, which takes a long time.

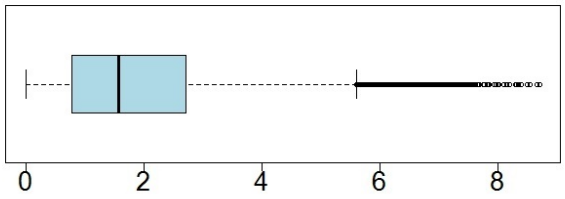


(a) (b)









(c) (d)

Figure 4.13: Summary of the distances

vector of dissimilarities for two consecutive years is computed only for players who played in both seasons.

Table 4.25 shows the summary of the correlations between the vector of dissimilarities for two consecutive years over different standardisation techniques. I additionally check how the correlation differs when transformation ($\log(x + c)$) is applied, and the constant value, c is computed in the same way as explained in Section 4.1.2. The same values are used for the weights of the variables, see Table 4.11. The summary results, which contain 6 different statistics for 8 data points, indicate that range standardisation with transformation gives the best values for the aggregation of four dissimilarities. Thus, our conclusion is to adopt this selection for the further analysis, see Chapter 8.

Table 4.25: Summary of the correlations between the vector of dissimilarities for two consecutive years from 2009-2010 to 2016-2017 football seasons over different standardisation techniques with or without transformation

Standardisation Technique	Transformation	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
AAD Standardisation	No	0.771	0.810	0.827	0.830	0.851	0.882
	Yes	0.784	0.810	0.830	0.831	0.852	0.878
Z-score Standardisation	No	0.767	0.808	0.828	0.830	0.852	0.882
	Yes	0.790	0.817	0.835	0.838	0.860	0.884
Range Standardisation	No	0.812	0.849	0.870	0.870	0.885	0.912
	Yes	0.851	0.871	0.882	0.884	0.900	0.918

Once range standardisation with transformation was applied to each distance, the appropriate weights should be assigned to each dissimilarity. The dissimilarity weight for the upper level count variables, the lower level compositions, and some other variables are assigned as the total weights of the variables in Table 4.11, which is 43.5, because I want to preserve the weights from those variables. For the position variables, $C(Y_{(15)})$ and $Y_{(11)}$ have 15 and 11 variables, respectively. The weights for these dissimilarities can be assigned as the total number of those variables, but at the same time I want two types of the position variables to have the same impact on the final dissimilarity matrix, so that the decision of the weight assignment for the two position distance measures is to assign the average of the total number of the variables for d_{pos1} and d_{pos2}, which is 13. Assigning the total number (3) of the team and league variables can be adopted here as well, but a player can have very different performance, if he transfers from a strong team to a less strong team, or vice versa; therefore, I need to assign weight to the dissimilarity such that players are discriminated in terms of their team and league information. I believe the weight should be higher than 3, but it should not be much higher, because otherwise players can be dominated by these

variables. Thus, I have decided to assign a weight that is equal to twice the total number of the team and league variables. Again, weight assignment is made based on my football interpretation, but this can be differentiated by users (e.g., managers or scouts, etc.) in terms of which dissimilarity information they are interested in more.

In conclusion, the overall dissimilarity to be used for clustering and MDS is defined as follows:

$$d_{fin}(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{4} \frac{w_k * d_k(\mathbf{x}, \mathbf{y})}{s_k}, \tag{4.24}$$

where the definitions of the parameters are given in Table 4.26.

Table 4.26: Definition of the parameters

	Performance	**Position-1**	**Position-2**	**Team and League**
Distances	$d_1(\mathbf{x}, \mathbf{y})$	$d_2(\mathbf{x}, \mathbf{y})$	$d_3(\mathbf{x}, \mathbf{y})$	$d_4(\mathbf{x}, \mathbf{y})$
Weights	$w_1 = 43.5$	$w_2 = 13$	$w_3 = 13$	$w_4 = 6$
s_k	Range	Range	Range	Range

4.4 Distance query

At the end of the distance construction of football players performance data, a final dissimilarity matrix was obtained. In further analysis, the aim is to determine how the grouping structure of football players should be conducted. Prior to this goal, I would like to show some examples that simply explore such players based on the similarity results obtained from the dissimilarity matrix. This sort of implementation can be characterised as "distance queries" of a player; in other words, the aim is to find players that have the smallest distances to a player of interest.

Figure 4.14, 4.15 and 4.16 display such examples with the distance queries of three famous players. `R Shiny` (Chang et al., 2015), which is simply the user interface of the `R` software, is used for the visualisation of distance queries of such a player. More information about `R Shiny` application can be found in Section 6.3. These examples are based on the 2016-2017 football season data set, containing approximately 1000 players. On the left hand side of the same figure, the weight assignment for different variables is designed, so that users can have the flexibility to play with variable weights of their interest, whereas on the right hand side, some characteristic variables are shown in filtering format, so that managers or scouts may consider such players with some kinds of range values for the variables of their interest. In addition, different standardisation

techniques for aggregating the dissimilarity matrix are inserted on the right-bottom segment of the figure. The box-plots in the middle show the smallest distances (first, second, third smallest and so on) of every player, so that we can compare all the smallest distances with the distance between the players of interest. In this respect, the similarity of players can be interpreted as very similar if the red point is on the left side of the box, or less similar if it is on the right side of the box. For example, the most similar player to Lionel Messi is Paolo Dybala, but their similarity is not very strong due to the position of the red point on the box plot. The same argument can be made for the similarity between Neymar and Memphis Depay. On the other hand, the similarity degree between Cristiano Ronaldo and Robert Lewandowski is considerably larger than the similarity of the other players on Figure 4.14 and 4.16. The results indicate that these three famous players (Lionel Messi, Cristiano Ronaldo and Neymar) can be interpreted as special players because even the most similar ones to these players are considerably far away from these three players based on the result of the dissimilarity matrix. Distance query for players discovery can be adequate to managers and scouts, but one could also consider clustering players to explore a list of players who are similar to a player of interest. Therefore, cluster analysis of football players will be investigated more in further analysis of this thesis.

One of the goals of this project is to use the distance query application to a football team in order to explore such specific players of their interest. In this respect, I contacted one Turkish team, Istanbul Basaksehir football club. We have been working together for the 2017-2018 football season to discover such specific players of interest for the next football season.

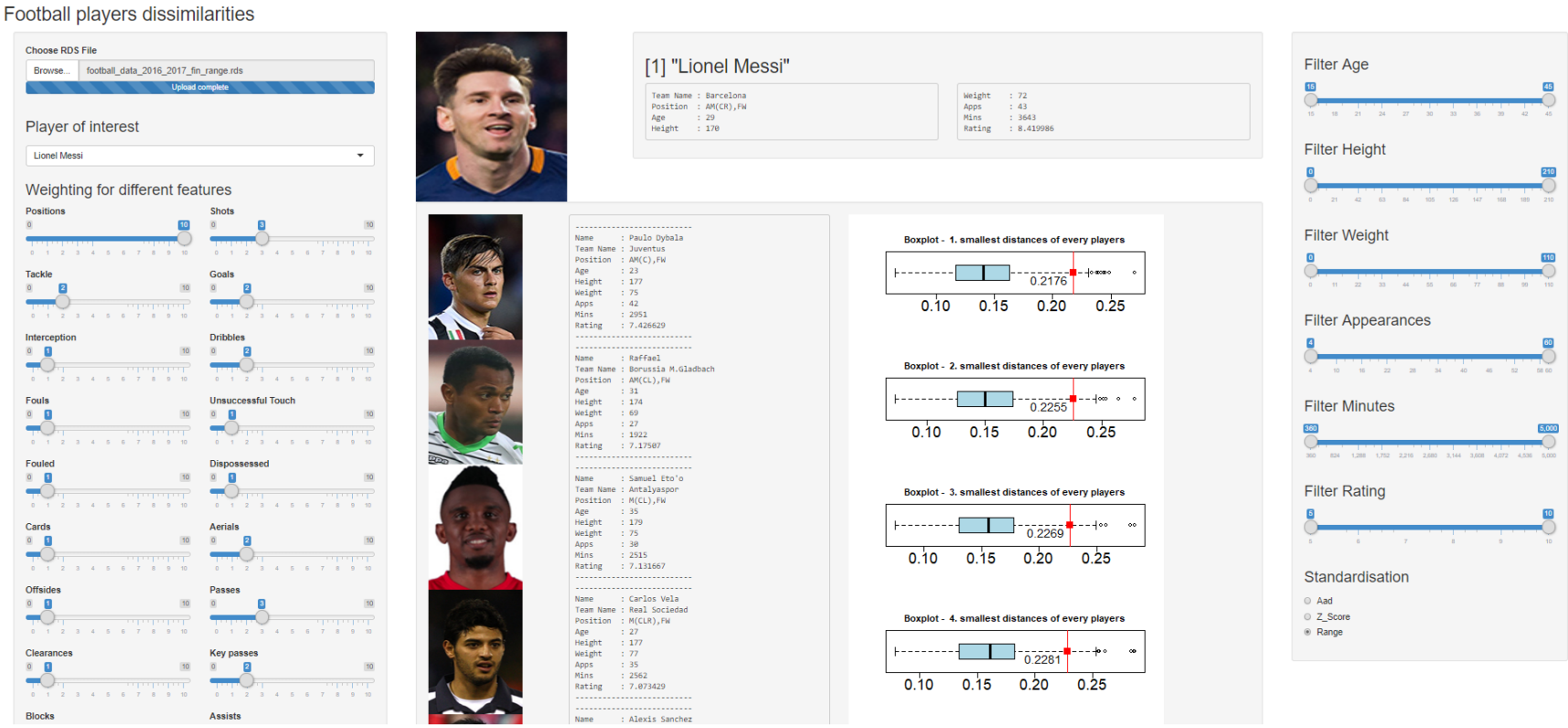


Figure 4.14: Distance query examples with the application of `R Shiny` - Lionel Messi

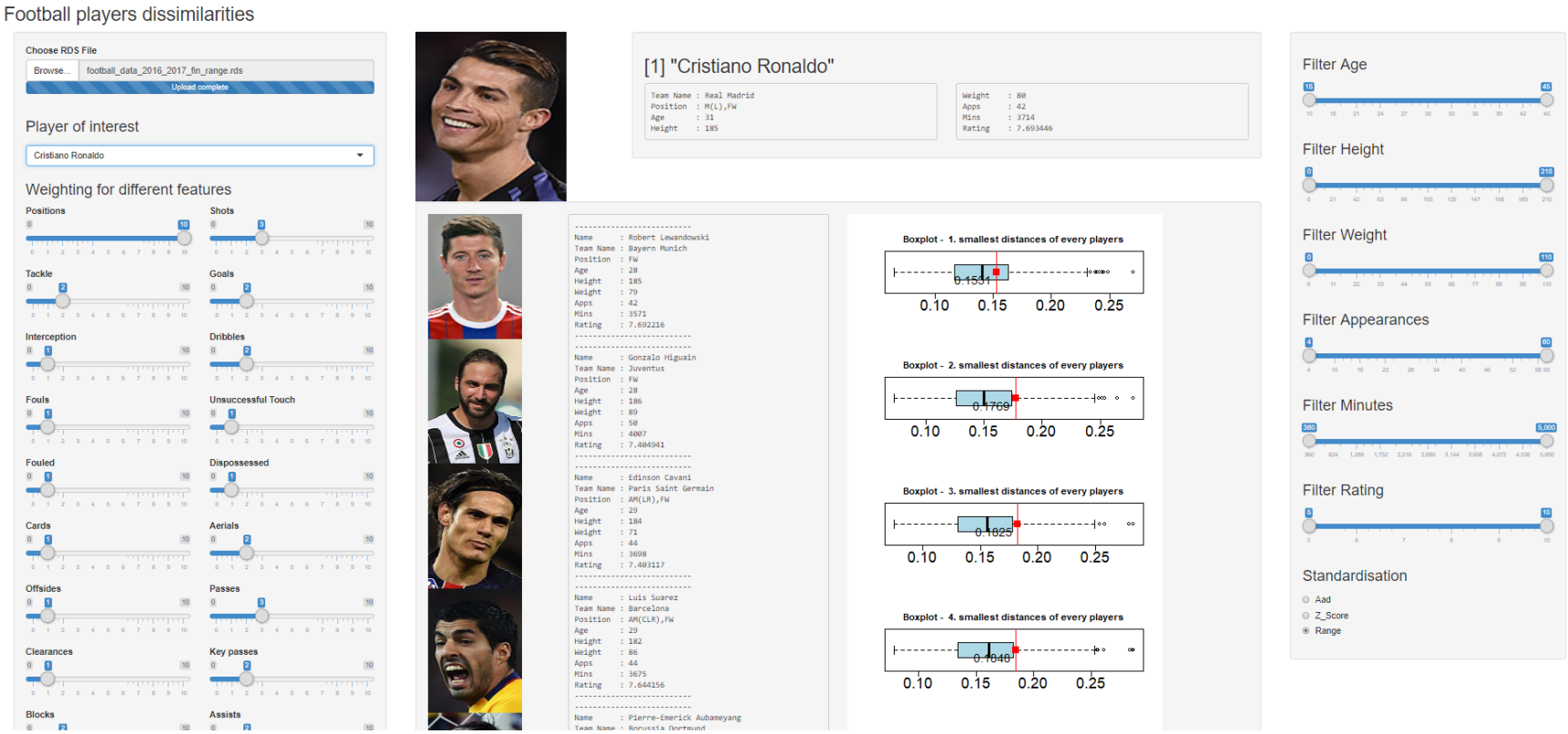


Figure 4.15: Distance query examples with the application of `R Shiny` - Cristiano Ronaldo

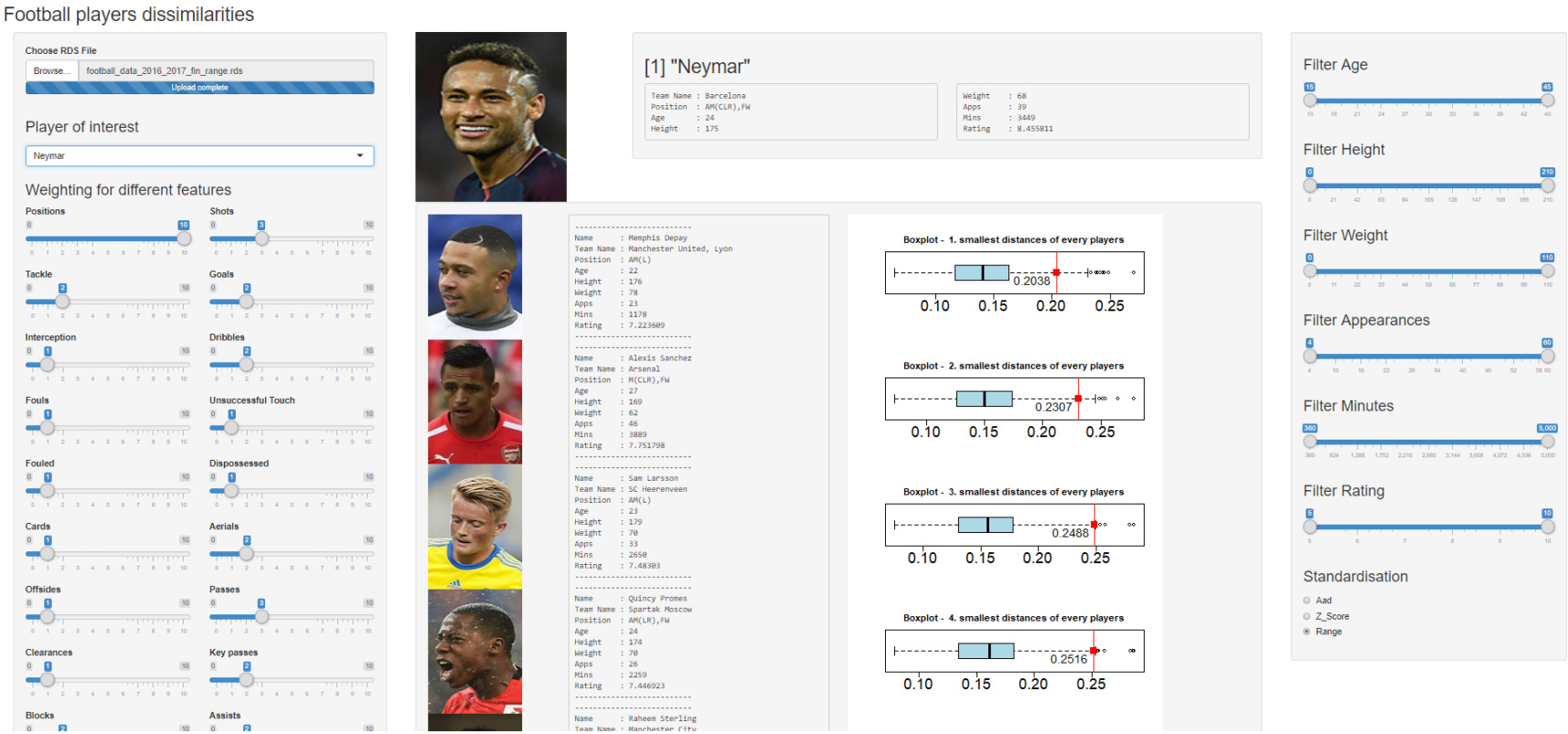
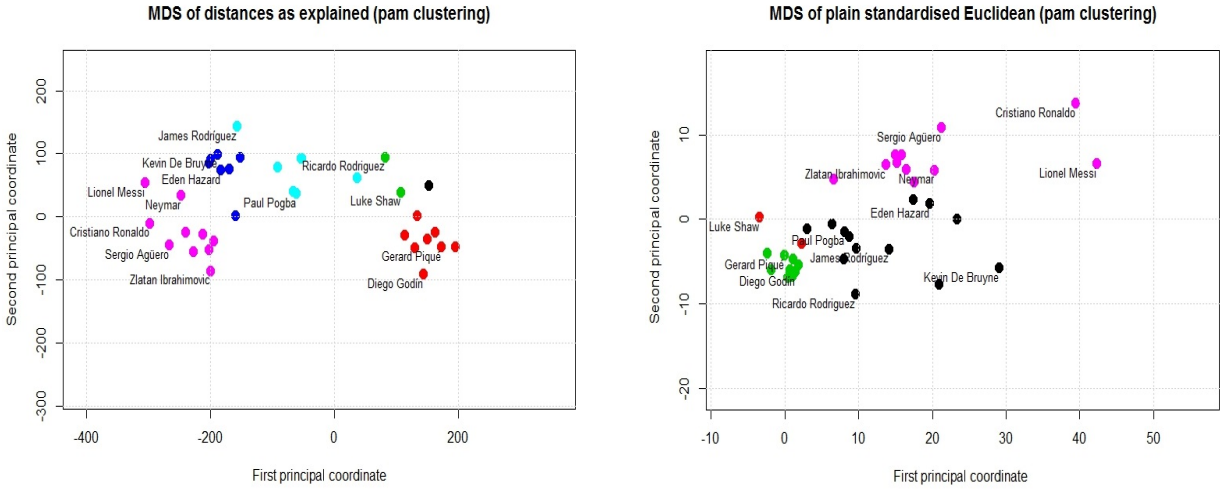


Figure 4.16: Distance query examples with the application of `R Shiny` - Neymar

4.5 Preliminary Results

In order to show a preliminary analysis for all the players in the data set based on the constructed distance including all variables, I show a) Kruskal's Multidimensional scaling (MDS) (Kruskal, 1964a) of the distances constructed as explained here, and b) Kruskal's MDS of plain standardised Euclidean distances for all variables, but only a test subset of players are used for visualisation, and I adopt PAM clustering (Kaufman and Rousseeuw, 1990) with number of clusters $K = 6$. PAM clustering (See Section 5.2.2) and Kruskal's MDS (See Section 5.3.2) will be scrutinised in the next chapter. Note that this is just a preliminary analysis, and finding an optimal K and comparing different clustering methods are discussed in the next chapters.

According to Figure 4.17b, Ricardo Rodriguez (*left back*) and Eden Hazard (*attacking midfielder*) are quite different, but in the same cluster in Figure 4.17a. Since both Ricardo Rodriguez and Luke Shaw play in the left back, they can be expected to be similar, which they are according to the distances constructed here, but in different clusters in plain Euclidean solution.



(a) (b)

Figure 4.17: MDS and PAM clustering ($K = 6$) for test subset of players based on all variables.

The result implies that clustering and mapping multivariate data are strongly affected by preprocessing decisions such as the choice of variables, transformation, standardisation, weighting and design of dissimilarity measures. The variety of options is huge, but the fundamental concept is to match the "interpretative dissimilarity" between objects as well as possible by the formal dissimilarity between objects. This is an issue involving subject-matter knowledge that cannot be decided by the data alone.

CHAPTER 5

OVERVIEW OF CLUSTER ANALYSIS

So far, I have explained the use of a dissimilarity measure with the aim of mapping and clustering football players' information in order to explore their similarity structure. In this respect, first the rules of football as well as its history were introduced, and some literature regarding football statistics, specifically in terms of clustering, were reviewed. Second, methodologies and literature were scrutinised for the sake of dissimilarity construction with data pre-processing steps, and in Chapter 4 I have pre-processed the football data and presented a dissimilarity measure which reflects players' characteristics by using all the available information in the data set.

For the second part of the thesis, my aim is to determine how to find an appropriate grouping structure between players and to present the structure in a appropriate mapping scheme that provides a more informative guide to the practitioner than a dissimilarity matrix of players. To achieve this aim, *Cluster analysis* will be adopted for partitioning football players, and *Multidimensional scaling* will be used for mapping and visualising players.

5.1 Introduction

In Section 3.1, a short overview of cluster analysis is given and the strategy of cluster analysis is listed in seven steps. The first four of them are presented in detail in Chapter 3. As a continuation of those steps this chapter reviews the literature and resources of cluster analysis and demonstrates the relevant techniques which are to be used in the application part of the thesis (Chapter 7). Two popular dimension reduction techniques (PCA and MDS) are introduced for visualising the clustering points in order to distinguish their mutual relations. Finally, some external clustering validation techniques are reviewed to validate clustering results. Note that the subjects of estimating the number of clusters and internal clustering validation indexes are included in a separate chapter

(Chapter 6), since the majority of the work in these subjects is original to this thesis.

Prior to introducing some relevant methodologies, the formal definition of cluster analysis is given in Definition 5.1.1. The notation in the definition will be used in further sections.

Definition 5.1.1. Given a set of objects, also called a data set, $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ containing n data points, the technique of cluster analysis is to group them into K disjoint subsets of $\mathcal{X}$, denoted by $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$, where $n_1, n_2, \ldots, n_K$ are the number of objects in each cluster. If a data point $\mathbf{x}_i$ belongs to *cluster* C_k. $1 \leq k \leq K$, then the label of $\mathbf{x}_i$ is k, and cluster labels are denoted by $\mathcal{L} = \{l_1, l_2, \ldots, l_n\}$; hence, $1 \leq l_i \leq K$, where $i = 1, \ldots, n$.

5.2 Clustering Methods

The choice of an appropriate technique is strongly dependent on the aim of clustering (Hennig, 2015b). The distinction between clustering methodologies can be interpreted in several different formats, such as what type of clustering, what data format, what clustering criterion or what probabilistic regime the users are interested in. The different categories in terms of clustering types are explained in greater detail in Hennig and Meila (2015).

Interpretation of data for choosing a clustering algorithm is a fundamentally important step in cluster analysis, and researchers should take a broad perspective when choosing a clustering method for their analysis. The following sections are concerned with the clustering methodologies used for this study.

5.2.1 Centroid-Based Clustering

The idea behind centroid based clustering is simply to find K centroid objects in such a way that all other objects are gathered around those K centroids in an optimal manner. Clustering of this type requires every object in a cluster to be close to the centroid, which tends to restrict the cluster shapes.

K-means clustering, which was first proposed by Steinhaus (1957), is a method for multivariate data in $\mathbb{R}^p$ $(p > 1)$ and is based on the least squares principle. For finding K clusters in a data set (K is the number of clusters and needs to pre-specified), K "centroid points" are positioned in $\mathbb{R}^p$, and every observation is assigned to the closest centroid point, in such a way that the sum of all squared Euclidean distances of the observations to the centroids is minimised. In order to achieve this, the centroid points have to be the (multivariate) means of the observations assigned to them, i.e. the clusters of which they are the centroids, hence the name K-means.

Four different types of K-means algorithms are available in the R software, and more information about the those listed algorithms can be found in Hartigan and Wong (1979), Lloyd (1982), Forgey (1965) and MacQueen et al. (1967). Lloyd (1982) presented the simplest version of these algorithms, which is described in Algorithm 5, Appendix A. The definition of K-means square error criterion to be used in the K-means algorithm is given below.

Definition 5.2.1. Let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ contain n observations, and $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$ be the K disjoint clusters of $\mathcal{X}$ with a set of cluster labels, $\mathcal{L} = \{l_1, l_2, \ldots, l_n\}$. The K-means square error criterion is defined by

$$\mathcal{S}(\mathcal{C}, \mathbf{m}_1, \ldots, \mathbf{m}_K) = \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{m}_{l_i}\|^2, \tag{5.1}$$

where $\mathbf{m}_1, \ldots, \mathbf{m}_K$ are the cluster centroids.

As specified in Algorithm 5, see Appendix A, the K-means algorithm does not explicitly use pairwise distances between data points. It amounts to repeatedly assigning points to the closest centroid thereby using Euclidean distance from data points to a centroid. However, the squared error criterion can also be formulated as:

$$\sum_{\mathbf{x}_i \in C_k} \|\mathbf{x}_i - \mathbf{m}_{C_k}\|^2 = \frac{1}{2n_k} \sum_{\mathbf{x}_i, \mathbf{x}_j \in C_k} d(\mathbf{x}_i, \mathbf{x}_j)^2. \tag{5.2}$$

where $n_k = |C_k|$ is the number of objects in C_k. Equation (5.2) illustrates that the K-means algorithm can also be used as a distance-based clustering technique, where $d(\mathbf{x}_i, \mathbf{x}_j)^2$ is the squared Euclidean distance. Although there are also many other equivalent reformulations of the K-means criterion, which give different perspectives leading to different approaches to optimization of the criterion, see Mirkin (2015), this research tends to focus more on the cluster validation indexes than the clustering algorithms in detail.

K-means works well when the clusters are in compact shapes that are well separated from one another. K-means has the disadvantage that it is not well suited for non-spherical shape clusters because distances in all directions from the center are treated in the same way. It is also sensitive to outliers, since the squared Euclidean distance penalises heavily large distances within clusters.

5.2.2 K-medoids clustering

Medoids are similar in concept to means or centroids, but medoids are always members of the data set. **Partitioning Around Medoids (PAM)** (Kaufman and Rousseeuw, 1990) is one of the most

well-known K-medoids clustering algorithms. PAM, which has a strong connection to K-means, aims at finding centroid objects for each of a given fixed number of K clusters. These centroid objects cannot be mean vectors, because the computation of mean vectors requires a Euclidean space and this cannot be done for general dissimilarity data (Steinley, 2015). Instead, for PAM, the centroid objects ("medoids") are members of the data set.

Definition 5.2.2. Let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be characterised by a dissimilarity measure $d : \mathcal{X}^2 \to \mathbb{R}_0^+$, and let $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$ be the K disjoint clusters of $\mathcal{X}$ with a set of cluster labels, $\mathcal{L} = \{l_1, l_2, \ldots, l_n\}$.

The **Partitioning Around Medoids (PAM)** clustering of $\mathcal{X}$ for given fixed K is defined by choosing $\mathbf{m}_1^{PK}, \ldots, \mathbf{m}_K^{PK}$ in such a way that they minimise

$$\mathcal{T}(\mathcal{C}, \mathbf{m}_1, \ldots, \mathbf{m}_K) = \sum_{i=1}^{n} d(\mathbf{x}_i, \mathbf{m}_{l_i}), \tag{5.3}$$

where $\mathbf{m}_1, \ldots, \mathbf{m}_K$ are the cluster medoids, and $d(\mathbf{x}_i, \mathbf{m}_{l_i})$ is the dissimilarity measure between objects $\mathbf{x}_i$ and medoids $\mathbf{m}_{l_i}$. Then, the PAM algorithm proceeds as shown in Algorithm 6, see Appendix A.

PAM has the flexibility that it can use a general dissimilarity as an input. PAM has the advantage of not penalising large within-cluster distances as strongly as K-means, so that it is somewhat better at finding non-spherical clusters, as long as the deviation from a spherical shape is not too strong. Hence, it can be interpreted as more robust than the K-means in the presence of noise and outliers. PAM has a disadvantage that it can be computationally expensive, especially for large data sets. Note that instead of using PAM for dealing with large data sets, another K-medoids algorithm, CLARA (Clustering Large Applications) is suggested, which works by applying PAM to several smaller subsets of the data set and classifying all further points to their closest centroids, picking the best solution among these according to T. CLARA is designed only for Euclidean data. For more information, see Kaufman and Rousseeuw (1990).

5.2.3 Hierarchical methods

The aim of hierarchical clustering is to set up a hierarchy of clusters. The mathematical definition of a hierarchy can be given as follows:

Definition 5.2.3. A **hierarchy** is a sequence of partitions $\mathcal{C} = \bigcup_{k=1}^{K} \mathcal{C}_k$, where $\mathcal{C}_k$, $i = k, \ldots, K$ are partitions with $n_1 = |\mathcal{C}_1| > \ldots > n_K = |\mathcal{C}_K|$ ($|C|$ denoting the number of elements of C) so

that for $C_k \in \mathcal{C}_k$ and $C_h \in \mathcal{C}_h$ with $k < h$ either $C_k \cap C_h = C_k$ or $C_k \cap C_h = \emptyset$, so that the sets of the lower levels are subsets of the sets of the higher levels.

Hierarchies can be visualised as trees ("dendrograms", see Section 1.5 for the definition). For finding a partition, users should determine either a cutting point (height of trees) or number of clusters from dendrograms.

There are two main types of methods to set up a hierarchy: 1) **agglomerative methods** start from an initial low level hierarchy in which every observations is a cluster on its own, and proceed by merging clusters upward until everything is merged, 2) **divisive methods** start with the whole data set as a single cluster and proceed downward by dividing clusters until everything is isolated. Most attention in the clustering literature has been paid to agglomerative methods, and I will only treat agglomerative methods here, which require solely a dissimilarity matrix. Many kinds of agglomerative methods exist in literature, but only the most widely used ones will be introduced in this section.

The general setup of the agglomerative hierarchical clustering algorithm can be seen in Algorithm 7, see Appendix A, which is integrated with three popular methods: single linkage, complete linkage and average linkage. In the next part of this section, these three methodologies will be summarised in terms of their structure and usage.

Single linkage

Single linkage, also known as the **nearest-neighbour technique**, was first described by Florek et al. (1951). The defining feature of the method is that the distance between groups is defined as that of the closest pair of individuals, where only pairs consisting of one individual from each cluster are considered (Everitt et al., 2011, chap. 4).

Single linkage focuses totally on separation (disregarding homogeneity), that is, keeping the closest points of different clusters apart from each other (Hennig, 2015a). It tends to produce unbalanced clusters, especially in large data sets, thus nearby items of the same cluster have small distances, whereas objects at opposite ends of a cluster may be much farther from each other than to elements of other clusters. On the other hand, single linkage has a potential benefit of identifying outliers (if they exist) (Everitt et al., 2011, chap. 4).

Complete linkage

Complete linkage, also known as the **furthest-neighbour technique**, is opposite to single linkage, was first introduced by Sørensen (1948), in the sense that distance between groups is now defined as that of the most distant pair of individuals, where only pairs consisting of one individual from each cluster are considered. Complete linkage focuses totally on keeping the largest distance within cluster low (disregarding separation). It tends to find compact clusters with approximately equal diameters.

Average linkage

Average linkage, is also called **Unweighted Pair Group Method with Arithmetic Mean (UPGMA)**, was first proposed by Sokal (1958), see Algorithm 7. For average linkage, the distance between two clusters is found by computing the average dissimilarity of each item in the first cluster to each in the second cluster. Average linkage and most other hierarchical methods compromise between single and complete linkage; between within-cluster homogeneity and between cluster separation (Hennig, 2015a).

Average linkage tends to join clusters with small variances, so that it is relatively more robust than other hierarchical algorithms (Everitt et al., 2011, chap. 4).

Ward's method

In addition to the hierarchical clustering algorithms outlined above, Ward Jr. (1963) proposed the use of an objective function in agglomerative hierarchical clustering, so that in each step of the algorithm, clusters are merged in order to give the best possible value of the objective function. Ward defined his method as a hierarchical version of K-means, however this method cannot be used for general dissimilarity measures. More recently Murtagh and Legendre (2014) presented an implementation of Ward's hierarchical clustering method for general dissimilarities, see the `R`-function `hclust` with `method="ward.D2"`. Therefore, the distance in Equation (5.2) must be the squared Euclidean distance to compute what is normally referred to as Ward's method, but other more general dissimilarity measures can be used instead of d_{L_2} using more recent implementations of the method.

For the sake of comparison with the standard K-means method, a disadvantage of Ward's method is that in many cases the value of S achieved for a given K based on Equation (5.1) and (5.2) is worse than what can be achieved by running the K-means algorithm several times.

In contrast, two advantages of Ward's method are: 1) the algorithm produces a hierarchy, 2) the algorithm works in a deterministic way and is not dependent on random initialisation.

5.2.4 Model based clustering

Model-based clustering is based on statistical models. The distributional assumptions for such models determine the clusters. The term "model-based clustering" is often used for **mixture models**. More often, this approach requires data set as an input, but it has been applied to models with assumptions on a latent data space for objects that come as similarity or network data, see Murphy (2015) for more information.

Scott and Symons (1971) were among the first to use the model-based approach to clustering. Since then, a considerable amount of literature has been published on model-based clustering, see Everitt and Hand (1981), Titterington et al. (1985), McLachlan and Basford (1988), McLachlan and Peel (2000) and Frühwirth-Schnatter (2006). The general form of a mixture model can be defined as follows:

Given data $\mathbf{X}$ with independent multivariate observations $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, the likelihood for a mixture model with K clusters is

$$f(\mathbf{x}; \boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k f_{\boldsymbol{\theta}_k}(\mathbf{x}), \tag{5.4}$$

where $\{f_{\boldsymbol{\theta}}, \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ defines a parametric family of distributions on $\mathbb{R}^p$, where $p \geq 1$ (e.g. $\mathbf{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$), and the π_k being known as mixing proportions, where $\pi_k \geq 0$, $k = 1, \ldots, K$, and $\sum_{j=1}^{K} \pi_j = 1$. The parameters $(\hat{\pi}_1, \hat{\boldsymbol{\theta}}_1), \ldots, (\hat{\pi}_K, \hat{\boldsymbol{\theta}}_K)$ are often estimated by Maximum Likelihood (ML), but it is often difficult to find the ML estimators of these parameters directly; in this case, the Expectation maximization (EM) algorithm is suggested, see McLachlan and Rahtnayake (2015) for more information. Having estimated the parameters of the assumed mixture distribution, objects can be associated with distinct clusters on the basis of the maximum value of the following estimated posterior probability:

$$P(c(i) = k|\mathbf{x}_i) = \frac{\hat{\pi}_k f_{\hat{\boldsymbol{\theta}}_k}(\mathbf{x}_i)}{\sum_{k=1}^{K} \hat{\pi}_k f_{\hat{\boldsymbol{\theta}}_k}(\mathbf{x}_i)}, \tag{5.5}$$

where $c(i)$ is a component membership for $\mathbf{x}_i$ ($i = 1, \ldots, n$), which is generated according to a multinomial distribution with component probabilities $\pi_1, \ldots, \pi_K$. Equation (5.5) defines a probabilistic clustering, and a crisp partition can be obtained by maximizing $P(c(i) = k|\mathbf{x}_i)$ for every

i. Equation (5.4) is the general form of a mixture model and can also be adopted for different data formats. A large and growing body of literature has investigated various data structures for interval and continuous data formats from various distributions (McLachlan and Rahtnayake, 2015), categorical and mixed-type data (Celeux and Govaert, 2015), linear models, functional data and time series models (Alfo and Viviani, 2015; Caiado et al., 2015; Hitchcock and Greenwood, 2015). Another way of estimating the posterior probability, $P(c(i) = k|\mathbf{x}_i)$ is to adopt a Bayesian prior for the estimation of the mixture parameters, $(\hat{\pi}_1, \hat{\boldsymbol{\theta}}_1), \ldots, (\hat{\pi}_K, \hat{\boldsymbol{\theta}}_K)$. The most recent study of the Bayesian approach has focused on a Dirichlet process prior, see (Rao, 2015) for more details.

The main advantage of model-based clustering is that it is based on formal models, whereas other clustering approaches which are not based on any statistical model are largely heuristic. Mixture densities often provide a sensible statistical base for performing cluster analysis and model-based clustering also has the flexibility that a large number of distributions can be adopted to model the mixture components. However, one disadvantage of the approach is that large sample sizes might be required in order to have good mixture parameter estimators (Everitt et al., 2011, chap. 6).

The model based approach is a very broad topic in cluster analysis, and much of the recent attention in cluster analysis has focused on this specific approach. Due to practical constraints, this thesis will not provide a comprehensive review of the model-based clustering approach.

5.2.5 Density-based clustering

Clusters can be defined as areas of higher density than the remainder of the data set. Although identifying clusters as high density areas seems to be very intuitive and directly connected to the term "clusters", the disadvantage of density-based clustering is that high density areas may vary in size, so that they may include very large dissimilarities and the variation between objects within clusters can be large (Hennig, 2015a).

Some density-based methods that perform kernel density estimation require the original variables; in other words, they cannot be used with dissimilarity data. More information about clustering methods based on kernel density estimators can be found in Carreira-Perpinan (2015). On the other hand, there are density-based methods that directly address the idea of discovering clusters without performing density estimation, such as one of the most well-known algorithms, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), which was proposed by Ester et al. (1996). DBSCAN is essentially a non-probabilistic clustering technique for finding clusters with a general dissimilarity measure. DBSCAN does not require the specification of the number of clusters in the data a priori, but clusters require a minimum number of points (MinPts) within a

maximum distance (ϵ), which can usually be determined by a domain expert. The method is able to find arbitrarily shaped clusters as well as clusters of different sizes. However, DBSCAN does not work well for large variances within cluster densities. In addition, determining the parameters of the DBSCAN is seldom easier than specifying the number of clusters.

Despite the fact that single linkage is defined as a hierarchical clustering method, it also has been used to obtain high-density clusters between density valleys by adopting the nearest neighbour technique (Hennig and Meila, 2015).

5.2.6 Spectral clustering

Spectral clustering, which at first deals with (dis)similarities data given in the form of an $n \times n$ similarity matrix, employs the eigenvectors of a matrix to find K clusters by adopting traditional clustering algorithms such as the K-means. In this paradigm, the structure is typically based on the idea of graph clustering theory, where the points are clustered with the similarity graph by using vertexes and edges. The definition of the spectral clustering approach can be given as follows.

Definition 5.2.4. Given a set of data points $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ or a matrix of (dis)similarity $\mathbf{D} = d(\mathbf{x}_i, \mathbf{x}_j)$, $(i = 1, \ldots, n)$, $\mathcal{X}$ or $\mathbf{D}$ is defined in the form of the *similarity graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is the vertex set, and $\mathcal{E}$ are the graph edges to be represented by the pair of vertices, (v_i, v_j).

Assume that the $\mathcal{G}$ is an *undirected* and *weighted* graph, where each edge between two vertices v_i and v_j carries a non-negative weight $w_{ij} \geq 0$. The weighted adjacency (similarity) matrix of the graph is the matrix $A = (w_{ij})$, $i, j = 1, \ldots, n$. The degree of a vertex $v_i \in \mathcal{V}$ is then defined as

$$p_i = \sum_{i=1}^{n} w_{ij}. \tag{5.6}$$

The *degree matrix* P is defined as the diagonal matrix with the degrees $p_1, \ldots, p_n$ on the diagonal. Different types of transformations are available for construction of the weighted adjacency matrix, A. The main target of this construction is to model the local neighbourhood relationship between data points. Three types of structures are listed below:

- **The ϵ neighbourhood graph:** The main idea here is that the weight assignment of A is only applied for the connection of all points whose pairwise dissimilarities are smaller than ϵ, while the weights for larger than ϵ are transformed to zero.

- k**-nearest neighbour graphs:** Each vertex, v_i is connected to its k-nearest neighbours where k is an integer number which controls the local relationships of data.

- **Fully connected graph:** All vertices having non-zero similarities are connected to each other in such a way that all edges are weighted by s_{ij}. This method for the construction of A typically uses one of the popular kernel functions, deemed appropriate by the users. The most popular choice is the Gaussian similarity kernel function, which can be adopted for both the $n \times p$ data sets or the $n \times n$ pairwise Euclidean distance between data objects, and is defined as:

$$w_{ij} = s(x_i, x_j) = \exp\left\{-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right\} = \exp\left\{-\frac{d(\mathbf{x}_i, \mathbf{x}_j)^2}{2\sigma^2}\right\}, \tag{5.7}$$

  where σ controls the width of the neighbourhoods and plays a similar role to the parameter ϵ in the case of the ϵ-neighbourhood graph (Von Luxburg, 2007). Some other widely used kernel functions are available in the `R` function `specc`, see the `R` package `kernlab`.

The calculation of parameters (e.g., k, ϵ, σ, etc.) for the graph types listed above is a separate issue and the results are difficult to obtain. However, there are some rules of thumb to approximate the parameters or a heuristic way can be used to determine suitable parameters, see Von Luxburg (2007) and Ng et al. (2002) for more information.

The main tools for carrying out spectral clustering are graph Laplacian matrices. I present three popular Laplacian matrix forms, see Table 5.1, but several other algorithms are available in the literature. Algorithm 8, see Appendix A in spectral clustering is designed to adopt the Laplacian matrix with the K-means algorithm. However, spectral clustering can also be constructed in a way such that partitions can be viewed as finding "cuts" in graphs by minimising some objective functions, see Definition 5.2.5.

Another way of constructing partitions in spectral clustering is graph clustering from point of view of the cut. The following definition explains how this procedure works.

Definition 5.2.5. Given a similarity graph with weighted adjacency matrix $\mathbf{W} = (w_{ij})_{i,j=1,\ldots,n}$, the value of the cut between partitions can be defined as $\mathcal{C} = \{C_1, \ldots, C_K\}$, $(C_k, \bar{C}_k) \subseteq \mathcal{V}$, $C_k \cap \bar{C}_k = \emptyset$, where $\bar{C}_k$ is the complement of C_k. For a given number K of subsets, the Cut approach simply consists in choosing the partition $\mathcal{C} = \{C_1, \ldots, C_K\}$ which minimizes

$$Cut(\mathcal{C}) = \frac{1}{2}\sum_{k=1}^{K} W(C_k, \bar{C}_k), \tag{5.8}$$

Table 5.1: The Laplacian Matrices

Name	**Definition**
Unnormalized Laplacian	$L = P - A$
Normalised Laplacian (Shi and Malik, 2000)	$L = I - P^{-1}A$ (Random walk matrix)
Normalised Laplacian (Ng et al., 2002)	$L = I - P^{-1/2}AP^{-1/2}$ (Symmetric matrix)

I is the identity matrix. Note that Ng et al. (2002) defined one additional step in Algorithm 8, where the rows of $\mathbf{U}$ are normalised to norm 1 by using $t_{ij} = u_{ij}/(\sum_k u_{ik}^2)^{1/2}$.

where $W(C_k, \bar{C}_k) = \sum_{i \in C_k} \sum_{j \in \bar{C}_k} w_{ij}$.

The Cut function can be solved efficiently, especially for $K = 2$, but is often problematic for a large number of clusters. This problem arises because in many cases the solution of Cut separates one individual vertex from the rest of the graph (Von Luxburg, 2007). To circumvent the issue for a large number of clusters and to balance the size of the clusters, some other objective functions are proposed. The most commonly used ones are as follows:

- The normalized cut, $NCut$ (Shi and Malik, 2000):

$$NCut(\mathcal{C}) = \frac{1}{2}\sum_{k=1}^{K} \frac{W(C_k, \bar{C}_k)}{d_{C_k}}, \tag{5.9}$$

  where $d_{C_k} = \sum_{i \in C_k} d_i$.

- The ratio cut, $RCut$ (Hagen and Kahng, 1992):

$$RCut(\mathcal{C}) = \frac{1}{2}\sum_{k=1}^{K} \frac{W(C_k, \bar{C}_k)}{|C_k|}, \tag{5.10}$$

  where $|C_k|$ is the number of vertices for the k^{th} cluster.

- The min-max cut, $Min - Max - Cut$ (Ding et al., 2001):

$$Min - Max - Cut(\mathcal{C}) = \frac{1}{2}\sum_{k=1}^{K} \frac{W(C_k, \bar{C}_k)}{W(C_k, C_k)}. \tag{5.11}$$

The main advantage of spectral clustering is that it has the flexibility to find clusters of irregular shapes. Ng et al. (2002) claimed that this approach effectively handles clusters whose overlap or

connectedness varies across clusters. It is also able to cluster "points" by using a (dis)similarity matrix, rather than directly clustering them in their native data space. Depending on the choice of kernel function, spectral clustering gives more consideration to between-cluster separation than within-cluster homogeneity in Euclidean space (Hennig, 2015a).

There is a considerable amount of literature regarding the concept of spectral clustering. Many other approaches to those discussed here are available in the spectral clustering literature, see for example Meila and Xu (2003), Azran and Ghahramani (2006) and more about spectral clustering can be found Von Luxburg (2007), Ng et al. (2002) and Meila (2015b).

5.2.7 Further clustering approaches

A large and growing body of clustering methodologies have been published over the past half-century, and several of the major clustering approaches are presented in the previous sections. In this section, some further clustering methods are described in brief.

Overlapping clustering is a concept where objects can belong to more than one cluster simultaneously. Overlapping clustering relaxes the requirement that the objects have to be assigned to one (and only one) cluster. Many different overlapping clustering techniques are available in the clustering literature. One of the earliest overlapping approaches was described by Shepard and Arabie (1979), in which additive clustering algorithm (ADCLUS) was presented. In this paper, it is assumed that the effective similarity between two objects is a weighted sum of their common features. Arabie and Carroll (1980) have developed a mathematical programming approach for fitting the ADCLUS model. These two techniques have been generalized for a clustering solution of individual differences (INDCLUS), see Carroll and Arabie (1983). The idea of overlapping clustering is also related to graph theory and social network clustering [1] in some studies. For instance, Shen et al. (2009) proposed an algorithm (EAGLE) to detect overlapping and hierarchical community structure in networks. This algorithm deals with the set of maximal cliques [2] and adopts an agglomerative framework. Several other overlapping clustering algorithms are also presented, see for example Latouche et al. (2009), Pérez-Suárez et al. (2012) and Yang et al. (2016).

Fuzzy clustering is a different type of clustering technique where membership grades are assigned to each of the data point. In other words, it is opposed to standard (classic) clustering which

[1] **Social Network Clustering** is a way of clustering objects where data points are represented in a graph with social structures through the use of networks, which can describe (a)symmetric relations between observations.

[2] A **clique**, c , in an undirected graph $\mathcal{G} = (V, E)$ is a subset of the vertices, $c \subseteq V$, such that every two distinct vertices are adjacent. A **maximal clique** is a clique that cannot be extended by including an additional adjacent vertex, that is, a clique which does not exist exclusively within the vertex set of a larger clique.

results in mutually exclusive clusters (Bezdek, 1981). The fuzzy C-means clustering, which is one of the most popular and the most widely used fuzzy clustering techniques, was developed by Dunn (1973) and improved by Bezdek (1981). The algorithm is very similar to the K-means algorithm, the only difference is that the the objects are represented as coefficients, w_{ik} ($0 \leq w_{ik} \leq 1$), for each cluster rather than assigning the objects to only one cluster. In particular, the fuzzy clustering approach was extended to different algorithms by changing its variants (e.g, distance variants, prototype variants, etc.). More information can be found in D'Urso (2015).

Consensus clustering (also known as ensemble clustering) can be defined as combining multiple partitioning of a set of data points into a single consolidated clustering without accessing data features or data points. Ghosh and Acharya (2015) presented consensus clustering under three categories: 1) probabilistic approach, which is based on statistical models, see for example Topchy et al. (2004), Wang et al. (2011a), Wang et al. (2011b); 2) pairwise similarity approach, which is simply the weighted average of co-association matrices, see Strehl and Ghosh (2002), Nguyen and Caruana (2007) for different algorithms of this type, and 3) direct and other heuristic methods.

Cluster analysis has also been developed for different types of data structures. Some of them have been mentioned in Section 5.2.4, and some others are described in different sources, such as categorical data (Andreopoulos, 2013), symbolic data (Brito, 2015), repeated measures data (Vichi, 2015) to name a few.

A huge amount of clustering methodologies are explained in many different sources, see for example Aggarwal and Reddy (2013), Everitt et al. (2011), Gan et al. (2007), Kaufman and Rousseeuw (1990). Several of them are very briefly described in this section, and some of the main clustering methods are reviewed in the previous sections regarding methodological and algorithmic themes.

5.3 Visual Exploration of Clusters

Graphical displays of multivariate data can provide insight into the structure of the data, which in turn can be useful for finding clustering structure (Everitt et al., 2011, chap. 2). Many of the potentially desired features of clustering such as separation between clusters, high density within clusters, and distributional shapes can be explored graphically in a more holistic (if subjective) way than by looking at index values (Hennig, 2015a). Standard visualisation techniques such as scatterplots, heatplots, etc. as well as interactive and dynamic graphics can be used to find and to validate clusters, see for example Theus and Urbanek (2008) and Cook and Swayne (2007). Furthermore, dendrograms, which are probably one of the most commonly used visualisation methods

for clustering, are usually used for ordering observations in heatplots. For use in cluster validation it is desirable to plot observations in the same cluster together, which is achieved by the use of dendrograms for ordering the observations (Hennig, 2015a). A brief overview of visualising clustering can be found in different sources, see Everitt et al. (2011, chap. 2), Déjean and Mothe (2015), and Xu and Wunsch (2008, chap. 9).

On the other hand, researcher's intuition for clusters is often moivated by visualisation using lower dimensional projections of multivariate data for graphical representation. In Section 3.2.5, a considerable amount of literature has been reviewed on low dimensional representation in multivariate data analysis for use in cluster analysis. In this section, two popular low dimensional representation methods are scrutinized to demonstrate their utility in visualising cluster points; these methods are to be used in a later chapter of this thesis.

5.3.1 Principal Component Analysis

The purpose of principal component analysis (PCA) in this study is to visualise the cluster points in a low dimensional space and a brief explanation of how the principal components are calculated is described here. PCA is a statistical procedure for transforming the variables in a multivariate data set into a few orthogonal linear combinations of the original variables that are linearly uncorrelated with each other. The new linearly uncorrelated variables (i.e. the principal components) account for decreasing proportions of the total variance of the original variables. Stated otherwise: the first principal component is the linear combination with the largest variance, and the second PC has the second largest variance, and so on.

Formally, given a data set with n observations and p variables

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_p \end{pmatrix}, \tag{5.12}$$

the sample covariance matrix of $\mathbf{X}$ is defined as

$$\text{var}(\mathbf{X}) = \Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1p} \\ \sigma_{21} & \sigma_2^2 & \dots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \dots & \sigma_p^2 \end{pmatrix}. \tag{5.13}$$

New variables (the principal components) $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_p$ are then defined as follows:

$$\begin{aligned} \mathbf{y}_1 &= u_{11}\mathbf{x}_1 + u_{12}\mathbf{x}_2 + \cdots + u_{1p}\mathbf{x}_p \\ \mathbf{y}_2 &= u_{21}\mathbf{x}_1 + u_{22}\mathbf{x}_2 + \cdots + u_{2p}\mathbf{x}_p \\ &\vdots \\ \mathbf{y}_p &= u_{p1}\mathbf{x}_1 + u_{p2}\mathbf{x}_2 + \cdots + u_{pp}\mathbf{x}_p \end{aligned}, \tag{5.14}$$

where each of these equations is a linear combination of $\mathbf{x}_i$'s which gives $\mathbf{y}_j$'s, $i, j = 1, \ldots, p$, and $u_{j1}, u_{j2}, \ldots, u_{jp}$ are the coefficients of the equations. The coefficients can be found from the spectral decomposition based on eigenvalues and eigenvectors of the covariance matrix, Σ as

$$\Sigma = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T, \tag{5.15}$$

where $\mathbf{\Lambda} = \text{diag}\,(\lambda_1, \ldots, \lambda_p)$ is the diagonal matrix of the ordered eigenvalues $\lambda_1 \geq \ldots \geq \lambda_p$, and $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_p)$ is a $p \times p$ orthogonal matrix containing the eigenvectors, where $\mathbf{u}_j$'s are the the columns of $\mathbf{U}$. Hence, the elements for these eigenvectors will be the coefficients of the principal components.

The variance of the j^{th} principal component is equal to the j^{th} eigenvalue so that

$$\text{var}(\mathbf{y}_j) = \text{var}(u_{j1}\mathbf{x}_1 + u_{j2}\mathbf{x}_2 + \ldots u_{jp}\mathbf{x}_p) = \lambda_j, \tag{5.16}$$

and the principal components are uncorrelated with each other, so that the covariance between $\mathbf{y}$'s is

$$\text{cov}\left(\mathbf{y}_j, \mathbf{y}_{j'}\right) = 0 \quad \forall\, j, j'. \tag{5.17}$$

Another important property of the eigenvalue decomposition is that the total variation can be defined as the sum of eigenvalues of the covariance matrix Σ,

$$\sum_{j=1}^{p} \lambda_j = \sum_{j=1}^{p} \text{trace}(\Sigma) = \sum_{j=1}^{p} \sigma_j^2 \tag{5.18}$$

and that the fraction

$$\sum_{j=1}^{t} \lambda_j / \text{trace}(\Sigma) \tag{5.19}$$

gives the cumulative proportion of the variance explained by the first t principal components. Here the parameter t can often be determined by user's intuition when the percentage of the overall variation obtained from the first t principal components is high. For example, if the first 3 principal components explain the overall variation as $\%90$, then the user can intuitively make the decision that $t = 3$ might be the suitable choice, since the percentage for the variance of the first t principal components is relatively large. One can also determine the parameter t by a threshold value, say λ_0 so that the decision can then be made by only keeping the eigenvectors such that their corresponding eigenvalues are greater than λ_0, see Jolliffe (1972) and Jackson (2005).

Although there are different ways to determine the parameter t, in this research it is convenient to define t as 2 or 3. That is because here the main aim of using the principal components is to visualise the cluster points and the visualisation in a single scatter-plot is not appropriate for more than 3 dimensions.

As with most multivariate data analysis, it is recommended when carrying out Principal Component Analysis to standardise the data set first. This is important because the principal components obtained can be influenced by the different scales of the variables.

5.3.2 Multidimensional Scaling

Multidimensional scaling (MDS) is a multivariate data analysis technique to detect meaningful underlying dimensions using $n \times n$ similarities or dissimilarities between the investigated objects rather than using a $n \times p$ multivariate data matrix. In this research and many others, the main purpose of MDS is to provide a visual representation of the pattern of proximities (i.e., similarities or dissimilarities) among a set of objects for identifying "cluster" points.

MDS has various procedures designed to arrive at an optimal low-dimensional configuration for a particular type of proximity data. In the following sections, I describe a number of MDS methods.

Classical Scaling

Classical Multidimensional scaling was first proposed by Torgerson (1952), Torgerson (1958) and Gower (1966). It is a dimension reduction technique using spectral decomposition to detect a t dimensional representation (referred to as *principal coordinates*) that explains the dissimilarity matrix in an optimal manner. Algorithm 9 in Appendix A describes how the classical MDS algorithm works.

Although the classical MDS algorithm assumes the dissimilarity matrix $\mathbf{D}$ to be Euclidean, which makes all eigenvalues of $\mathbf{B}$ (See Algorithm 9 in Appendix A) non-negative, one could use any kind of dissimilarity matrix as an input. However, negative eigenvalues may occur in case of using different types of dissimilarities. If the eigenvalues of $\mathbf{B}$ are not all non-negative, then some linear transformation on $\mathbf{B}$ can be considered in order to make $\mathbf{B}$ non-negative definite (Cox and Cox, 2000, chap.2). This can be achieved by either ignoring the negative eigenvalues (and associated eigenvectors) or adding a suitable constant to the dissimilarities (i.e., $d_{ij} \leftarrow d_{ij} + c$ if $i \neq j$, and unchanged otherwise). This is the additive constant issue, for more information see Cailliez (1983).

The inter-point distances between the principal coordinates ($d^*_{ij} = \|\mathbf{z}_i - \mathbf{z}_j\|$) are identical to the inter-point distances of $\mathbf{D}$ when $\mathbf{D}$ is Euclidean distance ($d^*_{ij} = d_{ij}$); in other words, the classical MDS is equivalent to PCA in which the principal coordinates are identical to the scores of the first t principal components of the $\mathbf{X}$. However, there are cases where $\mathbf{D}$ is not Euclidean, in which case the inter-point distances between the principal coordinates do not have to be equal to the inter-point distances of $\mathbf{D}$. Nevertheless, classical MDS solution still finds optimal coordinates for the inter-point distances of $\mathbf{Z}$ such that $d^*_{ij} \approx d_{ij}$,Borg and Groenen (2005, chap.19).

Assessing dimensionality is another major aspect of classical MDS that the user needs to determine. One of the usual strategies is to plot the ordered eigenvalues against the dimension and then to choose a dimension at which the eigenvalues become stable. On the other hand, if $\mathbf{B}$ is non-negative definite, then the number of positive eigenvalues can be the number of dimensions to be used as principal coordinates. However, for practical reasons, in this research t should be relatively small (say 2 or 3) for the sake of graphical interpretation.

Distance scaling

In distance scaling, the aim is to find an optimal configuration in a lower dimensional space such that the the inter-point distances of $\mathbf{Z}$ are approximated to the inter-point distances of $\mathbf{D}$ as closely as possible. In distance scaling, the relationship between $f(d_{ij})$ and d_{ij} is flexible so that a suitable configuration can be shown as

$$d_{ij} \approx f(d_{ij}), \tag{5.20}$$

where f is some monotone function. Two types of distance scaling methods (metric and non-metric MDS) exist in the literature. The use of "metric" and "non-metric" distance scaling depends upon the nature of the dissimilarities. In general, if the dissimilarities are quantitative (e.g., ratio or

interval scale), we use metric distance scaling, whereas if the dissimilarities are qualitative (e.g., ordinal), we use non-metric distance scaling (Izenman, 2008, chap.13).

- **Metric MDS (Torgerson, 1952):** The possible f is usually taken to be a parametric linear function, such as

$$f(d_{ij}) = \alpha + \beta d_{ij}, \tag{5.21}$$

where α and β are unknown positive coefficients. Borg and Groenen (2005, chap.9) stated that metric distance scaling is categorised under three different models:

1. Absolute MDS ($\alpha = 0, \beta = 1$)
2. Ratio MDS ($\alpha = 0, \beta > 0$)
3. Interval MDS ($\alpha \geq 0, \beta \geq 0$)

Absolute MDS is typically the same as the classical scaling. Interval MDS, which is the standard model of metric MDS, preserves the data linearly in the distances, while Ratio MDS searches for a solution that preserves the proximities up to a scaling factor β. The choice of model is often connected with the type of data set (Borg et al., 2012, chap.5).

The important question is how to find the principal coordinates in an optimal manner. A useful loss function ($\mathcal{L}_f$) in this context is called "stress", which is minimised over all t dimensional configurations $\{z_{ij}\}$ to find a monotone f yielding an optimal metric distance scaling solution.

$$\text{stress} = \mathcal{L}_f(d_{ij}) = \left(\frac{\sum_{i<j} \left(d_{ij} - f(d_{ij})\right)^2}{\sum_{i<j} d_{ij}^2} \right)^{1/2} \tag{5.22}$$

- **Non-metric MDS (Kruskal, 1964b):** In many applications of MDS, dissimilarities are considered with respect to the ranking of distances and in Non-metric MDS (also known as ordinal MDS) a low-dimensional representation with respect to the ranking of distances is found. The functional form in metric scaling is often defined as linear regression, whereas non-metric distance scaling uses isotonic (monotonic) regression, which yields transformed approximated distances also referred to as "disparities" in the MDS literature. Isotonic regression fits ideal distances in such a way that relative dissimilarities between points match the order of dissimilarities between points in an optimal way.

The isotonic regression and Non-metric distance scaling algorithm are summarised in Section A.6, Appendix A. The concept of isotonic regression is explained with an example provided by Izenman (2008, chap.13).

As the main goal of MDS here is visualisation, I consider a reasonably small number of dimensions t, typically 2 or 3, however, the solution must to MDS must be validated. Stress S measures the goodness of fit between configuration points and disparities. Various stress values can be considered in different manners to assess the global fit of any non-metric distance scaling solution. Kruskal (1964a) studied different types of real and simulated data and suggests a guideline to determine how well an MDS solution fits the suitable data structure, see Table 5.2. However, Izenman (2008, chap.13) warns that Kruskal's suggestion may not be appropriate in some situations, especially for noisy data or for data sets with large numbers of variables. Borg and Groenen (2005, chap.11) stated that while Kruskal's benchmarks provide a formal sense about the goodness of fit for MDS solution, the criteria may be misleading. Following a literature review and some simulation studies, they gave some remarks on this subject: 1) A higher number of objects usually gives higher stress, 2) interval MDS generally leads to higher stress than ordinal MDS.

Table 5.2: Evaluation of "stress"

Stress	Goodness of Fit
0.200	Poor
0.100	Fair
0.050	Good
0.025	Excellent
0.000	"Perfect"

More advanced approaches are available in the MDS literature. Constrained MDS (sometimes referred to Confirmatory MDS), which was first proposed by Borg and Lingoes (1980), is an MDS approach in which user-defined external information (or restriction) is incorporated directly in the distances. As the main target of this approach is to minimise the stress value, enforcing such additional properties onto the MDS model may provide a more accurate low dimensional representation than the standard MDS solution. However, Borg and Groenen (2005, chap.10) claimed that if the stress of a confirmatory MDS solution is not much higher than the stress of a standard (unconstrained) MDS solution, the former model can be accepted as an adequate model. Other references to constrained MDS can be found in De Leeuw and Heiser (1980), Weeks and Bentler (1982), Winsberg and Soete (1997). On the other hand, Cox and Cox (1991) proposed another constrained MDS approach, Spherical MDS, in which the points of a configuration from non-metric MDS can

be forced onto an exact circle, ellipse, hyperbola or parabola for a two dimensional surface, or a sphere or an ellipsoid for a three dimensional surface. Because the surface is spherical, the choice of measurement needs to be connected to such space, so that the geodesic distance, which is defined to be the length of the shortest geodesic along the surface can be used. See more information on Spherical MDS in Cox and Cox (2000, chap.4).

Another prominent variant of MDS is the dimensional weighting model, often referred as INDSCAL (INdividual Differences SCALing) model (Carroll and Chang, 1970). The goal of this model is also to minimise stress. Whereas the aforementioned MDS procedures analyse a single dissimilarity matrix (d_{ij}), the INDSCAL method incorporates one additional parameter which can be defined as different replications of dissimilarities, which can be obtained from several "individuals" who have different judgements. At the end, all these different dissimilarities are combined in such a way that the represented MDS dimensions are weighted in an optimal way, see the mathematical definition of this method in Carroll and Chang (1970).

More details about these methods and many other different MDS approaches can be found in Cox and Cox (2000), Borg and Groenen (2005) and Borg et al. (2012).

5.4 Cluster Validation

Cluster validation is about assessing the quality of a clustering on a data set of interest. In most applications, no "true" clustering is known with which the clustering to be assessed could be compared (Hennig, 2015b). Cluster validation is an essential step in the cluster analysis process, because the quality of the resulting clusters is often not obtained directly from clustering methods. There are several different approaches to cluster validation.

The context of "cluster validation" is known as evaluating the results of a clustering algorithm, which can be specified in either an 'internal' or 'external' scheme. Here *external* evaluation criteria, which are different ways of validating the data set with the external information of interest, will be analysed for the sake of comparing the clustering. In the next chapter, Chapter 6, I focus on internal validation criteria that decide about the number of clusters by measuring the quality of a clustering without reference to external information. For this thesis, the term "clustering validation indexes" or "clustering quality indexes" is used for internal validation criteria, while external validation indexes for clustering (e.g., adjusted Rand index) are addressed as the term "external clustering validation indexes".

The decision of how to use external information can be made in a formal or an informal way. Informally, subject matter experts can decide to what extent a clustering makes sense in terms of subject-matter reason, or new discoveries can be helpful as a cluster validation technique. Formally, some external information that is expected to be related to the clustering might be known prior to cluster analysis (Hennig, 2015a).

A formal mathematical definition of external validation measurement can be given as follows. Let $\mathcal{C} = \{C_1, \ldots, C_K\}$ be the clustering computed by using the given data set $\mathcal{X}$, and let $\mathcal{C}^{'} = \left\{C_1^{'}, \ldots, C_{K^{'}}^{'}\right\}$ be the partitions obtained from the external information. $P(\mathcal{C}, \mathcal{C}^{'})$ is defined as the proximity function for comparing these partitions.

Table 3.2 is a contingency table which demonstrates how the pairs of points are counted. As I described in Section 3.3.1, the distance between two partitions (or two parts of categories) can be defined in different functional forms. Here, the idea is very similar and the parameters as defined in the contingency table can be described as

N_{11}: The number of point pairs in the same cluster under both $\mathcal{C}$ and $\mathcal{C}^{'}$,

N_{00}: The number of point pairs in different cluster under both $\mathcal{C}$ and $\mathcal{C}^{'}$,

N_{10}: The number of point pairs in the same cluster under $\mathcal{C}$ but not under $\mathcal{C}^{'}$,

N_{01}: The number of point pairs in the same cluster under $\mathcal{C}^{'}$ but not under $\mathcal{C}$.

The number of points in the intersection of cluster C_k of $\mathcal{C}$ and $C_{k^{'}}^{'}$ of $\mathcal{C}^{'}$ can be defined as $n_{kk^{'}} = |C_k \cap C_k^{'}|$, and then the counts $n_{kk^{'}}$ satisfy

$$\sum_{k=1}^{K} n_{kk^{'}} = |C_{k^{'}}^{'}| = n_{k^{'}} \qquad \sum_{k^{'}=1}^{K^{'}} n_{kk^{'}} = |C_k| = n_k \tag{5.23}$$

so that $\sum_{k=1}^{K} n_k = \sum_{k^{'}=1}^{K^{'}} n_{k^{'}} = n$, and the four counts always satisfy $n(n-1)/2 = N_{11} + N_{00} + N_{10} + N_{01}$. These notations will be used for several classes of external validation criteria as shown in Table 5.3.

The Rand index ($\mathcal{R}$), which is one of the first external validation criteria defined in the clustering literature, is simply matching similarity between $\mathcal{C}$ and $\mathcal{C}^{'}$ over all pairs of points. The interpretation of values of $\mathcal{R}$ depends on the numbers and sizes of clusters in the two clusterings. Two random clusterings with large K and $K^{'}$ can be expected to have a rather high value of $\mathcal{R}$. A problem with the Rand index is that the expected value of the Rand index of two random clusterings does not take a constant value (say zero). Therefore, the adjusted version of the Rand Index

Table 5.3: Some of the external validation criteria

Code	Index	Formula
$(\mathcal{R})$	Rand Index (Rand, 1971)	$\mathcal{R}(\mathcal{C},\mathcal{C}^{'}) = \frac{N_{11}+N_{00}}{n(n-1)/2}$
$(\mathcal{AR})$	Adjusted Rand Index (Hubert and Arabie, 1985)	$\mathcal{AR}(\mathcal{C},\mathcal{C}^{'}) = \frac{\mathcal{R}(\mathcal{C},\mathcal{C}^{'})-E(\mathcal{R})}{1-E(\mathcal{R})}$
$(\mathcal{J})$	Jaccard Index (Jaccard, 1912)	$\mathcal{J}(\mathcal{C},\mathcal{C}^{'}) = \frac{N_{11}}{N_{11}+N_{01}+N_{10}}$
$(\mathcal{W})$	Wallace (1983) Index	$\mathcal{W}(\mathcal{C},\mathcal{C}^{'}) = \frac{N_{11}}{\sum_k n_k(n_k-1)/2}$
$(\mathcal{F})$	Fowlkes and Mallows (1983) Index	$\mathcal{F}(\mathcal{C},\mathcal{C}^{'}) = \sqrt{\mathcal{W}(\mathcal{C},\mathcal{C}^{'})\mathcal{W}(\mathcal{C}^{'},\mathcal{C})}$

was proposed. The $E(\mathcal{R})$ function from the equation of $\mathcal{AR}$ in Table 5.3 is the expected value of $\mathcal{R}$ assuming that objects are randomly assigned to clusters in such a way that numbers and sizes of clusters in $\mathcal{C}$ and $\mathcal{C}^{'}$ are held constant. A different and more detailed formulation of $\mathcal{AR}(\mathcal{C},\mathcal{C}^{'})$ is shown in Equation (5.24).

$$\mathcal{AR}(\mathcal{C},\mathcal{C}^{'}) = \frac{\sum_{k=1}^{K}\sum_{k'=1}^{K^{'}}\binom{n_{kk'}}{2} - \left[\sum_{k=1}^{K}\binom{n_k}{2}\right]\left[\sum_{k'=1}^{K^{'}}\binom{n_{k'}}{2}\right]/\binom{n}{2}}{\left[\sum_{k=1}^{K}\binom{n_k}{2} + \sum_{k'=1}^{K^{'}}\binom{n_{k'}}{2}\right]/2 - \left[\sum_{k=1}^{K}\binom{n_k}{2}\right]\left[\sum_{k'=1}^{K^{'}}\binom{n_{k'}}{2}\right]/\binom{n}{2}}. \tag{5.24}$$

For the later chapters, I will use the $\mathcal{AR}$ Index, which is the most widely used external validation criteria among statisticians. Milligan and Cooper (1986) analysed the external criteria in Table 5.3 by applying the hierarchical clustering over different numbers of clusters, and the result of their study indicated that the $\mathcal{AR}$ Index was found to be the best criteria based on counting pairs across different hierarchy levels.

Meila (2015a) addressed a few required properties of external validation of clustering as listed below:

- Any external validation criteria can be applied to any two partitions of the same data set; in other words, the number of clusters for different partitions (K and $K^{'}$) do not have to be equal.
- There is no assumption about how clusterings are generated.
- For the sake of interpretability, indices should be taking the values of a fixed range, and a reasonable range is $[0, 1]$. Although the $\mathcal{AR}$ Index covers the $[0, 1]$ range much more uniformly than the $\mathcal{R}$ Index, it can take a negative value (but not so often).

More discussions about cluster validation and some other external validation criteria are reviewed in Meila (2015a), Xiong and Li (2013) and Xu and Wunsch (2008, chap.10). The function `external_validation` in the `R` package "`ClusterR`" (Mouselimis, 2018) provides several external validation criteria including the ones in Table 5.3.

5.5 Summary

In this chapter, various clustering algorithms with their pros and cons have been discussed. Some of these only work with a data matrix as an input (e.g., K-means), whereas some others are characterised as distance based clustering algorithms (e.g., Hierarchical clustering). For the visualisation of cluster objects, the most widely used dimensional reduction techniques (PCA and MDS) have been introduced. Finally, some external validation criteria have been reviewed with regards to cluster validation, with the main focus on the adjusted Rand index, which is the suitable choice for the analysis of data sets in the later chapters of this thesis.

CHAPTER 6

ESTIMATING THE NUMBER OF CLUSTERS AND AGGREGATING INTERNAL CLUSTER VALIDATION INDEXES

In cluster analysis, two important questions are how to choose an appropriate clustering method and how to determine the best number of clusters. To decide on the appropriate cluster analysis methodology and the number of clusters, researchers should consider what are the desired data analytic characteristics of the clusters. For this aim, different clustering quality index values can be evaluated, the choice of which is crucially dependent on the aim of clustering.

A large number of clustering quality methods for finding different numbers of clusters or for choosing other tuning parameters (e.g., DBSCAN) are available for researchers who apply cluster analysis. Some of these are based on a single aspect of cluster quality (e.g., average within-cluster dissimilarities), and there are many clustering validation indexes (e.g., Caliński and Harabasz (1974) index) that are usually advertised as measures of global cluster validation in a univariate way. Some clustering validation indexes are designed to combine different complementary aspects of cluster quality. Several of these different types of measurements for choosing the number of clusters will be explained more in the next sections.

The subject of clustering quality indexes is scrutinized in many different sources. For example, Hennig (2017) introduced a number of validation criteria that refer to different desirable characteristics of a clustering. Halkidi et al. (2015) gave an overview of various clustering validation indexes. In the sections that follow, I will review several clustering quality aspects and their usages with reference to the sources above.

The results of clustering quality indexes can be in different ranges and different directions, it is therefore useful to define them in such a way that they point in the same direction and are in the

same range. In this respect, if necessary, these indexes will be standardised in a way that the index value must be in the range of $[0, 1]$, and that larger index values are always considered better.

Note that the notations in Definition 5.1.1 will be used for the formulations in clustering quality indexes below.

6.1 Clustering Validation Indexes

In this section, measurements for various aspects of cluster validity will be introduced.

6.1.1 Small within-cluster dissimilarities

Homogeneity within clusters is a major aspect of most cluster analysis applications. The measurement of homogeneity can often be computed by using dissimilarity between objects within clusters. Many different formulations can be explored, but here some of the important formulations are reviewed.

- **Average within-cluster dissimilarities** is the most straightforward way to formalise all objects within a cluster. The formulation of this aspect is defined as follows:

$$I_{ave.wit}(\mathcal{C}) = \frac{1}{\sum_{k=1}^{K} n_k(n_k - 1)} \sum_{k=1}^{K} \sum_{x_i \neq x_j \in C_k} d(x_i, x_j), \tag{6.1}$$

  where $d(x_i, x_j)$ is the dissimilarity between i^{th} and j^{th} objects from data $\mathcal{X}$. As implied by the title of this section, smaller within-cluster dissimilarities indicate better clustering quality. However, the standard definition is that the larger index values are always better and they must be in the range of $[0, 1]$. In this sense, the standardised version of $I_{ave.wit}$ is defined as

$$I^*_{ave.wit}(\mathcal{C}) = 1 - \frac{I_{ave.wit}(\mathcal{C})}{d_{max}} \in [0, 1], \tag{6.2}$$

  where $d_{max} = \max_{x_i, x_j \in \mathcal{X}} d(x_i, x_j)$ is the maximum value in the dissimilarity matrix.

- **Within-cluster sum of squares** is an alternative way of measuring within-cluster dissimilarities, which is originally constructed from the K-means objective function (See Equation (5.2)) in its formulation:

$$I_{within.ss}(\mathcal{C}) = \sum_{k=1}^{K} \frac{1}{2n_k} \sum_{x_i \neq x_j \in C_k} d(x_i, x_j)^2. \tag{6.3}$$

The standardised version of this is formed as

$$I^*_{within.ss}(\mathcal{C}) = 1 - \frac{I_{within.ss}(\mathcal{C})}{I_{overall.ss}(\mathcal{C})} \in [0, 1], \tag{6.4}$$

where $I_{overall.ss}(\mathcal{C}) = \frac{\sum_{i<j} d(x_i,x_j)^2}{n}$ is the overall sum of squares of dissimilarities. This measure gives more emphasis to the largest within-cluster dissimilarities than average within-cluster dissimilarities.

- **Widest gap:** Hennig (2017) introduced this clustering quality index as the dissimilarity of the widest within-cluster gap. Homogeneity within clusters implies that there are no "gaps" within a cluster, and that the cluster is well connected. A gap can be characterised as a split of a cluster into two sub-clusters so that the minimum dissimilarity between the two sub-clusters is large. The corresponding index measure is then provided as

$$I_{widest.gap}(\mathcal{C}) = \max_{C \in \mathcal{C},\, D,E:\, C=D \cup E} \quad \min_{x_i \in D,\, x_j \in E} d(x_i, x_j). \tag{6.5}$$

$I_{widest.gap} \in [0, d_{max}]$ and small values are good, so it is standardised as:

$$I^*_{widest.gap}(\mathcal{C}) = 1 - \frac{I_{widest.gap}(\mathcal{C})}{d_{max}} \in [0, 1]. \tag{6.6}$$

This section provides some of the major aspects for measuring within-cluster dissimilarities. One could also consider different functional forms of within-cluster dissimilarities (e.g., average of maximum within-cluster dissimilarities). On the other hand, if users are more interested in making the index less sensitive to large within-cluster dissimilarity, then quantiles or trimmed means can be used.

6.1.2 Between cluster separation

Between cluster separation, which measures how distinct or well-separated a cluster is from other clusters, is another important aspect in the clustering validation literature. Various formulations can be generated for examining the clustering quality of between-cluster separation. I review some of the between-cluster separation as listed below:

- **Average between-cluster dissimilarities:** The functional structure for between-cluster measurement can be first thought of as averaging all between-cluster dissimilarities and is formulated as follows:

$$I_{ave.bet}(\mathcal{C}) = \frac{1}{n(n-1) - \sum_{k=1}^{K} n_k(n_k - 1)} \sum_{k=1}^{K} \sum_{x_i \in C_k, x_j \notin C_k} d(x_i, x_j). \tag{6.7}$$

The standardised version of this is defined as:

$$I^*_{ave.bet}(\mathcal{C}) = \frac{I_{ave.bet}(\mathcal{C})}{d_{max}} \in [0, 1]. \tag{6.8}$$

- **Separation index:** Hennig (2017) proposed an alternative way of measuring between-cluster separation. He argued that averaging all between-cluster dissimilarities cannot be satisfactory, because between-cluster separation accounts for the smallest between-cluster dissimilarities, and the dissimilarities between pairs of farthest objects from different clusters should not contribute to this. The simplest way is to consider the minimum between-cluster dissimilarity, but this might be inappropriate, because in the case of there being more than two clusters the computation only depends on the two closest clusters. Thus, he proposed another index that takes into account a portion, p, of objects in each cluster that are closest to another cluster. Here is the definition:

For every object $x_i \in C_k$, $i = 1, \ldots, n$, $k \in 1, \ldots, K$ let $d_{k:i} = \min_{x_j \notin C_k} d(x_i, x_j)$. Let $d_{k:(1)} \leq \ldots \leq d_{k:(n_k)}$ be the values of $d_{k:i}$ for $x_i \in C_k$ ordered from the smallest to the largest, and let $[pn_k]$ be the largest integer $\leq pn_k$. Then, the **separation index** with the parameter p is defined as

$$I_{sep.index}(\mathcal{C}; p) = \frac{1}{\sum_{k=1}^{K} [pn_k]} \sum_{k=1}^{K} \sum_{i=1}^{[pn_k]} d_{k:(i)}, \tag{6.9}$$

and a suitable standardised form can be defined as follows:

$$I^*_{sep.index}(\mathcal{C}) = \frac{I_{sep.index}(\mathcal{C}; p)}{d_{max}} \in [0, 1]. \tag{6.10}$$

Here the proportion p can be chosen according to the user's interest. Hennig (2017) suggests choosing p to be 0.1.

One could also consider using the between-cluster sum of squares of dissimilarities, but since $I_{overall.ss}(\mathcal{C}) = I_{between.ss}(\mathcal{C}) + I_{within.ss}(\mathcal{C})$, and $I_{overall.ss}(\mathcal{C})$ is constant, it does not give any additional information apart from within-cluster sum of squares.

6.1.3 Representation of dissimilarity structure by clustering

Summarising dissimilarity structure and clustering information in a univariate form can be another alternative way of validating cluster quality. In this respect, Hubert and Schultz (1976) introduced a framework for data analysis indexes, which is simply the correlation of the vector of dissimilarities $\mathbf{d} = vec([d(x_i, x_j)]_{i<j})$ with the vector of "clustering induced dissimilarity" $\mathbf{c} = vec([c_{ij}]_{i<j})$, where $c_{ij} = \mathbf{1}(l_i \neq l_j)$, l_i, l_j are the cluster labels and $\mathbf{1}(\cdot)$ denotes the indicator function. With ρ denoting the sample Pearson correlation,

$$I_{Pearson.\Gamma}(\mathcal{C}) = \rho(\mathbf{d}, \mathbf{c}). \tag{6.11}$$

$I_{Pearson.\Gamma}(\mathcal{C}) \in [-1, 1]$ from the definition of correlation, so the standardised form of this index is therefore defined as

$$I^*_{Pearson.\Gamma}(\mathcal{C}) = \frac{I_{Pearson.\Gamma}(\mathcal{C}) + 1}{2} \in [0, 1]. \tag{6.12}$$

Halkidi et al. (2015) gave an overview of some alternative versions of the $I_{Pearson.\Gamma}(\mathcal{C})$ index. For example, Baker and Hubert (1975) proposed a similar idea, where Goodman and Kruskal (1954)'s rank correlation was used instead of the sample Pearson correlation. Although this structure is more robust against extreme observations, it is computationally very expensive compared with the sample Pearson correlation, and some useful information might be lost due to the rank transformation when the correlation is computed.

6.1.4 Uniformity for cluster sizes

In some clustering applications, analysts are interested in whether the clusters are roughly of the same size. This mainly depends on organisation within clustering structure. The **Entropy** (Shannon, 1948) is one of the well-known methodologies for measuring the uniformity of cluster sizes:

$$I_{entopy}(\mathcal{C}) = -\sum_{k=1}^{K} \frac{n_k}{n} \log(\frac{n_k}{n}). \tag{6.13}$$

Large values are good. The entropy is maximised for fixed K by $e_{max}(K) = -\log(\frac{1}{K})$, so the standardised form is defined as

$$I^*_{entopy}(\mathcal{C}) = \frac{I_{entopy}(\mathcal{C})}{e_{max}(K)}. \tag{6.14}$$

6.1.5 Some popular clustering quality indexes

A considerable number of clustering quality indexes have been published in recent decades. Some of them balance in some way within-cluster homogeneity against between-cluster separation, others are designed in different structures by using one of the single aspects (e.g., within or between-cluster dissimilarities), and a few others have different forms for different aims. In this section. I will present four popular commonly used indexes.

- **The Average silhouette width (ASW)** (Kaufman and Rousseeuw, 1990) is based on a compromise between within cluster homogeneity and between cluster separation. In particular, the dissimilarities of observations from other observations of the same cluster are compared with dissimilarities from observations of the nearest other cluster, which emphasizes separation between a cluster and their neighbouring clusters (Hennig and Liao, 2013). Mathematically, for a partition of $\mathbf{w}$ into clusters $C_1, \ldots, C_K$ let

$$s_i(k) = \frac{b_i(k) - a_i(k)}{\max\{a_i(k), b_i(k)\}} \in [-1, 1] \tag{6.15}$$

be the so-called 'silhouette width', where $i = 1, \ldots, n$, $1 \leq k \leq K$, and

$$a_i(k) = \frac{1}{n_k - 1} \sum_{\mathbf{x}_j \in C_k} d(\mathbf{x}_i, \mathbf{x}_j), \tag{6.16}$$

$$b_i(k) = \min_{h \neq k} \frac{1}{n_h} \sum_{\mathbf{x}_j \in C_h} d(\mathbf{x}_i, \mathbf{x}_j). \tag{6.17}$$

Finally, the ASW index is defined as

$$I_{ASW}(\mathcal{C}) = \frac{1}{n} \sum_{i=1}^{n} s_i(k) \tag{6.18}$$

Large values are better. Since $I_{ASW}(\mathcal{C}) \in [-1, 1]$, it can be standardised as

$$I^*_{ASW}(\mathcal{C}) = \frac{I_{ASW}(\mathcal{C}) + 1}{2} \in [0, 1]. \tag{6.19}$$

- **The Caliński and Harabasz (1974) index**: The proportion of squared within-cluster dissimilarities is compared with all between-cluster dissimilarities, which emphasizes within-cluster homogeneity more, and is through the use of squared dissimilarities more prohibitive

against large within-cluster dissimilarities (Hennig and Liao, 2013). In mathematical form, $I_{CH}(\mathcal{C})$ is defined as

$$I_{CH}(\mathcal{C}) = \frac{\mathbf{B}(K)(n-K)}{\mathbf{W}(K)(K-1)} \tag{6.20}$$

where

$$\mathbf{W}(K) = \sum_{k=1}^{K} \frac{1}{n_k} \sum_{\mathbf{x}_i, \mathbf{x}_j \in C_k} d(\mathbf{x}_i, \mathbf{x}_j)^2 \tag{6.21}$$

and

$$\mathbf{B}(K) = \frac{1}{n} \sum_{i,j=1}^{n} d(\mathbf{x}_i, \mathbf{x}_j)^2 - \mathbf{W}(K). \tag{6.22}$$

$\mathbf{W}(K)$ and $\mathbf{B}(K)$ are the criteria based on balancing the within-cluster and the between-cluster validation, respectively.

The CH Index was originally defined with the Euclidean distance as d, which connects it to the K-means objective function, but Milligan and Cooper (1985) examined different clustering quality indexes with their simulation studies and their findings indicated that the estimation of the number of clusters based on the CH Index, which can be found by maximising $I_{CH}(\mathcal{C})$, is quite successful with various clustering methods.

- **The Dunn (1974) Index** simply formalises the concept of the ratio between the minimal inter-cluster dissimilarity and the maximal intra-cluster dissimilarity. The index is given by the following equation.

$$I_{Dunn}(\mathcal{C}) = \frac{\min_{1 \le g < h \le K} d(C_g, C_h)}{\max_{1 \le k \le K} diam(C_k)} \in [0, 1], \tag{6.23}$$

where $d(C_g, C_h) = \min_{\mathbf{x}_i \in C_g, \mathbf{x}_j \in C_h} d(\mathbf{x}_i, \mathbf{x}_j)$ is the dissimilarity function between two clusters C_g and C_h, and $diam(C) = \max_{\mathbf{x}_i, \mathbf{x}_j \in C} d(\mathbf{x}_i, \mathbf{x}_j)$ is the diameter of a cluster, C, which might be considered as the spread of a cluster. If the data set contains well-separated and compact clusters at the same time, then the Dunn Index should be maximised. Otherwise, The Dunn Index might have difficulties to detect the appropriate number of clusters, because it only depends on the maximum and the minimum dissimilarities within clusters.

- **Clustering Validation Index Based on Nearest Neighbours (CVNN)** was proposed by Liu et al. (2013), and measures the separation that is based on how many of the κ nearest

neighbours of each observations are in the same cluster. The index is balancing separation and compactness statistics together in one single measure. The separation statistics is

$$I_{Sep}(\mathcal{C};\kappa) = \max_{1\leq k\leq K}\left(\frac{1}{n_k}\sum_{\mathbf{x}\in C_k}\frac{q_\kappa(\mathbf{x})}{\kappa}\right), \tag{6.24}$$

where $q_\kappa(\mathbf{x})$ is the number of observations among the κ nearest neighbours of $\mathbf{x}$ that are **not** in the same cluster. The compactness statistics ($I_{Com}(\mathcal{C})$) is just the average within-cluster dissimilarity as defined in Equation (6.1). For both statistics, small values are better. With this,

$$I_{CVNN}(\mathcal{C},\kappa) = \frac{I_{Sep}(\mathcal{C},\kappa)}{\max_{\mathcal{C}\in\mathcal{K}} I_{Sep}(\mathcal{C},\kappa)} + \frac{I_{Com}(\mathcal{C})}{\max_{\mathcal{C}\in\mathcal{K}} I_{Com}(\mathcal{C})}, \tag{6.25}$$

Here $\max_{\mathcal{C}\in\mathcal{K}} I_{Sep}(\mathcal{C})$ and $\max_{\mathcal{C}\in\mathcal{K}} I_{Com}(\mathcal{C})$ are computed over several clustering methods with different numbers of clusters $\mathcal{K} = \{\mathcal{C}_{K_{min}},\ldots,\mathcal{C}_{K_{max}}\}$.

One of the disadvantages of CVNN is that κ needs to be pre-specified by the user, so that one additional parameter has to be determined. In addition, Halkidi et al. (2015) pointed out that in the presence of outliers, CVNN penalises one-point clusters heavily, because such objects produce maximum possible $I_{Sep}(\mathcal{C};\kappa)$ value of 1.

The function `cluster.stats` in the `R` package "`fpc`" can assist users to compute most of the cluster validation indexes defined above. CVNN is the only function missing from the `fpc`, and I have manually implemented this function in `R`.

6.1.6 Stability

Clusterings are often interpreted as meaningful in the sense that they can be generalised as substantive patterns. This at least implicitly requires that they are stable (Hennig, 2017). Stability in cluster analysis can be used to estimate the number of clusters by implicitly defining the true clustering as the one with highest stability. This can be explored using some resampling techniques, such as bootstrap, splitting and jittering, see more information in Leisch (2015). I will review two popular approaches in detail, which have been defined using this principle.

Prediction strength

The prediction strength is defined according to Tibshirani and Walther (2005) as a tool to estimate the number of clusters using the idea of cluster stability by resampling methods. In particular, the

data set is randomly split into two halves, say $X_{[1]}$ and $X_{[2]}$, and two clusterings are obtained by using these two halves separately with the selected clustering technique and the number of clusters K. Then the points of $X_{[2]}$ are classified to the clusters of $X_{[1]}$, which is done by assigning every observation in $X_{[2]}$ to the closest cluster centroid in $X_{[1]}$. By using the idea of repeated cross-validation, the same is done with the points of $X_{[1]}$ relative to the clustering on $X_{[2]}$. For any pairs of observations in the same cluster in the same half, it is then checked whether or not they are predicted in the same clustering by the clustering on the other half. If this is the case, their co-memberships are correctly predicted. Finally, the prediction strength is defined by averaging the proportions of correctly predicted co-memberships for minimum clusters in each of the two halves.

Here is the mathematical definition. Let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a set of objects and $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ be a data set with n observations. For a candidate number of clusters K, let $n_1, \ldots, n_K$ be the number of observations in these clusters. Then prediction strength of the clustering is defined by

$$I_{PS}(\mathcal{C}) = \frac{1}{q}\sum_{t=1}^{q}\left\{\min_{1\le k\le K}\left(\frac{\sum_{i'\in A_k(i)} f_{ii'}(L_{[t]}, L^*_{[t]})\left[\left(\sum_{i'\in A_k(i)} f_{ii'}(L_{[t]}, L^*_{[t]})\right)-1\right]}{n_k(n_k-1)}\right)\right\} \in [0,1], \tag{6.26}$$

where $A_k(i)$ are the observations indexes $i^{'}$ such that $i \neq i^{'}$, q is the number of cross validation folds and

$$f_{ii'}(L_{[t]}, L^*_{[t]}) = \mathbf{1}\left(l_{i[t]} = l^*_{i[t]}\right), \tag{6.27}$$

where $\mathbf{1}(\cdot)$ is the indicator function, $L_{[t]} = \left\{l_{1[t]}, \ldots, l_{n_t[t]}\right\}$ and $L^*_{[t]} = \left\{l^*_{1[t]}, \ldots, l^*_{n_t[t]}\right\}$ are the set of cluster labels obtained from two different scenarios (n_t is the number of observations from the data set with the first t number of observations). The clustering labels are derived according to the equality, $L_{[t]} = C(X_{[t]}; K)$ using an appropriate clustering methodology (e.g., K-means) and by using t fold of the data set $X_{[t]}$, whereas $L^*_{[t]} = C^*(X; K, t)$ is derived from a classification technique connected with the clustering methodology by adopting both t part of the data set, $X_{[t]}$ and the complementary part, $X_{[-t]}$. Table 6.1 summarises various classification methodologies with their formulations of $C^*(X; K, t)$. Here the sets of two types of data are defined as

$$\mathcal{X}_{[t]} = \mathcal{X} \setminus \mathcal{X}_{[-t]} \quad \text{and} \quad \mathcal{X}_{[-t]} = \left\{\mathbf{x}_{(t-1)\left[\frac{n}{q}\right]+1}, \ldots, \mathbf{x}_{t\left[\frac{n}{q}\right]}\right\}$$

which essentially refers to q-fold cross validation. Tibshirani and Walther (2005) analysed two

different q-cross validation scenarios over the values of bias, variance and prediction error, and concluded that there is no advantage in using the five-fold over two-fold cross validation technique. For simplicity, I use two-fold cross validation ($q = 2$) in Equation (6.26).

The prediction rule recommended in Tibshirani and Walther (2005) is to predict observations into the cluster with the closest cluster centroid, which is appropriate for K-means. The software implementation of prediction strength can be found in `R` package `fpc` (Hennig, 2013), which provides alternative classification techniques (not provided in Tibshirani and Walther (2005)), as shown in Table 6.1, for different clustering algorithms. I will also contribute one additional classification approach (furthest neighbour distance) for the complete linkage method, which is not implemented in the `fpc` package.

Table 6.1: Classification of unclustered points

Classification approaches	**Formulations** $l^*_{i[t]}$
Centroid	$\underset{1\le k\le K}{\arg\min} \left\|\mathbf{x}_{i[t]} - \mathbf{m}_{c_k[t]}\right\|$
Nearest neighbour distance	$\underset{1\le k\le K}{\arg\min} \left(\min_{1\le j_k\le n_k} \left\|\mathbf{x}_{i[-t]} - \mathbf{x}_{j_k[t]}\right\|\right)$
Furthest neighbour distance	$\underset{1\le k\le K}{\arg\min} \left(\max_{1\le j_k\le n_k} \left\|\mathbf{x}_{i[-t]} - \mathbf{x}_{j_k[t]}\right\|\right)$
Average distance	$\underset{1\le k\le K}{\arg\min} \left(\frac{1}{n_k}\sum_{j_k=1}^{n_k} \left\|\mathbf{x}_{i[-t]} - \mathbf{x}_{j_k[t]}\right\|\right)$
QDA or LDA	$\underset{1\le k\le K}{\arg\max} \left\{\delta_k(\mathbf{x}_{i[-t]})\right\}$, where $\delta_k(\mathbf{x}_{i[t]}) = P(C_{i[t]} = k \mid \mathbf{x}_{j_k[t]})$

$\mathbf{m}_{C_k[t]}$ is the centroid of the k^{th} cluster in the data set $X_{[t]}$.

$\mathbf{x}_{j_k[t]}$ is the j_k^{th} object of the k^{th} cluster in the data set $X_{[t]}$.

$C_{i[t]}$ is the i^{th} cluster label of the data set $X_{[t]}$, where $i = 1, \ldots, \left(n - \left[\frac{n}{q}\right]\right)$.

$P(C_{i[t]} = k \mid \mathbf{x}_{j_k[t]})$ is the probability function, where $\mathbf{x}_{j_k[t]} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

In Table 6.1, "**centroid**" assigns observations to the cluster with the closest cluster centroid, which is associated to K-means, PAM, Ward's method, spectral clustering [1], since all these techniques are centroid based clustering algorithms. "**Nearest neighbour distance**", which is associated with single linkage clustering, classifies by the nearest neighbours. The idea of "**furthest neighbour distance**" is to choose from the furthest point of each clusterand classify to the nearest one. This clustering technique is connected with complete linkage clustering. "**Average distance**" assigns to the cluster to which an observation has the minimum average dissimilarity to all points in the cluster, which is associated to average linkage clustering, because the dissimilarity matrix is used as an input in their algorithms and the computation of the algorithms are made by averaging dissimilarity within clusters. **QDA** (Quadratic discriminant analysis) is associated with Gaussian clusters with flexible covariance matrices, which is appropriate for model based clustering, because $P(C_{i[t]} = k \mid \mathbf{x}_{j[t]})$ in Table 6.1 and $P(C_i = k \mid \mathbf{x}_i)$ in Equation (5.4) have the same structure in terms of estimating the probability function. **LDA** (Linear discriminant analysis) (Fisher, 1936) is an alternative approach for model based clustering as computational issues in `R` can arise using QDA when classes are small. The main difference between these two methodologies is that the covariance matrices for clusters K are all assumed to be equal ($\Sigma_k = \Sigma,\ \forall K$) in LDA, whereas the covariance matrices in QDA do not have to be equal. In Table 6.1, the LDA (or QDA) classification algorithm is estimated by using the probability function in Equation (5.5) modelled with the data set $X_{[t]}$, and the clustering labels for the complementary part, $X_{[-t]}$, is predicted by using this model.

The prediction strength measurement is applicable to both distance based and non-distance based clustering algorithms.

Based on Tibshirani and Walther (2005)'s findings from their experiments, the prediction strength value should be above 0.8 or 0.9 for choosing as the optimal number of clusters. In addition, they provided one additional parameter M, which is the number of iterations in which the data set is divided into two halves. The iterations contribute M different prediction strength values, and the final result is simply obtained by averaging these values.

The bootstrap method for estimating the number of clusters

In the previous section, the stability measurement for estimating the number of clusters is achieved by using the idea of cross validation, whereas here the bootstrap method [2], which is an alternative

[1] This type of spectral clustering procedure clusters points by adopting the K-means algorithm after Laplacian matrix transformation is applied, see Algorithm 8 for more details.

[2] The idea of bootstrap, which is simply the random sampling from the data set of interest with replacement, was first introduced by Efron (1981).

measurement for the stability assessment, is presented as explained in Fang and Wang (2012). The idea is simply to draw B times two bootstrap samples from the data, and the number of clusters is chosen by optimising an instability estimation from these pairs.

The mathematical definition is as follows. Let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a set of objects and $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ be a data set with n observations. The boot function for estimating the number of clusters is defined by

$$I_{Boot}(\mathcal{C}) = \frac{1}{B}\sum_{b=1}^{B}\left\{\frac{1}{n^2}\sum_{k=1}^{K}\left(\left[\sum_{i^{'}\in A_k(i)} f_{ii'}(L_{[1]}, L_{[2]})\right]^2 - \sum_{i^{'}\in A_k(i)}\left[f_{ii'}(L_{[1]}, L_{[2]})\right]^2\right)\right\} \in [0,1], \tag{6.28}$$

where $A_k(i)$ are the observations indexes $i^{'}$ such that $i \neq i^{'}$, and $f_{ii'}(L_{[1]}, L_{[2]})$ is the same as explained in Equation (6.27). Here both $L_{[1]}$ and $L_{[2]}$ are simply calculated as defined in Table 6.1. The only difference is that $X_{[1]}$ and $X_{[2]}$ are the bootstrapped and the non-bootstrapped sample points of the data set X, respectively. The optimal number of clusters can be estimated by choosing the minimum value of $I_{Boot}(\mathcal{C})$. As mentioned at the beginning of this chapter, we want to standardise the clustering quality indexes in such a way that large index values are always better. Thus, the $I_{Boot}(\mathcal{C})$ function is standardised by

$$I^{*}_{Boot}(\mathcal{C}) = 1 - I_{Boot}(\mathcal{C}) \in [0,1]. \tag{6.29}$$

All the arguments as explained in the previous section (for the usage of classification approaches over different clustering algorithms) can also be applied for $I_{Boot}(\mathcal{C})$, but there is no requirement that the values of $I^{*}_{Boot}(\mathcal{C})$ should be above or below some definite numbers for estimating the optimal number of clusters.

6.1.7 Further aspects and discussion

In the literature, researchers often suggest that the number of clusters should be "known" or otherwise it needs to be estimated from the data. However, Hennig (2015a) claimed that because finding the number of clusters in a certain application needs user input anyway, fixing the number of clusters is often as legitimate a user decision as the user input needed otherwise. Many different "objective" criteria for finding the best number of clusters exist in the literature and several of them are described in the previous sections. Based on all these arguments, in the given application the user needs to consider the following aspects for the decision of the number of clusters: 1)

estimating any underlying "true" number of clusters by using some of the objective criteria 2) implicitly defining what the best number of clusters is from the user's point of view for subject-matter reasons.

As stated in Section 3.1, there is no such thing as a universally "best clustering method", and different methods should be used for different aims of clustering. Researchers should also explicitly define their requirements and their definition of a "true cluster" based on their research aims, because clustering becomes scientific not through uniqueness but through transparent and open communication (Hennig, 2015b). This means that in order to decide about appropriate cluster analysis methodology, researchers should consider what data analytic characteristics the clusters they are aiming at are supposed to have. In this sense, (Hennig, 2015b) listed desirable characteristics which can be checked using the available data. Several of these are related with the "formal categorisation principles" listed in Van Mechelen et al. (1993, chap. 14). The desirable characteristics are:

1. Within-cluster dissimilarities should be small.
2. Between-cluster dissimilarities should be large.
3. The dissimilarity matrix of the data should be well represented by the clustering.
4. Low number of clusters is desirable.
5. Clusters should be stable.
6. Uniformity in cluster sizes is preferred.
7. Members of a cluster should be well represented by its centroid.
8. Clusters should be fitted well by certain homogeneous probability models such as the Gaussian or a uniform distribution, etc.
9. Clusters should correspond to connected areas in data space with high density.
10. The areas in data space corresponding to clusters should have certain characteristics.
11. The clusters should, if possible, be characterised by using a small number of variables
12. Clusters should correspond well to an externally given partition or values of one or more variables that were not used for computing the clustering.
13. Variables should be approximately independent within clusters.

Although all these potential characteristics are often important for constituting a good clustering, their level of importance may change depending on the clustering aim of the applications. For example, No.9 can be very important for image segmentation and No.8 may not be useful if users are interested in clustering of a data set without any statistical assumptions. All these arguments regarding the criteria that users may or may not consider give a direction to how their analysis will

proceed, such as which clustering methods and validation indexes are more appropriate for the clustering aim of a particular application.

Some of these characteristics are literally connected to the clustering quality indexes as I explained in the previous sections, but some others can be represented with different type of clustering quality validation indexes. For example, Hennig (2017) proposed several different indexes relevant to connected areas in data space with high density (No.9). These aspects simply measure density modes within clusters and density valleys between clusters. For a density-based cluster validation, Halkidi and Vazirgiannis (2008) introduced a cluster validity index, *CDbw*, which assesses the compactness and separation of clusters defined by a clustering algorithm. *CDbw* handles arbitrarily shaped clusters by representing each cluster with a number of points rather than by a single representative point. On the other hand, distribution-wise clustering quality assessment based on certain homogeneous probability models (No.8) are also discussed. For instance, Lago-Fernández and Corbacho (2010) introduced a clustering validity index which emphasises the cluster shape by using a high order characterization of its probability distribution, by looking at the neg-entropy distances [3] of the within-cluster distributions to the Gaussian distribution.

There might be some contradictions between some of these characteristics in a certain application. Connected areas with high density may have very large distances and may include undesired shapes. Roughly the same size clusters may produce large within-cluster dissimilarities if a cluster has a potential outlier.

A number of simulation studies have been presented for comparing different clustering quality indexes, see for example Milligan and Cooper (1988), Dimitriadou et al. (2002) and Arbelaitz et al. (2013). Although all these studies favour some of the clustering quality indexes (e.g., CH Index, ASW Index, etc.), which are usually performed with Gaussian data and some popular supervised classification data with known "truth", favouring one particular index may misguide when seeking the true number of clusters, because characteristics of clusters are varied in different applications. But users still need to make a decision about finding an appropriate number of clusters, therefore the decision should depend on the context and the clustering aim.

More indexes have been presented in different sources, see for example Halkidi et al. (2015) and many further indexes implemented in the "`NbClust`" package of `R`, see Charrad et al. (2014).

[3] Neg-entropy distance is a standard measure of the distance to normality which evaluates the difference between the cluster's entropy and the entropy of a normal distribution with the same covariance matrix (Lago-Fernández and Corbacho, 2010).

6.2 Aggregation of Clustering Quality Indexes

In the literature, a large number of methods for finding different numbers of clusters are available for researchers. Several of these are explained in the previous section in detail. Some of them attempt to balance a small within cluster heterogeneity and a large between-cluster heterogeneity, such as Average Silhouette Width (ASW) and Caliński and Harabasz (1974) index (CH), whereas some others have different goals; for example Pearson Gamma index emphasises good approximation of the dissimilarity structure by the clustering in the sense that whether observations are in different clusters should strongly be correlated with large dissimilarity. However, researchers may also prefer to implement several of these clustering quality indexes together in terms of their application of interest. For instance, one may aim to obtain a single value for different number of clusters in different clustering algorithms by aggregating several clustering quality index values. If this type of approach existed in the literature, it would be very valuable so that researchers may have the flexibility of which indexes they wish to use, as well as the clustering algorithm and what number of clusters they need to choose.

Hennig (2017) introduced a concept of aggregating the clustering quality indexes in such a way that the aggregated single criterion can be used to compare different clustering methods, different number of clusters and other possible parameter choices of clusterings. The following is the description of how this concept works.

The aggregation can be made by computing a weighted mean of selected indexes $I_1, \ldots, I_s$ with weights $w_1, \ldots, w_s > 0$, which are denoted as the relative importance of the different clustering quality indexes:

$$\mathcal{A}(\mathcal{C}) = \sum_{k=1}^{s} w_k I_k. \tag{6.30}$$

The weights can only be assigned to directly reflect the relative importance of the various clustering quality aspects if the values of the indexes are comparable in terms of their variations. As specified in the previous section, all indexes are standardised in the same value range $[0, 1]$. However, this may not be sufficient to establish comparability between indexes. For example, some of these indexes are scattered within roughly the whole value range whereas other indexes might be in a very small range (e.g., larger than 0.9) for all clusterings.

In order to provide a proper comparability between clustering quality indexes, researchers can consider standardising the index values that are derived from certain clustering algorithms with some random numbers generated from some random clustering algorithms from $\mathcal{X}$. Clustering

quality index values are then computed based on the result of the random clustering algorithms, so that users can construct some kind of distribution of the values for different number of clusters.

Generating completely random clusterings regardless of the information from the data set $\mathcal{X}$ is not suitable for this, because it can be expected that generating random clusterings without using any clustering algorithms will give completely unrelated and much worse results than for clusters that were generated by a clustering method. Hence, Hennig (2017) proposed two methods for generating random clusterings. These two random clustering algorithms are called "**Random K-centroids**" and "**Random nearest neighbour**". Here Random K-centroids is more connected with two popular centroid-based clustering algorithms, K-means and PAM, which generally produce compact clusters, whereas random nearest neighbour is related to single linkage which is more focused on cluster separation than on within-cluster homogeneity. Therefore, these two approaches generate clusterings that are in a certain sense opposite ways of clustering the data.

In addition to the algorithms above, I propose two additional random clustering algorithms: "**Random furthest neighbours**", which is more connected with complete linkage method, disregards separation and tends to keep the largest distance within clusters small. "**Random average neighbours**", which is related to average linkage method, compromises two random clustering algorithms (Random nearest and furthest neighbour clusterings) and focuses more on joining clusters with small variances. These additional algorithms will explore various overall ranges of clustering quality index values from different distributions as well as provide good collections of different random clusterings.

6.2.1 Random K-centroid

The algorithm of random K-centroid is very similar to the design of the K-means and the PAM algorithm. The random K-centroid works as follows. For a fixed number of clusters K, randomly select K cluster centroids from the data points, and assign every observation to the closest centroid.

6.2.2 Random K-neighbour

The algorithm of random neighbour clustering is typically similar to the structure of the hierarchical clustering algorithms (Single, complete and average linkage). The idea is that for a fixed number of clusters K randomly select K cluster initialisations from the data points and add the not yet assigned observation closest to any cluster to that cluster until all observations are clustered, see Algorithm 3.

Algorithm 2: The Random K-centroid Algorithm ($\mathcal{C}_{K-stupidcent}$)

input : $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ (set of objects for computing $\mathbf{D} = d(\mathbf{x}_i, \mathbf{x}_j)$ $(i, j = 1, \ldots, n)$, the matrix of dissimilarity to be clustered) or $\mathbf{D}$ (matrix of dissimilarity to be clustered), K (number of clusters)

output: $\mathcal{L} = \{l_1, l_2, \ldots, l_n\}$ (set of cluster labels)

 # initialise K random centroids, $\boldsymbol{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_K\} \in \mathcal{X}$

for $k \leftarrow 1$ **to** K **do**
  $\mathbf{m}_{C_k} \leftarrow \mathbf{s}_k$

for $i \leftarrow 1$ **to** n **do**
  # Assign every observations to the closest centroid:
  $l_i = \underset{1 \leq k \leq K}{\arg\min}\, d(\mathbf{x}_i, \mathbf{m}_{C_k}),\ i \in N_n$

return $\mathcal{L}$

6.2.3 Calibration

The random clusterings can be used in different ways to calibrate the clustering validation indexes. For a set of any value $\mathcal{K} = \{2, \ldots, K\}$ of interest, one could collect $4B + R$ clustering validation values of indexes as follows

$$\begin{aligned}\mathcal{C}_{\mathcal{K}-collection} = (\mathcal{C}_{\mathcal{K}:1}, \ldots, \mathcal{C}_{\mathcal{K}:4B+R}) = (&\mathcal{C}_{\mathcal{K}-real}(\mathcal{S}_1), \ldots, \mathcal{C}_{\mathcal{K}-real}(\mathcal{S}_R),\\ &\mathcal{C}_{\mathcal{K}-randomcen}(\mathcal{S}_1), \ldots, \mathcal{C}_{\mathcal{K}-randomcen}(\mathcal{S}_B),\\ &\mathcal{C}_{\mathcal{K}-randomsin}(\mathcal{S}_1), \ldots, \mathcal{C}_{\mathcal{K}-randomsin}(\mathcal{S}_B),\\ &\mathcal{C}_{\mathcal{K}-randomcom}(\mathcal{S}_1), \ldots, \mathcal{C}_{\mathcal{K}-randomcom}(\mathcal{S}_B),\\ &\mathcal{C}_{\mathcal{K}-randomave}(\mathcal{S}_1), \ldots, \mathcal{C}_{\mathcal{K}-randomave}(\mathcal{S}_B)),\end{aligned}$$

on generated $\mathcal{X}$. Here $\mathcal{C}_{\mathcal{K}-real}$ are clusterings over K number of clusters derived from a "real" clustering algorithm (e.g., K-means), and R is the number of clustering algorithms of interest. Four different types of random clusterings ($\mathcal{C}_{\mathcal{K}-random}$) over K number of clusters are collected from the random clustering algorithms, see Algorithm 2 and 3, and B is the number of clusterings of interest derived from a random clustering algorithm.

As mentioned previously, one could standardise the index values with a proper standardisation method for the sake of calibration over K clustering validation indexes of interest. The major argument here is how to standardise these index values. Two scenarios can be discussed:

Algorithm 3: The Random K-neighbour Clustering Algorithms

input : $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ (set of objects for computing $\mathbf{D} = d(\mathbf{x}_i, \mathbf{x}_j)$ $(i, j = 1, \ldots, n)$, the matrix of dissimilarity to be clustered) or $\mathbf{D}$ (matrix of dissimilarity to be clustered).

output: $\mathcal{C} = \{C_1, \ldots, C_K\}$ (set of clusters)

INITIALISATION:

Select random K initialization points, $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_K\} \in \mathcal{X}$. Let

$$\mathcal{C}(\mathcal{S}) = \{C_1, \ldots, C_K\} \leftarrow \{\{\mathbf{s}_1\}, \ldots, \{\mathbf{s}_K\}\}$$

$t \leftarrow 1$

$\mathbf{D}^{(t)} \leftarrow \mathbf{D}$

repeat

STEP 1:

Let $\mathcal{R}$ to be the set of the remaining points as $\mathcal{R} = \mathcal{X} \setminus \mathcal{S}$. Find the smallest dissimilarity between the remaining points and the initialisation points from $\mathbf{D}^{(t)}$ as

$$(g, h) \leftarrow \underset{\mathbf{x}_g \in \mathcal{R}, \mathbf{x}_h \in \mathcal{S}}{\arg\min}\, d^{(t)}(\mathbf{x}_g, \mathbf{x}_h)$$

STEP 2:

Adding points to clusters by: C_g, C_h: $C_l \leftarrow C_g \cup C_h$.

Mark g and h as unavailable: $\mathcal{S} \leftarrow \mathcal{S} \setminus \{\mathbf{x}_h\}$ and $\mathcal{R} \leftarrow \mathcal{R} \setminus \{\mathbf{x}_g\}$

STEP 3:

Mark l as available: $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathbf{x}_l\}$.

$\mathbf{x}_l$ is the new point in $\mathcal{S}$, where the dissimilarity measures with the other renaming points are computed as follows:

STEP 4:

foreach $\{\mathbf{x}_i\} \in \mathcal{R}$ **do**

Update dissimilarity matrix $d^{(t)}(\mathbf{x}_i, \mathbf{x}_l)$, if:

Random nearest: $\min\left\{d^{(t)}(\mathbf{x}_i, \mathbf{x}_g), d^{(t)}(\mathbf{x}_i, \mathbf{x}_h)\right\}$,

Random furthest: $\max\left\{d^{(t)}(\mathbf{x}_i, \mathbf{x}_g), d^{(t)}(\mathbf{x}_i, \mathbf{x}_h)\right\}$,

Random average: $(n_i n_g d^{(t)}(\mathbf{x}_i, \mathbf{x}_g) + n_i n_h d^{(t)}(\mathbf{x}_i, \mathbf{x}_h))/(n_i n_l)$, where n_i, n_l, n_g and n_h are the numbers of elements in clusters C_i, C_l, C_g and C_h respectively.

$t \leftarrow t + 1$

until $\mathcal{R} = \emptyset$

return $\mathcal{C}$

- It is possible to calibrate the indexes using real and random clusterings for the same K;
- An alternative version is to calibrate the indexes again using real and random clusterings for all values of K together.

I will give an example for better illustration of the scenarios above. Figure 6.1 shows CH Index values with $R = 8$ real clusterings and $4B = 400$ random clusterings for 9 different numbers of cluster scenarios. This is just an illustration, so we can disregard the values for this example. The calibration for the first scenario is simply to use the values of the first column for standardisation, then calibrate with another clustering validation index of interest for the same $K = 2$ and do this for the other columns (or other K's) in the same way for calibration. The second scenario uses whole values for standardisation instead of column by column calibration.

	2_Clusters	3_Clusters	4_Clusters	5_Clusters	6_Clusters	7_Clusters	8_Clusters	9_Clusters	10_Clusters
PAM	99.595	55.014	52.667	86.243	81.969	73.548	66.521	65.021	55.415
KMEANS	104.457	129.310	115.606	92.733	133.165	162.342	135.205	108.990	138.027
SINGLE	72.534	64.306	87.887	143.799	152.437	163.109	159.771	151.188	138.027
COMPLETE	93.846	103.437	157.289	169.130	181.697	168.902	185.805	181.910	192.345
AVERAGE	72.534	107.742	152.802	164.042	175.727	163.109	159.771	167.771	175.735
WARD	104.457	129.310	157.289	169.130	181.897	203.991	203.047	203.985	206.365
MCLUST	105.225	130.515	150.363	165.221	145.125	155.897	185.667	195.363	196.547
SPECTRAL	104.457	104.269	109.048	106.352	156.947	162.257	156.235	174.252	145.331
Random_centroid_1	55.635	31.895	7.558	6.112	6.055	51.105	19.992	20.895	5.029
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
Random_centroid_100	7.488	40.058	12.508	38.772	4.817	5.921	8.488	5.144	15.565
Random_single_1	4.325	5.409	5.108	4.245	11.628	3.963	6.026	14.807	6.132
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
Random_single_100	3.631	52.612	6.069	29.274	6.074	8.110	4.845	10.770	7.051
Random_complete_1	9.109	10.205	6.805	40.880	8.489	16.844	36.875	18.801	8.812
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
Random_ complete_100	8.093	5.641	6.676	6.028	5.611	16.482	17.674	19.320	7.092
Random_average_1	11.794	56.087	10.529	9.496	9.745	10.262	6.895	16.608	20.263
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
Random_average_100	5.595	7.113	8.004	18.412	15.147	33.112	6.105	8.699	4.468

Figure 6.1: Illustration of how a clustering validation values of indexes are generated

I will use the second scenario, which is to calibrate the indexes using real and random clusterings for all values of K together. One of the disadvantages of the first scenario can be explained as follows. It is possible that a weight that is derived from a standardisation method by using the index values for one specific number of clusters (say $K = 2$) can be much larger or much smaller than the standardisation weights for other number of clusters (say $K = 3, \ldots,$ or, 10). In this case, although the index values favour for $K = 2$ number of cluster without standardising the values, the index values over different number of clusters can be standardised with the division of standardisation weight so that one cannot identify the appropriate number of clusters. The second scenario does not affect the different number of clusters in one specific clustering validation index, because

all index values are used for computing a single weight for the sake of calibration. Figure 6.1 just illustrates an example of how clustering validation values of index are distributed for different clustering algorithms over different number of clusters. Several other clustering validation indexes of interest can be designed in the same way, then are calibrated with the appropriate standardisation methodology.

Now the question is how to choose a proper standardisation method for calibrating the index values. Consider I^* (Standardised version is used here since large values are good for all indexes) as a clustering quality index, and $\mathcal{C}_{\mathcal{K}-collection} = \{C_{k:j} : k \in \mathcal{K}; j = 1, \ldots, 4B + R\}$. With this, the following standardisation methodologies can be used for calibration.

- Z-score standardisation:

$$I_{Z-score}(\mathcal{C}) = \frac{I^*(\mathcal{C}) - m^*(\mathcal{C}_{\mathcal{K}-collection})}{\sqrt{\frac{1}{(4B+R)(K-1)-1}\sum_{k=2}^{K}\sum_{j=1}^{4B+R}\left(I^*(\mathcal{C}_{k:j}) - m^*(\mathcal{C}_{\mathcal{K}-collection})\right)^2}}, \tag{6.31}$$

  where

$$m^*(\mathcal{C}_{\mathcal{K}-collection}) = \frac{1}{(4B+R)(K-1)-1}\sum_{k=2}^{K}\sum_{j=1}^{4B+R} I^*(\mathcal{C}_{k:j}).$$

- Range standardisation:

$$I_{Range}(\mathcal{C}) = \frac{I^*(\mathcal{C}_{\mathcal{K}-collection}) - \min(\mathcal{C}_{\mathcal{K}-collection})}{\max(\mathcal{C}_{\mathcal{K}-collection}) - \min(\mathcal{C}_{\mathcal{K}-collection})}. \tag{6.32}$$

Hennig (2017) pointed out that using the rank standardisation in the set of clusterings can be another alternative to calibrate indexes. Although ranking the index values is probably more robust, some information might be lost with the rank standardisation. The argument for the choice of the standardisation technique is not different than the argument in Section 3.2.3, because the idea of standardisation is typically to make the values of interest comparable with each other. In Chapter 7, I will consider these standardisation techniques with the simulation studies.

6.3 Visualisation with R Shiny Application

This section explains how all the analysis in the next chapters will be conducted in a simple visualisation format, in which various considerations will be seen. For this task, the user interface implementation of `R` software, `R Shiny` has been used. `shiny` is an `R` package that makes it easy to build interactive web applications straight from `R`, see the web source `https://shiny.rstudio.com/` for more information.

In Appendix B, I present the `R Shiny` implementations as screen-shots for different data set scenarios. All these different visualisations will be described in detail. Figure B.1 is the demonstration of how all the different simulated data set scenarios are analysed including the visualisations for estimating the number of clusters by using the concept of the aggregation of clustering validation indexes. On the left-hand side, clustering validation indexes that are described in this chapter are presented, and different standardisation methodologies are shown at the bottom of the left side for the sake of aggregation of the selected indexes with the concept of random clustering calibration. On the top of the middle segment, graphical representations in two and three dimensions are displayed with their coordinates, which are obtained from Principal Component Analysis. The main title of these plots provides the proportion of variance explained based on the PCA solution. At the top of the right-hand side, the average of 50 simulated data set values of aggregation indexes is shown, while the choices of simulated data set, number of clusters and clustering algorithms are given on the right hand side. The summary table in the middle demonstrates the simulation results and the ARI values for different numbers of clusters and various clustering algorithms.

Figure B.2 and B.3 are very similar to the previous one, but I additionally implement the selection of clustering validation index results for real and random clustering algorithms. Figure B.3 also demonstrates the option for the weights of clustering validation indexes, so that users can have the flexibility to give different weights for the indexes. Note that the three dimensional scatter plots on these three figures are interactive plots so that users are able to rotate or zoom in or zoom out of the plots to better view the clustered points. Figure B.4 is the visualisation of distance query of a player of interest. The figure is described in Section 4.5 in detail.

6.4 Summary

Numerous clustering quality indexes have been described in terms of how to estimate the number of clusters. These indexes can be classified in such a way that calibrating the indexes with different goals can be legitimate and the aggregation of clustering quality indexes can be done by using the index values of random clustering algorithms. It is a new concept that researchers have the flexibility to select various clustering quality indexes as well as to estimate the number of clusters in an optimal way.

CHAPTER 7

EXAMINATION OF AGGREGATING CLUSTER VALIDATION INDEXES WITH VISUALISATION

The aim of this chapter is to investigate the new idea of aggregating clustering validation indexes by analysing some simulation scenarios and some real data sets. Different aggregated clustering validation index scenarios are established by using the clustering validation indexes obtained from random clustering algorithms, and they are compared with several single validation criteria as introduced in Chapter 6. In the next chapter, the dissimilarity matrix of the football performance data set that has been built in Chapter 4 will be examined for the sake of clustering football players.

The clustering algorithms introduced in Section 5.2 that are to be used for the analysis of aggregating clustering validation indexes are given in Table 7.1 along with their corresponding R implementations. The R functions that will be used for the clustering validation indexes are provided in Table 7.2 . Note that the R implementation of the CVNN Index and the random clustering algorithms were manually coded by myself. The functions of prediction strength and the bootstrap method for estimating the number of clusters were updated according to the additional implementation, which is the further neighbour distance as mentioned in Section 6.1.6.

For the analysis of different types of data sets the concept of aggregating clustering validation indexes is implemented with various calibration scenarios. In this respect, 8 real clustering algorithms (see Table 7.1) with 4 random clusterings each containing 100 objects are adopted. An illustration of how clustering validation index values are generated can be seen in Figure 6.1. In the calibration stage, the clustering validation indexes for real and random clusterings are aggregated with a choice of standardisation methodology (Z-score or Range) for all values of K together rather than for the same K. Many different calibration considerations can be generated from different clustering validation indexes. In the next sections, the choice of calibration for cluster analysis of different types of data sets is dependent on the clustering validation indexes which have dif-

Table 7.1: Clustering algorithms to be used for the data analysis

Clustering algorithms	**R package**	**R function**	**Details**
Partitioning Around Medoids (PAM)	`cluster` (Maechler et al., 2017)	`pam`	Parameters in the `R` function are all selected as default.
K-means	`stats` (R Core Team, 2013)	`kmeans`	Parameters in the `R` function are all selected as default.
Single linkage	`stats` (R Core Team, 2013)	`hclust`	`method = "single"` is selected. `cutree` function is used for partitioning the objects.
Complete linkage	`stats` (R Core Team, 2013)	`hclust`	`method = "complete"` is selected. `cutree` function is used for partitioning the objects.
Average linkage	`stats` (R Core Team, 2013)	`hclust`	`method = "average"` is selected. `cutree` function is used for partitioning the objects.
Ward's method	`stats` (R Core Team, 2013)	`hclust`	`method = "ward.D2"` is selected. `cutree` function is used for partitioning the objects.
Model based clustering	`mclust` (Scrucca et al., 2016)	`Mclust`	Gaussian distribution is used for the parameters of the assumed mixture distribution.
Spectral clustering	`kernlab` (Karatzoglou et al., 2004)	`specc`	Clustering is performed by embedding the data into the subspace of the eigenvectors of an affinity matrix, as explained in Algorithm 8.

Table 7.2: Clustering validation indexes to be used for the data analysis

Clustering validation indexes	**R package**	**R function**
Average within cluster dissimilarities, within cluster sum of squares, widest gap, average between cluster dissimilarities, separation index, entropy, PG Index, ASW Index, CH Index, Dunn Index	`fpc` (Hennig, 2013)	`cluster.stats`
Prediction strength		`prediction.strength`
The bootstrap method for estimating the number of clusters		`nselectboot`

ferent characteristics. In this sense, I will consider different combinations comprising one within cluster dissimilarity measure (e.g., average within cluster dissimilarities or widest gap, etc.), one between cluster dissimilarity measure (e.g., average between cluster dissimilarities or separation index), the Pearson Gamma Index (that simply measures the correlation between dissimilarities and clusterings) and one stability methodology (e.g., prediction strength or the bootstrap method). More details will be provided in the following sections.

7.1 Examination of Aggregating Clustering Validation Indexes on Simulated Data Sets

The purpose of this section is to describe various simulated data sets, which are to be replicated 50 times for the sake of simulation. The aim is to compare a number of different clustering algorithms and estimate the number of clusters by using various clustering validation indexes. The first four data set scenarios are simulated as explained in Tibshirani and Walther (2005), and the other scenarios are obtained from the `clusterSim` `R` package, see Walesiak et al. (2011).

The results of different combination scenarios of aggregated clustering validation indexes with different standardisation techniques are compared with various clustering validation indexes (called here "single criteria") as described in Section 6.1.5. In Section 7.1.8, all the results for different

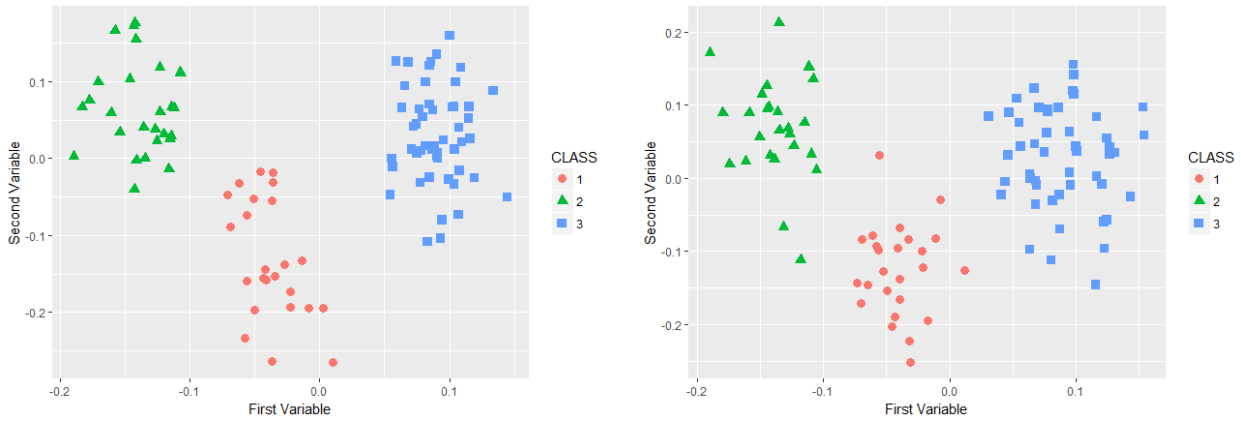
simulation scenarios are shown. In addition, Principal Component Analysis is used with 2 or 3 principal components for visualisation of the cluster points. Three dimensional plots do not fit on this two dimensional document, and there is no dynamic plot tool on the hard copy of this thesis with which the reader can explore the three dimensional plots. Instead, three dimensional graphs are rotated here and then presented as a two dimensional projection in such a way that readers can visualise the plots and identify the distinction between clusters in an optimal way. Note that 8 different clustering algorithms are analysed, but the results (See Section 7.1.8) are only plotted for the two clustering algorithms with the highest adjusted Rand index values based on the given true number of clusters in the simulation scenarios below. Numbers in the tables in Section 7.1.8 are counts out of 50 trials for different clustering validation indexes, and the number assigned corresponds to the number of times the values of clustering validation indexes are maximised for a given number of clusters. In addition, the average Rand index (ARI) value for each clustering validation criterion on the tables in 7.1.8 are computed by averaging the ARI values that are selected based on the maximum clustering validation index value over different numbers of clusters from each simulation.

7.1.1 Three clusters in two dimensions

- **Description:** This simulated data set is generated based on information from three clusters in two dimensions. The clusters are normally distributed variables with $(25, 25, 50)$ observations, centred at $(0, 0)$, $(0, 5)$, and $(5, -3)$, where each covariance matrix is the identity matrix, $\Sigma = I_2$.

- **Analysis:** Table 7.3 provides the average of the adjusted Rand index values from 50 simulations. The results indicate that PAM clustering and model based clustering algorithms have the two highest adjusted Rand index values, therefore clustering validation index results for these clustering methods will be analysed in detail. Figure 7.1 shows the solutions for $K = 3$ clusters where the cluster points of these two clustering algorithms are randomly selected from 50 replications.

  Next, clustering validation index results for two clustering algorithms are shown in Table 7.10 and 7.11. For the PAM algorithm, both the single criteria and the clustering validation indexes aggregated with the combination of average within and between cluster dissimilarities for both Z-score and range standardisations did quite well in estimating the correct number of clusters, whereas the aggregated ones with the widest gap and the separation index were not successful. The ARI values in Table 7.10 and 7.11, which are larger than 0.9 in most cases, also indicate that the majority of the clustering validation indexes perform



(a) PAM clustering

(b) Model based clustering

Figure 7.1: Three clusters in two dimensions - Two dimensional representation of a randomly selected simulated data set out of 50 replications

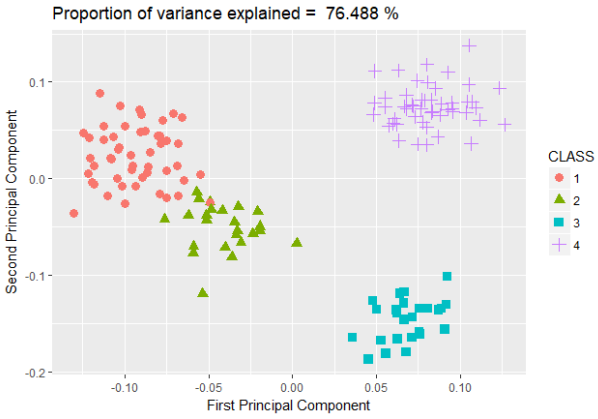
well. When model-based clustering is applied, the same conclusion can be made for most of the clustering validation indexes, but not for prediction strength and the bootstrap method, which simply measure the stability. Therefore, the stability measurement for model based clustering algorithm can be problematic to estimate the correct number of clusters for this specific simulated data set scenario.

7.1.2 Four clusters in 10 dimensions

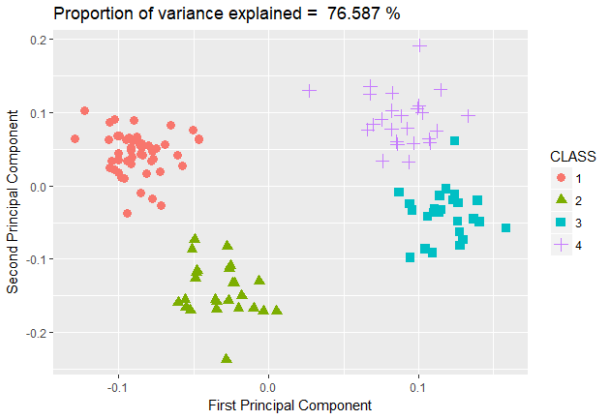
- **Description:** For this specific simulated scenario, four clusters are generated with 10 variables. Each cluster was randomly chosen to have 25 or 50 normally distributed observations, with centres randomly chosen as $N(0, 1.9I_{10})$. Any simulation with clusters having minimum distance less than 1.0 units between them was discarded. In this scenario, the settings are such that about one-half of the random realizations were discarded.

- **Analysis:** In this simulated setting, the number of variables is increased compared to the previous simulation scenario, and the simulated scenario was designed based on four numbers of clusters. The adjusted Rand index values in Table 7.4 indicate that PAM and model based clustering algorithms are the best solutions for this specific scenario. Figure 7.2 shows two dimensional representation and two dimensional projections of three dimensions of this data set for these two different clustering algorithms.

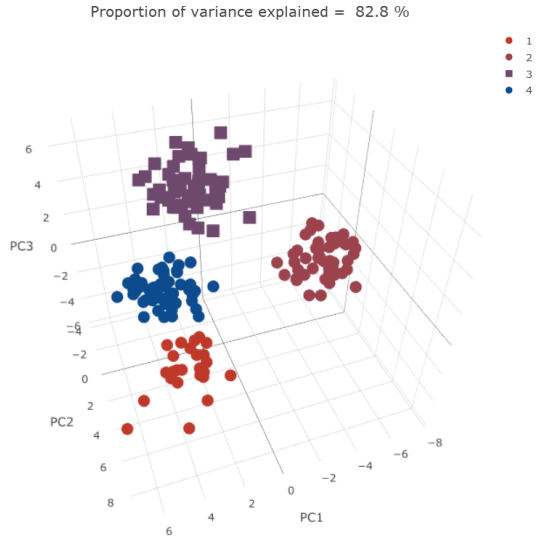
  Table 7.12 and 7.13 present various clustering validation index results. Based on any of the single criterion results, the majority of replications properly estimate the number of clusters



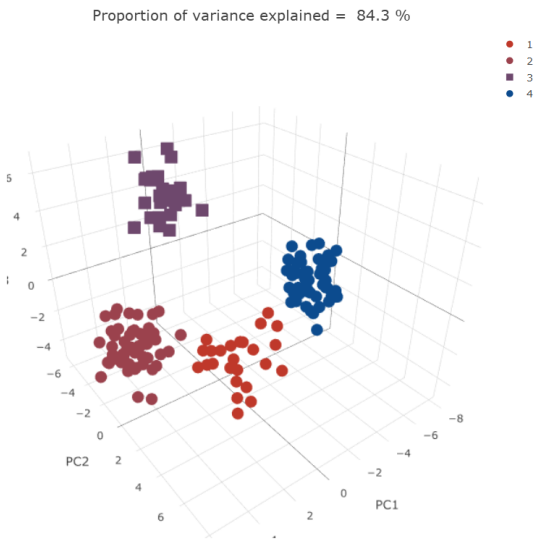
(a) PAM clustering - Two dimensional representation



(b) Model based clustering - Two dimensional representation



(c) PAM clustering - Two dimensional projections of three dimensions



(d) Model based clustering - Two dimensional projections of three dimensions

Figure 7.2: Four clusters in 10 dimensions - Two dimensional representation and two dimensional projections of three dimensions (PCA) of a randomly selected simulated data set out of 50 replications

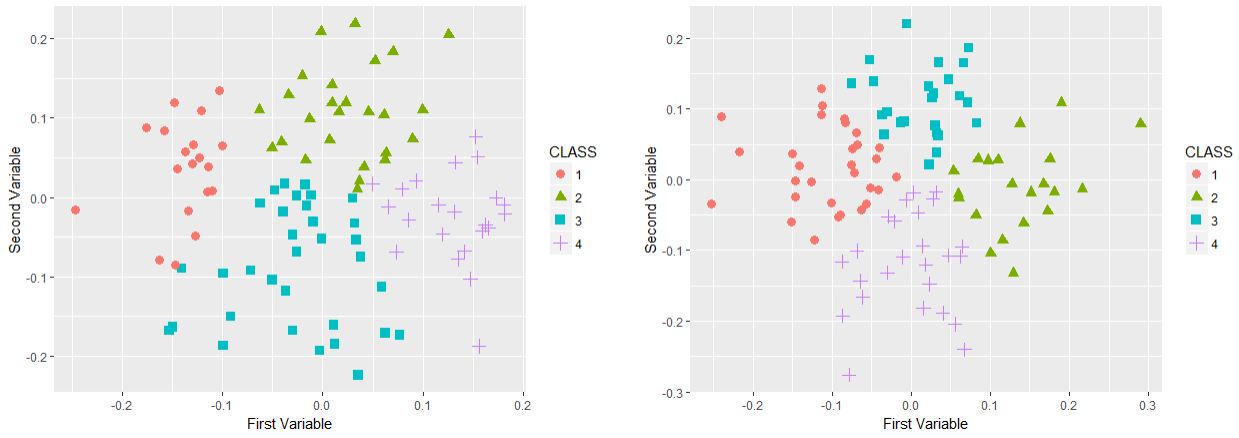
when the PAM algorithm is applied. For both clustering algorithms, the CVNN Index with different κ selections predicts the correct number of clusters well, while PG Index is the least successful comparing against the other validation criteria. For model based clustering, two stability measurements are again not able to estimate the correct number of clusters well. On the other hand, aggregation of clustering validation index scenarios, which are calibrated with average within and between cluster dissimilarities, indicate that range standardisation gives better estimation of the correct number of clusters than Z-score standardisation for both clustering algorithms, but PAM clustering gives even more accurate predictions than model based clustering. The aggregated scenarios with the widest gap and the separation index are usually not very successful in predicting the correct number of clusters, especially when model based clustering is applied for both standardisation techniques. The same conclusions can be made by looking at the ARI values.

7.1.3 Four clusters in two dimensions that are not well separated

- **Description:** This simulated scenario is more challenging than the other previous scenarios for estimating the given true number of clusters. The four numbers of clusters are generated in a two dimensional data set, but the clusters are not well separated at this time. Each cluster has 25 normally distributed observations, centred at $(0, 0)$, $(0, 2.5)$, $(2.5, 0)$ and $(2.5, 2.5)$, where each covariance matrix is the identity matrix, $\Sigma = I_2$.

- **Analysis:** Table 7.5 shows that the adjusted Rand index values of clustering algorithms against the true classes are greater than 0.5, which makes it difficult to predict the correct cluster labels based on the given scenario. However, as highlighted in Table 7.5, two clustering algorithms (PAM and K-means when $K = 4$) with the highest adjusted Rand index values will be chosen for the analysis of clustering validation indexes. Figure 7.3 shows that the simulated data set for both clustering algorithms are very homogeneous, so that it is not easy to detect the correct number of clusters for this specific type of data set.

  Tables 7.14 and 7.15 provide the clustering validation index results. For the PAM algorithm, the PG Index and the CVNN index ($\kappa = 10$) are both successful in predicting the majority of simulations. The other single criteria did not perform well. One of the interesting results is that prediction strength favours for a small number of clusters, whereas the other stability measurement, the bootstrap technique, estimates the largest number of clusters for this sort of homogeneous data set. Aggregation of different validation index results did not perform well in many instances with one exception, which is the aggregation of average within dissimilarities, average between dissimilarities, PG Index and prediction strength by applying



(a) PAM clustering

(b) K-means clustering

Figure 7.3: Four clusters in two dimensions that are not well separated - Two dimensional representation of a randomly selected simulated data set out of 50 replications

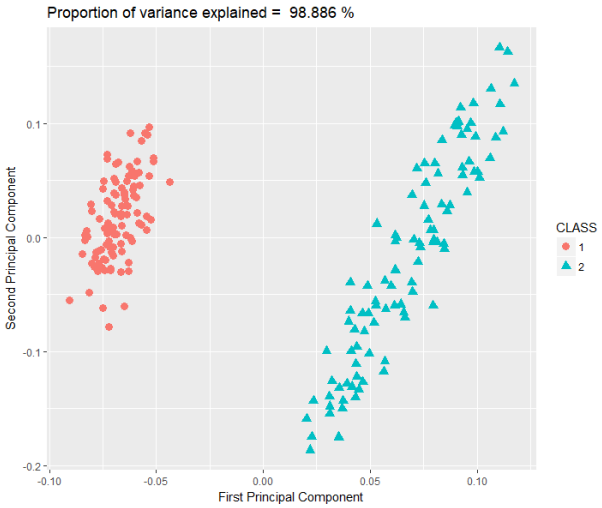
the range standardisation technique. For K-means algorithm, the same arguments (as made for PAM) can be made for many instances, but there are some little differences for some cases.

7.1.4 Two elongated clusters in three dimensions

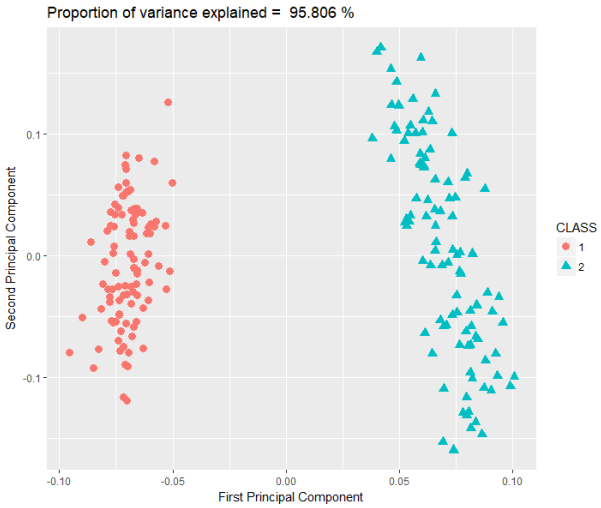
- **Description:** Two elongated clusters with three dimensional data are generated for this simulated scenario. Each cluster is generated as follows: Set $x_1 = x_2 = x_3 = t$ with t taking on 100 equally spaced values from $-.5$ to $.5$ ($t = \{-0.50, -0.49, \ldots, 0.49, 0.50\}$). Then Gaussian noise with standard deviation $.1$ is added to each feature ($\epsilon_i \sim N(0, 0.1)$ and $x_i = t + \epsilon_i$ for $i = 1, 2, 3$). Cluster 2 is generated in the same way, except that the value 1 is then added to each feature ($\epsilon_i \sim N(0, 1)$ and $x_i = t + \epsilon_i$ for $i = 1, 2, 3$).

- **Analysis:** The adjusted Rand index values in Table 7.6 show that the clustering algorithms almost perfectly predict the correct cluster labels. For analysis of clustering validation indexes, I selected two hierarchical clustering algorithms (complete and average linkage) in order to observe different clustering algorithms. Two and three dimensional scatter-plot representations are displayed in Figure 7.4.

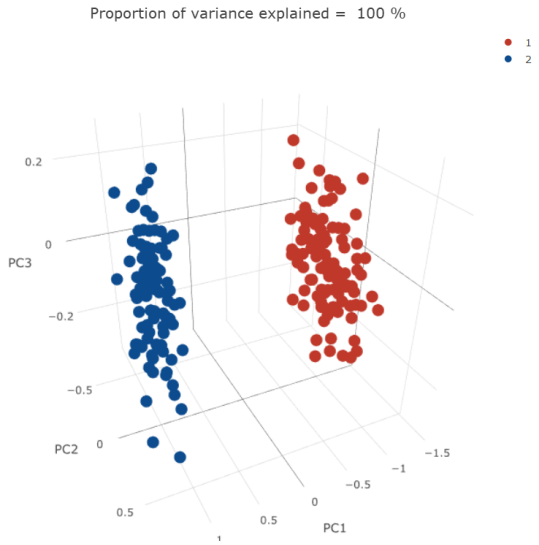
  Table 7.16 and 7.17 compares different validation indexes, and the results indicate that several of these criteria predict the correct number of clusters well, but the CVNN Index failed to estimate the true number of clusters. The results obtained from the aggregation of different clustering validation aspects almost certainly predict the correct number of clusters.



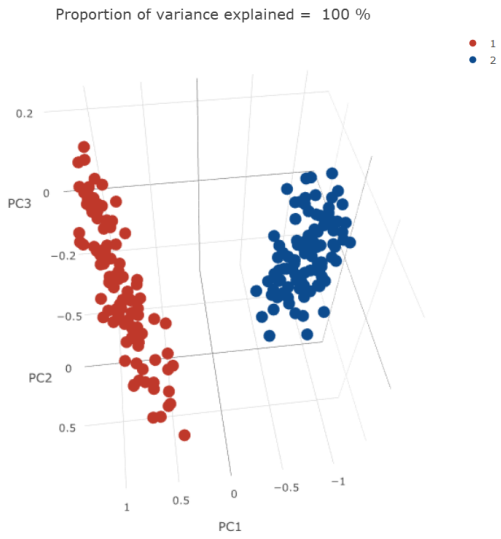
(a) Complete linkage - Two dimensional representation



(b) Average linkage - Two dimensional representation



(c) Complete linkage - Two dimensional projections of three dimensions



(d) Average linkage - Two dimensional projections of three dimensions

Figure 7.4: Two close and elongated clusters in three dimensions - Two dimensional representation and two dimensional projections of three dimensions (PCA) of a randomly selected simulated data set out of 50 replications

The next three simulated data sets have quite different cluster shapes than the previous ones. The simulation scenarios are distributed in such a way that the clusters are irregularly shaped.

7.1.5 Two clusters in two dimensions with ring shapes

- **Description:** For each point, firstly a random radius r is generated from a uniform distribution with a given interval, then a random angle $\alpha \sim U[0, 2\pi]$ is generated and finally the coordinates of points are calculated as $(r * \cos(\alpha), r * \sin(\alpha))$. The `shapes.circles2` `R` function is used, and the parameters in this function are all selected as default, such that number of objects in each cluster is 180, the range of r for the first (big) ring is $[0.75, 0.9]$, and the range of r for the second (small) ring is $[0.35, 0.5]$.

- **Analysis:** Figure 7.5 gives an illustration of this specific simulation. The two clustering algorithms are chosen based on the adjusted Rand index results in Table 7.7. The results indicate that there are substantial differences between the clustering algorithms, with single linkage and spectral clustering algorithms producing good results as they have the tendency to detect ring shaped clusters well in many situations.

(a) Single linkage

(b) Spectral clustering

Figure 7.5: Two clusters in two dimensions with ring shapes - Two dimensional representation of a randomly selected simulated data set out of 50 replications

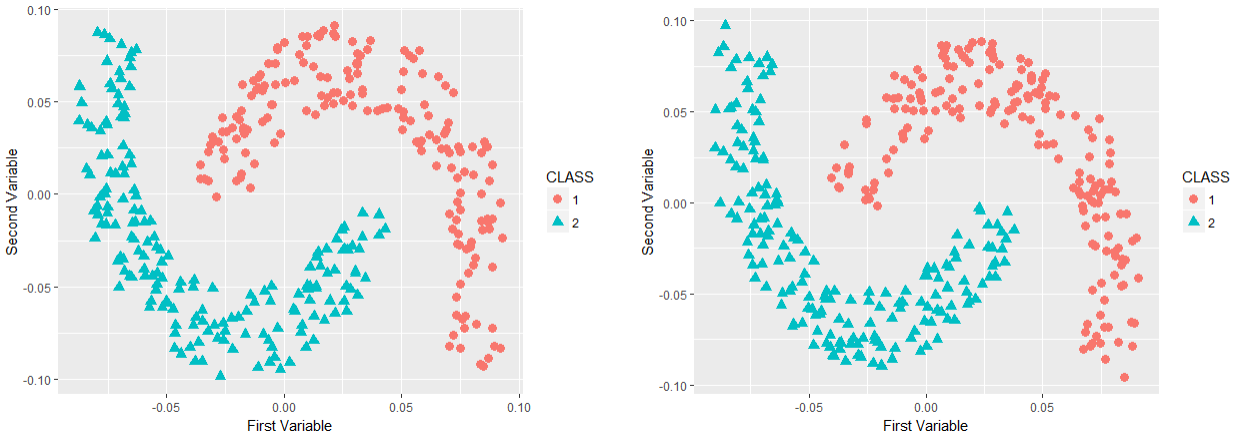
Tables 7.18 and 7.19 provide a summary of the clustering validation index results. For this and the next scenarios, the widest gap and the separation index combination gives better results than average within dissimilarities and average between dissimilarities. This is due to the fact that the two former indexes focus on the extreme points of clusters, therefore ring-shaped clusters can be identified in a better way by these clustering validation indexes. For

both clustering methods, the results indicate that the Dunn Index and the prediction strength predict the correct number of clusters well, while the other single criteria do not. That is because the other single criteria are more closely associated with compact clusters. On the other hand, the bootstrap method is quite successful for single linkage, but not for spectral clustering. The aggregation index and the ARI results in Table 7.18 and 7.19 show that the combination of the widest gap, the separation index and prediction strength methodologies predict the correct number of clusters well, whereas any combinations with the Pearson Gamma index does not in most cases.

7.1.6 Two clusters in two dimensions with two moon shapes

- **Description:** For each point first a random radius r is generated from a uniform distribution with a given interval, then a random angle $\alpha \sim U[0, 2\pi]$, and finally the coordinates of the points are calculated as $(a + |r * \cos(\alpha)|, r * \sin(\alpha))$ for the first shape and $(-|r * \cos(\alpha)|, r * \sin(\alpha) - b)$ for the second shape. The `shapes.two.moon R` function is used, and the parameters in this function are all selected as default, such that the number of objects in each cluster is 180, the range of r for the first and the second shapes are the same, $[0.8, 1.2]$, and $a = -0.4$ and $b = 1$.

- **Analysis:** Figure 7.6 visualises the cluster points on two dimensional plots for two clustering algorithms (single linkage and spectral clustering) which have the highest adjusted Rand index values, see Table 7.8.



(a) Single linkage

(b) Spectral clustering

Figure 7.6: Two clusters in two dimensions with two moon shapes - Two dimensional representation of a randomly selected simulated data set out of 50 replications

The results, as shown Table 7.20 and 7.21 give some idea of how various clustering validation index considerations performed. For single linkage, CH Index, ASW Index and CVNN Index with different κ values predict the correct number of clusters well, while the prediction by Pearson Gamma Index is not satisfactory. The Dunn Index and prediction strength give perfect solutions for both clustering algorithms. The indexes aggregated with the widest gap and the separation index almost surely predict the correct number of clusters well for both clustering algorithms when any standardisation method is applied. On the other hand, the clustering validation indexes aggregated with average within and between cluster dissimilarities are unsuccessful in estimating the true number of clusters in many instances, except the ones calibrated with the stability measurements when single linkage is used.

7.1.7 Two clusters in two dimensions with parabolic shapes

- **Description:** The final scenario is simulated based on parabolic shaped clusters (first is given by $y = x^2$, second by $y = -(x - a)^2 + b$ with distortion from $(-tol, +tol)$). The `shapes.worms R` function is used, and parameters in this function are all selected as default, such that the number of objects in each cluster is 180, the range on the x- axis for the first shape is $[-2, 2]$, and the second shape is $[-0.5, 2.5]$, and $a = 1.5$ and $b = 5.5$ with the tolerance parameter, $tol = 1$.

- **Analysis:** Figure 7.7 visualises these shapes when single linkage and spectral clustering (which again have two highest adjusted Rand index values, see Table 7.9) are applied. The clustering validation index results are shown in Tables 7.22 and 7.23. For single linkage, most of the single criteria perform fairly well in predicting the correct number of clusters. The aggregation of indexes is often successful in estimating the number of clusters, especially the ones aggregated with the widest gap and the separation index, but Pearson Gamma Index is not. For spectral clustering, the performances of the Dunn Index and the prediction strength perform well at predicting the correct number of clusters, while the other single criteria do not give good results. Different aggregated clustering validation index results are able to estimate the correct number of clusters well for the majority of the replications, except the ones calibrated with average within and between cluster dissimilarities.

(a) Single linkage

(b) Spectral clustering

Figure 7.7: Two clusters in two dimensions with parabolic shapes - Two dimensional representation of a randomly selected simulated data set out of 50 replications

7.1.8 Detailed results of simulated data sets

Adjusted Rand Index results

The following tables present the average of adjusted Rand index values from 50 simulated data sets."*" indicates the column corresponding to the correct number of clusters. The highlighted values are the two highest adjusted Rand index values based on the given true number of clusters in the simulation scenarios above.

Table 7.3: Three clusters in two dimensions.

Clustering Algorithm	**Estimate of Number of Clusters,** $\hat{k}$								
	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$
PAM	0.703	0.990*	0.726	0.619	0.533	0.463	0.406	0.366	0.332
K-means	0.726	0.884*	0.773	0.653	0.576	0.512	0.452	0.413	0.374
Single linkage	0.556	0.789*	0.873	0.893	0.905	0.892	0.891	0.874	0.857
Complete linkage	0.705	0.974*	0.840	0.735	0.635	0.562	0.490	0.445	0.410
Average linkage	0.739	0.984*	0.948	0.912	0.854	0.771	0.696	0.643	0.560
Ward's method	0.741	0.984*	0.753	0.647	0.564	0.485	0.429	0.393	0.357
Model based clustering	0.744	0.992*	0.882	0.792	0.711	0.630	0.547	0.481	0.449
Spectral clustering	0.671	0.907*	0.904	0.781	0.659	0.534	0.456	0.411	0.358

Table 7.4: Four clusters in 10 dimensions.

Clustering Algorithm	**Estimate of Number of Clusters, $\hat{k}$**								
	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$
PAM	0.397	0.715	0.943∗	0.825	0.714	0.637	0.574	0.518	0.475
K-means	0.420	0.688	0.843∗	0.829	0.767	0.683	0.621	0.586	0.569
Single linkage	0.263	0.406	0.522∗	0.565	0.586	0.618	0.624	0.634	0.647
Complete linkage	0.389	0.686	0.938∗	0.884	0.830	0.758	0.700	0.635	0.588
Average linkage	0.364	0.675	0.931∗	0.925	0.924	0.913	0.898	0.878	0.860
Ward's method	0.414	0.725	0.941∗	0.841	0.742	0.666	0.609	0.562	0.515
Model based clustering	0.416	0.743	0.955∗	0.861	0.770	0.706	0.654	0.605	0.561
Spectral Clustering	0.382	0.679	0.899∗	0.872	0.787	0.702	0.609	0.562	0.509

Table 7.5: Four clusters in two dimensions that are not well separated.

Clustering Algorithm	**Estimate of Number of Clusters, $\hat{k}$**								
	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$
PAM	0.253	0.391	0.487∗	0.437	0.393	0.356	0.321	0.296	0.275
K-means	0.291	0.394	0.478∗	0.428	0.388	0.343	0.316	0.294	0.282
Single linkage	0.000	0.000	0.003∗	0.017	0.018	0.020	0.025	0.036	0.049
Complete linkage	0.195	0.313	0.351∗	0.347	0.344	0.327	0.308	0.296	0.277
Average linkage	0.099	0.221	0.326∗	0.359	0.362	0.364	0.349	0.338	0.318
Ward's method	0.234	0.358	0.412∗	0.395	0.373	0.349	0.322	0.301	0.281
Model based clustering	0.240	0.341	0.444∗	0.427	0.388	0.354	0.335	0.315	0.290
Spectral clustering	0.150	0.302	0.380∗	0.398	0.354	0.338	0.310	0.294	0.273

Table 7.6: Two close and elongated clusters in three dimensions.

Clustering Algorithm	**Estimate of Number of Clusters, $\hat{k}$**								
	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$
PAM	0.995∗	0.751	0.539	0.424	0.343	0.299	0.262	0.233	0.210
K-means	0.995∗	0.750	0.597	0.472	0.430	0.353	0.291	0.274	0.229
Single linkage	1.000∗	0.990	0.979	0.968	0.957	0.935	0.923	0.901	0.884
Complete linkage	1.000∗	0.768	0.563	0.456	0.381	0.334	0.293	0.262	0.240
Average linkage	1.000∗	0.778	0.598	0.508	0.436	0.396	0.358	0.334	0.310
Ward's method	1.000∗	0.761	0.542	0.435	0.364	0.312	0.270	0.241	0.218
Model based clustering	1.000∗	0.771	0.539	0.434	0.366	0.320	0.286	0.262	0.243
Spectral clustering	1.000∗	0.791	0.624	0.487	0.413	0.358	0.315	0.278	0.255

Table 7.7: Two clusters in two dimensions with untypical ring shapes.

Clustering Algorithm	**Estimate of Number of Clusters, $\hat{k}$**								
	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$
PAM	0.000∗	0.001	0.002	0.022	0.059	0.096	0.132	0.156	0.164
K-means	0.001∗	0.001	0.001	0.003	0.015	0.051	0.077	0.104	0.125
Single linkage	1.000∗	0.824	0.733	0.690	0.663	0.642	0.624	0.612	0.597
Complete linkage	0.006∗	0.007	0.007	0.009	0.013	0.022	0.032	0.046	0.057
Average linkage	0.008∗	0.011	0.017	0.037	0.057	0.083	0.116	0.144	0.173
Ward's method	0.006∗	0.009	0.013	0.023	0.045	0.072	0.100	0.129	0.147
Model based clustering	0.001∗	0.001	0.018	0.054	0.110	0.152	0.169	0.186	0.192
Spectral clustering	1.000∗	0.773	0.642	0.491	0.421	0.360	0.325	0.276	0.258

Table 7.8: Two clusters in two dimensions with two moon shapes.

Clustering Algorithm	**Estimate of Number of Clusters,** $\hat{k}$								
	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$
PAM	0.338∗	0.339	0.382	0.371	0.335	0.290	0.253	0.228	0.203
K-means	0.293∗	0.308	0.333	0.341	0.327	0.295	0.262	0.235	0.209
Single linkage	1.000∗	0.925	0.825	0.784	0.725	0.655	0.626	0.584	0.555
Complete linkage	0.420∗	0.351	0.298	0.279	0.302	0.282	0.266	0.237	0.212
Average linkage	0.405∗	0.334	0.391	0.396	0.361	0.311	0.268	0.239	0.215
Ward's method	0.394∗	0.345	0.392	0.384	0.350	0.305	0.264	0.235	0.210
Model based clustering	0.390∗	0.263	0.502	0.407	0.338	0.299	0.259	0.231	0.208
Spectral clustering	1.000∗	0.811	0.600	0.494	0.396	0.366	0.315	0.284	0.239

Table 7.9: Two clusters in two dimensions with untypical parabolic shapes.

Clustering Algorithm	**Estimate of Number of Clusters,** $\hat{k}$								
	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$
PAM	0.509∗	0.401	0.366	0.350	0.303	0.264	0.237	0.214	0.198
K-means	0.525∗	0.433	0.407	0.361	0.312	0.284	0.259	0.228	0.207
Single linkage	0.980∗	0.931	0.902	0.858	0.823	0.787	0.750	0.724	0.693
Complete linkage	0.506∗	0.473	0.441	0.401	0.368	0.350	0.322	0.278	0.246
Average linkage	0.514∗	0.498	0.496	0.432	0.407	0.364	0.342	0.317	0.291
Ward's method	0.518∗	0.443	0.405	0.367	0.348	0.327	0.280	0.246	0.225
Model based clustering	0.390∗	0.405	0.437	0.400	0.348	0.297	0.266	0.236	0.215
Spectral clustering	1.000∗	0.863	0.700	0.591	0.477	0.410	0.356	0.323	0.286

Clustering validation index results

The tables below demonstrate numerous single criteria and various aggregated clustering validation index considerations for different simulation scenarios. Numbers are counts out of 50 trials for different clustering validation indexes. "*" indicates column corresponding to correct number of clusters. The ARI's are the average of adjusted Rand index values that are selected by the maximum clustering validation index value over different numbers of clusters from each simulation.

Table 7.10: Three clusters in two dimensions - PAM Algorithm.

Clustering Validation Index	**ARI**	**Estimate of Number of Clusters**, $\hat{k}$								
		$\hat{k}=2$	$\hat{k}=3$	$\hat{k}=4$	$\hat{k}=5$	$\hat{k}=6$	$\hat{k}=7$	$\hat{k}=8$	$\hat{k}=9$	$\hat{k}=10$
Single Criterion										
CH Index	0.9904	0	50∗	0	0	0	0	0	0	0
ASW Index	0.9609	6	44∗	0	0	0	0	0	0	0
Dunn Index	0.9372	11	39∗	0	0	0	0	0	0	0
Pearson gamma (PG)	0.9904	0	50∗	0	0	0	0	0	0	0
Prediction strength (PS)	0.9659	5	45∗	0	0	0	0	0	0	0
N select boot (NSB)	0.9904	0	50∗	0	0	0	0	0	0	0
CVNN ($\kappa = 5$)	0.9593	0	45∗	4	1	0	0	0	0	0
CVNN ($\kappa = 10$)	0.9904	0	50∗	0	0	0	0	0	0	0
CVNN ($\kappa = 20$)	0.9904	0	50∗	0	0	0	0	0	0	0
Z-score aggregation										
Ave.wit + Ave.bet	0.9904	0	50∗	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG	0.9904	0	50∗	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PS	0.9904	0	50∗	0	0	0	0	0	0	0
Ave.wit + Ave.bet + NSB	0.9904	0	50∗	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + PS	0.9904	0	50∗	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + NSB	0.9904	0	50∗	0	0	0	0	0	0	0
Wid.gap + Sep.Ind	0.9231	14	36∗	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG	0.9372	11	39∗	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PS	0.9423	10	40∗	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	0.9423	10	40∗	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	0.9469	9	41∗	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.9469	9	41∗	0	0	0	0	0	0	0
Range aggregation										
Ave.wit + Ave.bet	0.9904	0	50∗	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG	0.9904	0	50∗	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PS	0.9904	0	50∗	0	0	0	0	0	0	0
Ave.wit + Ave.bet + NSB	0.9904	0	50∗	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + PS	0.9904	0	50∗	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + NSB	0.9904	0	50∗	0	0	0	0	0	0	0
Wid.gap + Sep.Ind	0.8923	20	30∗	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG	0.9028	18	32∗	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PS	0.9372	11	39∗	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	0.9322	12	38∗	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	0.9271	13	37∗	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.9322	12	38∗	0	0	0	0	0	0	0

Ave.wit = Average within dissimilarities, Ave.bet = Average between dissimilarities

Wid.gap = Widest gap within dissimilarities, Sep.Ind = Separation Index

Table 7.11: Three clusters in two dimensions - Model based clustering.

Clustering Validation Index	**ARI**	**Estimate of Number of Clusters**, $\hat{k}$								
		$\hat{k}=2$	$\hat{k}=3$	$\hat{k}=4$	$\hat{k}=5$	$\hat{k}=6$	$\hat{k}=7$	$\hat{k}=8$	$\hat{k}=9$	$\hat{k}=10$
Single Criterion										
CH Index	0.9915	0	50∗	0	0	0	0	0	0	0
ASW Index	0.9624	0	44∗	0	0	0	0	0	0	0
Dunn Index	0.9227	14	32∗	4	0	0	0	0	0	0
Pearson gamma (PG)	0.9911	0	49∗	1	0	0	0	0	0	0
Prediction strength (PS)	0.8299	33	17∗	0	0	0	0	0	0	0
N select boot (NSB)	0.8841	22	28∗	0	0	0	0	0	0	0
CVNN ($\kappa = 5$)	0.9735	0	47∗	2	1	0	0	0	0	0
CVNN ($\kappa = 10$)	0.9915	0	50∗	0	0	0	0	0	0	0
CVNN ($\kappa = 20$)	0.9915	0	50∗	0	0	0	0	0	0	0
Z-score aggregation										
Ave.wit + Ave.bet	0.9911	0	49∗	1	0	0	0	0	0	0
Ave.wit + Ave.bet + PG	0.9911	0	49∗	1	0	0	0	0	0	0
Ave.wit + Ave.bet + PS	0.9864	1	48∗	1	0	0	0	0	0	0
Ave.wit + Ave.bet + NSB	0.9911	0	49∗	1	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + PS	0.9911	0	49∗	1	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + NSB	0.9911	0	49∗	1	0	0	0	0	0	0
Wid.gap + Sep.Ind	0.8713	24	21∗	3	2	0	0	0	0	0
Wid.gap + Sep.Ind + PG	0.9176	15	31∗	3	1	0	0	0	0	0
Wid.gap + Sep.Ind + PS	0.8598	27	21∗	2	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	0.8838	22	26∗	2	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	0.8940	20	28∗	2	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.9184	15	33∗	2	0	0	0	0	0	0
Range aggregation										
Ave.wit + Ave.bet	0.9911	0	49∗	1	0	0	0	0	0	0
Ave.wit + Ave.bet + PG	0.9911	0	49∗	1	0	0	0	0	0	0
Ave.wit + Ave.bet + PS	0.9911	1	48∗	1	0	0	0	0	0	0
Ave.wit + Ave.bet + NSB	0.9864	0	49∗	1	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + PS	0.9911	0	49∗	1	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + NSB	0.9911	0	49∗	1	0	0	0	0	0	0
Wid.gap + Sep.Ind	0.8339	32	15∗	3	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG	0.8543	28	20∗	2	0	0	0	0	0	0
Wid.gap + Sep.Ind + PS	0.8552	28	22∗	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	0.8405	31	19∗	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	0.8501	29	21∗	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.8644	26	24∗	0	0	0	0	0	0	0

Ave.wit = Average within dissimilarities, Ave.bet = Average between dissimilarities

Wid.gap = Widest gap within dissimilarities, Sep.Ind = Separation Index

Table 7.12: Four clusters in 10 dimensions - PAM Algorithm.

Clustering Validation Index	**ARI**	**Estimate of Number of Clusters**, $\hat{k}$								
		$\hat{k}=2$	$\hat{k}=3$	$\hat{k}=4$	$\hat{k}=5$	$\hat{k}=6$	$\hat{k}=7$	$\hat{k}=8$	$\hat{k}=9$	$\hat{k}=10$
Single Criterion										
CH Index	0.8609	7	6	37∗	0	0	0	0	0	0
ASW Index	0.8065	10	8	32∗	0	0	0	0	0	0
Dunn Index	0.7945	11	3	33∗	1	0	0	0	1	1
Pearson gamma (PG)	0.7957	9	11	30∗	0	0	0	0	0	0
Prediction strength (PS)	0.8098	12	3	35∗	0	0	0	0	0	0
N select boot (NSB)	0.8477	9	2	39∗	0	0	0	0	0	0
CVNN ($\kappa = 5$)	0.9172	1	5	44∗	0	0	0	0	0	0
CVNN ($\kappa = 10$)	0.9063	1	8	41∗	0	0	0	0	0	0
CVNN ($\kappa = 20$)	0.8996	2	8	40∗	0	0	0	0	0	0
Z-score aggregation										
Ave.wit + Ave.bet	0.7484	12	12	26∗	0	0	0	0	0	0
Ave.wit + Ave.bet + PG	0.7723	10	12	28∗	0	0	0	0	0	0
Ave.wit + Ave.bet + PS	0.7837	13	6	31∗	0	0	0	0	0	0
Ave.wit + Ave.bet + NSB	0.8098	10	7	33∗	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + PS	0.7888	12	7	31∗	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + NSB	0.7999	10	8	32∗	0	0	0	0	0	0
Wid.gap + Sep.Ind	0.7053	18	8	23∗	1	0	0	0	0	0
Wid.gap + Sep.Ind + PG	0.7433	16	6	28∗	0	0	0	0	0	0
Wid.gap + Sep.Ind + PS	0.7483	17	4	29∗	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	0.7658	15	3	32∗	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	0.7870	14	3	33∗	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.8006	12	3	35∗	0	0	0	0	0	0
Range aggregation										
Ave.wit + Ave.bet	0.8871	1	7	39∗	2	0	0	0	0	1
Ave.wit + Ave.bet + PG	0.8525	4	8	37∗	1	0	0	0	0	0
Ave.wit + Ave.bet + PS	0.8546	4	5	41∗	0	0	0	0	0	0
Ave.wit + Ave.bet + NSB	0.8901	1	5	43∗	0	0	0	0	0	1
Ave.wit + Ave.bet + PG + PS	0.9097	7	5	38∗	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + NSB	0.8982	2	5	42∗	1	0	0	0	0	0
Wid.gap + Sep.Ind	0.7053	18	8	23∗	1	0	0	0	0	1
Wid.gap + Sep.Ind + PG	0.7259	17	7	26∗	0	0	0	0	0	0
Wid.gap + Sep.Ind + PS	0.7640	16	3	31∗	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	0.7427	18	3	29∗	0	0	0	0	0	1
Wid.gap + Sep.Ind + PG + PS	0.7747	15	2	33∗	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.7776	14	3	33∗	0	0	0	0	0	0

Ave.wit = Average within dissimilarities, Ave.bet = Average between dissimilarities

Wid.gap = Widest gap within dissimilarities, Sep.Ind = Separation Index

Table 7.13: Four clusters in 10 dimensions - Model based clustering.

Clustering Validation Index	**ARI**	**Estimate of Number of Clusters**, $\hat{k}$								
		$\hat{k}=2$	$\hat{k}=3$	$\hat{k}=4$	$\hat{k}=5$	$\hat{k}=6$	$\hat{k}=7$	$\hat{k}=8$	$\hat{k}=9$	$\hat{k}=10$
Single Criterion										
CH Index	0.8781	6	6	38∗	0	0	0	0	0	0
ASW Index	0.8153	9	9	32∗	0	0	0	0	0	0
Dunn Index	0.7958	12	5	32∗	1	0	0	0	1	1
Pearson gamma (PG)	0.7986	7	15	28∗	0	0	0	0	0	0
Prediction strength (PS)	0.6324	28	8	14∗	0	0	0	0	0	0
N select boot (NSB)	0.7488	13	5	9∗	23	0	0	0	0	0
CVNN ($\kappa = 5$)	0.9357	0	5	45∗	0	0	0	0	0	0
CVNN ($\kappa = 10$)	0.9344	1	4	45∗	0	0	0	0	0	0
CVNN ($\kappa = 20$)	0.9188	2	7	41∗	0	0	0	0	0	0
Z-score aggregation										
Ave.wit + Ave.bet	0.7541	11	15	24∗	0	0	0	0	0	0
Ave.wit + Ave.bet + PG	0.7762	9	15	26∗	0	0	0	0	0	0
Ave.wit + Ave.bet + PS	0.6737	22	11	17∗	0	0	0	0	0	0
Ave.wit + Ave.bet + NSB	0.7924	11	8	26∗	5	0	0	0	0	0
Ave.wit + Ave.bet + PG + PS	0.7417	16	9	25∗	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + NSB	0.7775	12	10	25∗	3	0	0	0	0	0
Wid.gap + Sep.Ind	0.5838	28	12	9∗	0	0	0	0	1	0
Wid.gap + Sep.Ind + PG	0.7193	16	12	21∗	1	0	0	0	0	0
Wid.gap + Sep.Ind + PS	0.6017	31	5	14∗	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	0.7088	18	11	20∗	1	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	0.6925	20	11	18∗	1	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.7555	13	13	23∗	1	0	0	0	0	0
Range aggregation										
Ave.wit + Ave.bet	0.8944	1	7	38∗	1	0	1	0	1	1
Ave.wit + Ave.bet + PG	0.8375	4	13	32∗	0	1	0	0	0	0
Ave.wit + Ave.bet + PS	0.7889	14	6	30∗	0	0	0	0	0	0
Ave.wit + Ave.bet + NSB	0.7836	4	5	28∗	13	0	0	0	0	1
Ave.wit + Ave.bet + PG + PS	0.8791	14	5	31∗	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + NSB	0.8623	5	7	32∗	6	0	0	0	0	0
Wid.gap + Sep.Ind	0.5597	30	13	7∗	0	0	0	0	1	0
Wid.gap + Sep.Ind + PG	0.6949	17	14	18∗	1	0	0	0	0	0
Wid.gap + Sep.Ind + PS	0.6868	21	10	19∗	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	0.5787	33	5	12∗	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	0.6895	20	10	19∗	1	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.7269	16	12	21∗	1	0	0	0	0	0

Ave.wit = Average within dissimilarities, Ave.bet = Average between dissimilarities

Wid.gap = Widest gap within dissimilarities, Sep.Ind = Separation Index

Table 7.14: Four clusters in two dimensions that are not well separated - PAM Algorithm.

Clustering Validation Index	**ARI**	**Estimate of Number of Clusters**, $\hat{k}$								
		$\hat{k}=2$	$\hat{k}=3$	$\hat{k}=4$	$\hat{k}=5$	$\hat{k}=6$	$\hat{k}=7$	$\hat{k}=8$	$\hat{k}=9$	$\hat{k}=10$
Single Criterion										
CH Index	0.3821	0	1	16∗	1	2	5	3	10	12
ASW Index	0.4177	3	15	18∗	1	3	3	1	1	5
Dunn Index	0.3434	1	2	3∗	3	4	9	11	6	11
Pearson gamma (PG)	0.4642	0	12	29∗	8	1	0	0	0	0
Prediction strength (PS)	0.2792	43	5	2∗	0	0	0	0	0	0
N select boot (NSB)	0.2774	1	0	0∗	0	0	0	0	2	47
CVNN ($\kappa=5$)	0.4313	2	9	18∗	7	5	2	6	1	0
CVNN ($\kappa=10$)	0.4542	0	10	28∗	6	4	0	2	0	0
CVNN ($\kappa=20$)	0.4424	2	23	22∗	2	1	0	0	0	0
Z-score aggregation										
Ave.wit + Ave.bet	0.3657	5	18	6∗	5	1	2	0	1	12
Ave.wit + Ave.bet + PG	0.4429	2	18	21∗	7	1	1	0	0	0
Ave.wit + Ave.bet + PS	0.3494	24	20	6∗	0	0	0	0	0	0
Ave.wit + Ave.bet + NSB	0.2989	1	0	2∗	1	1	1	1	9	34
Ave.wit + Ave.bet + PG + PS	0.4158	7	25	16∗	2	0	0	0	0	0
Ave.wit + Ave.bet + PG + NSB	0.4252	1	5	20∗	8	5	3	2	3	3
Wid.gap + Sep.Ind	0.3136	1	1	1∗	0	4	4	4	8	27
Wid.gap + Sep.Ind + PG	0.3886	0	4	10∗	11	4	8	3	3	7
Wid.gap + Sep.Ind + PS	0.3681	17	9	12∗	2	2	3	1	1	3
Wid.gap + Sep.Ind + NSB	0.2903	1	0	0∗	0	1	2	3	8	35
Wid.gap + Sep.Ind + PG + PS	0.4241	8	16	17∗	6	2	1	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.3495	1	0	6∗	4	3	7	5	8	16
Range aggregation										
Ave.wit + Ave.bet	0.2765	0	0	0∗	0	0	0	0	5	45
Ave.wit + Ave.bet + PG	0.3205	0	0	3∗	3	3	7	3	10	21
Ave.wit + Ave.bet + PS	0.4546	8	11	17∗	5	3	1	2	1	2
Ave.wit + Ave.bet + NSB	0.4126	0	0	0∗	0	0	0	0	2	48
Ave.wit + Ave.bet + PG + PS	0.2757	3	12	27∗	5	3	0	0	0	0
Ave.wit + Ave.bet + PG + NSB	0.2806	0	0	2∗	0	0	1	0	15	34
Wid.gap + Sep.Ind	0.2984	0	0	1∗	0	1	3	3	12	30
Wid.gap + Sep.Ind + PG	0.3775	0	4	9∗	9	3	7	4	4	10
Wid.gap + Sep.Ind + PS	0.4098	12	20	13∗	4	1	0	0	0	0
Wid.gap + Sep.Ind + NSB	0.3752	19	16	11∗	1	1	1	0	0	1
Wid.gap + Sep.Ind + PG + PS	0.2834	0	0	0∗	0	0	1	3	5	41
Wid.gap + Sep.Ind + PG + NSB	0.3051	1	0	2∗	2	0	2	3	12	28

Ave.wit = Average within dissimilarities, Ave.bet = Average between dissimilarities

Wid.gap = Widest gap within dissimilarities, Sep.Ind = Separation Index

Table 7.15: Four clusters in two dimensions that are not well separated - K-means algorithm.

Clustering Validation Index	**ARI**	**Estimate of Number of Clusters**, $\hat{k}$								
		$\hat{k}=2$	$\hat{k}=3$	$\hat{k}=4$	$\hat{k}=5$	$\hat{k}=6$	$\hat{k}=7$	$\hat{k}=8$	$\hat{k}=9$	$\hat{k}=10$
Single Criterion										
CH Index	0.3816	0	1	14∗	4	3	6	7	9	6
ASW Index	0.4190	2	13	18∗	3	3	4	3	1	3
Dunn Index	0.3421	1	0	4∗	6	2	9	4	11	13
Pearson gamma (PG)	0.4383	0	12	20∗	14	3	1	0	0	0
Prediction strength (PS)	0.2998	46	3	1∗	0	0	0	0	0	0
N select boot (NSB)	0.2819	0	0	0∗	0	0	0	0	1	49
CVNN ($\kappa=5$)	0.4328	0	3	21∗	8	8	7	3	0	0
CVNN ($\kappa=10$)	0.4572	0	7	25∗	8	8	2	0	0	0
CVNN ($\kappa=20$)	0.4642	2	14	28∗	2	3	1	0	0	0
Z-score aggregation										
Ave.wit + Ave.bet	0.3985	2	6	13∗	11	5	2	4	6	1
Ave.wit + Ave.bet + PG	0.4217	0	12	15∗	15	4	3	1	0	0
Ave.wit + Ave.bet + PS	0.3829	20	21	9∗	0	0	0	0	0	0
Ave.wit + Ave.bet + NSB	0.3466	0	0	9∗	5	2	2	1	12	19
Ave.wit + Ave.bet + PG + PS	0.4272	7	20	20∗	3	0	0	0	0	0
Ave.wit + Ave.bet + PG + NSB	0.4141	0	4	14∗	15	4	5	3	5	0
Wid.gap + Sep.Ind	0.3114	3	1	1∗	2	0	2	6	13	22
Wid.gap + Sep.Ind + PG	0.3756	0	3	9∗	10	3	4	7	8	6
Wid.gap + Sep.Ind + PS	0.3563	25	8	8∗	3	0	1	1	0	4
Wid.gap + Sep.Ind + NSB	0.2952	0	0	0∗	1	0	1	3	11	34
Wid.gap + Sep.Ind + PG + PS	0.4267	7	14	19∗	8	2	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.3493	0	2	7∗	5	1	2	6	9	18
Range aggregation										
Ave.wit + Ave.bet	0.2891	0	0	0∗	0	1	1	1	13	34
Ave.wit + Ave.bet + PG	0.3437	0	0	3∗	8	5	10	6	10	8
Ave.wit + Ave.bet + PS	0.4440	4	15	23∗	5	2	1	0	0	0
Ave.wit + Ave.bet + NSB	0.4424	0	0	0∗	0	0	0	0	6	44
Ave.wit + Ave.bet + PG + PS	0.2841	2	16	24∗	5	2	1	0	0	0
Ave.wit + Ave.bet + PG + NSB	0.3055	0	0	2∗	1	0	2	4	14	27
Wid.gap + Sep.Ind	0.2998	1	1	1∗	1	0	0	6	13	27
Wid.gap + Sep.Ind + PG	0.3636	0	3	8∗	8	2	3	7	8	11
Wid.gap + Sep.Ind + PS	0.4158	12	15	17∗	4	2	0	0	0	0
Wid.gap + Sep.Ind + NSB	0.3520	28	12	7∗	2	0	0	0	0	1
Wid.gap + Sep.Ind + PG + PS	0.2907	0	0	0∗	0	0	0	2	12	36
Wid.gap + Sep.Ind + PG + NSB	0.3220	0	0	5∗	1	0	3	5	13	23

Ave.wit = Average within dissimilarities, Ave.bet = Average between dissimilarities

Wid.gap = Widest gap within dissimilarities, Sep.Ind = Separation Index

Table 7.16: Two close and elongated clusters in three dimensions - Complete linkage.

Clustering Validation Index	**ARI**	**Estimate of Number of Clusters**, $\hat{k}$								
		$\hat{k}=2$	$\hat{k}=3$	$\hat{k}=4$	$\hat{k}=5$	$\hat{k}=6$	$\hat{k}=7$	$\hat{k}=8$	$\hat{k}=9$	$\hat{k}=10$
Single Criterion										
CH Index	0.7548	23∗	9	8	4	6	0	0	0	0
ASW Index	1.0000	50∗	0	0	0	0	0	0	0	0
Dunn Index	1.0000	50∗	0	0	0	0	0	0	0	0
Pearson gamma (PG)	0.9950	49∗	1	0	0	0	0	0	0	0
Prediction strength (PS)	0.9953	49∗	1	0	0	0	0	0	0	0
N select boot (NSB)	0.9748	45∗	4	1	0	0	0	0	0	0
CVNN ($\kappa = 5$)	0.5191	1∗	8	19	11	8	3	0	0	0
CVNN ($\kappa = 10$)	0.5160	1∗	6	24	9	7	3	0	0	0
CVNN ($\kappa = 20$)	0.5790	3∗	10	29	3	5	0	0	0	0
Z-score aggregation										
Ave.wit + Ave.bet	0.9903	48∗	2	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG	0.9950	49∗	1	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PS	0.9950	49∗	1	0	0	0	0	0	0	0
Ave.wit + Ave.bet + NSB	0.9950	49∗	1	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + PS	0.9950	49∗	1	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + NSB	0.9950	49∗	1	0	0	0	0	0	0	0
Wid.gap + Sep.Ind	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PS	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	1.0000	50∗	0	0	0	0	0	0	0	0
Range aggregation										
Ave.wit + Ave.bet	0.9903	48∗	2	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG	0.9950	49∗	1	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PS	0.9950	49∗	1	0	0	0	0	0	0	0
Ave.wit + Ave.bet + NSB	0.9950	49∗	1	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + PS	0.9950	49∗	1	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + NSB	0.9950	49∗	1	0	0	0	0	0	0	0
Wid.gap + Sep.Ind	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PS	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	1.0000	50∗	0	0	0	0	0	0	0	0

Ave.wit = Average within dissimilarities, Ave.bet = Average between dissimilarities

Wid.gap = Widest gap within dissimilarities, Sep.Ind = Separation Index

Table 7.17: Two close and elongated clusters in three dimensions - Average linkage.

Clustering Validation Index	**ARI**	**Estimate of Number of Clusters**, $\hat{k}$								
		$\hat{k}=2$	$\hat{k}=3$	$\hat{k}=4$	$\hat{k}=5$	$\hat{k}=6$	$\hat{k}=7$	$\hat{k}=8$	$\hat{k}=9$	$\hat{k}=10$
Single Criterion										
CH Index	0.7427	22∗	8	11	3	3	2	1	0	0
ASW Index	1.0000	50∗	0	0	0	0	0	0	0	0
Dunn Index	1.0000	50∗	0	0	0	0	0	0	0	0
Pearson gamma (PG)	0.9950	48∗	2	0	0	0	0	0	0	0
Prediction strength (PS)	1.0000	50∗	0	0	0	0	0	0	0	0
N select boot (NSB)	0.9952	49∗	1	0	0	0	0	0	0	0
CVNN ($\kappa=5$)	0.5214	3∗	6	19	8	7	3	4	0	0
CVNN ($\kappa=10$)	0.5327	3∗	7	20	8	8	2	2	0	0
CVNN ($\kappa=20$)	0.5754	3∗	11	25	6	4	1	0	0	0
Z-score aggregation										
Ave.wit + Ave.bet	0.9903	48∗	2	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG	0.9952	49∗	1	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PS	0.9952	49∗	1	0	0	0	0	0	0	0
Ave.wit + Ave.bet + NSB	0.9952	49∗	1	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + PS	0.9952	49∗	1	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + NSB	0.9952	49∗	1	0	0	0	0	0	0	0
Wid.gap + Sep.Ind	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PS	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	1.0000	50∗	0	0	0	0	0	0	0	0
Range aggregation										
Ave.wit + Ave.bet	0.9903	48∗	2	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG	0.9952	49∗	1	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PS	0.9952	49∗	1	0	0	0	0	0	0	0
Ave.wit + Ave.bet + NSB	0.9952	49∗	1	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + PS	0.9952	49∗	1	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + NSB	0.9952	49∗	1	0	0	0	0	0	0	0
Wid.gap + Sep.Ind	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PS	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	1.0000	50∗	0	0	0	0	0	0	0	0

Ave.wit = Average within dissimilarities, Ave.bet = Average between dissimilarities

Wid.gap = Widest gap within dissimilarities, Sep.Ind = Separation Index

Table 7.18: Two clusters in two dimensions with untypical ring shapes - Single linkage.

Clustering Validation Index	**ARI**	**Estimate of Number of Clusters**, $\hat{k}$								
		$\hat{k}=2$	$\hat{k}=3$	$\hat{k}=4$	$\hat{k}=5$	$\hat{k}=6$	$\hat{k}=7$	$\hat{k}=8$	$\hat{k}=9$	$\hat{k}=10$
Single Criterion										
CH Index	0.6458	$0*$	4	8	10	11	6	5	3	0
ASW Index	0.7106	$2*$	17	13	8	6	2	1	1	0
Dunn Index	1.0000	$50*$	0	0	0	0	0	0	0	0
Pearson gamma (PG)	0.6171	$0*$	0	0	3	7	4	11	15	0
Prediction strength (PS)	1.0000	$50*$	0	0	0	0	0	0	0	0
N select boot (NSB)	0.9006	$37*$	0	0	0	0	3	2	5	0
CVNN ($\kappa=5$)	0.7003	$2*$	8	20	6	9	3	1	1	0
CVNN ($\kappa=10$)	0.7359	$5*$	15	16	7	5	2	0	0	0
CVNN ($\kappa=20$)	0.7849	$11*$	20	12	4	3	0	0	0	0
Z-score aggregation										
Ave.wit + Ave.bet	0.6188	$0*$	0	0	3	7	4	13	15	8
Ave.wit + Ave.bet + PG	0.6184	$0*$	0	0	3	7	4	13	14	9
Ave.wit + Ave.bet + PS	0.9257	$36*$	9	3	2	0	0	0	0	0
Ave.wit + Ave.bet + NSB	0.6178	$0*$	0	0	0	4	11	9	13	13
Ave.wit + Ave.bet + PG + PS	0.6728	$0*$	9	7	14	10	2	7	1	0
Ave.wit + Ave.bet + PG + NSB	0.6160	$0*$	0	0	0	4	9	11	13	13
Wid.gap + Sep.Ind	0.9989	$49*$	1	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG	0.6399	$0*$	0	0	6	11	9	7	9	8
Wid.gap + Sep.Ind + PS	1.0000	$50*$	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	0.9829	$47*$	0	0	1	1	0	1	0	0
Wid.gap + Sep.Ind + PG + PS	0.9854	$47*$	3	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.6342	$0*$	0	1	2	6	11	10	12	8
Range aggregation										
Ave.wit + Ave.bet	0.6137	$0*$	0	0	1	7	5	10	15	12
Ave.wit + Ave.bet + PG	0.6145	$0*$	0	0	2	7	4	10	16	11
Ave.wit + Ave.bet + PS	0.6565	$0*$	5	6	13	11	3	9	3	0
Ave.wit + Ave.bet + NSB	0.9257	$35*$	12	3	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + PS	0.6157	$0*$	0	0	0	3	10	10	13	14
Ave.wit + Ave.bet + PG + NSB	0.6142	$0*$	0	0	0	4	10	8	13	15
Wid.gap + Sep.Ind	0.9989	$49*$	1	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG	0.6418	$0*$	0	0	8	12	8	7	8	7
Wid.gap + Sep.Ind + PS	0.9902	$48*$	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	1.0000	$50*$	0	0	0	1	0	1	0	0
Wid.gap + Sep.Ind + PG + PS	0.9883	$48*$	2	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.6334	$0*$	0	0	2	7	11	11	11	8

Ave.wit = Average within dissimilarities, Ave.bet = Average between dissimilarities

Wid.gap = Widest gap within dissimilarities, Sep.Ind = Separation Index

Table 7.19: Two clusters in two dimensions with untypical ring shapes - Spectral clustering.

Clustering Validation Index	**ARI**	**Estimate of Number of Clusters**, $\hat{k}$								
		$\hat{k}=2$	$\hat{k}=3$	$\hat{k}=4$	$\hat{k}=5$	$\hat{k}=6$	$\hat{k}=7$	$\hat{k}=8$	$\hat{k}=9$	$\hat{k}=10$
Single Criterion										
CH Index	0.3241	0∗	2	2	0	5	4	11	12	14
ASW Index	0.2695	0∗	0	0	0	2	2	9	16	21
Dunn Index	1.0000	50∗	0	0	0	0	0	0	0	0
Pearson gamma (PG)	0.4287	0∗	1	6	8	6	5	8	7	9
Prediction strength (PS)	1.0000	50∗	0	0	0	0	0	0	0	0
N select boot (NSB)	0.2570	0∗	0	0	0	0	0	0	1	49
CVNN ($\kappa=5$)	0.4168	0∗	6	7	5	4	9	9	7	3
CVNN ($\kappa=10$)	0.3884	0∗	4	6	6	8	7	10	4	5
CVNN ($\kappa=20$)	0.3998	0∗	3	9	10	7	7	7	5	2
Z-score aggregation										
Ave.wit + Ave.bet	0.4507	0∗	7	6	6	7	4	4	7	9
Ave.wit + Ave.bet + PG	0.4190	0∗	3	5	8	7	3	6	7	11
Ave.wit + Ave.bet + PS	0.5206	0∗	14	9	7	4	3	2	6	5
Ave.wit + Ave.bet + NSB	0.2654	0∗	0	0	0	0	1	2	15	32
Ave.wit + Ave.bet + PG + PS	0.4856	0∗	11	6	8	7	3	2	7	6
Ave.wit + Ave.bet + PG + NSB	0.3067	0∗	0	1	1	2	2	7	14	23
Wid.gap + Sep.Ind	0.9989	49∗	1	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG	0.7049	2∗	18	20	6	3	1	0	0	0
Wid.gap + Sep.Ind + PS	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	0.7520	16∗	4	15	5	2	1	2	1	4
Wid.gap + Sep.Ind + PG + PS	0.9470	40∗	7	3	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.5526	0∗	6	14	9	3	4	3	5	6
Range aggregation										
Ave.wit + Ave.bet	0.3169	0∗	1	3	1	4	3	7	12	19
Ave.wit + Ave.bet + PG	0.3504	0∗	1	4	2	6	3	7	11	16
Ave.wit + Ave.bet + PS	0.4087	0∗	5	5	3	6	4	5	10	12
Ave.wit + Ave.bet + NSB	0.3877	0∗	7	4	1	4	2	4	13	15
Ave.wit + Ave.bet + PG + PS	0.2605	0∗	0	0	0	0	1	1	15	33
Ave.wit + Ave.bet + PG + NSB	0.2822	0∗	0	1	0	0	1	7	15	26
Wid.gap + Sep.Ind	0.9989	49∗	1	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG	0.7184	3∗	19	19	6	2	1	0	0	0
Wid.gap + Sep.Ind + PS	0.9422	39∗	8	3	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	0.7432	15∗	4	16	5	2	1	1	2	4
Wid.gap + Sep.Ind + PG + NSB	0.5709	0∗	6	16	9	3	4	2	4	6

Ave.wit = Average within dissimilarities, Ave.bet = Average between dissimilarities

Wid.gap = Widest gap within dissimilarities, Sep.Ind = Separation Index

Table 7.20: Two clusters in two dimensions with two moon shapes - Single linkage.

Clustering Validation Index	**ARI**	**Estimate of Number of Clusters**, $\hat{k}$								
		$\hat{k}=2$	$\hat{k}=3$	$\hat{k}=4$	$\hat{k}=5$	$\hat{k}=6$	$\hat{k}=7$	$\hat{k}=8$	$\hat{k}=9$	$\hat{k}=10$
Single Criterion										
CH Index	0.8744	37∗	2	6	0	0	0	4	1	0
ASW Index	0.9073	40∗	1	6	0	1	1	1	0	0
Dunn Index	1.0000	50∗	0	0	0	0	0	0	0	0
Pearson gamma (PG)	0.6262	0∗	3	6	6	4	12	4	10	5
Prediction strength (PS)	1.0000	50∗	0	0	0	0	0	0	0	0
N select boot (NSB)	1.0000	50∗	0	0	0	0	0	0	0	0
CVNN ($\kappa = 5$)	0.8860	32∗	7	10	0	0	0	1	0	0
CVNN ($\kappa = 10$)	0.8935	32∗	8	10	0	0	0	0	0	0
CVNN ($\kappa = 20$)	0.9058	35∗	7	8	0	0	0	0	0	0
Z-score aggregation										
Ave.wit + Ave.bet	0.6552	3∗	7	4	5	3	9	4	11	4
Ave.wit + Ave.bet + PG	0.6343	0∗	7	4	5	4	11	5	10	4
Ave.wit + Ave.bet + PS	1.0000	50∗	0	0	0	0	0	0	0	0
Ave.wit + Ave.bet + NSB	0.9405	41∗	6	0	1	0	0	1	1	0
Ave.wit + Ave.bet + PG + PS	0.9838	46∗	4	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + NSB	0.7363	16∗	6	4	5	2	6	4	5	2
Wid.gap + Sep.Ind	0.9999	49∗	1	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG	0.9616	40∗	8	1	0	0	0	1	0	0
Wid.gap + Sep.Ind + PS	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	0.9999	49∗	1	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.9998	48∗	2	1	0	0	0	0	0	0
Range aggregation										
Ave.wit + Ave.bet	0.6156	1∗	6	4	2	2	9	5	13	8
Ave.wit + Ave.bet + PG	0.6185	0∗	6	5	3	2	10	6	12	6
Ave.wit + Ave.bet + PS	0.9905	48∗	2	0	0	0	0	0	0	0
Ave.wit + Ave.bet + NSB	1.0000	50∗	0	0	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + PS	0.9355	41∗	5	1	1	0	0	1	1	0
Ave.wit + Ave.bet + PG + NSB	0.8156	28∗	6	2	3	0	1	2	5	3
Wid.gap + Sep.Ind	0.9999	49∗	1	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG	0.9998	48∗	2	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PS	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	0.9999	49∗	1	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.9999	49∗	1	0	0	0	0	0	0	0

Ave.wit = Average within dissimilarities, Ave.bet = Average between dissimilarities

Wid.gap = Widest gap within dissimilarities, Sep.Ind = Separation Index

Table 7.21: Two clusters in two dimensions with two moon shapes - Spectral clustering.

Clustering Validation Index	**ARI**	**Estimate of Number of Clusters**, $\hat{k}$								
		$\hat{k}=2$	$\hat{k}=3$	$\hat{k}=4$	$\hat{k}=5$	$\hat{k}=6$	$\hat{k}=7$	$\hat{k}=8$	$\hat{k}=9$	$\hat{k}=10$
Single Criterion										
CH Index	0.2955	$0*$	0	3	3	8	5	7	11	13
ASW Index	0.3487	$0*$	0	6	10	12	6	6	7	3
Dunn Index	1.0000	$50*$	0	0	0	0	0	0	0	0
Pearson gamma (PG)	0.4518	$0*$	0	14	21	10	2	3	0	0
Prediction strength (PS)	1.0000	$50*$	0	0	0	0	0	0	0	0
N select boot (NSB)	0.2450	$0*$	0	0	0	0	0	0	6	44
CVNN ($\kappa=5$)	0.3328	$0*$	0	3	8	12	8	6	9	4
CVNN ($\kappa=10$)	0.3377	$0*$	0	5	9	11	7	5	8	5
CVNN ($\kappa=20$)	0.3727	$0*$	2	6	11	12	6	5	5	3
Z-score aggregation										
Ave.wit + Ave.bet	0.4046	$0*$	4	3	13	6	9	4	5	6
Ave.wit + Ave.bet + PG	0.4382	$0*$	3	7	19	9	6	5	1	0
Ave.wit + Ave.bet + PS	0.9034	$41*$	0	3	5	1	0	0	0	0
Ave.wit + Ave.bet + NSB	0.2964	$0*$	0	2	2	4	6	8	14	14
Ave.wit + Ave.bet + PG + PS	0.5290	$3*$	5	14	20	6	0	2	0	0
Ave.wit + Ave.bet + PG + NSB	0.3730	$0*$	0	2	13	13	9	7	5	1
Wid.gap + Sep.Ind	1.0000	$50*$	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG	0.9838	$46*$	4	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PS	1.0000	$50*$	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	1.0000	$50*$	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	1.0000	$50*$	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.9838	$48*$	1	0	0	1	0	0	0	0
Range aggregation										
Ave.wit + Ave.bet	0.3185	$0*$	1	2	6	6	7	6	11	11
Ave.wit + Ave.bet + PG	0.3798	$0*$	1	2	15	11	8	6	6	1
Ave.wit + Ave.bet + PS	0.4964	$4*$	4	10	16	10	3	3	0	0
Ave.wit + Ave.bet + NSB	0.7062	$25*$	0	4	11	7	1	2	0	0
Ave.wit + Ave.bet + PG + PS	0.2684	$0*$	0	0	2	1	4	7	16	20
Ave.wit + Ave.bet + PG + NSB	0.3198	$0*$	0	2	5	8	7	7	10	11
Wid.gap + Sep.Ind	1.0000	$50*$	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG	1.0000	$50*$	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PS	1.0000	$50*$	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	1.0000	$50*$	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	1.0000	$50*$	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	1.0000	$50*$	0	0	0	0	0	0	0	0

Ave.wit = Average within dissimilarities, Ave.bet = Average between dissimilarities

Wid.gap = Widest gap within dissimilarities, Sep.Ind = Separation Index

Table 7.22: Two clusters in two dimensions with untypical parabolic shapes - Single linkage.

Clustering Validation Index	**ARI**	**Estimate of Number of Clusters**, $\hat{k}$								
		$\hat{k}=2$	$\hat{k}=3$	$\hat{k}=4$	$\hat{k}=5$	$\hat{k}=6$	$\hat{k}=7$	$\hat{k}=8$	$\hat{k}=9$	$\hat{k}=10$
Single Criterion										
CH Index	0.9931	49∗	0	0	0	0	0	1	0	0
ASW Index	0.9931	49∗	0	0	0	0	0	1	0	0
Dunn Index	0.9947	49∗	0	0	0	0	1	0	8	0
Pearson gamma (PG)	0.7350	0∗	0	1	5	5	9	6	0	16
Prediction strength (PS)	0.9800	50∗	0	0	0	0	0	0	0	2
N select boot (NSB)	0.9618	28∗	16	2	1	1	0	0	0	0
CVNN ($\kappa=5$)	0.9240	24∗	15	5	5	0	0	1	0	0
CVNN ($\kappa=10$)	0.9590	40∗	4	3	3	0	0	0	0	0
CVNN ($\kappa=20$)	0.9744	45∗	3	0	1	0	0	1	0	0
Z-score aggregation										
Ave.wit + Ave.bet	0.7495	1∗	2	5	4	3	5	7	8	15
Ave.wit + Ave.bet + PG	0.7453	0∗	1	4	3	5	7	7	8	15
Ave.wit + Ave.bet + PS	0.9989	49∗	0	1	0	0	0	0	0	0
Ave.wit + Ave.bet + NSB	0.9294	23∗	13	4	0	3	3	1	0	3
Ave.wit + Ave.bet + PG + PS	0.9976	46∗	3	1	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + NSB	0.7974	10∗	1	3	3	6	4	6	3	14
Wid.gap + Sep.Ind	0.9726	36∗	10	3	1	0	0	0	0	0
Wid.gap + Sep.Ind + PG	0.9552	24∗	14	9	2	1	0	0	0	0
Wid.gap + Sep.Ind + PS	0.9800	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	0.9684	34∗	15	0	1	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	0.9794	48∗	2	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.9561	29∗	15	2	3	1	0	0	0	0
Range aggregation										
Ave.wit + Ave.bet	0.7495	1∗	2	5	4	3	5	7	8	15
Ave.wit + Ave.bet + PG	0.7428	0∗	1	4	3	5	6	8	7	16
Ave.wit + Ave.bet + PS	0.9989	49∗	0	1	0	0	0	0	0	0
Ave.wit + Ave.bet + NSB	0.9989	49∗	0	1	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + PS	0.9227	23∗	12	4	0	3	3	2	0	3
Ave.wit + Ave.bet + PG + NSB	0.8245	14∗	3	3	0	7	4	6	1	12
Wid.gap + Sep.Ind	0.9738	35∗	12	2	1	0	0	0	0	0
Wid.gap + Sep.Ind + PG	0.9594	22∗	9	11	4	2	1	1	0	0
Wid.gap + Sep.Ind + PS	0.9787	47∗	3	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	0.9800	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	0.9666	33∗	14	1	2	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.9686	27∗	15	3	3	2	0	0	0	0

Ave.wit = Average within dissimilarities, Ave.bet = Average between dissimilarities

Wid.gap = Widest gap within dissimilarities, Sep.Ind = Separation Index

Table 7.23: Two clusters in two dimensions with untypical parabolic shapes - Spectral clustering.

Clustering Validation Index	**ARI**	**Estimate of Number of Clusters**, $\hat{k}$								
		$\hat{k}=2$	$\hat{k}=3$	$\hat{k}=4$	$\hat{k}=5$	$\hat{k}=6$	$\hat{k}=7$	$\hat{k}=8$	$\hat{k}=9$	$\hat{k}=10$
Single Criterion										
CH Index	0.3787	3∗	0	1	4	7	7	6	11	11
ASW Index	0.5571	14∗	0	1	8	7	3	5	8	4
Dunn Index	0.9951	49∗	0	1	0	0	0	0	0	0
Pearson gamma (PG)	0.5785	0∗	3	8	17	13	6	2	1	0
Prediction strength (PS)	1.0000	50∗	0	0	0	0	0	0	0	0
N select boot (NSB)	0.3119	2∗	0	0	0	0	0	1	7	40
CVNN ($\kappa = 5$)	0.4157	0∗	0	8	7	7	9	9	8	2
CVNN ($\kappa = 10$)	0.3750	2∗	0	4	3	5	11	9	8	8
CVNN ($\kappa = 20$)	0.3983	3∗	2	6	8	4	7	7	7	8
Z-score aggregation										
Ave.wit + Ave.bet	0.5329	0∗	9	7	8	6	4	7	5	4
Ave.wit + Ave.bet + PG	0.5752	0∗	4	8	14	12	7	4	1	0
Ave.wit + Ave.bet + PS	0.9774	46∗	2	2	0	0	0	0	0	0
Ave.wit + Ave.bet + NSB	0.3989	0∗	0	4	3	3	8	8	14	10
Ave.wit + Ave.bet + PG + PS	0.9178	35∗	8	4	2	1	0	0	0	0
Ave.wit + Ave.bet + PG + NSB	0.5181	0∗	2	6	10	11	8	6	5	2
Wid.gap + Sep.Ind	0.9995	49∗	1	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG	0.9843	40∗	8	1	1	0	0	0	0	0
Wid.gap + Sep.Ind + PS	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	0.9783	42∗	6	0	2	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.9597	34∗	11	3	2	0	0	0	0	0
Range aggregation										
Ave.wit + Ave.bet	0.4714	0∗	3	7	7	6	6	9	8	4
Ave.wit + Ave.bet + PG	0.5782	0∗	4	9	14	10	7	4	2	0
Ave.wit + Ave.bet + PS	0.9267	38∗	5	4	2	1	0	0	0	0
Ave.wit + Ave.bet + NSB	0.9774	46∗	2	2	0	0	0	0	0	0
Ave.wit + Ave.bet + PG + PS	0.3787	0∗	0	2	3	3	7	8	14	13
Ave.wit + Ave.bet + PG + NSB	0.4922	0∗	1	5	10	10	8	6	6	4
Wid.gap + Sep.Ind	0.9979	47∗	3	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG	0.9719	37∗	9	3	1	0	0	0	0	0
Wid.gap + Sep.Ind + PS	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + NSB	1.0000	50∗	0	0	0	0	0	0	0	0
Wid.gap + Sep.Ind + PG + PS	0.9645	36∗	10	2	2	0	0	0	0	0
Wid.gap + Sep.Ind + PG + NSB	0.9470	32∗	10	4	4	0	0	0	0	0

Ave.wit = Average within dissimilarities, Ave.bet = Average between dissimilarities

Wid.gap = Widest gap within dissimilarities, Sep.Ind = Separation Index

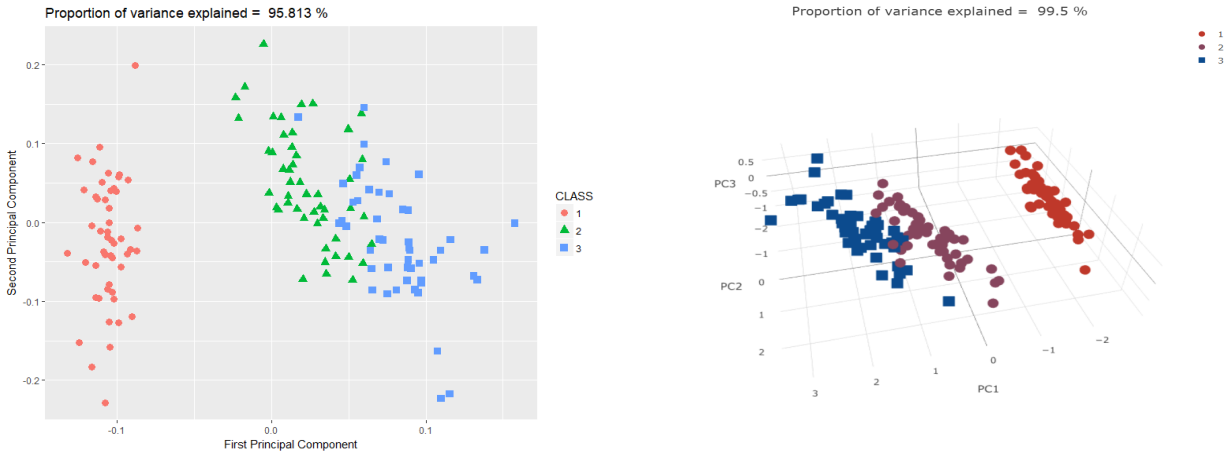
7.2 Examination of Aggregating Clustering Validation Indexes on Real Data Sets

In the previous section, different simulation scenarios that are generated from various types of distributional settings are examined, whereas here some real data set examples that are not based on any type of statistical distribution will be considered. In this respect, three popular data sets obtained from the University of California Irvine Machine Learning Repository (Dheeru and Karra Taniskidou, 2017) will be analysed for the sake of estimating the number of clusters by using different clustering validation index considerations. The data sets are visualised in two and three dimensional graphical representation by using Principal Component Analysis. For each data set, the adjusted Rand index values are calculated between the given classes and the clusters from different clustering algorithms. The given number of classes for these data sets does not precisely imply the correct number of clusters because the cluster analysis is performed without knowing the true classes. However, these given labels will give us a guidance to externally validate the clustering results. Many different clustering validation index considerations are presented on the line charts from which users are able to determine for themselves which clustering algorithms and what number of clusters are the most appropriate. In addition to all this, the adjusted Rand index values between the given true class labels and three best clusterings from different clustering algorithms based on the clustering validation index results are computed in order to compare which clustering validation indexes most adequately estimate the given true class labels. Then, for each clustering validation index the average of three adjusted Rand index values are calculated to show a more balanced representation in the event one of these clusterings is much better than the other two. All these results are presented in Section 7.2.4.

7.2.1 Iris data set

This is one of the best known data sets to be found in the cluster analysis literature. It was first introduced in Fisher's (1936) paper. The data set contains 4 continuous variables and 1 categorical variable with 3 classes containing 50 members each, where each class refers to a type of iris plant. Two and three dimensional representations of the data set with the given classes are shown in Figure 7.8, which indicate that one class is separable from the other two; while the latter two are not separable from each other.

Figure 7.14 displays several single criterion index results, whereas Figure 7.15 and 7.16 give the solution of different aggregation of clustering validation indexes by applying Z-score and range



(a) Two dimensional representation

(b) Two dimensional projections of three dimensions

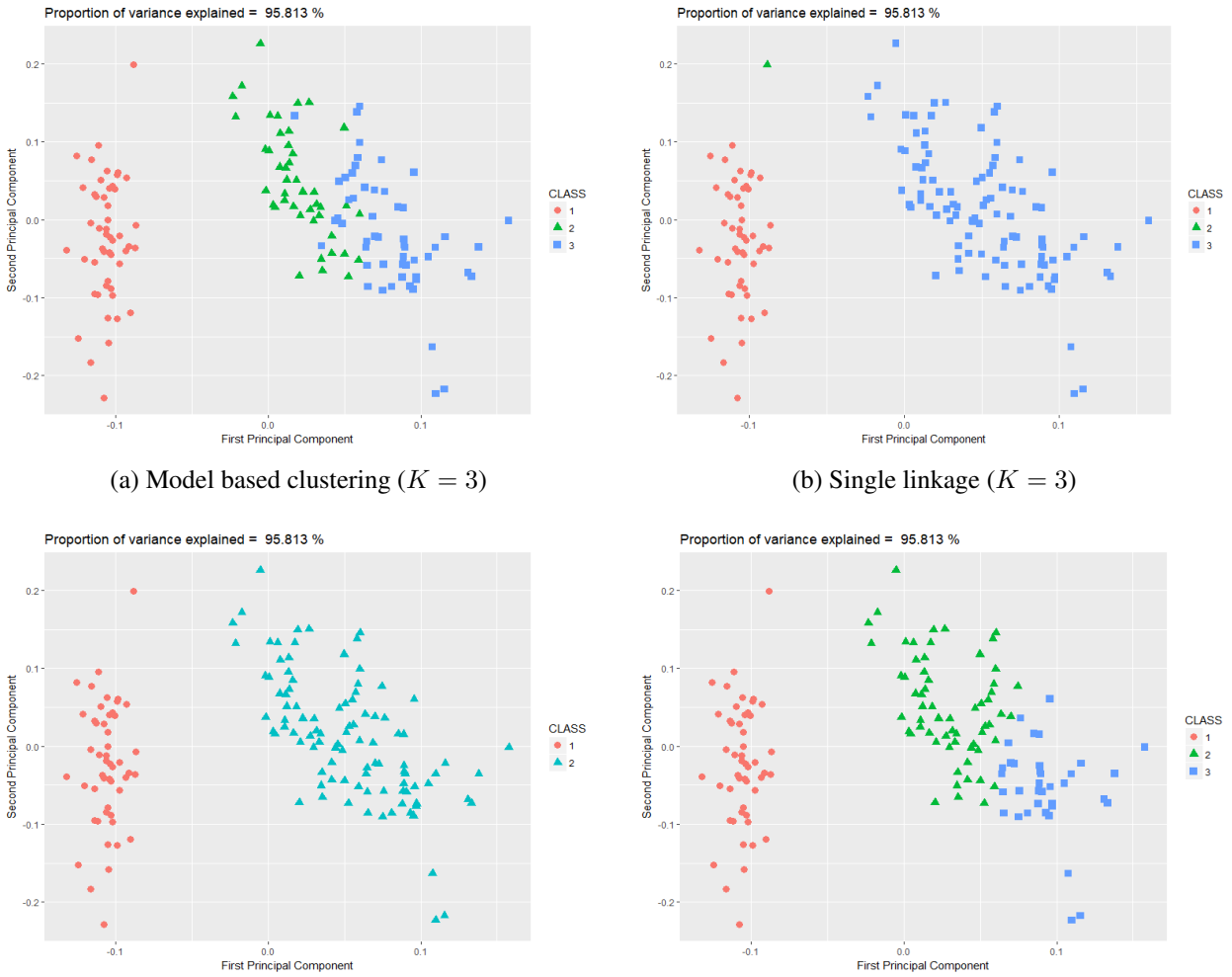
Figure 7.8: Two dimensional representation and two dimensional projections of three dimensions (PCA) of IRIS data set with true class labels

standardisation. CH Index and CVNN ($\kappa = 10$) Index results indicate that Ward's method and model based clustering for $K = 3$ are the best choices. These two clustering choices have the two highest adjusted Rand index values according to Table 7.24. ASW Index, Dunn Index and prediction strength results favour solutions with a small number of cluster in many instances. The other clustering validation indexes indicate that single linkage for $K = 3$ is the best choice.

Figure 7.9 displays some of the clustering solutions according to the cluster validation index results. Ward's method and model based clustering for $K = 3$ are very similar to true class solution as shown in Figure 7.8, however the data set looks more like two clusters rather than a solution with three clusters. Two solution for $K = 2$ is the best choice in many cases, especially based on the aggregated clustering validation results, see Figure 7.15 and 7.16. According to Table 7.24, most of the clustering solutions for $K = 2$ give the same result, where only Ward's method is shown. The aggregated index results favour either a two clustering solution or single linkage for $K = 3$. Single linkage for $K = 3$ is actually very similar to the two clustering solution, but the difference is that one outlying point is clustered as a third cluster.

For the comparison of clustering validation indexes, Table 7.27 presents different adjusted Rand index values which are computed between the three best clustering solutions and the given true classes. The results indicate that CH Index and CVNN Index are able to estimate the true class labels better than the other indexes. The adjusted Rand index values for aggregated indexes most likely give the same results because they favour the clustering solution for $K = 2$, and the different clustering algorithm for $K = 2$ lead to the same cluster label solutions in most cases. However, as explained previously, the data points are distributed in such a way that two clustering solution

appears to be more appropriate based on the graphical representation of the data set.



(a) Model based clustering ($K = 3$)

(b) Single linkage ($K = 3$)

(c) Ward's method ($K = 2$)

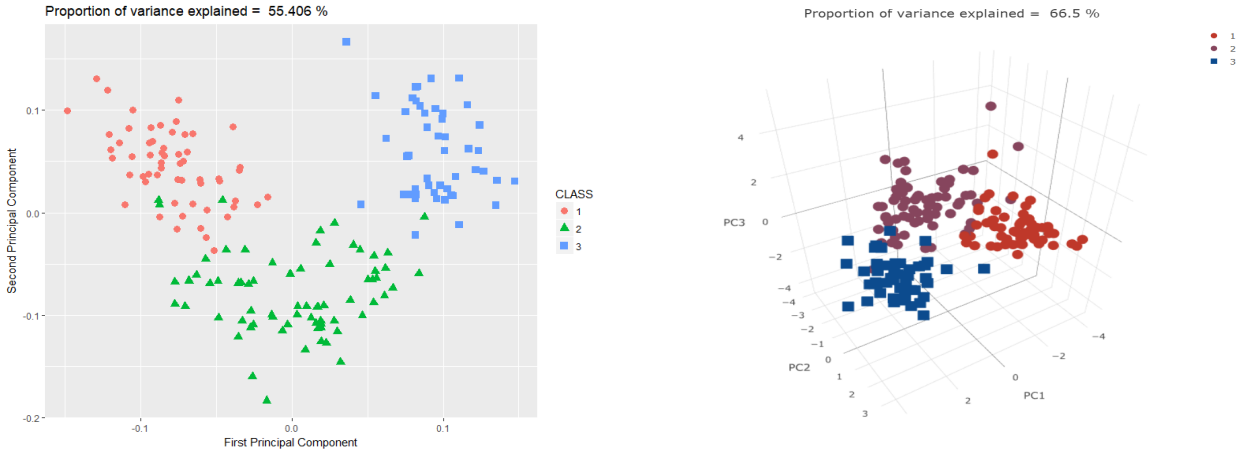
(d) Ward's method ($K = 3$)

Figure 7.9: Two dimensional representation (PCA) of IRIS data set for different clustering scenarios

7.2.2 Wine data set

This data set is based on the results of a chemical analysis of wines grown in the same region in Italy, but derived from three different types of wines. The wine data set was first investigated in Forina et al. (1988). The data set contains 13 continuous variables and 1 categorical variable with 3 classes, where the number of objects in the classes are 48, 59 and 71. Figure 7.10 presents an overview of two and three dimensional visualisations of this data set, where the colours of the points represent the given classes. According to the figures, the objects are very homogeneous, so that it might be difficult to detect distinct cluster shapes by looking at the PCA plots; correspond-

ingly, the proportion of variance explained is low at around 66%, indicating that it is challenging to observe such clusters. For this type of homogeneous data set, one could consider the given class labels for validating the results of the clustering algorithms. Table 7.25 provides the adjusted Rand index results for various clustering algorithms with different number of clusters.



(a) Two dimensional representation

(b) Two dimensional projections of three dimensions

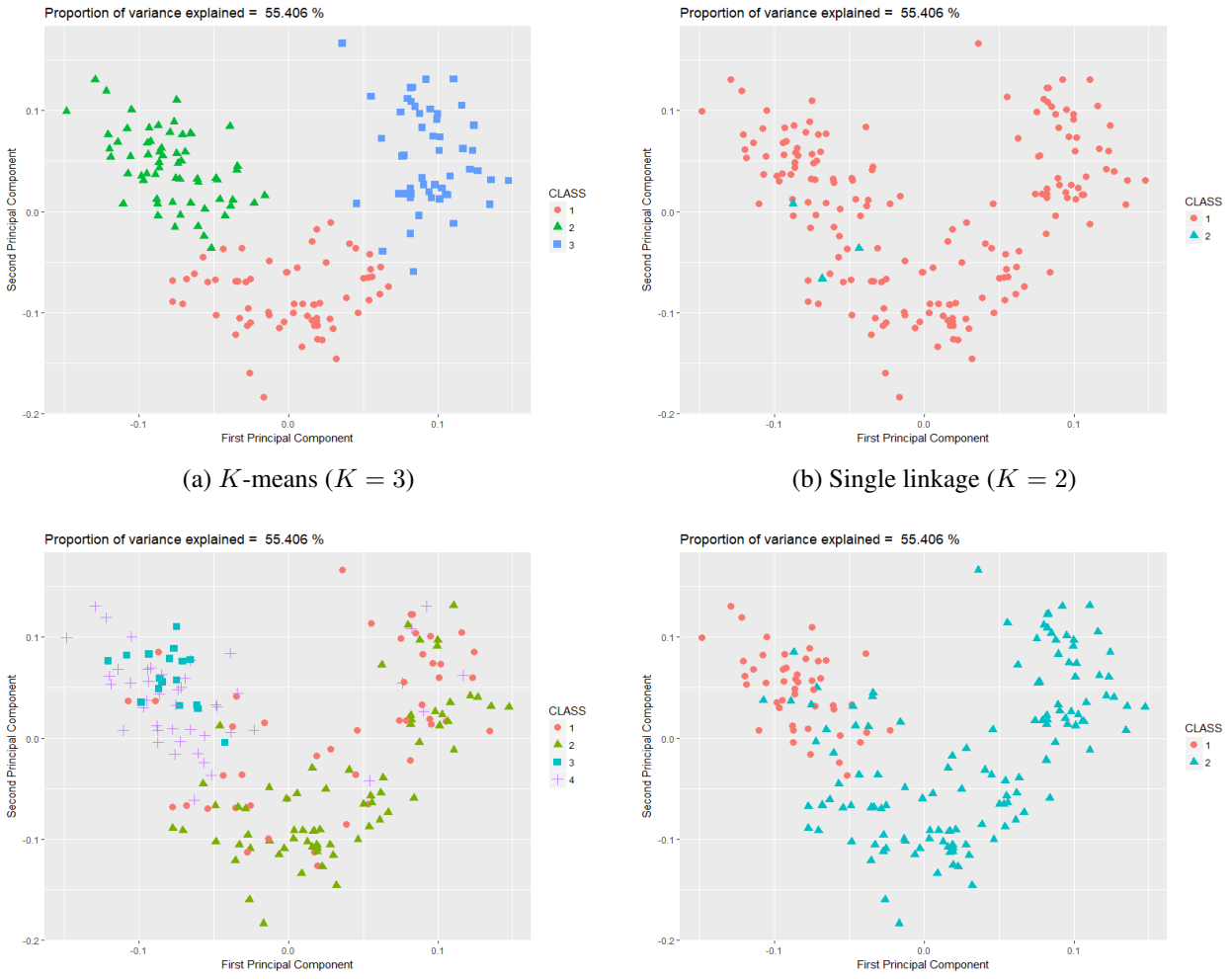
Figure 7.10: Two dimensional representation and two dimensional projections of three dimensions (PCA) of WINE data set with true class labels

Figure 7.17 shows different single criteria results. ASW Index, Dunn Index and PG Index results indicate that Ward's method for $K = 2$ is the best choice of clustering algorithm scenario. Interestingly, for this specific data set CH Index results favour the largest possible number of clusters when using Ward's method, but one could consider spectral clustering algorithm for either $K = 7$ or $K = 9$ due to the presence of two considerable spikes. CVNN Index results also point out that spectral clustering for either $K = 4, 5$ or 6 could be the best selection out of the clustering choices. Two stability methodologies for estimating the number of clusters favour the smallest possible number of clusters for single linkage, but if single linkage is disregarded, prediction strength results show that 3-means is another good choice, having the highest adjusted Rand index values based on the given classes.

The different calibration scenarios are shown in Figure 7.18 and 7.19, and the scenarios aggregated with the bootstrap method are quite successful at predicting the correct number of clusters when using K-means clustering with $K = 3$. The scenarios aggregated with the prediction strength technique favour the smallest possible number of clusters, but the 3-means can be considered another possible optimal solution due to the presence of a peak for $K = 3$. Two dimensional PCA plots are presented in Figure 7.11 with the clusterings that give good solutions based on clustering validation index results. 3-means appears to be a satisfactory selection, because it better partitions

the cluster points than the other clustering solutions when considering the PCA plots. This is also the best choice based on the adjusted Rand index values in Table 7.25.

Table 7.28 demonstrates different adjusted Rand index values when comparing various clustering validation indexes. The results indicate that most of the single criteria are not very successful at estimating the true class labels compared to the aggregated clustering validation indexes. As previously pointed out, the bootstrap method, average within and average between dissimilarities combinations of aggregated indexes are better able to estimate the true class labels than the other aggregated index combinations.



(a) K-means ($K = 3$)

(b) Single linkage ($K = 2$)

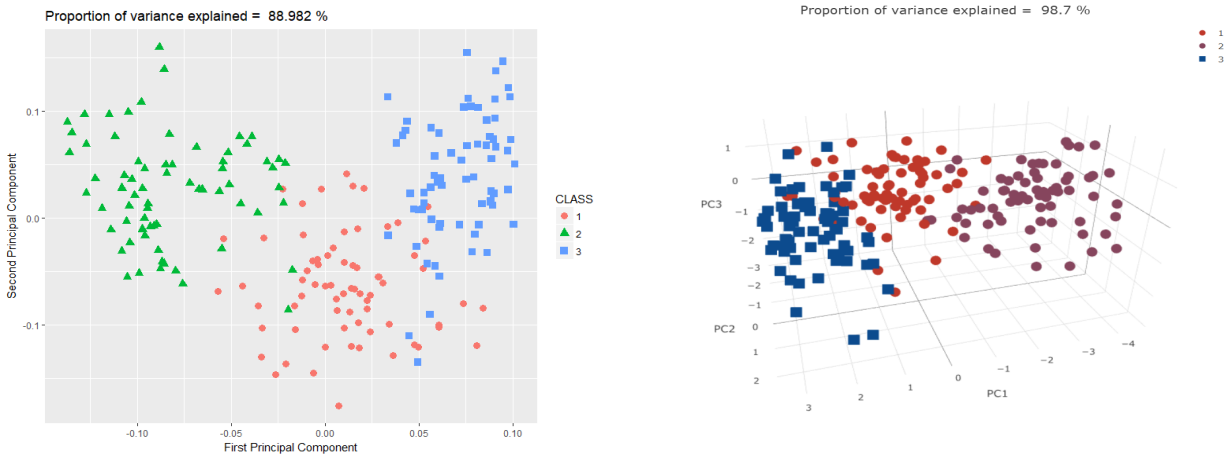
(c) Spectral clustering ($K = 4$)

(d) Ward's method ($K = 2$)

Figure 7.11: Two dimensional representation (PCA) of WINE data set for different clustering scenarios

7.2.3 Seed data set

The seed data set, which was first used in Charytanowicz et al. (2010), is based on measurements of geometrical properties of kernels belonging to three different varieties of wheat. It contains 7 geometric features of wheat kernels (continuous variables) and 1 categorical variable with 3 classes each containing 50 objects, where each class refers to a type of wheat. Two and three dimensional representations of the data set are visualised in Figure 7.12 by using Principal Component Analysis. The three different colours of points represent class labels given by three categories in the data set. The two scatter plots indicate that the points are homogeneously distributed so that it is not easy to distinguish such cluster patterns, however the given classes can still guide us on how to identify such clusters. Table 7.26 is the summary of the adjusted Rand index results, which are calculated with the given classes against cluster labels obtained from different clustering algorithms with different numbers of clusters. The table results indicate that PAM, K-means, Ward's method and model-based clustering for $K = 3$ are the scenarios with the highest adjusted Rand index values.



(a) Two dimensional representation

(b) Two dimensional projections of three dimensions

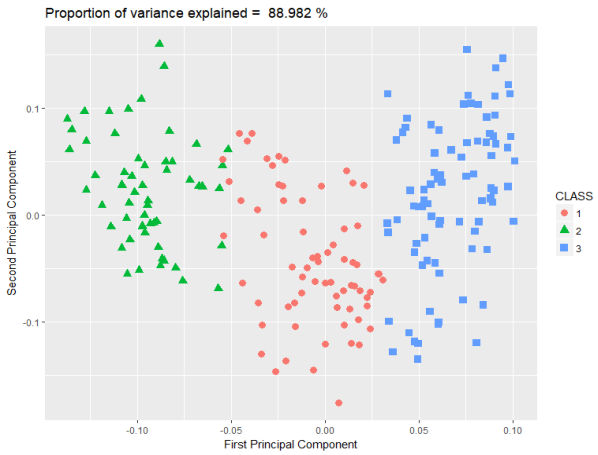
Figure 7.12: Two dimensional representation and two dimensional projections of three dimensions (PCA) of SEED data set with true class labels

Different single criteria results are shown in Figure 7.20. The CH Index and PG Index values suggest that PAM, K-means, Ward's method and model based clustering for $K = 3$ are the best choices, and are reasonable choices based on the adjusted Rand index values. CVNN Index values for different κ selections give similar solutions, but favour model-based clustering for $K = 3$ to the other three clustering solutions. The ASW Index results again have a tendency to predict the number of clusters as the smallest possible one ($K = 2$) for different clustering algorithms. The two stability methodologies again favour single linkage for $K = 2$, but the bootstrap method also

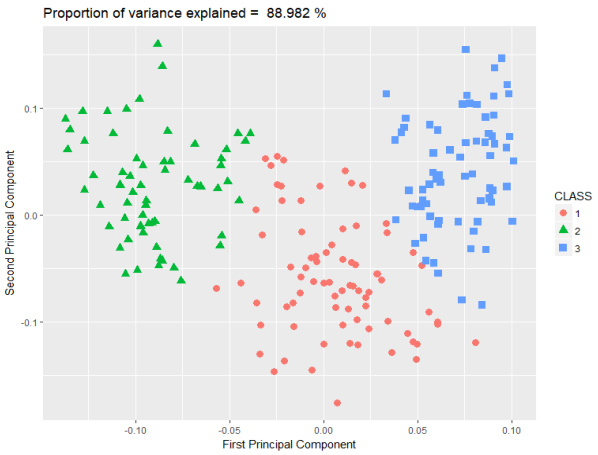
indicates that PAM for $K = 3$ is another optimal solution.

Based on the line charts in Figure 7.21 and 7.22, the results for any calibration with average within and average between dissimilarities indicate that PAM and model based clustering for $K = 3$ are the two optimal solutions in many instances. Because the points on the data set are more likely to be homogeneously distributed, the aggregated indexes for the combination of the widest gap and the separation index do not lead to the clustering solution for $K = 3$ in most cases, and those aggregated indexes usually favour a single linkage solution. Figure 7.13 provides various clustering solutions that are obtained from clustering validation index results. PAM for $K = 3$, which has the highest adjusted Rand index value in Table 7.25, appears to be a reasonable solution according to the shape of cluster points in Figure 7.13 as well as the results of aggregated clustering validation aspects.

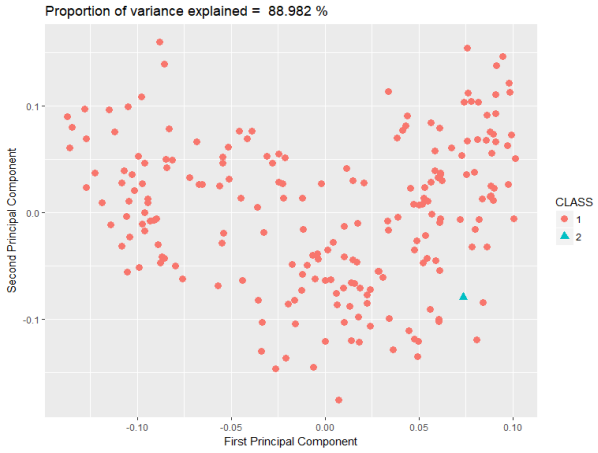
The adjusted Rand index values between the three best clusterings and the given true cluster labels for different clustering validation indexes are shown in Table 7.29 for the sake of comparison. The CH Index and CVNN Index estimate the true class labels better than the other single criteria. On the other hand, any combinations of the bootstrap method, average within and average between dissimilarities give the best ARI values based on the adjusted Rand index values between the first best clustering selection and the given class labels.



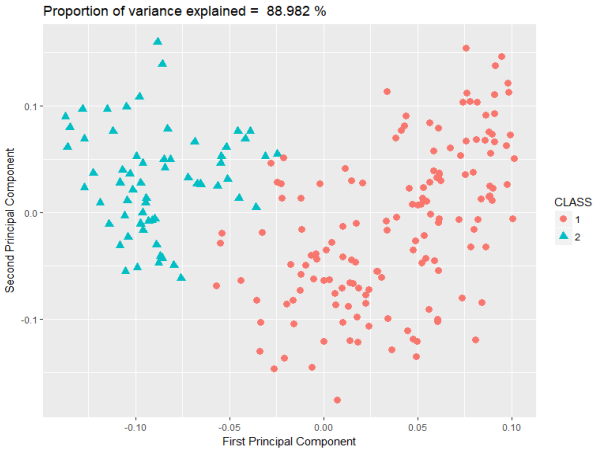
(a) Model based clustering ($K = 3$)



(b) PAM ($K = 3$)



(c) Single linkage ($K = 2$)



(d) Ward's method ($K = 2$)

Figure 7.13: Two dimensional representation (PCA) of SEED data set for different clustering scenarios

7.2.4 Detailed results of real data sets

Adjusted Rand Index results

The following tables provide the adjusted Rand index values of the class labels obtained from different real data sets for various clustering algorithms with different numbers of clusters. "*" indicates column corresponding to the given number of classes.

Table 7.24: IRIS data set

Clustering Algorithm	**Estimate of Number of Clusters, $\hat{k}$**								
	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$
PAM	0.568	0.641*	0.586	0.468	0.396	0.408	0.362	0.323	0.315
K-means	0.568	0.620*	0.558	0.439	0.424	0.468	0.344	0.259	0.351
Single linkage	0.568	0.558*	0.552	0.551	0.549	0.546	0.538	0.536	0.533
Complete linkage	0.410	0.572*	0.533	0.462	0.450	0.444	0.472	0.502	0.378
Average linkage	0.568	0.562*	0.552	0.550	0.509	0.553	0.510	0.501	0.372
Ward's method	0.568	0.731*	0.660	0.595	0.435	0.448	0.425	0.402	0.342
Model based Clustering	0.568	0.903*	0.841	0.682	0.614	0.525	0.517	0.536	0.513
Spectral clustering	0.568	0.563*	0.815	0.651	0.596	0.584	0.596	0.626	0.599

Table 7.25: WINE data set

Clustering Algorithm	**Estimate of Number of Clusters, $\hat{k}$**								
	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$
PAM	0.357	0.741*	0.662	0.538	0.439	0.395	0.355	0.347	0.325
K-means	0.374	0.897*	0.674	0.704	0.501	0.475	0.518	0.471	0.387
Single linkage	−0.005	−0.006*	−0.008	−0.009	−0.011	−0.012	−0.079	−0.008	−0.009
Complete linkage	0.295	0.577*	0.642	0.677	0.663	0.687	0.687	0.642	0.553
Average linkage	−0.001	−0.005*	−0.010	0.431	0.424	0.411	0.792	0.792	0.783
Ward's method	0.326	0.368*	0.281	0.215	0.229	0.213	0.216	0.185	0.171
Model based clustering	0.582	0.880*	0.762	0.797	0.727	0.502	0.408	0.407	0.373
Spectral clustering	0.369	0.311*	0.313	0.239	0.248	0.216	0.288	0.178	0.231

Table 7.26: SEED data set

Clustering Algorithm	**Estimate of Number of Clusters, $\hat{k}$**								
	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$
PAM	0.444	0.746*	0.636	0.524	0.424	0.380	0.340	0.311	0.287
K-means	0.480	0.773*	0.667	0.552	0.514	0.459	0.404	0.415	0.336
Single linkage	0.000	0.000*	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Complete linkage	0.488	0.686*	0.654	0.582	0.450	0.431	0.364	0.368	0.377
Average linkage	0.491	0.685*	0.681	0.674	0.587	0.588	0.555	0.498	0.493
Ward's method	0.452	0.713*	0.727	0.637	0.577	0.520	0.440	0.411	0.326
Model based clustering	0.459	0.629*	0.610	0.507	0.457	0.413	0.372	0.328	0.391
Spectral clustering	0.501	0.383*	0.640	0.432	0.590	0.415	0.503	0.375	0.398

Clustering validation index results

Table 7.24, 7.25 and 7.26 present the ARI values between three best clusterings from different clustering algorithms and the given true class labels based on various clustering validation index results. Then, the average of the ARI values from those three best clusterings is computed and shown in the next column in order to give a more balanced view in the event that one clustering is far better than the other two for one of the clustering validation indexes. The last three columns present those three clustering algorithms with their number of clusters. The aim of these tables is to show which clustering validation indexes are the most capable of estimating the given true class labels.

The meaning of the abbreviations to be used in the figures below are given as follows: CH: Calinski and Harabasz, ASW: Average silhouette width, PG: Pearson gamma, CVNN: Clustering validation on nearest neighbour, AW: Average within dissimilarities, AB: Average between dissimilarities, NSB: The bootstrap method, WG: Widest gap, SI: Separation Index.

Table 7.27: IRIS data set

Clustering Validation Index	ARI values			Average	Best clusterings in order		
	First	Second	Third	of ARI	First	Second	Third
Single Criterion							
CH Index	0.731	0.641	0.620	0.620	Ward (K=3)	PAM (K=3)	K-means (K=3)
ASW Index	0.568	0.568	0.568	0.568	PAM (K=2)	K-means (K=2)	Single (K=2)
Dunn Index	0.568	0.568	0.568	0.568	PAM (K=2)	K-means (K=2)	Single (K=2)
Pearson gamma (PG)	0.558	0.563	0.568	0.563	Single (K=3)	Spectral (K=3)	PAM (K=2)
Prediction strength (PS)	0.568	0.568	0.568	0.568	K-means (K=2)	Ward (K=2)	PAM (K=2)
N select boot (NSB)	0.568	0.568	0.558	0.564	K-means (K=2)	PAM (K=2)	Single (K=3)
CVNN ($\kappa = 5$)	0.904	0.620	0.641	0.721	Mclust (K=3)	K-means (K=3)	PAM (K=3)
CVNN ($\kappa = 10$)	0.904	0.641	0.620	0.721	Mclust (K=3)	PAM (K=3)	K-means (K=3)
CVNN ($\kappa = 20$)	0.904	0.641	0.620	0.721	Mclust (K=3)	PAM (K=3)	K-means (K=3)
Z-score aggregation							
Ave.wit + Ave.bet	0.568	0.568	0.568	0.568	PAM (K=2)	K-means (K=2)	Single (K=2)
Ave.wit + Ave.bet + PG	0.568	0.568	0.568	0.568	PAM (K=2)	K-means (K=2)	Single (K=2)
Ave.wit + Ave.bet + PS	0.568	0.568	0.568	0.568	K-means (K=2)	Ward (K=2)	PAM (K=2)
Ave.wit + Ave.bet + NSB	0.568	0.568	0.568	0.566	K-means (K=2)	PAM (K=2)	Average (K=2)
Ave.wit + Ave.bet + PG + PS	0.568	0.568	0.568	0.568	K-means (K=2)	Ward (K=2)	PAM (K=2)
Ave.wit + Ave.bet + PG + NSB	0.568	0.568	0.568	0.566	K-means (K=2)	PAM (K=2)	Average (K=2)
Wid.gap + Sep.Ind	0.568	0.568	0.568	0.568	PAM (K=2)	K-means (K=2)	Single (K=2)
Wid.gap + Sep.Ind + PG	0.568	0.568	0.568	0.568	PAM (K=2)	K-means (K=2)	Single (K=2)
Wid.gap + Sep.Ind + PS	0.568	0.568	0.568	0.568	K-means (K=2)	Ward (K=2)	PAM (K=2)
Wid.gap + Sep.Ind + NSB	0.568	0.568	0.568	0.566	K-means (K=2)	PAM (K=2)	Average (K=2)
Wid.gap + Sep.Ind + PG + PS	0.568	0.568	0.568	0.568	K-means (K=2)	Ward (K=2)	PAM (K=2)
Wid.gap + Sep.Ind + PG + NSB	0.568	0.568	0.568	0.566	K-means (K=2)	PAM (K=2)	Average (K=2)
Range aggregation							
Ave.wit + Ave.bet	0.568	0.568	0.568	0.568	PAM (K=2)	K-means (K=2)	Single (K=2)
Ave.wit + Ave.bet + PG	0.568	0.568	0.568	0.568	PAM (K=2)	K-means (K=2)	Single (K=2)
Ave.wit + Ave.bet + PS	0.568	0.568	0.568	0.568	K-means (K=2)	Ward (K=2)	PAM (K=2)
Ave.wit + Ave.bet + NSB	0.568	0.568	0.568	0.566	K-means (K=2)	PAM (K=2)	Average (K=2)
Ave.wit + Ave.bet + PG + PS	0.568	0.568	0.568	0.568	K-means (K=2)	Ward (K=2)	PAM (K=2)
Ave.wit + Ave.bet + PG + NSB	0.568	0.568	0.568	0.566	K-means (K=2)	PAM (K=2)	Average (K=2)
Wid.gap + Sep.Ind	0.568	0.568	0.568	0.568	PAM (K=2)	K-means (K=2)	Single (K=2)
Wid.gap + Sep.Ind + PG	0.568	0.568	0.568	0.568	PAM (K=2)	K-means (K=2)	Single (K=2)
Wid.gap + Sep.Ind + PS	0.568	0.568	0.568	0.568	K-means (K=2)	Ward (K=2)	PAM (K=2)
Wid.gap + Sep.Ind + NSB	0.568	0.568	0.568	0.566	K-means (K=2)	PAM (K=2)	Average (K=2)
Wid.gap + Sep.Ind + PG + PS	0.568	0.568	0.568	0.568	K-means (K=2)	Ward (K=2)	PAM (K=2)
Wid.gap + Sep.Ind + PG + NSB	0.568	0.568	0.568	0.566	K-means (K=2)	PAM (K=2)	Average (K=2)

Ave.wit = Average within dissimilarities, Ave.bet = Average between dissimilarities

Wid.gap = Widest gap within dissimilarities, Sep.Ind = Separation Index, Mclust = Model based clustering

Table 7.28: WINE data set

Clustering Validation Index	ARI values			Average	Best clusterings in order		
	First	Second	Third	of ARI	First	Second	Third
Single Criterion							
CH Index	0.171	0.216	0.582	0.323	Ward (K=10)	Spectral (K=7)	Mclust (K=2)
ASW Index	0.327	0.369	0.582	0.426	Ward (K=2)	Spectral (K=2)	Mclust (K=2)
Dunn Index	0.327	0.179	-0.005	0.167	Ward (K=2)	Spectral (K=9)	Single (K=2)
Pearson gamma (PG)	0.327	0.369	0.582	0.426	Ward (K=2)	Spectral (K=2)	Mclust (K=2)
Prediction strength (PS)	-0.007	0.897	-0.002	0.296	Single (K=3)	K-means (K=3)	Average (K=2)
N select boot (NSB)	-0.005	0.358	0.374	0.242	Single (K=2)	PAM (K=2)	K-means (K=2)
CVNN ($\kappa = 5$)	0.216	0.582	0.171	0.323	Spectral (K=7)	Mclust (K=2)	Ward (K=10)
CVNN ($\kappa = 10$)	0.314	0.582	0.171	0.355	Spectral (K=4)	Mclust (K=2)	Ward (K=10)
CVNN ($\kappa = 20$)	0.314	0.582	0.171	0.355	Spectral (K=4)	Mclust (K=2)	Ward (K=10)
Z-score aggregation							
Ave.wit + Ave.bet	0.327	0.369	0.582	0.426	Ward (K=2)	Spectral (K=2)	Mclust (K=2)
Ave.wit + Ave.bet + PG	0.327	0.369	0.582	0.426	Ward (K=2)	Spectral (K=2)	Mclust (K=2)
Ave.wit + Ave.bet + PS	0.327	0.369	0.582	0.426	Ward (K=2)	Spectral (K=2)	Mclust (K=2)
Ave.wit + Ave.bet + NSB	0.897	0.325	0.784	0.668	K-means (K=3)	PAM (K=10)	Average (K=10)
Ave.wit + Ave.bet + PG + PS	0.327	0.369	0.582	0.426	Ward (K=2)	Spectral (K=2)	Mclust (K=2)
Ave.wit + Ave.bet + PG + NSB	0.327	0.369	0.897	0.531	Ward (K=2)	Spectral (K=2)	K-means (K=3)
Wid.gap + Sep.Ind	0.327	0.311	-0.008	0.210	Ward (K=2)	Spectral (K=3)	Single (K=9)
Wid.gap + Sep.Ind + PG	0.327	0.369	0.582	0.426	Ward (K=2)	Spectral (K=2)	Mclust (K=2)
Wid.gap + Sep.Ind + PS	-0.008	0.327	-0.002	0.105	Single (K=4)	Ward (K=2)	Average (K=2)
Wid.gap + Sep.Ind + NSB	-0.008	0.358	0.374	0.241	Single (K=4)	PAM (K=2)	K-means (K=2)
Wid.gap + Sep.Ind + PG + PS	0.327	0.369	0.582	0.426	Ward (K=2)	Spectral (K=2)	Mclust (K=2)
Wid.gap + Sep.Ind + PG + NSB	0.897	0.327	0.311	0.511	K-means (K=3)	Ward (K=2)	Spectral (K=3)
Range aggregation							
Ave.wit + Ave.bet	0.327	0.369	0.582	0.426	Ward (K=2)	Spectral (K=2)	Mclust (K=2)
Ave.wit + Ave.bet + PG	0.327	0.369	0.582	0.426	Ward (K=2)	Spectral (K=2)	Mclust (K=2)
Ave.wit + Ave.bet + PS	0.327	0.369	0.582	0.426	Ward (K=2)	Spectral (K=2)	Mclust (K=2)
Ave.wit + Ave.bet + NSB	0.897	0.325	0.408	0.543	K-means (K=3)	PAM (K=10)	Mclust (K=9)
Ave.wit + Ave.bet + PG + PS	0.327	0.369	0.582	0.426	Ward (K=2)	Spectral (K=2)	Mclust (K=2)
Ave.wit + Ave.bet + PG + NSB	0.897	0.327	0.216	0.480	K-means (K=3)	Ward (K=2)	Spectral (K=7)
Wid.gap + Sep.Ind	0.327	-0.009	0.179	0.165	Ward (K=2)	Single (K=10)	Spectral (K=9)
Wid.gap + Sep.Ind + PG	0.327	0.369	0.582	0.426	Ward (K=2)	Spectral (K=2)	Mclust (K=2)
Wid.gap + Sep.Ind + PS	-0.008	-0.002	0.897	0.295	Single (K=4)	Average (K=2)	K-means (K=3)
Wid.gap + Sep.Ind + NSB	-0.008	0.358	0.374	0.241	Single (K=4)	PAM (K=2)	K-means (K=2)
Wid.gap + Sep.Ind + PG + PS	0.327	0.369	0.582	0.426	Ward (K=2)	Spectral (K=2)	Mclust (K=2)
Wid.gap + Sep.Ind + PG + NSB	0.897	0.784	0.741	0.807	K-means (K=3)	Average (K=10)	PAM (K=3)

Ave.wit = Average within dissimilarities, Ave.bet = Average between dissimilarities

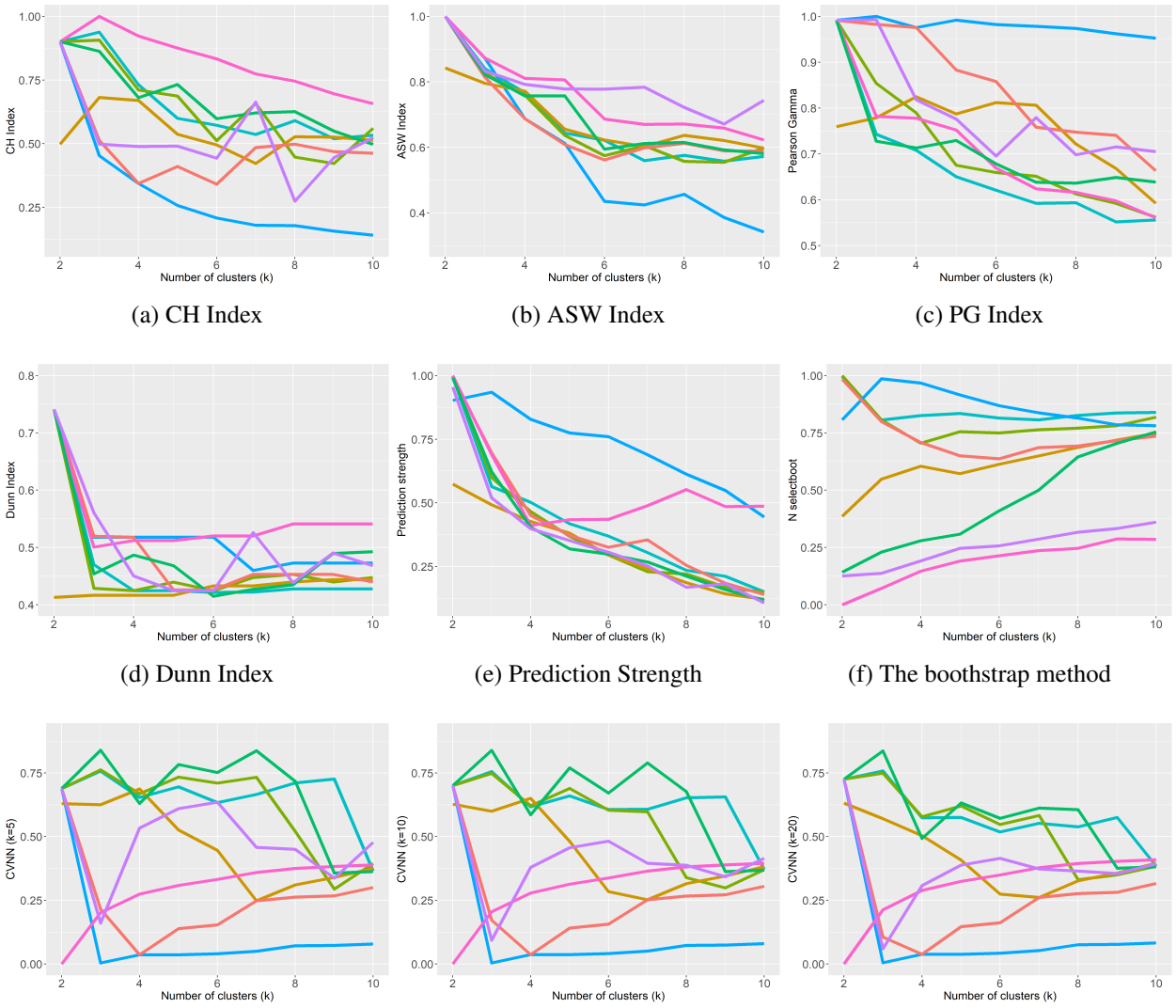
Wid.gap = Widest gap within dissimilarities, Sep.Ind = Separation Index, Mclust = Model based clustering

Table 7.29: SEED data set

Clustering Validation Index	ARI values			Average	Best clusterings in order		
	First	Second	Third	of ARI	First	Second	Third
Single Criterion							
CH Index	0.630	0.713	0.773	0.705	Mclust (K=3)	Ward (K=3)	K-means (K=3)
ASW Index	0.444	0.481	0.502	0.475	PAM (K=2)	K-means (K=2)	Spectral (K=2)
Dunn Index	0.440	0.433	0.773	0.548	Ward (K=8)	Spectral (K=5)	K-means (K=3)
Pearson gamma (PG)	0.630	0.444	0.713	0.595	Mclust (K=3)	PAM (K=2)	Ward (K=3)
Prediction strength (PS)	0.000	0.491	0.481	0.324	Single (K=2)	Average (K=2)	K-means (K=2)
N select boot (NSB)	0.000	0.747	0.773	0.506	Single (K=2)	PAM (K=3)	K-means (K=3)
CVNN ($\kappa = 5$)	0.630	0.686	0.773	0.696	Mclust (K=3)	Complete (K=3)	K-means (K=3)
CVNN ($\kappa = 10$)	0.630	0.686	0.747	0.687	Mclust (K=3)	Complete (K=3)	PAM (K=3)
CVNN ($\kappa = 20$)	0.630	0.773	0.747	0.716	Mclust (K=3)	K-means (K=3)	PAM (K=3)
Z-score aggregation							
Ave.wit + Ave.bet	0.630	0.713	0.747	0.696	Mclust (K=3)	Ward (K=3)	PAM (K=3)
Ave.wit + Ave.bet + PG	0.630	0.713	0.747	0.696	Mclust (K=3)	Ward (K=3)	PAM (K=3)
Ave.wit + Ave.bet + PS	0.481	0.453	0.747	0.560	K-means (K=2)	Ward (K=2)	PAM (K=3)
Ave.wit + Ave.bet + NSB	0.747	0.481	0.491	0.573	PAM (K=3)	K-means (K=2)	Average (K=2)
Ave.wit + Ave.bet + PG + PS	0.481	0.453	0.747	0.560	K-means (K=2)	Ward (K=2)	PAM (K=3)
Ave.wit + Ave.bet + PG + NSB	0.747	0.481	0.491	0.573	PAM (K=3)	K-means (K=2)	Average (K=2)
Wid.gap + Sep.Ind	0.000	0.453	0.433	0.295	Single (K=4)	Ward (K=2)	Spectral (K=5)
Wid.gap + Sep.Ind + PG	0.453	0.433	0.481	0.455	Ward (K=2)	Spectral (K=5)	K-means (K=2)
Wid.gap + Sep.Ind + PS	0.000	0.453	0.481	0.311	Single (K=2)	Ward (K=2)	K-means (K=2)
Wid.gap + Sep.Ind + NSB	0.000	0.481	0.493	0.324	Single (K=4)	K-means (K=2)	Average (K=10)
Wid.gap + Sep.Ind + PG + PS	0.453	0.481	0.502	0.478	Ward (K=2)	K-means (K=2)	Spectral (K=2)
Wid.gap + Sep.Ind + PG + NSB	0.453	0.000	0.747	0.409	K-means (K=2)	Single (K=4)	PAM (K=3)
Range aggregation							
Ave.wit + Ave.bet	0.630	0.713	0.747	0.696	Mclust (K=3)	Ward (K=3)	PAM (K=3)
Ave.wit + Ave.bet + PG	0.630	0.713	0.747	0.696	Mclust (K=3)	Ward (K=3)	PAM (K=3)
Ave.wit + Ave.bet + PS	0.481	0.453	0.747	0.560	K-means (K=2)	Ward (K=2)	PAM (K=3)
Ave.wit + Ave.bet + NSB	0.747	0.481	0.491	0.573	PAM (K=3)	K-means (K=2)	Average (K=2)
Ave.wit + Ave.bet + PG + PS	0.481	0.453	0.747	0.560	K-means (K=2)	Ward (K=2)	PAM (K=3)
Ave.wit + Ave.bet + PG + NSB	0.747	0.481	0.491	0.573	PAM (K=3)	K-means (K=2)	Average (K=2)
Wid.gap + Sep.Ind	0.000	0.453	0.433	0.295	Single (K=4)	Ward (K=2)	Spectral (K=5)
Wid.gap + Sep.Ind + PG	0.453	0.433	0.481	0.455	Ward (K=2)	Spectral (K=5)	K-means (K=2)
Wid.gap + Sep.Ind + PS	0.000	0.453	0.481	0.311	Single (K=2)	Ward (K=2)	K-means (K=2)
Wid.gap + Sep.Ind + NSB	0.000	0.481	0.747	0.409	Single (K=2)	K-means (K=2)	PAM (K=3)
Wid.gap + Sep.Ind + PG + PS	0.453	0.481	0.444	0.459	Ward (K=2)	K-means (K=2)	PAM (K=2)
Wid.gap + Sep.Ind + PG + NSB	0.481	0.747	0.488	0.572	K-means (K=2)	PAM (K=3)	Complete (K=2)

Ave.wit = Average within dissimilarities, Ave.bet = Average between dissimilarities

Wid.gap = Widest gap within dissimilarities, Sep.Ind = Separation Index, Mclust = Model based clustering



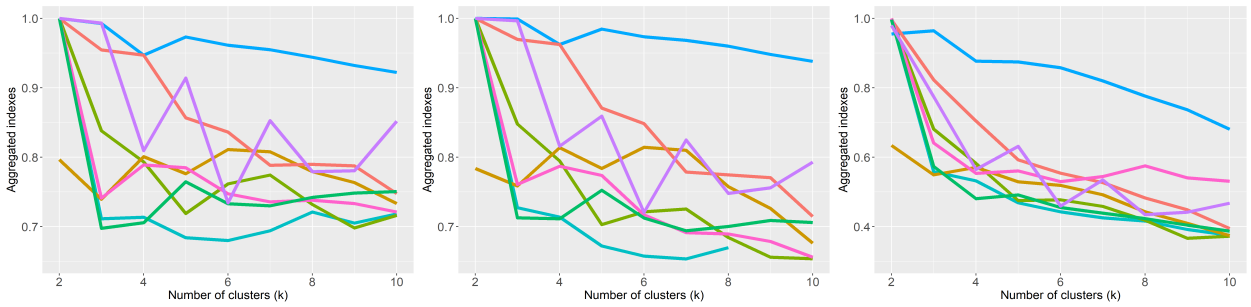
(a) CH Index (b) ASW Index (c) PG Index

(d) Dunn Index (e) Prediction Strength (f) The boothstrap method

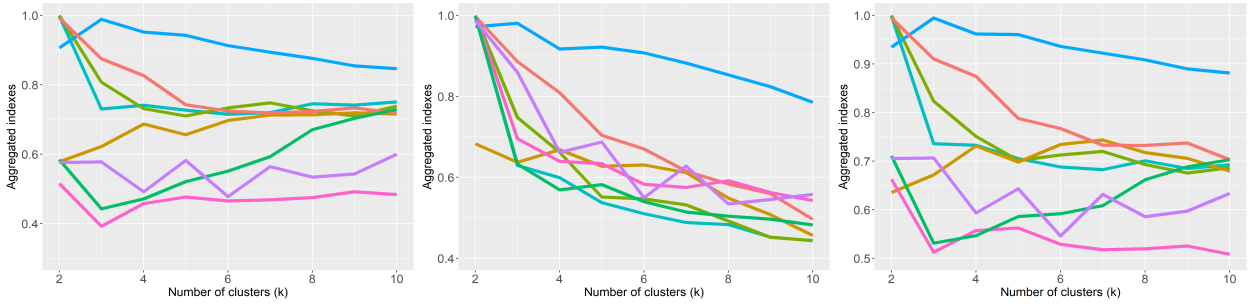
(g) CVNN ($\kappa = 5$) (h) CVNN ($\kappa = 10$) (i) CVNN ($\kappa = 20$)

Figure 7.14: Various single criteria for IRIS data set.

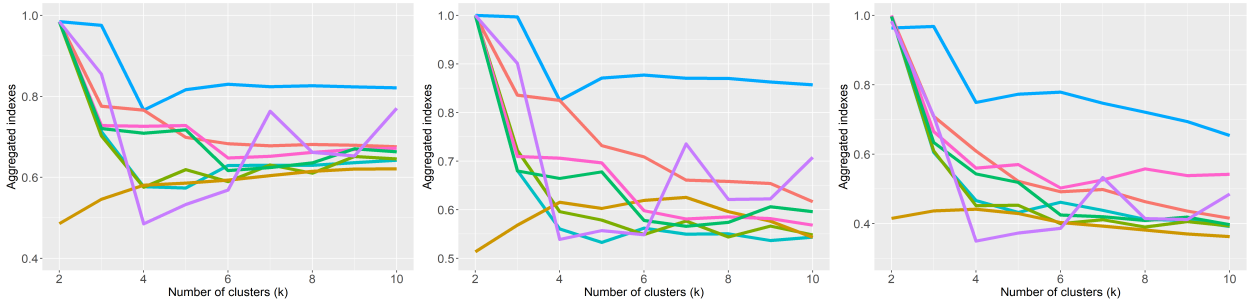
Average *Complete* *Kmeans* *Mclust* *PAM* *Single* *Spectral* *Ward*



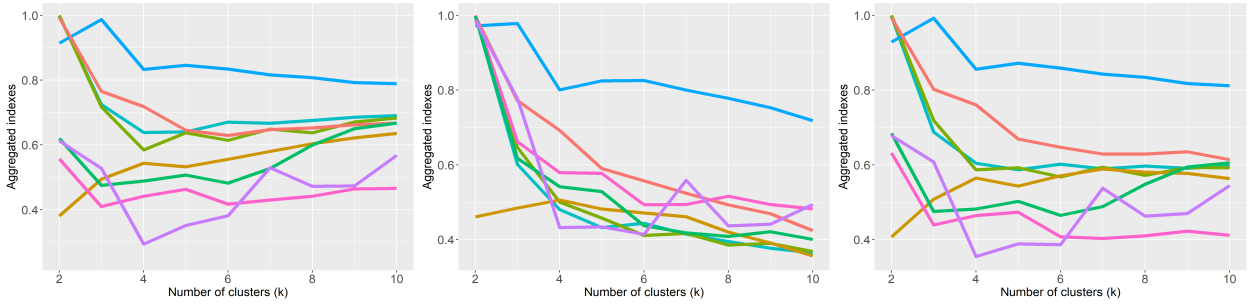
(a) AW + AB (b) AW + AB + PG (c) AW + AB + PS



(d) AW + AB + NSB (e) AW + AB + PG + PS (f) AW + AB + PG + NSB



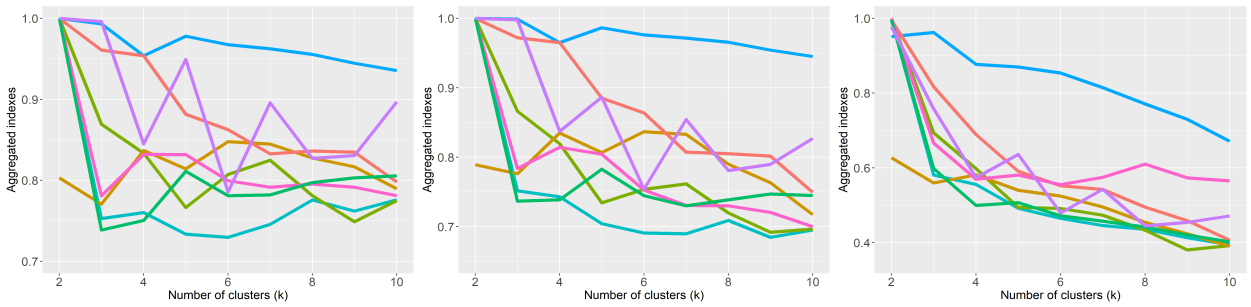
(g) WG + SI (h) WG + SI + PG (i) WG + SI + PS



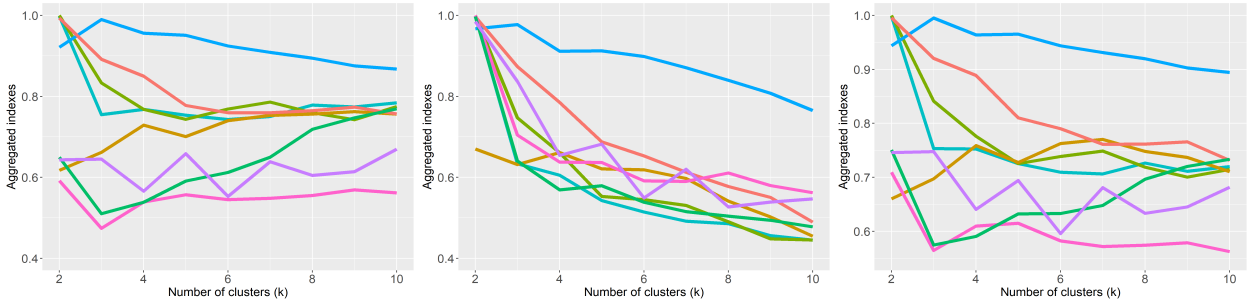
(j) WG + SI + NSB (k) WG + SI + PG + PS (l) WG + SI + PG + NSB

Figure 7.15: Different aggregated index considerations with Z-score standardisation for IRIS data set.

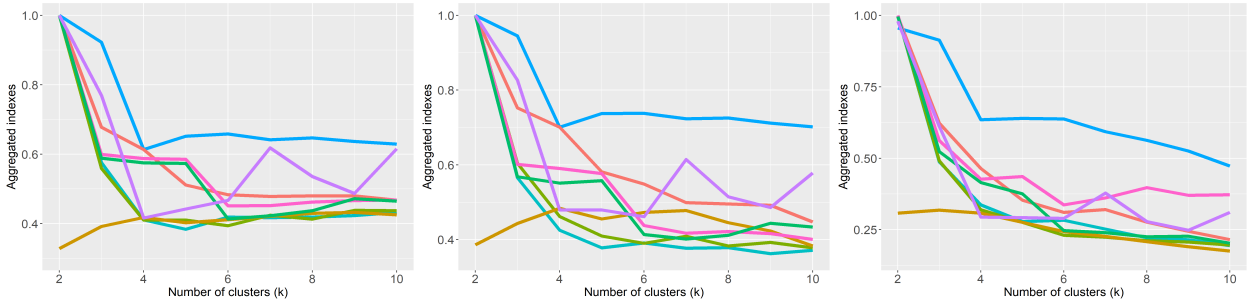
Average *Complete* *Kmeans* *Mclust* *PAM* *Single* *Spectral* *Ward*



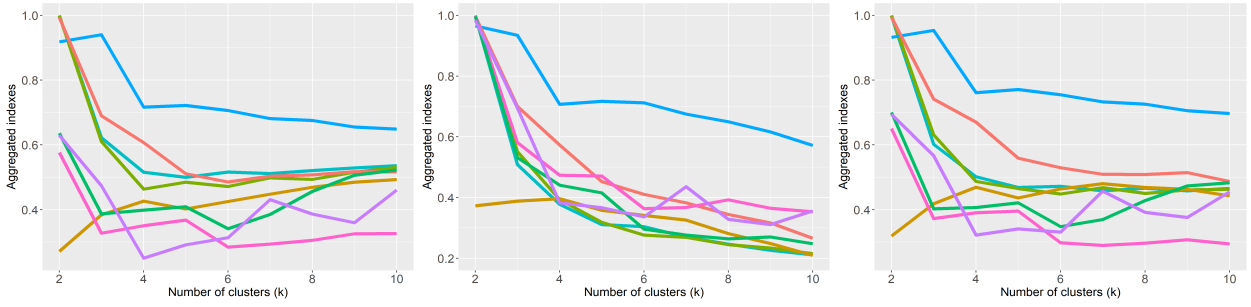
(a) AW + AB (b) AW + AB + PG (c) AW + AB + PS



(d) AW + AB + NSB (e) AW + AB + PG + PS (f) AW + AB + PG + NSB



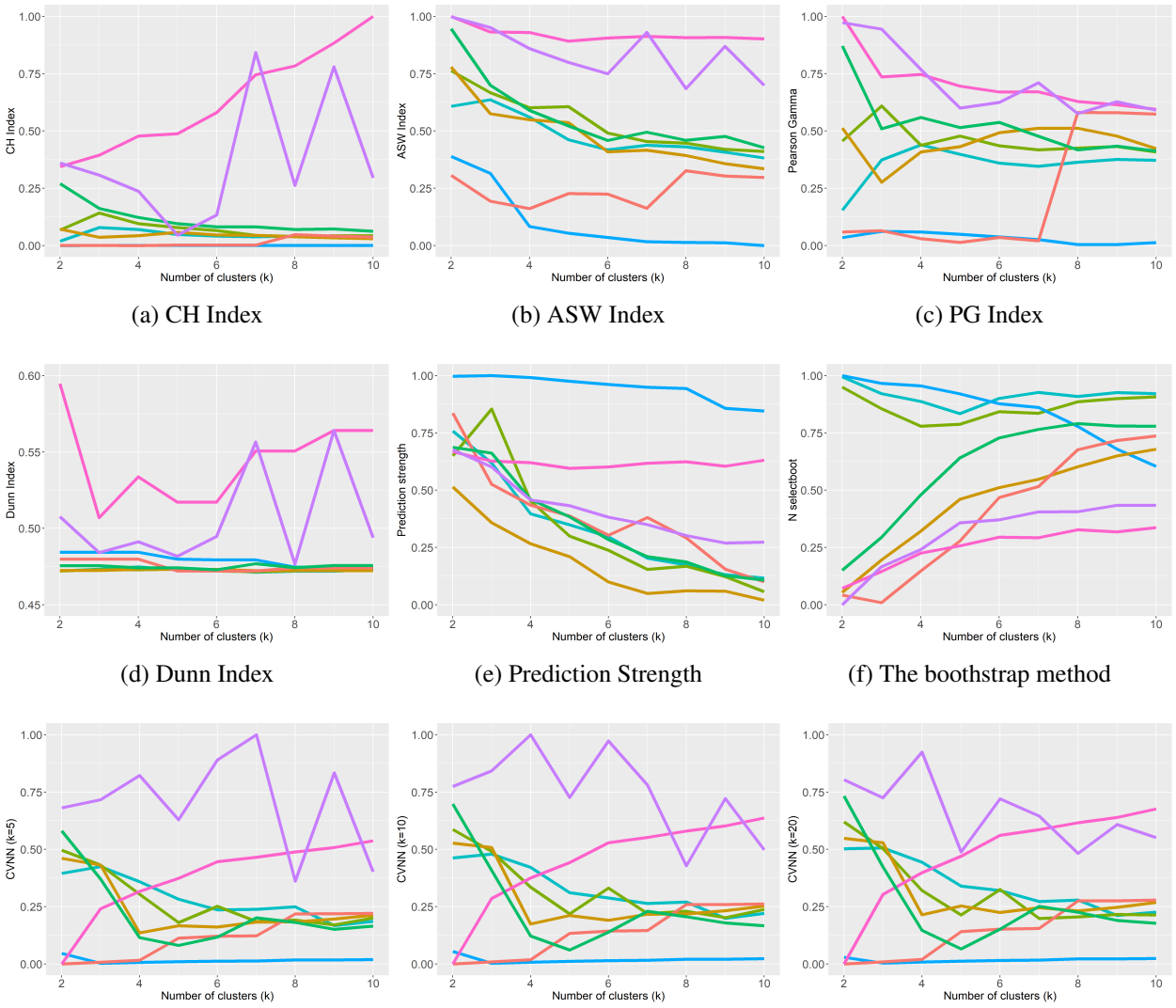
(g) WG + SI (h) WG + SI + PG (i) WG + SI + PS



(j) WG + SI + NSB (k) WG + SI + PG + PS (l) WG + SI + PG + NSB

Figure 7.16: Different aggregated index considerations with Range standardisation for IRIS data set.

Average *Complete* *Kmeans* *Mclust* *PAM* *Single* *Spectral* *Ward*



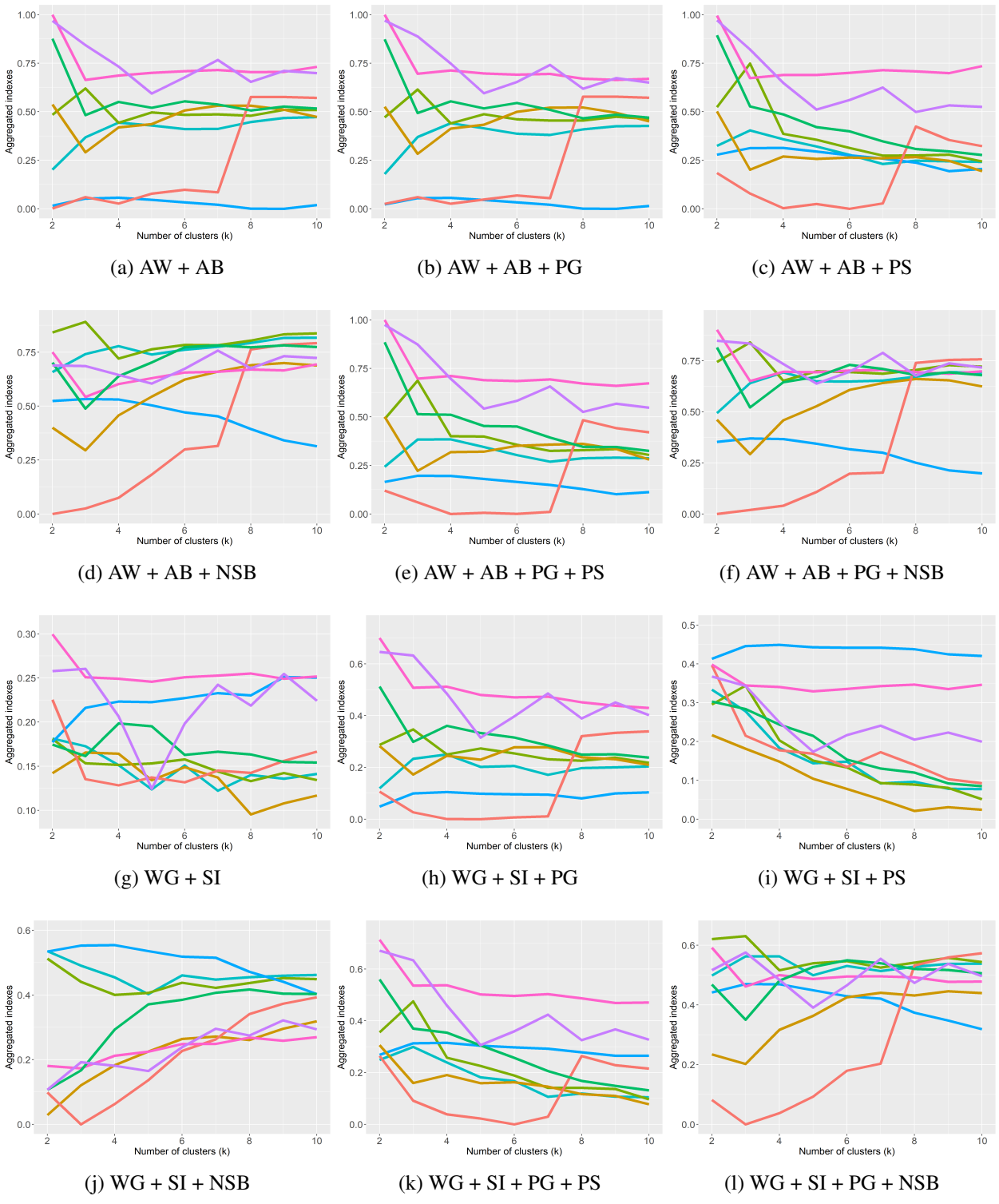
(a) CH Index (b) ASW Index (c) PG Index

(d) Dunn Index (e) Prediction Strength (f) The boothstrap method

(g) CVNN ($\kappa = 5$) (h) CVNN ($\kappa = 10$) (i) CVNN ($\kappa = 20$)

Figure 7.17: Various single criteria for WINE data set.

Average *Complete* *Kmeans* *Mclust* *PAM* *Single* *Spectral* *Ward*



(a) AW + AB (b) AW + AB + PG (c) AW + AB + PS

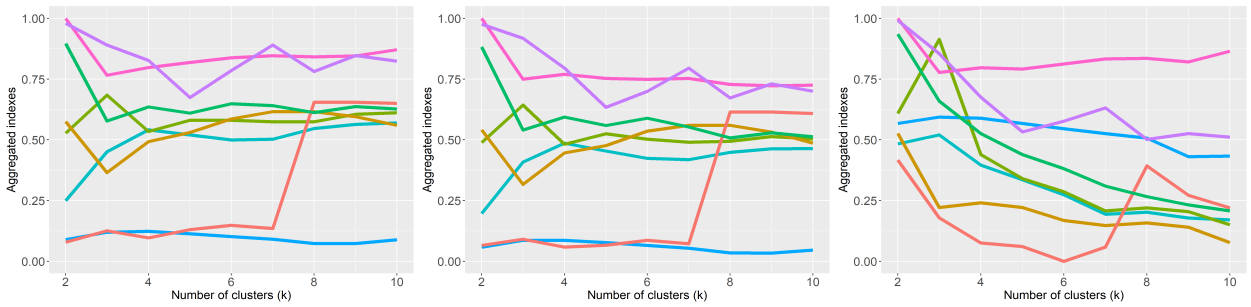
(d) AW + AB + NSB (e) AW + AB + PG + PS (f) AW + AB + PG + NSB

(g) WG + SI (h) WG + SI + PG (i) WG + SI + PS

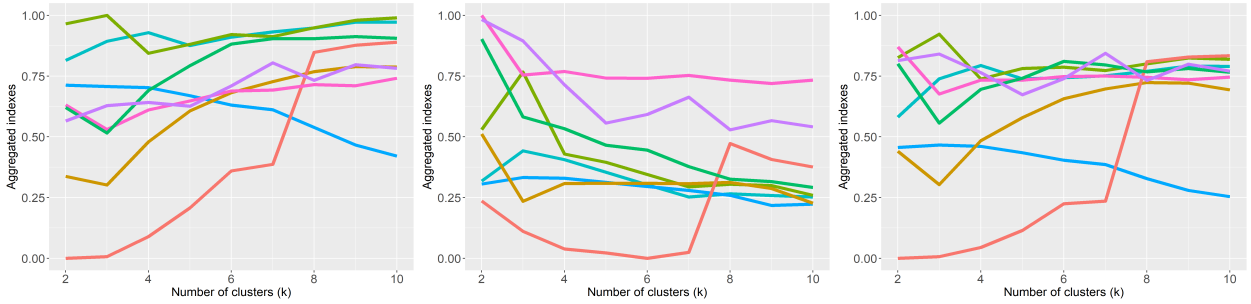
(j) WG + SI + NSB (k) WG + SI + PG + PS (l) WG + SI + PG + NSB

Figure 7.18: Different aggregated index considerations with Z-score standardisation for WINE data set.

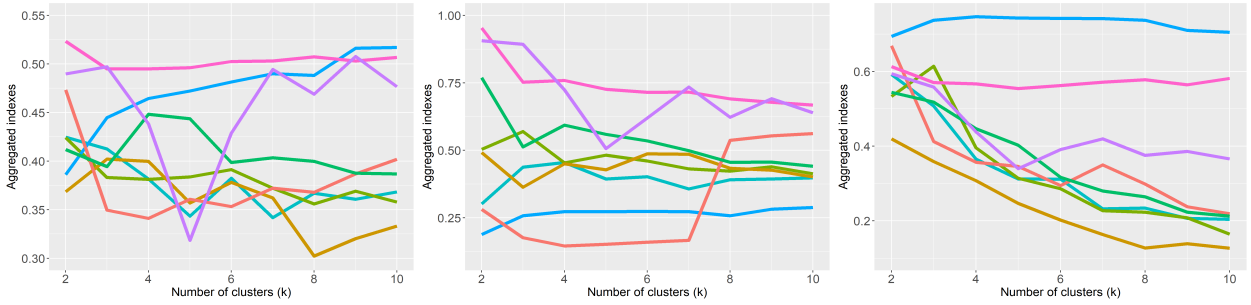
Average *Complete* *Kmeans* *Mclust* *PAM* *Single* *Spectral* *Ward*



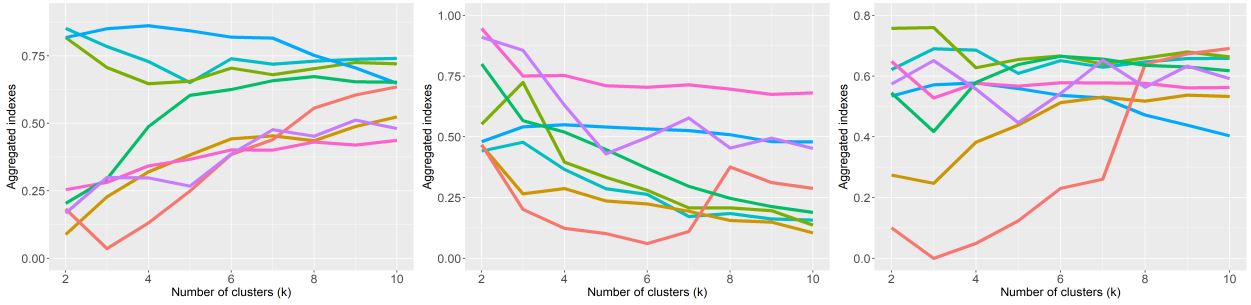
(a) AW + AB (b) AW + AB + PG (c) AW + AB + PS



(d) AW + AB + NSB (e) AW + AB + PG + PS (f) AW + AB + PG + NSB



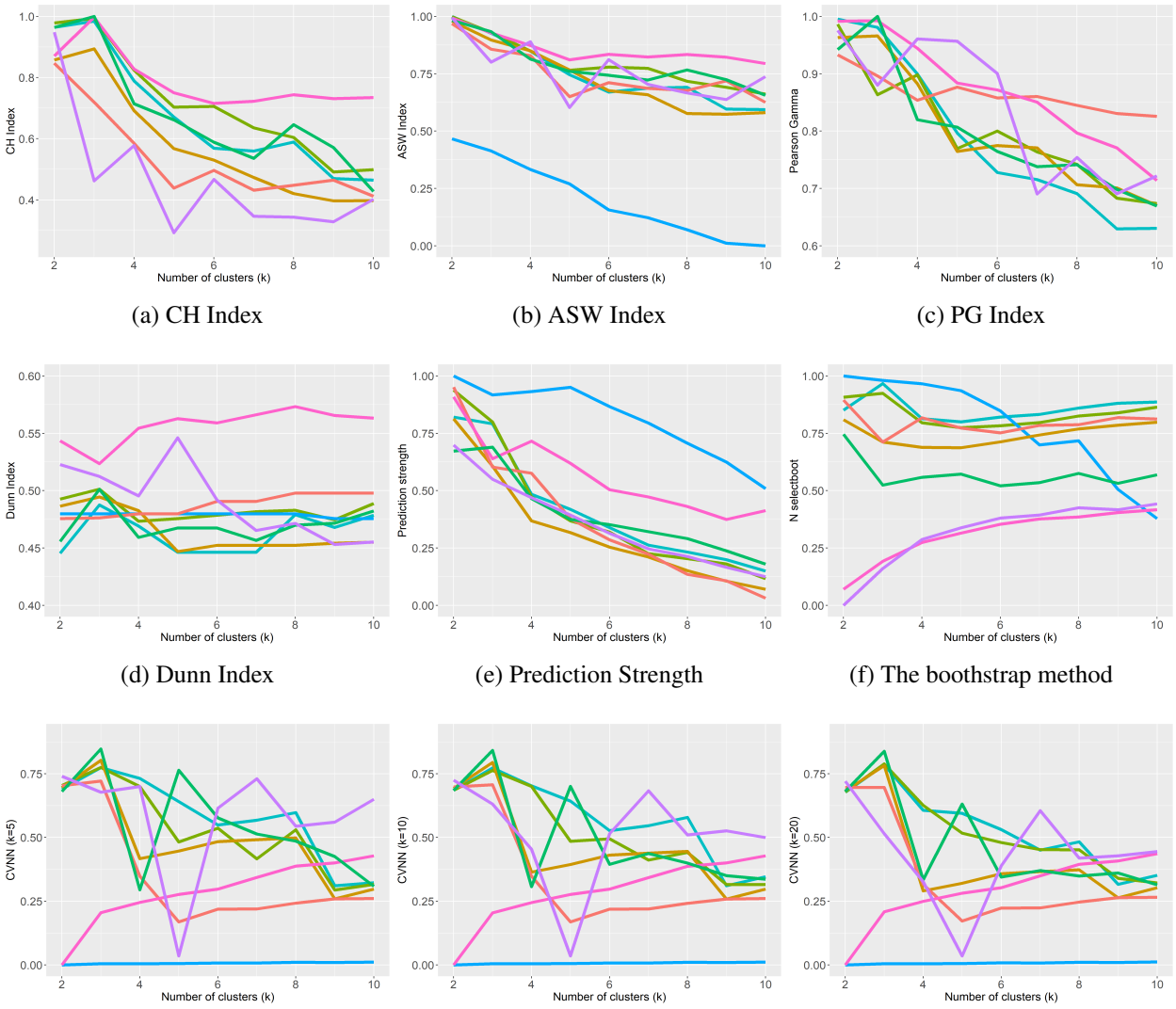
(g) WG + SI (h) WG + SI + PG (i) WG + SI + PS



(j) WG + SI + NSB (k) WG + SI + PG + PS (l) WG + SI + PG + NSB

Figure 7.19: Different aggregated index considerations with Range standardisation for WINE data set.

Average *Complete* *Kmeans* *Mclust* *PAM* *Single* *Spectral* *Ward*



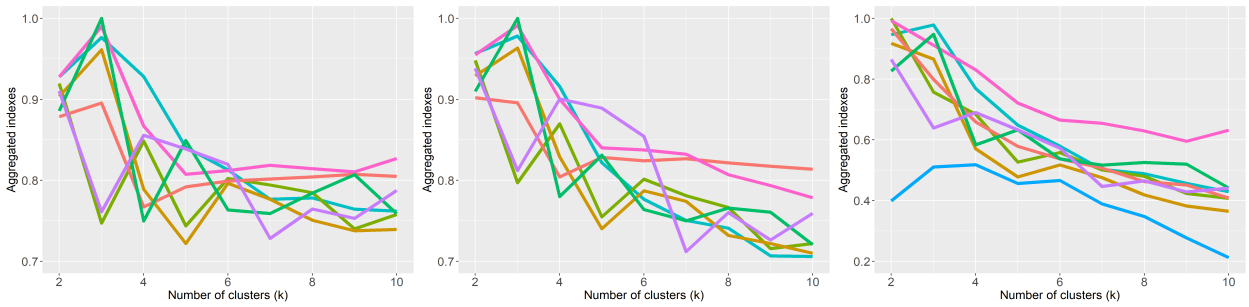
(a) CH Index (b) ASW Index (c) PG Index

(d) Dunn Index (e) Prediction Strength (f) The boothstrap method

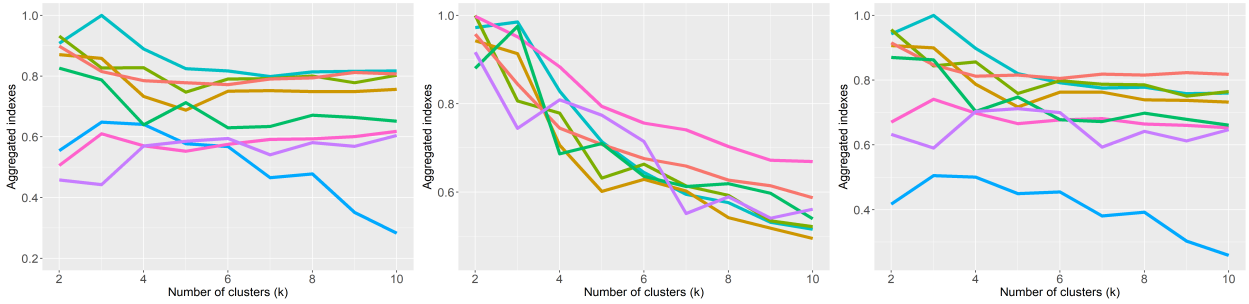
(g) CVNN ($\kappa = 5$) (h) CVNN ($\kappa = 10$) (i) CVNN ($\kappa = 20$)

Figure 7.20: Various single criteria for SEED data set.

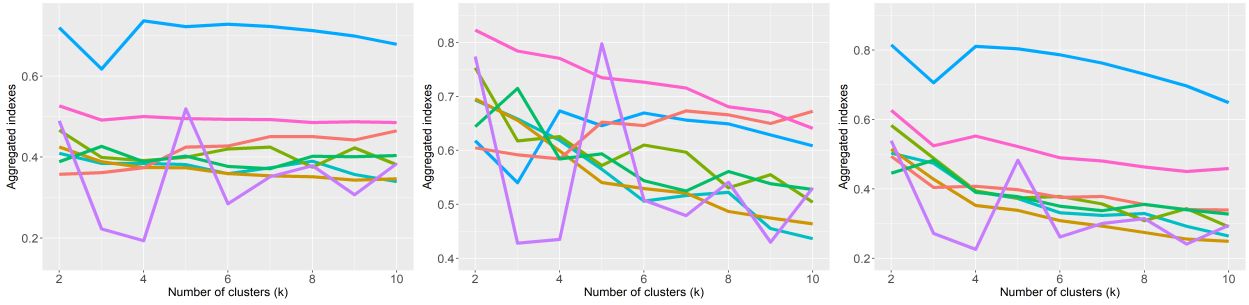
Average *Complete* *Kmeans* *Mclust* *PAM* *Single* *Spectral* *Ward*



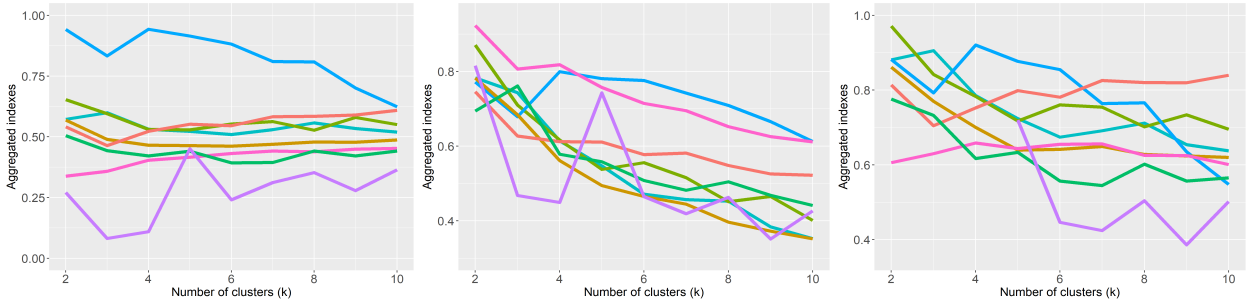
(a) AW + AB (b) AW + AB + PG (c) AW + AB + PS



(d) AW + AB + NSB (e) AW + AB + PG + PS (f) AW + AB + PG + NSB



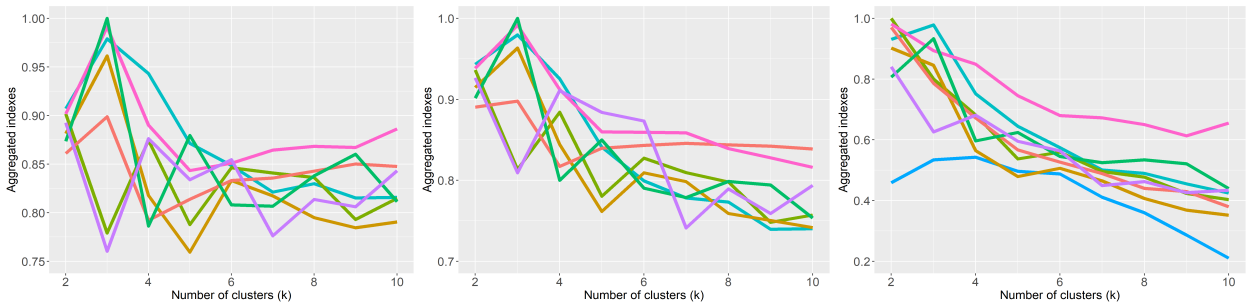
(g) WG + SI (h) WG + SI + PG (i) WG + SI + PS



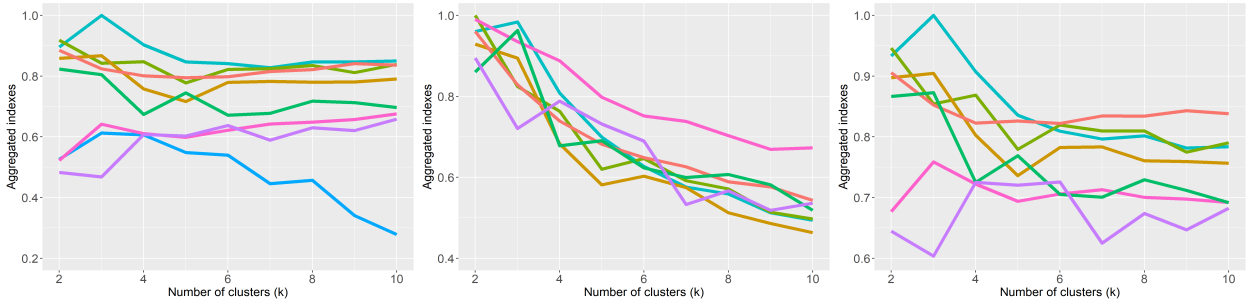
(j) WG + SI + NSB (k) WG + SI + PG + PS (l) WG + SI + PG + NSB

Figure 7.21: Different aggregated index considerations with Z-score standardisation for SEED data set.

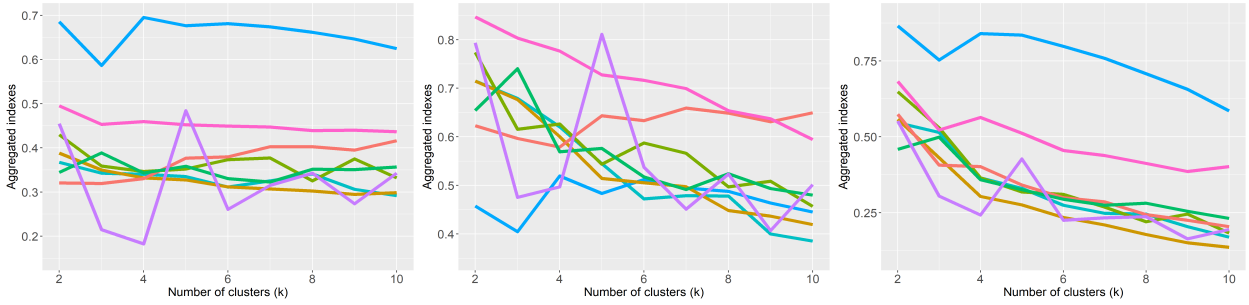
Average *Complete* *Kmeans* *Mclust* *PAM* *Single* *Spectral* *Ward*



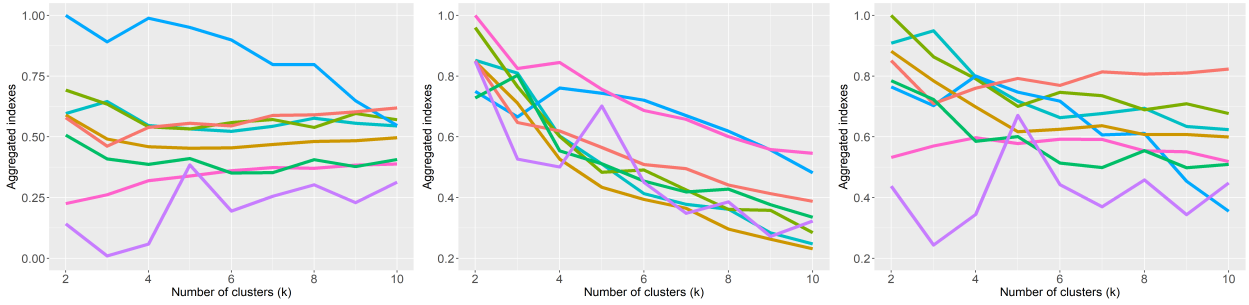
(a) AW + AB (b) AW + AB + PG (c) AW + AB + PS



(d) AW + AB + NSB (e) AW + AB + PG + PS (f) AW + AB + PG + NSB



(g) WG + SI (h) WG + SI + PG (i) WG + SI + PS



(j) WG + SI + NSB (k) WG + SI + PG + PS (l) WG + SI + PG + NSB

Figure 7.22: Different aggregated index considerations with Range standardisation for SEED data set.

Average *Complete* *Kmeans* *Mclust* *PAM* *Single* *Spectral* *Ward*

7.3 Final Comments

In this chapter, I run several simulation studies to investigate how successful the aggregation of clustering validation index results are when compared with some popular clustering validation criteria. Furthermore, some real data sets are analysed with the same purpose. The analysis in this chapter leads to some conclusions regarding the use of clustering validation in estimating the number of clusters. In this sense, some final comments will be given with respect to the clustering validation criteria results.

- For clusters with approximately normally distributed points, clustering methods which are usually based on some kind of optimization algorithms (K-means, PAM, model based clustering, etc.) estimate the given cluster labels well, see the ARI results in Sections 7.1.8 and 7.2.4. Combinations of clustering validation aspects, such as the calibration of average within dissimilarities and average between dissimilarities, which often focuses on the homogeneity, have better capability to estimate the correct number of clusters based on the given class labels than the aspects that tend to focus more on the extreme points, such as the combination of widest gap and separation index, see Tables 7.10, 7.11, 7.12 and 7.13.

- Single linkage and spectral clustering algorithms perform better for irregularly shaped clusters than the other clustering algorithms. For this kind of cluster, clustering validation criteria, which focus more on the extreme points, such as the calibration of widest gap and separation index, or Dunn Index, accurately predict the number of clusters, see Table 7.18, 7.19, 7.20, 7.21, 7.22 and 7.23.

- In general, the stability methodologies (Prediction strength or the bootstrap method) often fail to estimate the appropriate number of clusters when model based clustering is applied.

- For clusters that are not well separated, see for example Section 7.1.3, the stability measurements are not very successful in finding the correct number of clusters compared to the other clustering validation criteria.

- The CVNN Index often estimates the correct number of clusters well in many instances, except for the irregularly shaped clusters.

- For these different types of simulated data sets, our findings indicate that range aggregation of different aspects usually works better than Z-score aggregation for predicting the correct number of clusters.

- For real data set examples, different aggregated clustering validation index scenarios are used.
  - For the iris data set, the results for various calibration selections indicate that the $K = 2$ solution for different clustering algorithms gives good results in terms of estimating the appropriate number of clusters, which can also be validated from the graphical representation, despite the given true classes having a $K = 3$ solution. On the other hand, the selections calibrated with a stability measurement indicate that single linkage is reasonable for estimating the number of clusters in some cases. That is because single linkage usually generates one cluster with one or two points and the remaining points are in the other clusters, resulting in a more stable solution than for other scenarios, since there are no substantial changes in the sampling stage between the points in the clusters.
  - For the wine and the seed data sets, the homogeneity within clusters is more distinct than the homogeneity within clusters for the iris data set according to the PCA plots in Figure 7.8, 7.10 and 7.12. The clustering validation index results (especially the calibration of average within dissimilarities and average between dissimilarities) point out that clustering algorithms, which usually have better capability to deal with homogeneously distributed and not well separated clusters, such as K-means, PAM, model based clustering or Ward's method, accurately find the true number of clusters in many situations.

The selection of clustering validation aspects for estimating the number of clusters is not an easy task. Many different combinations can be chosen, but the user does not know in advance which cluster solution is the most appropriate. As mentioned at the beginning of this chapter, the selection of calibration can be based on the different characteristics of the clustering validation indexes and according to subject matter knowledge, which reduces the range of the user's selection. On the other hand, one could apply PCA (or MDS for dissimilarities) to the data set of interest in order to visualise the distribution of the cluster points in a low dimensional projection. The graphical representation gives the user guidance on what clustering validation indexes may be preferable. For example, if clusters are distributed more homogeneously, then the choice of calibration can depend on the indexes that focus more on homogeneity, such as average within dissimilarities and average between dissimilarities, see the data sets in Sections 7.1.1, 7.1.2, 7.1.3, or 7.2.2. In contrast, clustering validation aspects, which focuses more on separation between clusters (e.g., the widest gap and the separation index), can be selected for the clusterings that project irregular shapes on a low dimensional space, see for example the data sets in Sections 7.1.5, 7.1.6, or 7.1.7.

Alternatively, the user can apply all different aggregated index versions for their analysis (as implemented in this chapter), and optimise different aggregation results by choosing one of the best selections out of all choices. As mentioned in the previous section, `R Shiny` implementations, which are shown as screen-shots in Appendix B, can be used for this sake. However, firstly the user has no idea what the right choices are in real situations, and secondly it is not the case that the same combination always performs well; in other words, there is no combination that can be recommended universally. On the other hand, the results of simulation studies and the analysis of the real data sets give us a direction that some of the calibration scenarios perform better than all the other observed combinations. Thus, one could say that any combinations of average within dissimilarities, average between dissimilarities and the bootstrap method usually give a satisfactory result especially for data sets in which clusters are homogeneously distributed, see Table 7.12, Figure 7.19, 7.18,7.22, 7.21, etc. It is important to note that not every data set needs a different aggregation, and the same combinations may be appropriate over a good range of datasets. Consequently, the suggestions above give the user guidance in how to choose the most appropriate clustering solution.

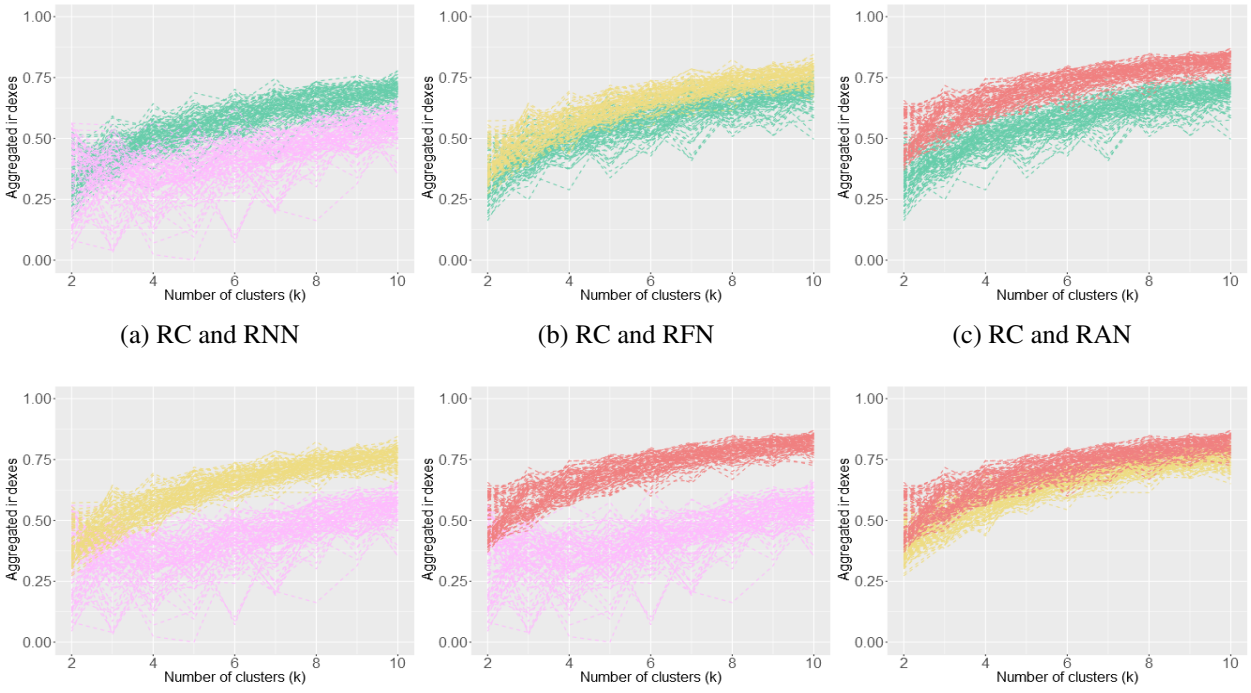
One of the fundamental contributions of this thesis is the new clustering algorithms, which are the random furthest neighbours and the random average neighbours. As stated in Section 6.2.3, these new algorithms explore some other regions of possible solutions that are not covered by the random centroid and the random nearest neighbours, which leads to obtaining different scale value of standardisation. As an illustration, Figure 7.23 displays a different combination of random clustering algorithms. For this case, a different real data set, Movement data set (Dias et al., 2009) is used. The data set contains 15 classes of 24 instances each, where each class references to a hand movement type in LIBRAS [1]. Here, the data set is represented with 90 features, where the number of variables is much larger than the previous real data set examples.

The combination of average within dissimilarities, average between dissimilarities, Pearson Gamma index and the bootstrap method, which is the suitable choice from the aggregation of clustering validation index results, is applied for the random clustering algorithms. The results in Figure 7.23 indicate that the solutions for the random nearest neighbour cover a wider range of smaller values compared to the other random clustering algorithms. The values obtained from the random average neighbour and the random furthest neighbour are larger than the other random clustering index values, but the index values obtained from the random average neighbour is a bit larger than the index values generated from the random furthest neighbour. All the illustrations in Figure 7.23 point out that these random clustering algorithms give a different range of index values

[1] LIBRAS is the Brazilian Sign Language (Libras - from the original name in Portuguese "Lingua BRAsileira de Sinais").

that make the aggregation process more suitable by contributing more consistent scale value for standardisation. In addition, the new random clustering algorithms have the potential to deliver a distinct space of index values that the other random clustering algorithms do not yield.

In many cases, the results indicate that the aggregation of clustering validation indexes is quite successful in finding the correct number of clusters. For researchers, I strongly recommend this methodology for finding the appropriate number of clusters, K.



(a) RC and RNN (b) RC and RFN (c) RC and RAN

(d) RNN and RFN (e) RNN and RAN (f) RFN and RAN

Figure 7.23: Clustering validation index results for different random clustering algorithms scenarios of MOVEMENT data set. The index results are based on the aggregated indexes of average within and between dissimilarities, Pearson Gamma Index and the bootstrap method

Random centroid (RC) *Random nearest (RNN)* *Random furthest (RFN)* *Random average (RAN)*

CHAPTER 8

EXAMINATION OF AGGREGATING CLUSTERING VALIDATION INDEXES ON THE FOOTBALL PLAYERS PERFORMANCE DATA SET

I started this thesis with the idea of designing a dissimilarity measure between football players based on their performance information. Then, the aim was to group the players in such a way that managers or scout teams can identify and explore players who are similar to a player of their interest. In this respect, I work on making the decision about the number of clusters by considering the two aspects that are explained in the first paragraph of Section 6.1.7. The introduced objective criteria (e.g., ASW Index, CH Index, etc.) in Chapter 6 can be adapted for the aim of finding an optimal number of clusters, but this does not mean that the optimal number of clusters is necessarily informative and practicable for football squads and managers. For example, consider the football player data set to be used for this project, where $n \approx 3000$, from which users want to identify specific types of players similar to another player of interest. If the number of clusters is relatively small, say less than 10, then the size of clusters for those specific players to be obtained from the clustering is most likely large, which may not be very useful. Managers would prefer to observe smaller groups of relevant players, making their jobs more efficient. For computational reasons, I analyse a subset of the data containing 1500 players ($n = 1500$). More consistent information is provided by players who have played a greater number of minutes in a season, therefore the chosen subset contains players who have played more than the median total number of minutes. Following the argument above, I decide to set the maximum number of clusters as $K = 150$, since it would be more interesting to a have relatively small size (say between $10 < n_K < 20$) for each cluster from the user's point of view.

In this section, I will discuss estimating the number of clusters in two respects together: 1) from

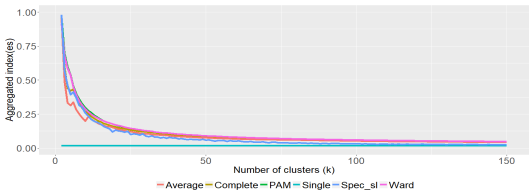
a statistical point of view and 2) based on subject-matter reasons. In Section 4.3, the sensitivity analysis results indicate that the appropriate choice for aggregating different dissimilarity measures is range standardisation when applying the log $(x + c)$ transformation. Note that the discussion for the choice of transformation with a suitable constant, c, is made in Section 4.1.2.

Here K-means and model-based clustering algorithms are not used, since the original versions of these algorithms require a raw data set as an input, whereas a dissimilarity measure of football players information is adopted in this application as an input. The `R` function of spectral clustering (see Table 7.1) requires the matrix of data to be clustered. I implement my own function for spectral clustering to be used for a dissimilarity measurement as an input based on Algorithm 8. To compute the degree matrix P in the same algorithm, since the aim is to adopt the original dissimilarity measures between players, there is no such kernel function (e.g., Gaussian Kernel) used for this setting. The Normalised Laplacian with random walk matrix is preferred for the choice of Laplacian Matrix, see in Table 5.1.

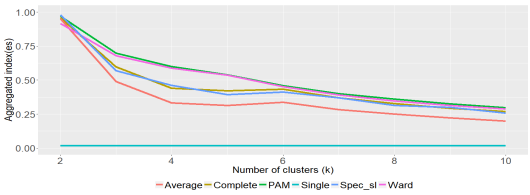
Figure 8.1 and 8.2 display the results of various objective criteria and the aggregated clustering validation indexes with Z-score and range standardisations for estimating the number of clusters on the dissimilarity matrix of football players. The plots on the left-hand side show how different criteria for different clustering algorithms act for different number of clusters over the range of $K \sim [2 : 150]$, whereas the plots on the right hand side provide a narrow range of numbers of clusters to better view the optimum choice of a clustering algorithm. Additionally, clustering validation index results of random clustering algorithms are presented in Figures 8.1 and 8.2.

The results in Figure 8.1 indicate that the majority of the indexes favour a small numbers of clusters, except the Dunn Index and the bootstrap method. The single linkage result in Figure 8.1k has a different pattern from the other clustering algorithms, and indicates that as the number of cluster increases, single linkage becomes worse based on the result of the bootstrap method. For the other clustering algorithms, although the index values obtained from the bootstrap method give better results for large numbers of clusters, it is difficult to detect a peak from the index results over the range of different numbers of clusters, because the index values become more stable as the number of cluster increases. CVNN Index results show that Ward's method for $K = 4$ is the optimal solution when compared to numerous clustering algorithms for different numbers of clusters, whereas the PG Index results assert that average linkage for $K = 4$ is the optimum choice. The Dunn Index, which focuses on the large values as shown in Equation (6.23), predicts the best choice as complete linkage for $K = 145$.

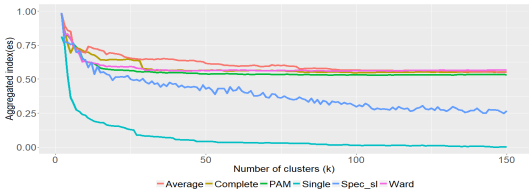
For this data set, the selection of clustering validation indexes for the sake of calibration relies upon two aspects. In Chapter 7, different aggregation scenarios are examined with the simulated



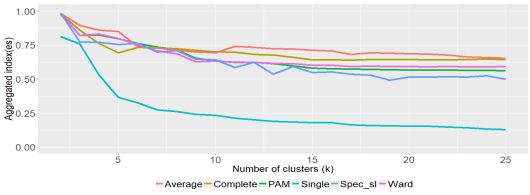
(a) CH Index $K \sim [2:150]$



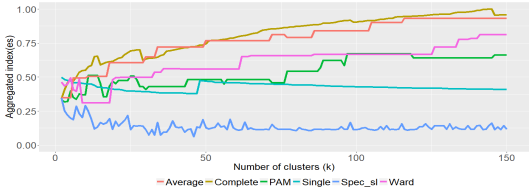
(b) CH Index $K \sim [2:10]$



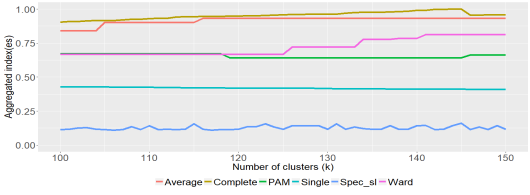
(c) ASW Index $K \sim [2:150]$



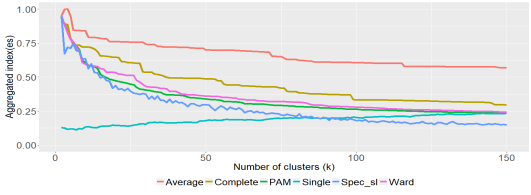
(d) ASW Index $K \sim [2:25]$



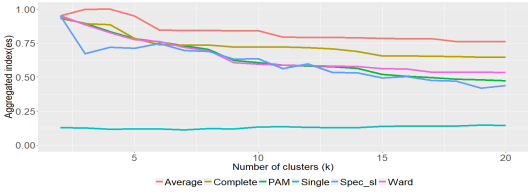
(e) Dunn Index $K \sim [2:150]$



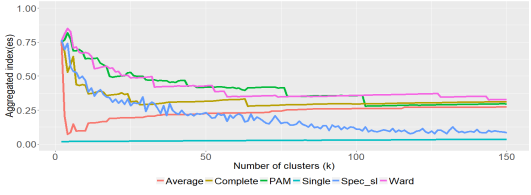
(f) Dunn Index $K \sim [100:150]$



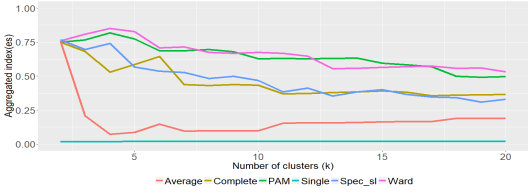
(g) PG Index $K \sim [2:150]$



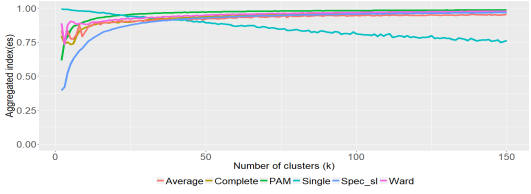
(h) PG Index $K \sim [2:20]$



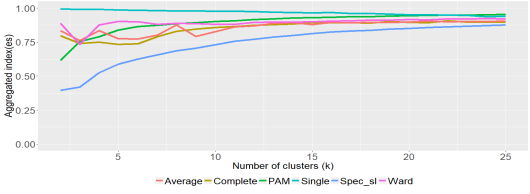
(i) CVNN ($\kappa = 10$) Index $K \sim [2:150]$



(j) CVNN ($\kappa = 10$) Index $K \sim [2:20]$



(k) The bootstrap method $K \sim [2:150]$



(l) The bootstrap method $K \sim [2:25]$

Figure 8.1: Various cluster validation indexes for FOOTBALL data set.

Average *Complete* *PAM* *Single* *Spectral* *Ward*

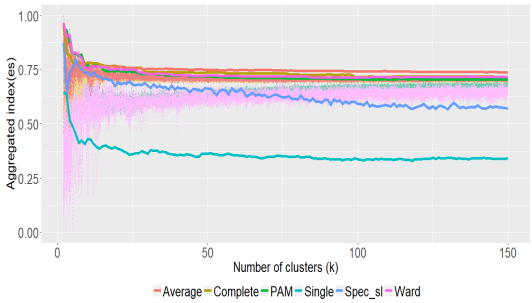
and the real data sets. As previously pointed out, the results indicate that the calibration of average within dissimilarities (AW), average between dissimilarities (AB), Pearson Gamma Index (PG), and the bootstrap method (NSB) is reasonable in most situations. Figure 8.3 indicates that the players information is very likely homogeneously distributed, which motivates the homogeneous clustering validation aspects for the sake of aggregation, such as average within and between cluster dissimilarities rather than adopting widest gap or separation index. Therefore, the calibration scenario ($I_{AW} + I_{AB} + I_{PG} + I_{NSB}$) will also be considered for the football data set from a statistical point of view. Secondly, the subjective relation between the choice of clustering validation indexes and football knowledge should be considered. In terms of average within and average between dissimilarities, the characteristic of keeping within-cluster distances small is more important than the characteristic of keeping between-cluster distances large. That is because the aim is to make the players in the same cluster as similar as possible. On the other hand, having large between-cluster distances is not primarily important, since one of the other aims is to have a large number of clusters. In this sense, the weights can be assigned to directly reflect the relative importance of the various clustering quality aspects, so that I decide to give 0.5 weight for average between dissimilarities and 1 weight for average within dissimilarities. Pearson Gamma Index is also important, because the similarity between players should well reflect the underlying dissimilarity structure. Investigating the stability using the bootstrap method is certainly of interest in order to measure the stability of the clustering structure of football players information. Finally, entropy, which simply measures the uniformity of cluster sizes, is included in different calibration considerations, because the aim for the football performance data set is to obtain balanced cluster sizes. For example, if football scouts wish to explore similar players to two specific players, then very different numbers of players in the respective clusters (e.g., 10 and 100) would be impractical from their point of view.

As explained in Section 6.2, the aggregation can be made by computing a weighted mean of selected indexes $I_1, \ldots, I_s$ with weights $w_1, \ldots, w_s > 0$, which are denoted as the relative importance of the different clustering quality indexes. For the football data set, the formula for aggregation of clustering validation indexes is given as follows.

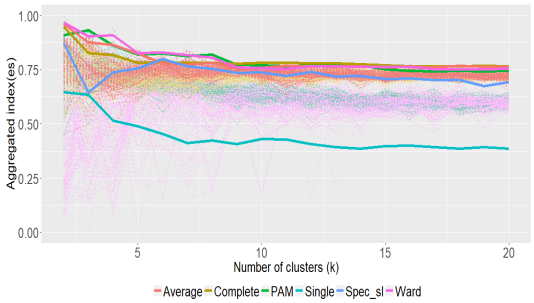
$$\mathcal{A}(\mathcal{C}) = I_{AW} + 0.5I_{AB} + I_{PG} + I_{Ent} + I_{NSB}. \tag{8.1}$$

Prior to the aggregation of all the indexes in Equation (8.1), Z-score and range standardisations are separately applied for each clustering validation index to make them comparable in terms of their variation.

The aggregated clustering validation index results in Figure 8.2 indicate that Ward's method

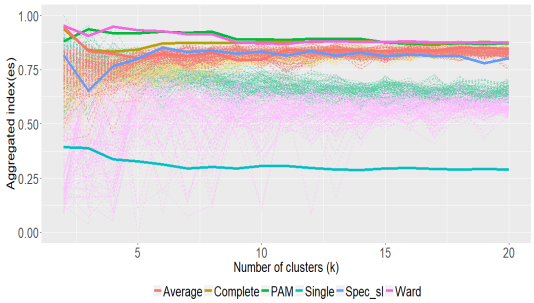


(a) $Z-score$: AW+(0.5)AB+Ent+PG+NSB $K \sim [2:150]$



(b) $Z-score$: AW+(0.5)AB+Ent+PG+NSB $K \sim [2:20]$

(c) Range: AW+(0.5)AB+Ent+PG+NSB $K \sim [2:150]$



(d) Range: AW+(0.5)AB+Ent+PG+NSB $K \sim [2:20]$

Figure 8.2: Various calibration of clustering validation index scenarios for FOOTBALL data set. Z-score and range standardisation are used for calibration. PG: Pearson gamma, AW: Average within dissimilarities, AB: Average between dissimilarities, NSB: The bootstrap method, Ent:Entropy.

Average *Complete* *PAM* *Single* *Spectral* *Ward*
Random centroid *Random nearest* *Random furthest* *Random average*

for $K = 4$ may be preferred as the best choice compared to the other clustering results. The choice of PAM for $K = 3$ may be another good selection for some calibration considerations, due to the existence of a peak point. Again, it is difficult to identify an optimum solution due to the stable index results over the range of large numbers of clusters. Moreover, as shown in the same figures, most of the random clustering validation index results are worse than the clustering validation index results of real clustering algorithms. This is reasonable since random clustering algorithms are designed based on the selection of random K centroids or K initialisation points rather than using some type of optimising algorithm to estimate optimum choice of these K points.

One of the aims of this project is to visualise football players in a low dimensional space in order to present their information in a mapping form that is practicable for football squads and managers. Multidimensional scaling (MDS), as described in Section 5.3.2, is the method for visualising the level of similarity of players. Non-metric MDS seems to be more useful for mapping football player information, because the dissimilarity matrix of the football players performance data set is not a metric, and non-metric MDS (Ordinal MDS) can be the appropriate choice for

this specific application. Figure 8.3 displays several MDS plots with two principal coordinates for each two dimensional plot. The stress value is nearly 10%, which implies a fair result based on Table 5.2. MDS graphical representations with a fair stress value guide us on how to interpret the players information for the aim of clustering. The clustering solutions in Figure 8.3 can be disregarded at this point, because we are only considering how the football players are distributed in a two dimensional plot.

(a) Average linkage ($K = 4$)

(b) Ward's method ($K = 4$)

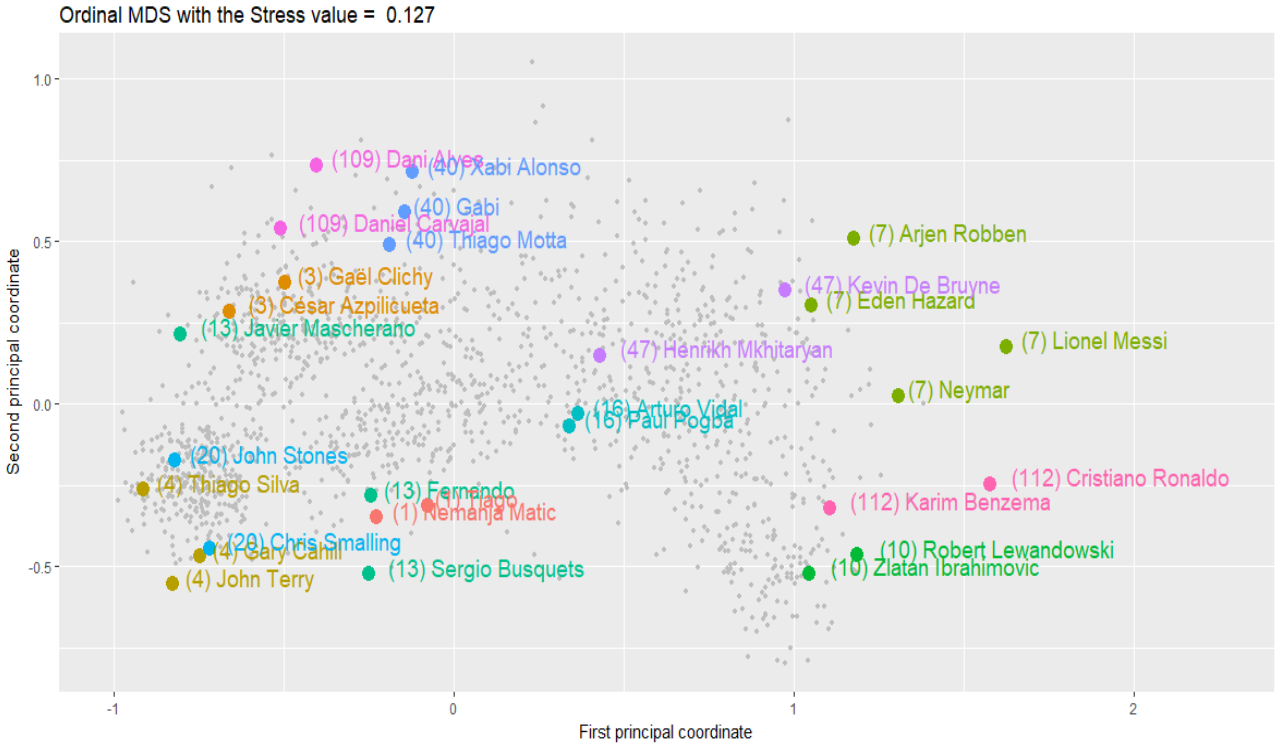
(c) PAM ($K = 3$)

(d) Ward's method ($K = 2$)

Figure 8.3: Two dimensional representation (MDS) of FOOTBALL data set for different clusterings

As a result of the clustering validation index results for the the football players performance data set, Ward's method for $K = 4$ can be selected as the best one, since the calibration scenarios and some single criteria perform best with this selection. The graphical validation in Figure 8.3 also gives the impression that Ward's method for K = 4 is the one that looks most appropriate compared to the other cluster patterns in Figure 8.3. However, this clustering consideration is not an optimal

solution from the user's point of view, because having a large number of players in one cluster is not desirable to football squads and managers. On the other hand, the Dunn Index result, which is the only distinct clustering validation index for the choice of large number of clusters, can give us a direction to choose which clustering algorithm and what number of clusters. The definition of the Dunn Index is basically the ratio between separation and homogeneity, and the maximum of within cluster distances should be small, see Section 6.1.5 for more information. Complete linkage for $K = 145$ gives the best solution according to Figure 8.1e. Figure 8.4 provides a grouping structure of some famous players based on the solution of complete linkage for $K = 145$.



(a) Complete linkage ($K = 145$)

Figure 8.4: Two dimensional representation (MDS) of FOOTBALL data set with some famous players. Small gray points represent other players location on this MDS scatter plot. Although there are 145 cluster solutions, 13 cluster solutions are shown only for the famous players. The numbers in parenthesis with player's name represent the cluster number of that player to avoid confusion in case of the similarity between colors.

8.1 External Validation

Although complete linkage for $K = 145$ can be chosen as the optimal clustering, this consideration is not directly related to the similarity of football players information from the user's point of view. None of these criteria (except the Dunn Index) give us a strong differentiation between clusterings for a large number of clusters, because as demonstrated previously, the optimal number of clusters estimated by clustering validation criteria is small and may not be informative to football squads and managers. Therefore, one could maintain a clustering algorithm for one of the selections with a large number of clusters, say between $K = 100$ and $K = 150$, as this would give us clusters where the players within are similar enough to each other. While this is not a statistically optimal solution, it is a rather pragmatic approach. In this respect, I am also interested in validating the selection of different clustering solutions by considering some external information. The decisions will be informed by interviews with some football experts, who are especially experienced with analysis of football player performance, and by questioning people with different opinions who are familiar with this specific context.

I conducted a survey with seven questions based on different grouping structures of selected famous players as shown in Figure 8.4. This survey is conducted for clustering validation on football players performance data sets. Different clustering solutions correspond to the multiple choices in each question, and each selection is based on a different clustering algorithm(s) over various numbers of cluster solutions (e.g., PAM solution with $K \sim [100 : 150]$). This is done because the cluster points for these specific groups of players give the same clustering solution for different numbers of clusters. The respondents, who are football experts (e.g., football managers, journalists, scouts, etc.), answer each question by ranking different clustering solutions in order of importance from 1 to the number of multiple choices of that question (e.g., 5 for the first question), where 1 is the most appropriate, 2 is the second most appropriate, and so on. Appendices C presents the questions with the details of how this survey is conducted. As an illustration, the questions in Appendices C are responded by myself. I also present which clustering techniques and what number of clusters are used for the clustering solutions in each question, but this is not presented to the respondents.

As mentioned in Section 4.4, I have collaborated with The Istanbul Basaksehir football club to apply the distance query algorithm to find such specific players of their interest. At the same time, for the sake of clustering validation the survey questions were asked to 13 football experts including the head coach, the assistant coaches, the football analysts and the scouts of this club, and some Turkish journalists who are especially experienced with European football.

The ranking responses of the survey questions are systematized in such a way that I assign scores for each rank in each question, where the score assignment is made in a balanced way, because each question has a different number of multiple choices. Table 8.1 is the reference of how the given scores are assigned for each question with different selection of multiple choices.

Table 8.1: Score assignment for the survey questions

The selection of multiple choices	**1. Rank**	**2. Rank**	**3. Rank**	**4. Rank**	**5. Rank**
For 5 multiple choices	30	24	18	12	6
For 3 multiple choices	30	20	10	-	-
For 2 multiple choices	30	15	-	-	-

The result of the survey (Table 8.2) based on the questions from each respondent indicates that the PAM clustering algorithm for $K \sim [130:133, 137:146]$ has the highest total points, which implies the choice of this clustering is the best solution according to the responses of the experts.

Table 8.2: This survey is conducted on 13 football experts including the head coach, the assistant coaches, the football analysts and the scouts of Istanbul Basaksehir football club, and some Turkish journalists who are especially experienced with European football. The numbers are the total scores of the seven questions for different clustering selections from each participant.

Respondents	Selection 1 PAM ($K \sim [100:113]$)	Selection 2 PAM ($K \sim [114:118]$)	Selection 3 PAM ($K \sim [119:129, 134:136, 147:150]$)	Selection 4 PAM ($K \sim [130:133, 137:146]$)	Selection 5 Ward's method ($K \sim [100:147]$)	Selection 6 Ward's method ($K \sim [148:150]$)	Selection 7 Complete linkage ($K \sim [100:150]$)	Selection 8 Average linkage ($K \sim [100:150]$)
Head coach	138	138	162	162	148	160	109	125
Assistant coach - 1	138	138	144	144	144	166	109	137
Assistant coach - 2	125	115	127	137	109	121	136	134
Goalkeeping coach	148	118	130	160	152	176	109	125
Individual performance coach	166	136	148	178	146	152	109	119
Physical performance coach	159	149	119	129	125	137	116	168
Football Analyst	132	132	144	144	166	154	123	139
Chief Scout	176	166	166	176	134	128	117	155
Scout - 1	144	144	150	150	154	148	99	97
Scout - 2	113	143	155	125	133	145	142	168
Scout - 3	148	118	100	130	132	126	115	129
Journalist - 1	129	149	161	141	95	123	150	156
Journalist - 2	154	134	116	166	136	160	117	145
TOTAL	1870	1780	1822	1942	1774	1896	1531	1797

To evaluate whether the participants give consistent responses for the sake of validating their expertise, I also consider a testing procedure, which will examine the variation of the total sum of the scores. Efron and Tibshirani (1994, chap.16) described bootstrap methods that are directly designed for hypothesis testing, and one of these bootstrap testing procedures will be adopted here. Algorithm 4 gives the steps of how this testing procedure is established. The idea here is to investigate whether the variation of the total scores is significantly greater than the variation of the total scores that are randomly generated from the bootstrap methodology. This would imply that

the rater variation in the original data set is stronger than what could be explained by random variation alone. The bootstrap methodology formalises the situation in which if the total scores from different clustering solutions are most likely the same, then experts essentially give the random assessment. This is not a satisfying result, since no solution is favoured. On the other hand, if the variation is considerably large enough, then the result can be distinguished from the random result, which means that the expert responses are relatively different among each other.

Formally, we observe two independent samples: $\mathbf{z} = (z_1, \ldots, z_s)$ is the vector of sums obtained from the survey results according to the rules as explained in 8.1, and $\mathbf{y} = (y_1, \ldots, y_s)$ is the distribution of the sums from the same number of virtual respondents which is generated randomly according to the same rule. $\mathbf{z}$ and $\mathbf{y}$ are possibly drawn from two different probability distributions F and G,

$$\begin{aligned} F &\rightarrow \mathbf{z} = (z_1, \ldots, z_s) \\ G &\rightarrow \mathbf{y} = (y_1, \ldots, y_s)\,. \end{aligned}$$

Having observed $\mathbf{z}$ and $\mathbf{y}$, I wish to test the null hypothesis H_0 of no difference between F and G.

$$\begin{aligned} H_0 &: F = G, \\ H_A &: F \neq G. \end{aligned}$$

The test statistic is the variance of total scores as explained in Algorithm 4. Note that for the alternative hypothesis we are specifically interested in detecting cases for which $F \neq G$ in the sense that there is more concentration in the scores of F; in other words, the alternative is testing whether the total scores of the survey result, F have a higher variance than the total scores of the random one, G.

The test result gives us $\widehat{ASL}_{boot} = 0.048$, which is less than the critical value $\alpha = 0.05$ indicating that the best clustering, which is PAM clustering algorithm for $K \sim [130 : 133, 137 : 146]$, is significantly better than all other clustering selections.

One selection out of different numbers of clusters has to be chosen for the final solution. Since the aim is to find a small group of players in each cluster, the largest possible number of cluster solutions can be more useful for football managers or scouts. For this reason, PAM for $K = 146$ solution is chosen as the best selection based on the survey results from the football experts point of view, see the MDS solution in Figure 8.5 for a visualisation. Note that as mentioned in Section 4.4,

Algorithm 4: Computation of bootstrap test

input : $\mathcal{X}_{org} = \{x_{ij} | i = 1, \ldots, p; j = 1, \ldots, s\}$, where $p = 13$, $s = 8$, and x_{ij} is the total sums of the i^{th} participant and the j^{th} clustering selection obtained from seven questions

output: $\widehat{ASL}_{boot}$ (Achieved significance level obtained from the bootstrap samples)

STEP 1: Generate $B = 2000$ datasets from a model in which all participants randomly assign ranks to the original seven questions.

STEP 2: Compute the scores as explained in Table 8.1:

$$\mathcal{X}_{boot} = \{\mathcal{X}(1), \ldots, \mathcal{X}(B)\},$$
$$\mathcal{X}(b) = \{x(b)_{ij} | i = 1, \ldots, p;\ j = 1, \ldots, s;\ b = 1, \ldots, B\},$$

where $p = 13$, $s = 8$, and $x(b)_{ij}$ is the total sums score of the b^{th} bootstrap, the i^{th} participant and the j^{th} clustering selection obtained from seven question

STEP 3: Evaluate $T(\cdot)$ on each dataset, which is the variance of the total sums from different clustering selection

$$\begin{aligned} T(\mathcal{X}(b)) &= Var(\sum_{i=1}^{p} \mathbf{x}(b)_i) \\ &= Var(\sum_{i=1}^{p}\sum_{j=1}^{s} x(b)_{ij}) \\ &= \frac{1}{s-1}\left(\sum_{i=1}^{p} \mathbf{x}(b)_i - \frac{1}{ps}\sum_{i=1}^{p}\sum_{j=1}^{s} x(b)_{ij}\right) \end{aligned} \tag{8.2}$$

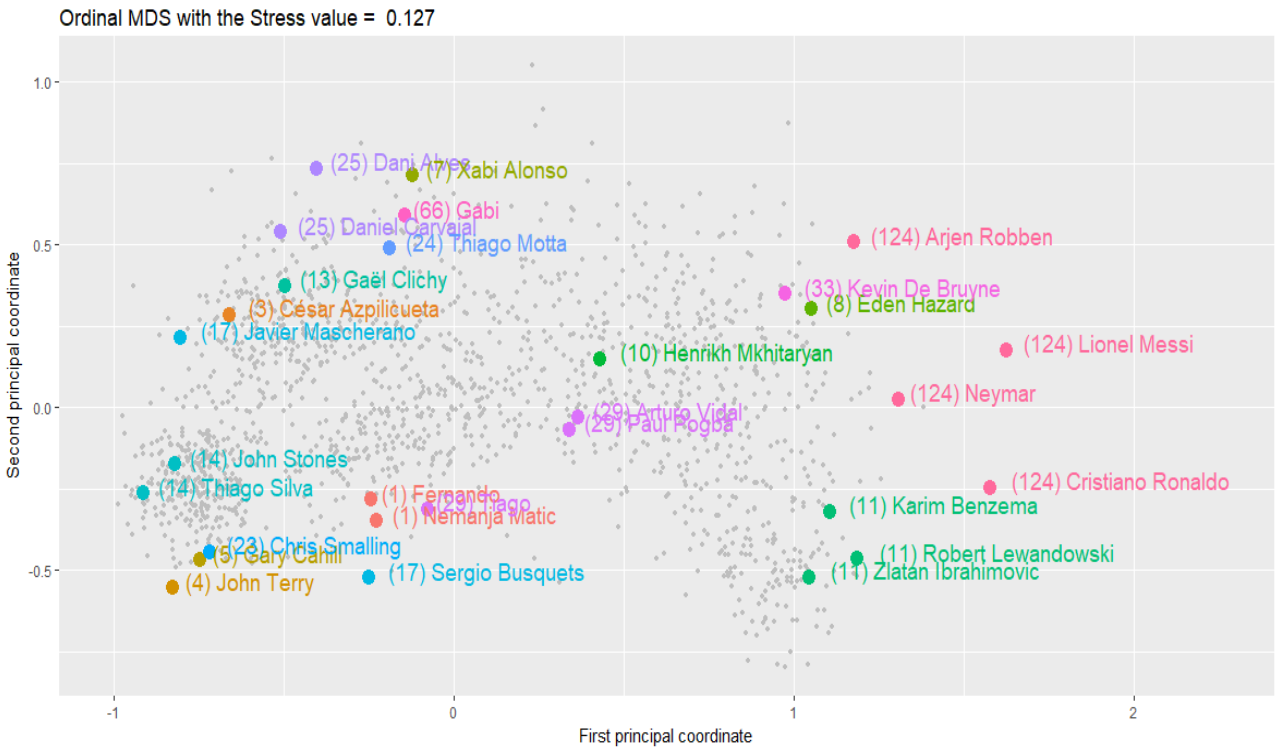
Also, evaluate the observed value, $T_{obs} = T(\mathcal{X}_{org})$, which is the variance of the total scores for the original dataset

$$\begin{aligned} T_{obs} &= Var(\sum_{i=1}^{p} \mathbf{x}_i) \\ &= \frac{1}{s-1}\left(\sum_{i=1}^{p} \mathbf{x}_i - \frac{1}{ps}\sum_{i=1}^{p}\sum_{j=1}^{s} x_{ij}\right) \end{aligned} \tag{8.3}$$

STEP 4: Approximate ASL_{boot} by $\widehat{ASL}_{boot} = \#\{T(\mathcal{X}(b)) \geq T_{obs}\}/B$.

return $\widehat{ASL}_{boot}$

a distance query is another alternative way of exploring such players of interest rather than using the clustering result from the user's point of view.



(a) PAM Algorithm ($K = 146$)

Figure 8.5: Two dimensional representation (MDS) of FOOTBALL data set with some famous players. Small gray points represent other players location on this MDS scatter plot. Although there are 146 cluster solutions, 18 cluster solutions are shown only for the famous players. The numbers in parenthesis with player's name represent the cluster number of that player to avoid confusion in case of the similarity between colors.

8.2 Comparisons of different clustering solutions

All the analytic decisions for football players performance data set can be summarised as follows. First, four different dissimilarity matrices are aggregated with range standardisation when log $(x + c)$ transformation is applied based on the sensitivity analysis results in Section 4.3. The four dissimilarities, which are obtained from the variables that have different football characteristics, are performance distance, position distance-1, position distance-2, and league and team distance. The final clustering solution is determined as PAM for $K = 146$ based on all different decisions and analysis as explained in Chapter 4 and 7. Now all these different selections from all different

analyses will be examined for the sake of comparing with the final clustering solution in order to check the stability of the information in the final solution. Table 8.3 presents the ARI values that are computed between the final solution and different decisions.

The PAM solution for $K = 146$ is applied on the four dissimilarities separately. They are compared with the aggregated dissimilarity matrix based on the clusterings of PAM for $K = 146$. In general, the overall clustering is not dominated by any of the individual clusterings obtained from four different dissimilarities, and these individual clusterings are quite different, so that they capture very different aspects of player dissimilarity. The ARI results indicate that although there is a slight positive relationship between the final dissimilarity matrix and the other four dissimilarities, substantial differences are detected based on small ARI values. Specifically, the ARI value between the final clustering solution and the clustering solution for league and team distance is very low, because league and team distance only depends on league and team information of players, and no player's individual information exists in these variables.

The second comparison in Table 8.3 is based on the choice of standardisation methodologies for aggregating four dissimilarities with or without performing variable transformation. The final clustering solution, PAM for $K = 146$ is again applied on different standardisation decisions with or without transformation separately. The results indicate that the Z-score with transformation has the lowest ARI value compared to the other standardisation choices, whereas all other selections give positive relationship with the higher ARI values than the Z-score with transformation. The highest ARI value is the range standardisation without variable transformation, which is reasonable because the range standardisation is also applied for the final clustering solution. On the other hand, the ARI value, 0.5222 is not as high as the clustering solutions from different clustering selections in the last column (3.Comparison) in Table 8.3, which indicates that variable transformation considerably affects the final clustering solution.

The third comparison relies on different clustering algorithms which are achieved by using the dissimilarity matrix that is obtained from the aggregation of four dissimilarities with range standardisation when $\log(x + c)$ transformation is applied. Complete linkage for $K = 145$, which is chosen based on the Dunn Index result, and two best selections from the survey results are determined for the sake of comparison with the PAM solution for $K = 146$. Complete linkage for $K = 145$ and Ward's method for $K = 149$ give the two lowest ARI values comparing with the other choices. The PAM solution for different numbers of clusters are highly related to the final clustering solution, which is reasonable because same clustering algorithm is used, and the ARI values are increasing when it gets closer to the number of clusters, $K = 146$. One of the interesting results is that Ward's method for $K = 148$ and $K = 150$ give very high ARI values compared to the other clustering solutions, although Ward's method is different than the PAM

algorithm. Since both algorithms are centroid-based clustering, high ARI values between these two clustering techniques makes sense from the clustering structural point of view.

The sensitiveness between different decisions and the final clustering solution based on PAM for $K = 146$ is examined. The ARI values are presented in Table 8.3 to analyse how sensitive all these decisions are. The interpretation of all these results are presented with the proper explanations above.

Table 8.3: PAM for $K = 146$ is applied for different choices of decisions. For the sake of comparison with all the decision below, the ARI values are computed between the clustering solution of PAM for $K = 146$ based on all different decisions and the PAM solution for $K = 146$ based on the final dissimilarity matrix that is achieved by the aggregation of four different dissimilarities by range standardisation with $log(x + c)$ transformation.

1.Comparison		*2.Comparison*		*3.Comparison*	
Dissimilarity measures	**ARI**	**Standardisation techniques**	**ARI**	**Clustering selections**	**ARI**
Performance distance	0.1572	AAD	0.4381	Complete linkage ($K = 145$)	0.2240
Position distance-1	0.1380	Z-score	0.2165	PAM ($K = 130$)	0.8631
Position distance-2	0.1387	AAD (NT)	0.4510	PAM ($K = 131$)	0.8640
League and team distance	0.0098	Z-score (NT)	0.4559	PAM ($K = 132$)	0.8711
		Range (NT)	0.5222	PAM ($K = 133$)	0.8793
				PAM ($K = 137$)	0.8846
				PAM ($K = 138$)	0.9003
				PAM ($K = 139$)	0.9075
				PAM ($K = 140$)	0.9024
				PAM ($K = 141$)	0.8977
				PAM ($K = 142$)	0.9120
				PAM ($K = 143$)	0.9978
				PAM ($K = 144$)	0.9988
				PAM ($K = 145$)	0.9995
				Ward's method ($K = 148$)	0.9585
				Ward's method ($K = 149$)	0.3536
				Ward's method ($K = 150$)	0.9983

NT (No transformation) is applied for the upper level variables when constructing the dissimilarity matrix.

CHAPTER 9

CONCLUDING REMARKS AND FUTURE RESEARCH DIRECTIONS

9.1 Concluding Remarks

This dissertation describes the design and development of a dissimilarity measure for mixed type data sets and the calibration of various clustering validation indexes for finding appropriate clustering methodology and numbers of clusters. Specifically in the first stage, a football players performance data set has been used for the aim of constructing a dissimilarity matrix in order for scouts and managers to be able to explore players of interest. In the second stage, the goal is determining how to choose a suitable cluster methodology with the appropriate number of clusters based on the dissimilarity measurements of football players. In this sense, clustering validation indexes have been used, and a new concept of aggregation of different indexes has been introduced and demonstrated with several results based on different types of data sets.

The key findings of the research are given as follows:

- In Section 3.2.2, I have discussed how variable transformation should be made when designing a dissimilarity measure for clustering. The choice of transformation and how this is applied are discussed from the interpretative dissimilarity point of view regarding the application manner of this thesis. Specifically, a guiding principle of the transformation selection can be the stabilisation of the variables between different seasons for the same player, see more discussion in Section 4.1.2.
- Several different standardisation techniques have been introduced in Section 3.2.3, and the usage of these techniques are discussed in terms of how they behave on differently dis-

tributed types of variables. In the application part of this thesis, average absolute deviation has been selected in order to aggregate different types of variables for the sake of making them comparable, because every value of the variables should have the same impact when computing a dissimilarity measurement based on subject matter reasons. For compositional data sets (percentages), the pooled average absolute deviation from all categories belonging to the same composition has been used. The reason for this is that a certain difference in percentages between two observations has the same meaning in all categories, which does not depend on the individual variance of the category variable.

- In Section 3.4, Compositional data analysis has been reviewed for the sake of representation of percentage variables in the football data set, and some conclusions are provided below:
  - For dealing with zero values in percentages, the Bayesian approach, which was used for adjusting these values, has been incorporated with my prior selection in order to represent football information in a more suitable way, see more discussion in Section 4.1.1.
  - Because the general principle is that differences in different variables should be treated in the same way, the resulting distances between percentages should be counted in the same way regardless of which part of the compositions they are from. In this sense, Theory 4.1.1 gives the formal argument with a mathematical proof of why either the Euclidean or the Manhattan distances can be considered as a selection of distance measure for percentage variables, but the Aitchison distance may not.
  - As discussed in Section 3.4.5, the Aitchison distance can be problematic for very small percentage values. Equation (3.28) demonstrates in mathematical form why the Aitchison distance is not adequate for this specific application, since the Aitchison distance is dominated by differences between small percentages in an inappropriate manner, see the example in Section 4.2.2 for a better illustration.
- For constructing a distance measure for two types of position variables, two different methodologies have been developed in Section 4.2.4.
  - A new type of dissimilarity measurement has been built for percentage variables (represented as $Y_{(15)}$). Specifically, I designed a three dimensional coordinate system in which a player's location is represented with one point on the football field, and then I computed the distance measures between these points, where the Great-circle distance has been incorporated.
  - For the other position variables ($Y_{(11)}$) which were represented as binary, a similar

distance measure to what Hennig and Hausdorf (2006) proposed was adapted for computing the distances between player information.

- In Section 4.3, several considerations have been given to aggregate four different types of dissimilarity measures in order to reach one single dissimilarity matrix for clustering players. In this respect, the first argument was how the distributional shape of the dissimilarities can be helpful for finding a proper standardisation approach in order to aggregate these dissimilarities. However, in the end a sensitivity analysis has been conducted for finding a suitable standardisation technique by observing the correlation of the vector of dissimilarities between two consecutive years information. The result indicates that range standardisation is the proper choice for aggregation of the dissimilarities.

- For choosing an appropriate clustering methodology and determining the best number of clusters, various clustering quality indexes have been described in Chapter 6. The main goal was to calibrate several clustering validation indexes with different aims. Following this, Hennig (2017) proposed two random clustering algorithms (Random K-centroids and Random nearest neighbour) to generate random clustering validation index values for the sake of calibrating these indexes. I proposed two additional random clustering algorithms (Random furthest neighbours and Random average neighbours) to contribute different distributional shapes for calibration, see the whole discussion in Section 6.2.

- For stability measurements, Tibshirani and Walther (2005) and Fang and Wang (2012) estimated the number of clusters using the idea of cluster stability by resampling methods. Both articles used the K-means algorithm for clustering, and the closest centroid approach for classification. The `R` package `fpc` (Hennig, 2013) provides alternative classification techniques over different clustering algorithms. In this thesis, I contribute one additional classification approach (furthest neighbour distance) for the complete linkage method. More discussion can be found in Section 6.1.6.

- In Chapter 7, calibrating different clustering validation indexes has been examined with different types of simulated and real data sets, and the results indicate that the calibration of clustering validation indexes is quite successful in the estimation of the correct number of clusters in most circumstances.

- For the dissimilarity measure obtained from the football data set, the clustering validation index results favour small numbers of clusters, which is not satisfactory from the user's point of view for subject-matter reasons. In this sense, a survey, involving different clustering solutions (for large K) in each question, has been conducted among football experts to select

an appropriate clustering methodology with the suitable number of clusters. This is a clustering validation step by checking external information obtained from the survey result, see more details in Section 8.

- In Section 4.4, a distance queries approach has been introduced for exploring a player of interest based on a final dissimilarity matrix obtained from the distance construction of football players performance data. R Shiny implementation has been used for the visualisation of distance queries of players with the aim to find players that have the smallest distances to a player of interest.

9.2 Future Research Directions

For the research conducted in the development of this thesis, a large amount of considerations and literature regarding distance construction and cluster analysis, specifically estimating the number of clusters using clustering validation techniques, has been reviewed. Several new contributions have been made. It is anticipated that there is significant scope for further progress in this field. To expedite this progress, the following recommendations with respect to future work are given:

- In Chapter 4, I discussed how to pre-process the football data set and how to build a dissimilarity measure in terms of reflecting players' characteristics by using all the available information in the data set. The data information was collected for the of $2014-2015$ football seasons, so that distance construction for the football data set has only been made for one year of information, which simply implies that no time series component is involved. Data information collected from sports events may not be stable and could be change year by year. In this respect, one could be interested in investigating football data information in different years and make some connections between these years. From the statistical point of view, this could be done by time series analysis. Time series analysis can be useful not only for making the connections between players in consecutive years, but also for discovering potential talented football players by checking their trends from the first year to the last year of available information. The first step for performing time series analysis is to collect historical player information, which is very challenging due to the difficulties of data collection from the specific websites (e.g., www.whoscored.com). Once this issue is resolved, another challenge is how to apply time series analysis in distance construction for analysis of mixed data types as well as time series data in cluster analysis. In the literature, time series analysis of a dissimilarity matrix is commonly studied with the aim of clustering, and there seems to be an increased interest in time series clustering in the recent years. Many different articles

exist in this specific area. For having a general idea about this topic, one could consider Liao (2005), which is a survey article summarising previous works that investigated the clustering of time series data in various application domains. I intend to work on these sorts of studies as a new research topic for the future, specifically from the direction of this thesis' specific application.

- In Section 3.2.4, the concept of weighting is described as multiplying variables with an appropriate constant which is determined by user's judgement. This kind of judgement is subjective and can be made from different perspectives. As discussed previously, the statistical approaches (e.g., PCA or some of the feature selection methodologies) for defining variable weights may not be suitable for dissimilarity design and clustering of the football data set, because the issue here involves subject-matter knowledge that cannot be decided from the data alone. Therefore, I gave different arguments from a subjective point of view that lead to assigning appropriate weights to the variables of the football data set, see Section 4.1.4. In future work, the determination of weight assignment for these types of football variables can be investigated from different settings, and the inclusion of external information (e.g., expert knowledge) can be a reasonable way of assigning appropriate weights. Multi criteria decision method (MCDM) [1] is one way to determine variable weights by rating the criteria (e.g., variables) based on experts' responses for how well it satisfies a particular interest or which criteria are more important from the decision makers' point of view. The idea is similar to the score assignment as explained in Section 8.1. Various schools of thought have developed for solving MCDM problems, see the book Hwang and Masud (2012) for more details.

- Football players in different leagues and teams have very different levels in terms of their performance, skills, physical conditions, and other such features. For example, two players who have very similar statistical data information from various football performance variables can be selected for checking how similar they are, but these two players can play in two different leagues and two different teams (say Barcelona, Spain and Galatasaray, Turkey). We expect that the dissimilarity between these two players should be small based on the information obtained from their performance and position variables, but some differences can be expected due to the effect of team and league variables. In the data pre-processing steps in Chapter 4, team and league variables (x_l, x_{tp} and x_{tc}) are incorporated into the dissimilarity matrix in an independent way; in other words, these variables are not linked with any other variables in the football data set. One could think of integrating team and league variables into all the other variables in such a way that the dissimilarity between football players can be interpreted in a

[1] Multiple criteria decision making is one of techniques in operational research that deals with finding optimal results in complex scenarios including different indicators.

more intelligent way. For example, the company **InStat**, which is a website platform storing a huge football database and match videos, created an index used to evaluate player's performances in matches during the entire season. **InStat Indexes** are designed in such a way that team and league information are linked with the other performance variables when calculating the player index value; in other words, as explained in the company website, it correlates with the tournament level, but at the same time does not depend on it directly. For example, there might be a higher index value for a player in Turkish League than a player in Spanish League. More information about the calculation of **The InStat Index** can be found in the web-link, http://instatsport.com/en/football-3/instat-index-en/. This concept can be reviewed and developed with different thoughts and viewpoints from both a statistical and football point of view, which is planned to be studied more in the future.

- In Section 6.1.6, two stability measurements have been introduced for estimating the number of clusters, and the stability assessment for clusters is measured by some resampling techniques. For these types of settings, various classification methodologies are proposed for the sake of measuring stability in clusters. The choice of integration between different classification techniques and different clustering methodologists is determined simply by considering the connection of these techniques based on their structures (e.g., furthest neighbour distance is related to complete linkage). The argument of why the selections are made is based on the explanations obtained from different sources (See Tibshirani and Walther (2005), Hennig (2013) and Fang and Wang (2012)). On the other hand, one could consider a theoretical proof for the selection of the classification methodologies in Table 6.1. Wang (2010) provided the proof of a consistency theorem for centroid and nearest neighbour assignments. In future work, I plan to contribute a similar idea of consistency theorem for the other classification methodologies, such as furthest neighbour distance and average distance.

Appendices

APPENDIX A

Algorithms

A.1 K-means algorithm

Algorithm 5: The K-means Algorithm (Lloyd, 1982)

input : $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ (set of objects to be clustered), K (number of clusters), $iter.max$ (the maximum number of iterations)

output: $\mathbf{m}_C = (\mathbf{m}_{C_1}, \mathbf{m}_{C_2}, \ldots, \mathbf{m}_{C_K})$ (set of cluster centroids), $\mathcal{L} = \{l_1, l_2, \ldots, l_n\}$ (set of cluster labels)

 # initialise K random centroids, $\boldsymbol{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_K\}$ from $\mathcal{X}$

for $k \leftarrow 1$ **to** K **do**
    $\mathbf{m}_{C_k} \leftarrow \mathbf{s}_k$

$iter \leftarrow 0$

while $iter \leq iter.max$ **do**
    for $i \leftarrow 1$ **to** n **do**
        # Assign every observations to the closest centroid:
        $l_i = \underset{1 \leq k \leq K}{\arg\min} \|\mathbf{x}_i - \mathbf{m}_{C_k}\|, i \in N_n$
    for $k \leftarrow 1$ **to** K **do**
        # Compute the means by minimising Equation (5.1)
        $\mathbf{m}_{C_k} = \frac{1}{n_k} \sum_{l_i = k} \mathbf{x}_i$
    $iter \leftarrow iter + 1$

return $\mathbf{m}_C$ *and* $\mathcal{L}$

A.2 PAM algorithm

Algorithm 6: Partitioning Around Medoids (PAM) Algorithm

input : $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ (set of objects for computing $\mathbf{D} = d(\mathbf{x}_i, \mathbf{x}_j)$ $(i, j = 1, \ldots, n)$, the matrix of dissimilarity to be clustered) or $\mathbf{D}$ (matrix of dissimilarity to be clustered), K (number of clusters)

output: $\mathbf{m}_C = (\mathbf{m}_{C_1}, \mathbf{m}_{C_2}, \ldots, \mathbf{m}_{C_K})$ (set of cluster medoids), $\mathcal{L} = \{l_1, l_2, \ldots, l_n\}$ (set of cluster labels)

STEP 0: *# initialise K random medoids, $\boldsymbol{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_K\}$ from $\mathcal{X}$*

for $k \leftarrow 1$ **to** K **do**
  $\mathbf{m}_{C_k} \leftarrow \mathbf{s}_k$

$q \leftarrow 1$

repeat

  STEP 1:

  for $i \leftarrow 1$ **to** n **do**

    # Assign every observations to the closest medoid:

    $l_i = \underset{1 \le k \le K}{\arg\min}\, d(\mathbf{x}_i, \mathbf{m}_{C_k})$, $i \in N_n$

  # Compute the total cost, $\mathcal{T}^{(0)} = \mathcal{T}(\mathcal{C}, \mathbf{m}_1, \ldots, \mathbf{m}_K)$ by using Equation (5.3),

  STEP 2: $(\mathbf{m}_1^*, \ldots, \mathbf{m}_K^*) = (\mathbf{m}_1^{IK}, \ldots, \mathbf{m}_K^{IK})$

  for $i \leftarrow 1$ **to** n **do**

    for $k \leftarrow 1$ **to** K **do**

      $\mathbf{x}_i \notin \{\mathbf{m}_1^*, \ldots, \mathbf{m}_K^*\}$ and $\mathbf{m}_k^* \in \{\mathbf{m}_1^*, \ldots, \mathbf{m}_K^*\}$

      # Compute $\mathcal{T}_{ik} = \mathcal{T}(\mathcal{C}^{ik}, \mathbf{m}_1^{ik}, \ldots, \mathbf{m}_K^{ik})$, where $(\mathbf{m}_1^{ik}, \ldots, \mathbf{m}_K^{ik})$ are $(\mathbf{m}_1^*, \ldots, \mathbf{m}_K^*)$ but with $\mathbf{m}_k^*$ replaced by $\mathbf{x}_i$, and $\mathcal{C}^{ik}$ assigns every object to the closest centroid in $\{\mathbf{m}_1^*, \ldots, \mathbf{m}_K^*\}$

  STEP 3: $(g, h) = \underset{(i,k)}{\arg\min}\, \mathcal{T}_{ik}$, $\mathcal{T}^{(q)} = \mathcal{T}_{gh}$

    $(\mathbf{m}_{C_1}, \ldots, \mathbf{m}_{C_K}) = (\mathbf{m}_1^{gh}, \ldots, \mathbf{m}_K^{gh})$

    $q \leftarrow q + 1$

until $\mathcal{T}^{(q)} \ge \mathcal{T}^{(q-1)}$

return $\mathbf{m}_C$ *and* $\mathcal{L}$

A.3 Hierarchical clustering

Algorithm 7: Agglomerative Hierarchical Clustering Algorithm

input : $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ (set of objects for computing $\mathbf{D} = d(\mathbf{x}_i, \mathbf{x}_j)$ $(i, j = 1, \ldots, n)$, the matrix of dissimilarity to be clustered) or $\mathbf{D}$ (matrix of dissimilarity to be clustered).

output: List of which clusters are merged at each step, which depends on the number of cluster or height of the dissimilarity for each merge.

STEP 0: Every object is a cluster on its own:
$\{C_1, \ldots, C_n\} \leftarrow \{\{1\}, \ldots, \{n\}\}$
Initialise set of clusters available for merging: $S \leftarrow \{1, \ldots, n\}$
$t \leftarrow 1$; $\mathbf{D}^{(t)} \leftarrow \mathbf{D}$

repeat

STEP 1: Find two most similar clusters (by checking the smallest dissimilarity from $\mathbf{D}^{(t)}$) to merge

$$(g, h) \leftarrow \arg\min_{g,h \in S} d^{(t)}(\mathbf{x}_g, \mathbf{x}_h)$$

STEP 2: Merge clusters C_g, C_h: $C_l \leftarrow C_g \cup C_h$
Mark g and h as unavailable: $S \leftarrow S \setminus \{g, h\}$

STEP 3:

if $C_l \neq \{1, \ldots, n\}$ **then**
Mark l as available: $S \leftarrow S \cup \{l\}$

STEP 4:

foreach $i \in S$ **do**
Update dissimilarity matrix $d^{(t)}(\mathbf{x}_i, \mathbf{x}_l)$, if:
Single linkage: $\min\left\{d^{(t)}(\mathbf{x}_i, \mathbf{x}_g), d^{(t)}(\mathbf{x}_i, \mathbf{x}_h)\right\}$,
Complete linkage: $\max\left\{d^{(t)}(\mathbf{x}_i, \mathbf{x}_g), d^{(t)}(\mathbf{x}_i, \mathbf{x}_h)\right\}$,
Average linkage:
$(n_i n_g d^{(t)}(\mathbf{x}_i, \mathbf{x}_g) + n_i n_h d^{(t)}(\mathbf{x}_i, \mathbf{x}_h))/(n_i n_l)$,
where n_i, n_l, n_g and n_h are the numbers of elements in clusters C_i, C_l, C_g and C_h respectively.

$t \leftarrow t + 1$

until *No more clusters are available for merging*

A.4 Spectral clustering

Algorithm 8: Spectral clustering algorithm

input : $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ (set of objects) or $\mathbf{D} = d(\mathbf{x}_i, \mathbf{x}_j)$ (matrix of (dis)similarity), K (number of clusters)

output: $\mathbf{m}_C = (\mathbf{m}_{C_1}, \mathbf{m}_{C_2}, \ldots, \mathbf{m}_{C_K})$ (set of cluster centroids), $\mathcal{L} = \{l_1, l_2, \ldots, l_n\}$ (set of cluster labels)

STEP 1: As described in Definition 5.2.4, define the weighted adjacency matrix (A), using a function (e.g., Gaussian Kernel) as explained previously, then compute the degree matrix P

STEP 2: Using the matrices A and P, construct the Laplacian matrix (L) from one of the choices in Table 5.1

STEP 3: Compute the largest K eigenvalues $\lambda_1 \geq \ldots \geq \lambda_K$ and corresponding eigenvectors $\mathbf{U} = \{\mathbf{u}_1, \ldots \mathbf{u}_K\}$ of L, where $\mathbf{U}$ is a matrix, and $\mathbf{u}_i$'s are the columns of $\mathbf{U}$.

STEP 4: Apply the K-means algorithm on $\mathbf{U}$ by using Algorithm 5.

return $\mathbf{m}_C$ *and* $\mathcal{L}$

A.5 Classical scaling algorithm

Algorithm 9: The classical scaling algorithm

input : $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ (set of objects for computing $\mathbf{D} = (d_{ij}) = d(\mathbf{x}_i, \mathbf{x}_j)$ $(i, j = 1, \ldots, n)$, the dissimilarity matrix) or $\mathbf{D}$ (the dissimilarity matrix).

output: $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_t)$ (the principal coordinates)

STEP 1: Square the dissimilarity matrix:

$\mathbf{A} = (a_{ij})$, where $a_{ij} = -\frac{1}{2} d_{ij}^2$

STEP 2: Form the "doubled centred" symmetric matrix:

$\mathbf{B} = \mathbf{HAH}$, where $\mathbf{H} = \mathbf{I}_n - \frac{1}{n}\mathbf{J}_n$ and $\mathbf{J}_n = \mathbf{1}_n \mathbf{1}_n^T$ is an $(n \times n)$-matrix of ones.

STEP 3: Compute eigenvalues and eigenvectors of $\mathbf{B}$:

$\mathbf{B} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{\Lambda} = \text{diag}\,(\lambda_1, \ldots, \lambda_n)$ is the diagonal matrix, and $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_n)$ is a $n \times n$ orthogonal matrix containing the eigenvectors.

STEP 4: Take the first t $(t < n)$ eigenvalues of $\mathbf{B}$ greater than 0, where $\mathbf{\Lambda}_+ = \text{diag}\,(\lambda_1, \ldots, \lambda_t)$ and the remaining $n - t$ will be zero, and let $\mathbf{U}_+ = (\mathbf{u}_1, \ldots, \mathbf{u}_t)$ be the corresponding first t eigenvectors of $\mathbf{B}$. Then,

$$\mathbf{B} = \mathbf{U}_+ \mathbf{\Lambda}_+ \mathbf{U}_+^T = \left(\mathbf{U}_+ {\mathbf{\Lambda}_+}^{1/2}\right)\left({\mathbf{\Lambda}_+}^{1/2} \mathbf{U}_+\right) = \mathbf{Z}\mathbf{Z}^T$$

where $\mathbf{Z} = \mathbf{U}_+ {\mathbf{\Lambda}_+}^{1/2} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)^T$ are the principal coordinates of the $t \times n$ matrix $\mathbf{Z}^T$.

return Z

A.6 Distance scaling algorithm

Table A.1 shows the estimation of disparities ($\hat{d}^*_{ij}$) by isotonic regression for an artificial data set $\mathbf{X}$ with $n = 6$ objects and $m = 15$ dissimilarities. The column of 'pair' provides the names of pairwise objects and the rank information is obtained by the rank order of the 15 dissimilarities (d_{ij}). d^*_{ij} is the Euclidean distance between the configuration points [1] estimated by the MDS solution. The columns I, II, III, IV, V, VI display a sequence of trial solutions for the disparities and the cells in red indicate the active block at each trial solution. A "block" is a consecutive set of dissimilarities that have to be set equal to each other to maintain monotonicity. We partition the estimated dissimilarities into blocks, and at each step of the algorithm one of these blocks becomes "active". A trial solution consists of averaging the values within the active block.

Table A.1: Artificial example for isotonic regression

Pair	d_{ij}	Rank of d_{ij}	d^*_{ij}	I	II	III	IV	V	VI	$\hat{d}^*_{ij}$
A-B	2.1	1	2.3	2.3	2.3	2.30	2.30	2.30	2.30	2.30
A-C	2.4	2	2.7	2.7	2.7	2.70	2.70	2.70	2.70	2.70
A-D	4.7	3	8.1	8.1	6.9	6.67	6.67	6.67	6.67	6.67
A-E	4.9	4	5.7	5.7	6.9	6.67	6.67	6.67	6.67	6.67
A-F	5.8	5	6.2	6.2	6.2	6.67	6.67	6.67	6.67	6.67
B-C	7.5	6	8.1	8.1	8.1	8.10	8.13	7.80	7.80	7.80
B-D	8.5	7	8.6	8.6	8.6	8.60	8.13	7.80	7.80	7.80
B-E	8.9	8	7.7	7.7	7.7	7.70	8.13	7.80	7.80	7.80
B-F	9.1	9	6.8	6.8	6.8	6.80	6.80	7.80	7.80	7.80
C-D	9.9	10	9.3	9.3	9.3	9.30	9.30	9.30	9.30	9.30
C-E	10.4	11	10.5	10.5	10.5	10.50	10.50	10.50	10.15	10.10
C-F	10.6	12	9.8	9.8	9.8	9.80	9.80	9.80	10.15	10.10
D-E	10.8	13	10.0	10.0	10	10.00	10.00	10.00	10.00	10.10
D-F	12.5	14	12.6	12.6	12.6	12.60	12.60	12.60	12.60	12.60
E-F	12.7	15	12.8	12.8	12.8	12.80	12.80	12.80	12.80	12.60

We can see that the first three d^*_{ij}'s are increasing (2.3, 2.7, 8.1) from the second column of Table A.1. The next distance (5.7) is smaller only than the preceding 8.1, so the active block is (8.1, 5.7) with an average of 6.9. The next distance 6.2 is smaller than the two previous 6.9s, so

[1] The configuration points are the principal coordinates, which can be obtained by the classical scaling method as a starting phase of non-metric distance scaling algorithm. Starting configurations can be determined by either randomly or using the classical MDS as implemented in the `smacof` package of `R` (De Leeuw and Mair, 2011).

the active block is (6.9, 6.9, 6.2), whose values are averaged to get 6.67. The two distances (8.1, 8.6) are increasing, but the next one (7.7) is smaller than the preceding two distances. The active block is now (8.1, 8.6, 7.7) with an average value of 8.13. The next distance (6.8) is smaller than the three 8.13s, so the active block is (8.13, 8.13, 8.13, 6.80), and their average value is 7.80. The next two distances (9.3, 10.5) are increasing, but 9.8 is smaller than 10.5. Hence, we average the two distances (10.5, 9.8) to get 10.15. The next distance 10.0 is smaller than the two 10.15s, so we average the three values to get 10.1. The remaining distances satisfy the monotonicity requirement, and the algorithm stops.

The implementation of isotonic regression is a part of the non-metric distance scaling procedure that is presented in Algorithm 10. The stress function for non-metric MDS is a bit different than the stress function of classical MDS, see Equation (5.22). As shown in Equation (A.1), the loss function is only estimated for the distance between the configuration points (d^*_{ij}) rather than using the dissimilarities (d_{ij}).

$$\text{stress} = \mathcal{S}(d^*_{ij}) = \left(\frac{\sum_{i<j} \left(d^*_{ij} - \hat{d}^*_{ij}) \right)^2}{\sum_{i<j} (d^*_{ij})^2} \right)^{1/2} . \tag{A.1}$$

From the last column of Table A.1, the disparities ($\hat{d}^*_{ij}$) are approximated by using a step-like function, see the left panel of Figure A.1. As an alternative to isotonic regression, Ramsay (1988) described a monotone spline (non-linear) transformation, which is smoother than a step function while preserving the non-decreasing property. The basic illustration of monotone spline can be seen on the right panel of the Figure A.1, while the conceptual idea is quite sophisticated so that the transformation from dissimilarities into disparities cannot be characterised by one simple function. Splines are piecewise polynomial functions and the pieces are determined by two additional parameters: interior knots and the spline degree. I will not get into the details about the mathematical explanation of monotone splines, but to those with a particular interest in splines, I recommend two general references: De Boor et al. (1978) and (Schumaker, 2007).

In Algorithm 10, the gradient search algorithm is a first-order iterative optimization algorithm to find a local minimum of a function, in which the configuration in a direction is determined by the partial derivatives of $S^{[r]}$ with respect to $\mathbf{z}$. Here $\mathbf{z} = (z_{11}, \ldots, z_{1t}, \ldots, z_{nt})$ is a vector form of $\mathbf{Z}$. Hence, given the configuration $\mathbf{z}^{[r]}$ at the r^{th} iteration, an updated configuration at the next iteration is calculated as

$$\mathbf{z}^{[r+1]} = \mathbf{z}^{[r]} - a_{r+1}\mathbf{v}, \tag{A.2}$$

Algorithm 10: Non-metric distance scaling algorithm

input : $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ (set of objects for computing $\mathbf{D} = (d_{ij}) = d(\mathbf{x}_i, \mathbf{x}_j)$ $(i, j = 1, \ldots, n)$, the dissimilarity matrix) or $\mathbf{D}$ (the dissimilarity matrix), t (number of dimensions), r_{max} (maximum number of iterations), and ϵ (convergence criterion).

output: $\mathbf{Z}^{[r]} = (\mathbf{z}_1, \ldots, \mathbf{z}_t)$ (the configuration points), and $S^{[r]}(d^*_{ij})$ (the stress value).

STEP 1: Order the $m = \frac{1}{2}n(n-1)$ dissimilarities (d_{ij}) from smallest to largest: $d_{i_1 j_1} \leq d_{i_2 j_2} \leq \ldots \leq d_{i_m j_m}$

STEP 2: Choose an initial configuration of points $\mathbf{z}_i \in \mathbb{R}^t$ $i = 1, \ldots, n$ with the fixed number of t dimensions.

$r \leftarrow 0$ and $S^{[r]}(d^*_{ij}) \leftarrow 0$

while $r \leq r_{max}$ *and* $\Delta < \epsilon$ **do**

STEP 3: Compute the Euclidean distances $\left(d^*_{ij}\right)$ between the initial configuration points ($\mathbf{Z}^{[r]}$)

$$d^*_{ij} = \|\mathbf{z}_i - \mathbf{z}_j\| = \left\{(\mathbf{z}_i - \mathbf{z}_j)^T (\mathbf{z}_i - \mathbf{z}_j)\right\}^{1/2}$$

STEP 4: Produce fitted values (disparities) $(\hat{d}^*_{ij})$ by using an isotonic regression algorithm

$$\hat{d}^*_{i_1 j_1} \leq \hat{d}^*_{i_2 j_2} < \ldots \leq \hat{d}^*_{i_m j_m}$$

STEP 5: Compute the stress value, see Equation (5.22) and Δ is given by:

$$\Delta = S^{[r+1]}(d^*_{ij}) - S^{[r]}(d^*_{ij})$$

STEP 6: Change the configuration points $\mathbf{z}_i$'s by applying an iterative gradient search algorithm (method of steepest descent): $\mathbf{Z}^{[r+1]} \leftarrow \mathbf{Z}^{[r]}$

$r \leftarrow r + 1$

return $\mathbf{Z}^{[r]}$ *and* $S^{[r]}(d_{ij})$

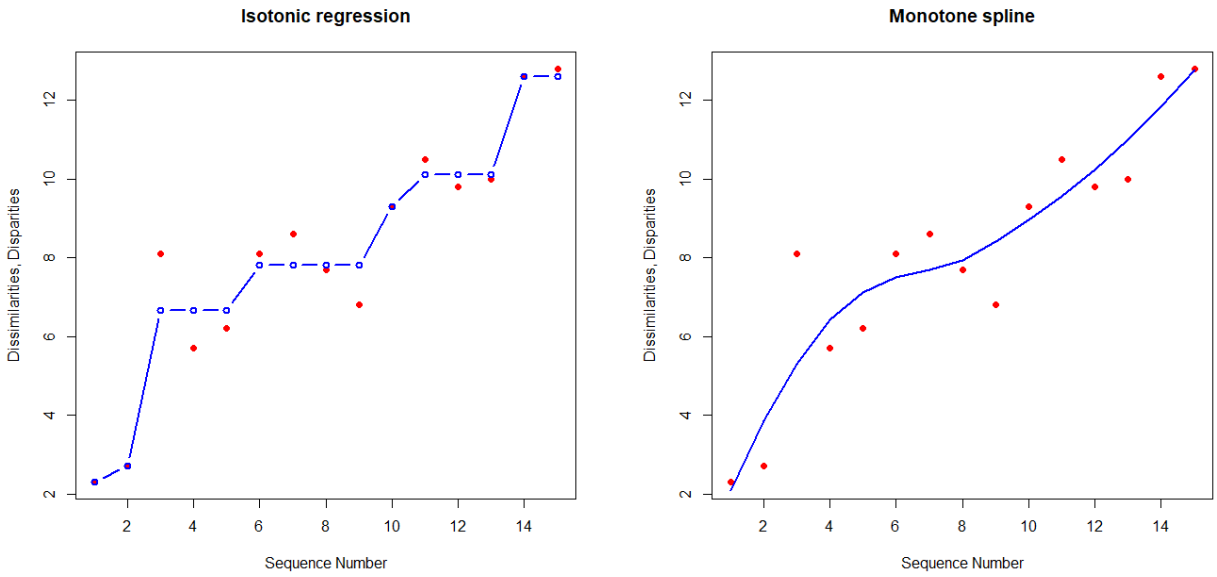


Figure A.1: Diagrams for the artificial example

where a_{r+1} is the step size constant that is updated at each iteration to accelerate the algorithm. $\mathbf{v}$ is the (standardised) gradient function is given by:

$$\mathbf{v} = \frac{\partial S}{\partial \mathbf{z}} \Big/ \left| \frac{\partial S}{\partial \mathbf{z}} \right| . \tag{A.3}$$

Kruskal (1964b) provided the explicit formula of $\mathbf{v}$ and suggested that the initialisation constant (a_0) can be 0.2, see also Cox and Cox (2000) for more details.

APPENDIX B

R Shiny implementations

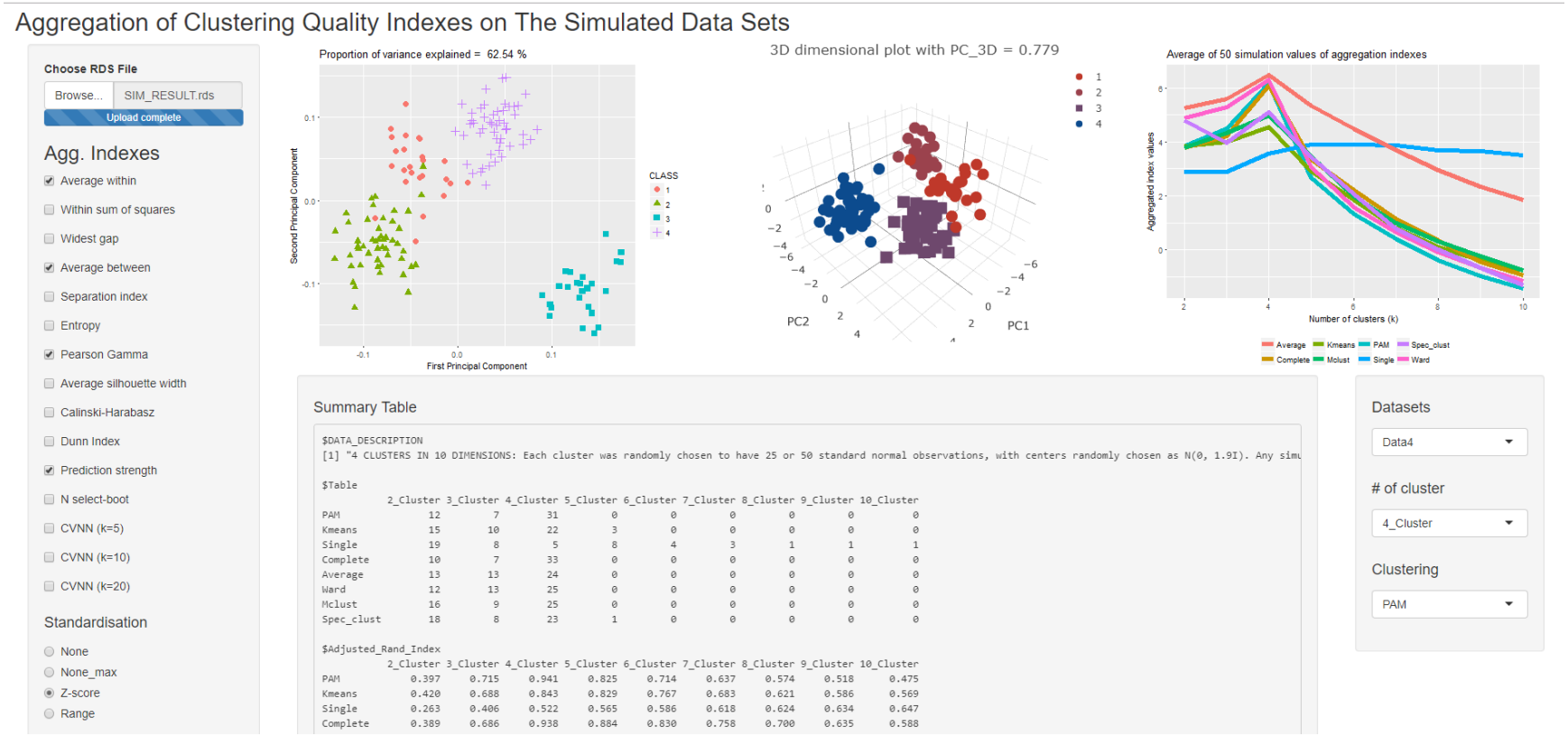


Figure B.1: R Shiny implementation for aggregation of clustering quality indexes on the simulated data set

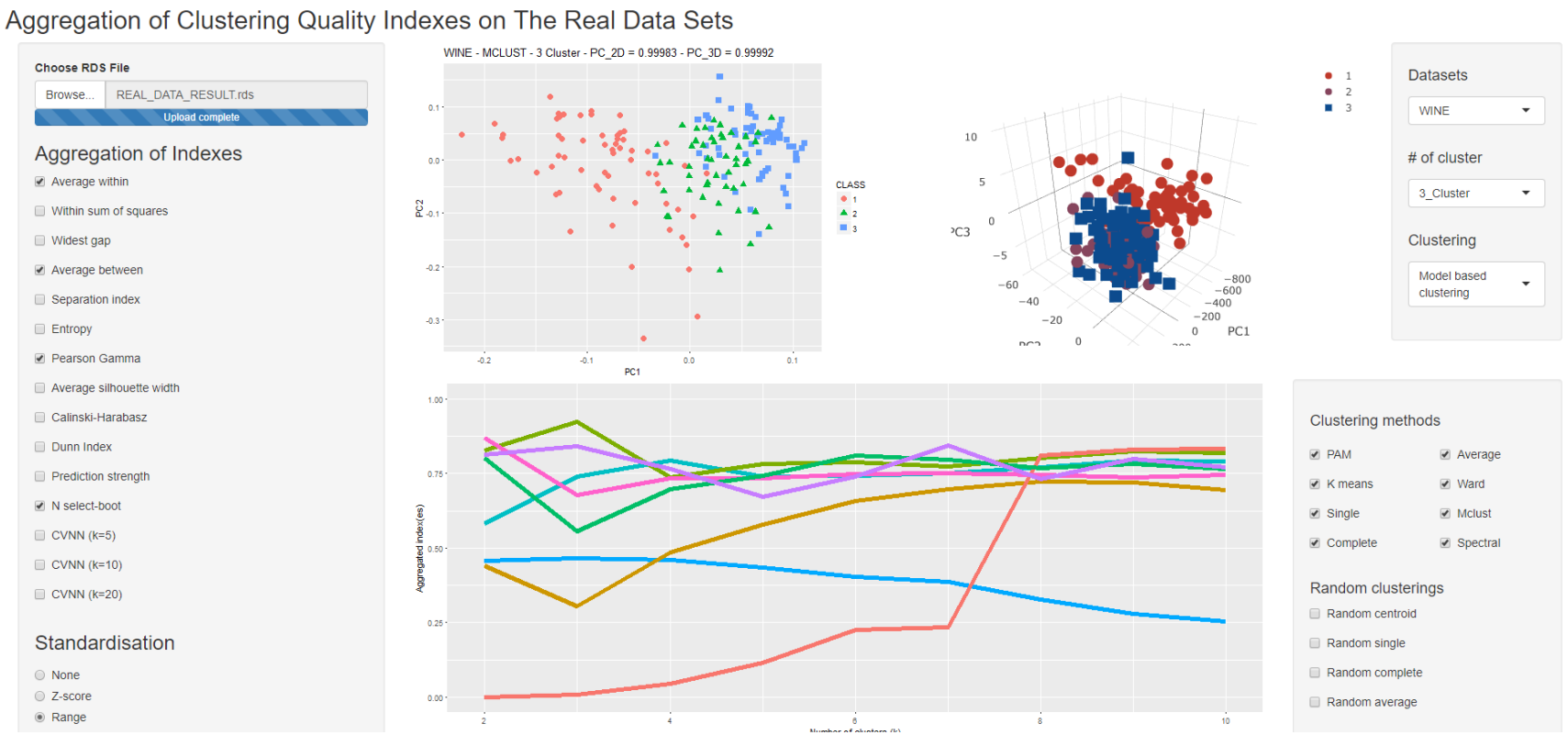


Figure B.2: R Shiny implementation for aggregation of clustering quality indexes on the real data sets

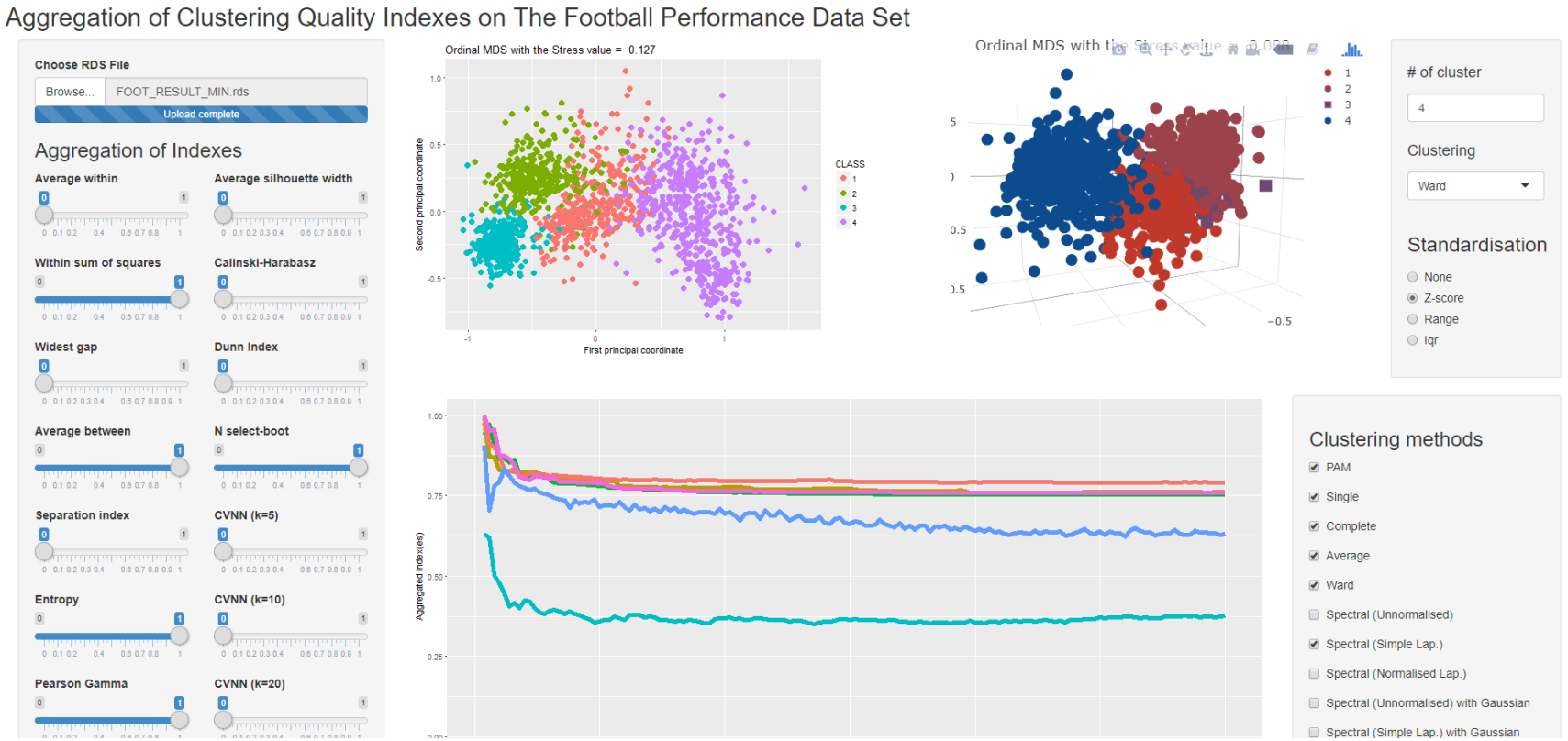


Figure B.3: R Shiny implementation for aggregation of clustering quality indexes on the football data sets

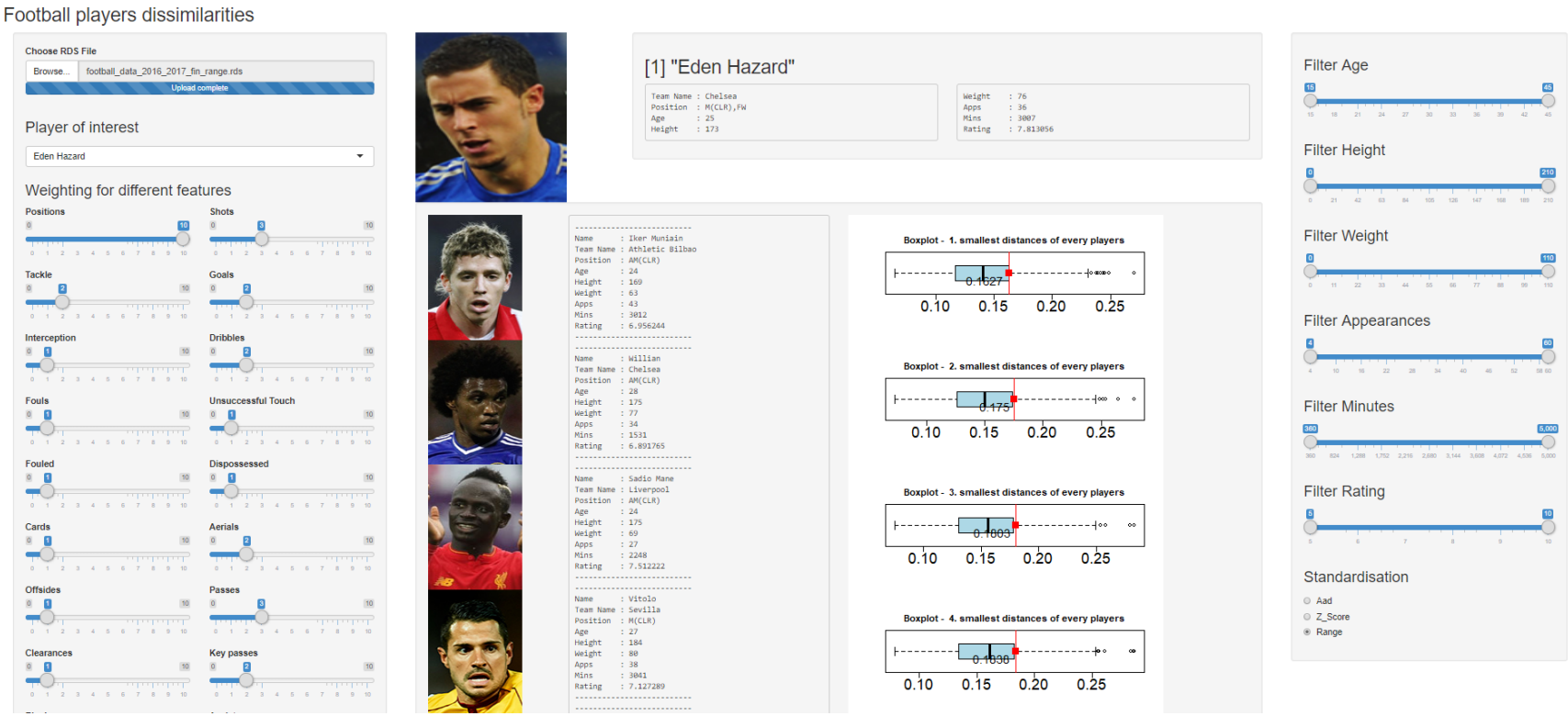


Figure B.4: R Shiny implementation for distance queries of football players

APPENDIX C

Survey for clustering solutions of football players performance data set

Table C.1: Question 1: This group of players are centre-defenders. Please rank the following in order of importance from 1 to 5 where 1 is the most appropriate to you and 5 is the least appropriate to you.

Clustering solutions	1. Group	2. Group	3. Group	4. Group	Rank
Selection 1 PAM ($K \sim [100:150]$)	John Terry	Gary Cahill	Chris Smalling	John Stones Thiago Silva	5
Selection 2 Ward's method ($K \sim [100:147]$)	John Terry Gary Cahill John Stones Thiago Silva	Chris Smalling	—	—	3
Selection 3 Ward's method ($K \sim [148:150]$)	John Terry Gary Cahill Thiago Silva	Chris Smalling	John Stones	—	1
Selection 4 Complete linkage ($K \sim [100:150]$)	John Terry Gary Cahill Thiago Silva	Chris Smalling John Stones	—	—	2
Selection 5 Average linkage ($K \sim [100:150]$)	John Terry Gary Cahill	Thiago Silva Chris Smalling John Stones	—	—	4

Table C.2: Question 2: This group of players are right or left defenders. Please rank the following in order of importance from 1 to 2 where 1 is the most appropriate to you and 2 is the least appropriate to you.

Clustering solutions	1. Group	2. Group	3. Group	Rank
Selection 1 PAM and Ward's method ($K \sim [100:150]$)	Cesar Azpilicueta	Gael Clichy	Dani Alves Daniel Carvajal	2
Selection 2 Complete and average linkage ($K \sim [100:150]$)	Cesar Azpilicueta Gael Clichy	Dani Alves Daniel Carvajal	—	1

Table C.3: Question 3: This group of players are defensive midfileders. Please rank the following in order of importance from 1 to 3 where 1 is the most appropriate to you and 3 is the least appropriate to you.

Clustering solutions	1. Group	2. Group	3. Group	Rank
Selection 1 PAM ($K \sim [100:113, 130:133, 137:146]$)	Nemanja Matic Fernando	Sergio Busquets Javier Mascherano	—	1
Selection 2 Ward's method ($K \sim [100:150]$)	Nemanja Matic	Fernando	Sergio Busquets Javier Mascherano	2
Selection 3 PAM ($K \sim [114:129, 134:136, 147:150]$), Complete and average linkage ($K \sim [100:150]$)	Nemanja Matic	Fernando Sergio Busquets Javier Mascherano	—	3

Table C.4: Question 4: This group of players are midfileders. Please rank the following in order of importance from 1 to 3 where 1 is the most appropriate to you and 3 is the least appropriate to you.

Clustering solutions	1. Group	2. Group	3. Group	4. Group	Rank
Selection 1 PAM ($K \sim [100:113, 130:133, 137:146]$)	Gabi	Tiago	Xabi Alonso	Thiago Motta	2
Selection 2 PAM ($K \sim [114:129, 134:136, 147:150]$)	Gabi	Tiago	Xabi Alonso Thiago Motta	—	1
Selection 3 Ward's method, complete and average linkage ($K \sim [100:150]$)	Gabi Xabi Alonso Thiago Motta	Tiago	—	—	3

Table C.5: Question 5: This group of players are defensive midfileders. Please rank the following in order of importance from 1 to 3 where 1 is the most appropriate to you and 3 is the least appropriate to you.

Clustering solutions	1. Group	2. Group	3. Group	4. Group	Rank
Selection 1 PAM and average linkage ($K \sim [100:150]$)	Paul Pogba Arturo Vidal	Kevin De Bruyne	Henrikh Mkhitaryan	—	1
Selection 2 Ward's method ($K \sim [100:150]$)	Paul Pogba	Arturo Vidal	Kevin De Bruyne	Henrikh Mkhitaryan	3
Selection 3 Complete linkage ($K \sim [100:150]$)	Paul Pogba Arturo Vidal	Kevin De Bruyne Henrikh Mkhitaryan	—	—	2

Table C.6: Question 6: This group of players are attacking midfileders. Please rank the following in order of importance from 1 to 5 where 1 is the most appropriate to you and 5 is the least appropriate to you.

Clustering solutions	1. Group	2. Group	3. Group	Rank
Selection 1 PAM ($K \sim [100:118]$)	Lionel Messi Neymar Arjen Robben	Eden Hazard	Cristiano Ronaldo	3
Selection 2 PAM ($K \sim [119:150]$)	Lionel Messi Neymar Arjen Robben Cristiano Ronaldo	Eden Hazard	—	4
Selection 3 Ward's method ($K \sim [100:150]$)	Lionel Messi Arjen Robben Cristiano Ronaldo	Eden Hazard Neymar	—	5
Selection 4 Complete linkage ($K \sim [100:150]$)	Lionel Messi Arjen Robben Eden Hazard Neymar	Cristiano Ronaldo	—	1
Selection 5 Average linkage ($K \sim [100:150]$)	Lionel Messi Eden Hazard Neymar	Cristiano Ronaldo	Arjen Robben	2

Table C.7: Question 7: This group of players are forwards. Please rank the following in order of importance from 1 to 5 where 1 is the most appropriate to you and 5 is the least appropriate to you.

Clustering solutions	**1. Group**	**2. Group**	**3. Group**	**4. Group**	**Rank**
Selection 1 PAM ($K \sim [100:118]$)	Cristiano Ronaldo Karim Benzema	Robert Lewandowski	Zlatan Ibrahimovic	—	5
Selection 2 PAM ($K \sim [119:150]$)	Cristiano Ronaldo	Robert Lewandowski Zlatan Ibrahimovic	Karim Benzema	—	2
Selection 3 Ward's method ($K \sim [100:150]$)	Cristiano Ronaldo	Robert Lewandowski Zlatan Ibrahimovic Karim Benzema	—	—	1
Selection 4 Complete linkage ($K \sim [100:150]$)	Cristiano Ronaldo Karim Benzema	Robert Lewandowski Zlatan Ibrahimovic	—	—	4
Selection 5 Average linkage ($K \sim [100:150]$)	Cristiano Ronaldo	Karim Benzema	Robert Lewandowski	Zlatan Ibrahimovic	3

References

FIFA - Laws of the game, 2015/2016. http://www.fifa.com/mm/Document/FootballDevelopment/Refereeing/02/36/01/11/LawsofthegamewebEN_Neutral.pdf, Zurich, Switzerland, 2015.

Aggarwal, C. C. and Reddy, C. K. *Data clustering: algorithms and applications*. CRC press, 2013.

Aggarwal, C. C., Hinneburg, A., and Keim, D. A. *Database Theory — ICDT 2001: 8th International Conference London, UK, January 4–6, 2001 Proceedings*, chapter On the Surprising Behavior of Distance Metrics in High Dimensional Space, pages 420–434. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001. ISBN 978-3-540-44503-6. doi: 10.1007/3-540-44503-X_27. URL http://dx.doi.org/10.1007/3-540-44503-X_27.

Aitchison, J. *The Statistical Analysis of Compositional Data*. Chapman & Hall, Ltd., London, UK, 1986. ISBN 0-412-28060-4.

Aitchison, J. On criteria for measures of compositional difference. *Mathematical Geology*, 24(4): 365–379, 1992.

Aitchison, J., Barceló-Vidal, C., Martín-Fernández, J., and Pawlowsky-Glahn, V. Logratio analysis and compositional distance. *Mathematical Geology*, 32(3):271–275, 2000.

Aitchison, J., Barceló-Vidal, C., Martín-Fernández, J., and Pawlowsky-Glahn, V. Reply to letter to the editor by S. Rehder and U. Zier. *Mathematical Geology*, 33(7):849–860, 2001.

Aitchison, J., Kay, J. W., et al. Possible solution of some essential zero problems in compositional data analysis. 2003.

Alelyani, S., Tang, J., and Liu, H. Feature selection for clustering: A review. *Data Clustering: Algorithms and Applications*, 29, 2013.

Alfo, M. and Viviani, S. Finite mixtures of structured models. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 217–240. CRC Press, 2015.

Anderberg, M. *Cluster analysis for applications*. Probability and mathematical statistics. Academic Press, 1973.

Anders, S. and Huber, W. Differential expression analysis for sequence count data. *Genome biol*, 11(10):R106, 2010.

Andreopoulos, B. Clustering categorical data. In Aggarwal, C. C. and Reddy, C. K., editors, *Data clustering: algorithms and applications*, pages 277–303. CRC Press, 2013.

Anscombe, F. J. The transformation of poisson, binomial and negative-binomial data. *Biometrika*, 35(3/4):246–254, 1948.

Arabie, P. and Carroll, J. D. Mapclus: A mathematical programming approach to fitting the adclus model. *Psychometrika*, 45(2):211–235, 1980.

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., and Perona, I. An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243–256, 2013.

Art, D., Gnanadesikan, R., and Kettenring, J. Data-based metrics for cluster analysis. *Utilitas Mathematica A*, 21:75–99, 1982.

Azran, A. and Ghahramani, Z. Spectral methods for automatic multiscale data clustering. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 190–197. IEEE, 2006.

Bacon Shone, J. Modelling structural zeros in compositional data. Universitat de Girona. Departament d'Informàtica i Matemàtica Aplicada, 2003. URL http://hdl.handle.net/10722/123816.

Bacon-Shone, J. A short history of compositional data analysis. In Pawlowsky-Glahn, V. and Buccianti, A., editors, *Compositional Data Analysis: Theory and Applications*, pages 3–11. John Wiley & Sons, 2011.

Baker, F. B. and Hubert, L. J. Measuring the power of hierarchical cluster analysis. *Journal of the American Statistical Association*, 70(349):31–38, 1975.

Bezdek, J. C. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981. ISBN 0306406713.

Bhattachayya, A. On a measure of divergence between two statistical population defined by their population distributions. *Bulletin Calcutta Mathematical Society*, 35:99–109, 1943.

Bolton, R. and Krzanowski, W. Projection pursuit clustering for exploratory data analysis. *Journal of Computational and Graphical Statistics*, 2012.

Borg, I. and Groenen, P. J. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

Borg, I. and Lingoes, J. C. A model and algorithm for multidimensional scaling with external constraints on the distances. *Psychometrika*, 45(1):25–38, 1980.

Borg, I., Groenen, P. J., and Mair, P. *Applied multidimensional scaling*. Springer Science & Business Media, 2012.

Bray, J. R. and Curtis, J. T. An ordination of the upland forest communities of southern wisconsin. *Ecological monographs*, 27(4):325–349, 1957.

Brito, P. Clustering of symbolic data. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 469–496. CRC Press, 2015.

Burden, R. L. and Faires, J. D. 2.1 the bisection algorithm. *Numerical Analysis. Prindle, Weber & Schmidt, Boston, MA., pp. x*, 676, 1985.

Cai, L., Huang, H., Blackshaw, S., Liu, J., Cepko, C., Wong, W., et al. Clustering analysis of sage data using a poisson approach. *Genome biology*, 5(7):2004–5, 2004.

Caiado, J., Maharaj, A. E., and D'Urso, P. Time-series clustering. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 241–264. CRC Press, 2015.

Cailliez, F. The analytical solution of the additive constant problem. *Psychometrika*, 48(2):305–308, 1983.

Cajori, F. *A history of mathematical notations*, volume 1. Courier Corporation, 1928.

Caliński, T. and Harabasz, J. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

Carreira-Perpinan, M. A. Clustering methods based on kernel density estimators: Means shift algorithms. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 383–418. CRC Press, 2015.

Carroll, J. D. and Arabie, P. Indclus: An individual differences generalization of the adclus model and the mapclus algorithm. *Psychometrika*, 48(2):157–169, 1983.

Carroll, J. D. and Chang, J.-J. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika*, 35(3):283–319, 1970.

Celeux, G. and Govaert, G. Latent class models for categorical data. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 173–194. CRC Press, 2015.

Chang, W., Cheng, J., Allaire, J. J., Xie, Y., and McPherson, J. Shiny: web application framework for r. *R package version 0.11*, 1(4):106, 2015.

Charrad, M., Ghazzali, N., Boiteau, V., and Niknafs, A. NbClust: An R package for determining the relevant number of clusters in a data set. *Journal of Statistical Software*, 61(6):1–36, 2014. URL http://www.jstatsoft.org/v61/i06/.

Charytanowicz, M., Niewczas, J., Kulczycki, P., Kowalski, P. A., Łukasik, S., and Żak, S. Complete gradient clustering algorithm for features analysis of x-ray images. In *Information technologies in biomedicine*, pages 15–24. Springer, 2010.

Chayes, F. On correlation between variables of constant sum. *Journal of Geophysical research*, 65 (12):4185–4193, 1960.

Choi, S.-S., Cha, S.-H., and Tappert, C. C. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, 8(1):43–48, 2010.

Clarke, K. R., Somerfield, P. J., and Chapman, M. G. On resemblance measures for ecological studies, including taxonomic dissimilarities and a zero-adjusted bray–curtis coefficient for denuded assemblages. *Journal of Experimental Marine Biology and Ecology*, 330(1):55–80, 2006.

Conover, W. J. *Practical Nonparametric Statistics*. John Wiley, New York, NY, second edition, 1980.

Cook, D. and Swayne, D. F. *Interactive and dynamic graphics for data analysis: with R and GGobi*. Springer Science & Business Media, 2007.

Cox, T. F. and Cox, M. A. Multidimensional scaling on a sphere. *Communications in Statistics-Theory and Methods*, 20(9):2943–2953, 1991.

Cox, T. F. and Cox, M. A. *Multidimensional scaling*. CRC press, 2000.

Daunis-i Estadella, J., Martín-Fernández, J., and Palarea-Albaladejo, J. Bayesian tools for count zeros in compositional data. *Proceedings of CODAWORK*, 8:8, 2008.

De Boor, C., De Boor, C., Mathématicien, E.-U., De Boor, C., and De Boor, C. *A practical guide to splines*, volume 27. Springer-Verlag New York, 1978.

De Leeuw, J. and Heiser, W. J. Multidimensional scaling with restrictions on the configuration. *Multivariate analysis*, 5:501–522, 1980.

De Leeuw, J. and Mair, P. Multidimensional scaling using majorization: Smacof in r. *Department of Statistics, UCLA*, 2011.

de Mendoza, J. et al. Memoria sobre algunos métodos nuevos de calcular la longitud por las distancias lunares y aplicación de su teórica a la solución de otros problemas de navegación. 1795.

De Soete, G. Optimal variable weighting for ultrametric and additive tree clustering. *Quality and Quantity*, 20(2-3):169–180, 1986.

Déjean, S. and Mothe, J. Visual clustering for data analysis and graphical user interfaces. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 679–702. CRC Press, 2015.

Dheeru, D. and Karra Taniskidou, E. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Di Salvo, V., Baron, R., Tschan, H., Calderon Montero, F., Bachl, N., and Pigozzi, F. Performance characteristics according to playing position in elite soccer. *International Journal of Sports Medicine*, 28(3):222, 2007.

Dias, D. B., Madeo, R. C., Rocha, T., Bíscaro, H. H., and Peres, S. M. Hand movement recognition for brazilian sign language: a study using distance-based neural networks. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pages 697–704. IEEE, 2009.

Dimitriadou, E., Dolničar, S., and Weingessel, A. An examination of indexes for determining the number of clusters in binary data sets. *Psychometrika*, 67(1):137–159, 2002.

Ding, C., He, X., Zha, H., Gu, M., and Simon, H. Spectral min-max cut for graph partitioning and data clustering. *Lawrence Berkeley National Lab. Tech. report*, 47848, 2001.

Donoho, D. 50 years of data science. In *The Tukey Centennial workshop*, Princeton, NJ, 2015.

Dunn, J. C. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. 1973.

Dunn, J. C. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1): 95–104, 1974.

D'Urso, P. Fuzzy clustering. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 544–574. CRC Press, 2015.

Edwards, C. J. *The historical development of the calculus*. Springer Science & Business Media, 2012.

Efron, B. Nonparametric estimates of standard error: the jackknife, the bootstrap and other methods. *Biometrika*, 68(3):589–599, 1981.

Efron, B. and Tibshirani, R. J. *An introduction to the bootstrap*. CRC press, 1994.

Egozcue, J. J., Pawlowsky-Glahn, V., Mateu-Figueras, G., and Barcelo-Vidal, C. Isometric logratio transformations for compositional data analysis. *Mathematical Geology*, 35(3):279–300, 2003.

Embretson, S. E. and Reise, S. P. *Item response theory*. Psychology Press, 2013.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

Everitt, B. S. and Hand, D. J. *Finite mixture distributions*. Chapman & Hall, London, 1981.

Everitt, B. S., Landau, S., Leese, M., and Stahl, D. *Cluster Analysis*, volume 20. John Wiley & Sons, 2011.

Faith, D. P., Minchin, P. R., and Belbin, L. Compositional dissimilarity as a robust measure of ecological distance. *Vegetatio*, 69(1-3):57–68, 1987.

Fang, Y. and Wang, J. Selection of the number of clusters via the bootstrap method. *Computational Statistics & Data Analysis*, 56(3):468–477, 2012.

Fisher, R. A. The use of multiple measurements in taxonomic problems. *Annals of human genetics*, 7(2):179–188, 1936.

Florek, K., Łukaszewicz, J., Perkal, J., Steinhaus, H., and Zubrzycki, S. Sur la liaison et la division des points d'un ensemble fini. In *Colloquium Mathematicae*, volume 2, pages 282–285. Institute of Mathematics Polish Academy of Sciences, 1951.

Fodor, I. K. A survey of dimension reduction techniques. Technical report, Technical Report UCRL-ID-148494, Lawrence Livermore National Laboratory, 2002.

Forgey, E. Cluster analysis of multivariate data: Efficiency vs. interpretability of classification. *Biometrics*, 21(3):768–769, 1965.

Forina, M., Leardi, R., Armanino, C., Lanteri, S., Conti, P., and Princi, P. Parvus: An extendable package of programs for data exploration, classification and correlation. *Journal of Chemometrics*, 4(2):191–193, 1988.

Fowlkes, E., Gnanadesikan, R., and Kettenring, J. R. Variable selection in clustering. *Journal of classification*, 5(2):205–228, 1988.

Fowlkes, E. B. and Mallows, C. L. A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, 78(383):553–569, 1983.

Friedman, J. H. and Meulman, J. J. Clustering objects on subsets of attributes (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(4):815–849, 2004.

Frühwirth-Schnatter, S. *Finite mixture and Markov switching models*. Springer, Berlin, 1st edition, 2006.

Fry, T. R., Chong, D., et al. A tale of two logits, compositional data analysis and zero observations. Universitat de Girona, Girona (Spain), 2005. URL http://ima.udg.es/Activitats/CoDaWork05/.

Gan, G., Ma, C., and Wu, J. *Data clustering: theory, algorithms, and applications*, volume 20. ASA-SIAM series on statistics and applied probability, 2007.

Gaudreau, P. and Blondin, J.-P. Different athletes cope differently during a sport competition: A cluster analysis of coping. *Personality and Individual Differences*, 36(8):1865–1877, 2004.

Gelman, A. and Hennig, C. Beyond subjective and objective in statistics. *ArXiv e-prints*, Aug. 2015.

Ghosh, J. and Acharya, A. A survey of consensus clustering. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 497–518. CRC Press, 2015.

Gnanadesikan, R., Kettenring, J., and Tsao, S. Weighting and selection of variables for cluster analysis. *Journal of Classification*, 12(1):113–136, 1995.

Goodman, L. A. and Kruskal, W. H. Measures of association for cross classifications. *Journal of the American Statistical Association*, 49(268):732–764, 1954.

Gordon, A. Constructing dissimilarity measures. *Journal of Classification*, 7(2):257–269, 1990.

Gower, J. C. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53(3-4):325–338, 1966.

Gower, J. C. A general coefficient of similarity and some of its properties. *Biometrics*, pages 857–871, 1971.

Gower, J. C. Euclidean distance geometry. *Mathematical Scientist*, 7(1):1–14, 1982.

Gower, J. C. Properties of euclidean and non-euclidean distance matrices. *Linear Algebra and its Applications*, 67:81–97, 1985.

Gower, J. C. and Legendre, P. Metric and euclidean properties of dissimilarity coefficients. *Journal of classification*, 3(1):5–48, 1986.

Greenacre, M. *Correspondence analysis in practice*. CRC press, 2017.

Greenacre, M. and Primicerio, R. *Multivariate analysis of ecological data*. Fundacion BBVA, 2014.

Hagen, L. and Kahng, A. B. New spectral methods for ratio cut partitioning and clustering. *IEEE transactions on computer-aided design of integrated circuits and systems*, 11(9):1074–1085, 1992.

Halkidi, M. and Vazirgiannis, M. A density-based cluster validity approach using multi-representatives. *Pattern Recognition Letters*, 29(6):773–786, 2008.

Halkidi, M., Vazirgiannis, M., and Hennig, C. Method-independent indices for cluster validation and estimating the number of clusters. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 595–618. CRC Press, 2015.

Hand, D. J. Statistics and the theory of measurement. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, pages 445–492, 1996.

Hartigan, J. A. Representation of similarity matrices by trees. *Journal of the American Statistical Association*, 62(320):1140–1158, 1967.

Hartigan, J. A. and Wong, M. A. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

Hausdorf, B. and Hennig, C. The influence of recent geography, palaeogeography and climate on the composition of the fauna of the central aegean islands. *Biological Journal of the Linnean Society*, 84(4):785–795, 2005.

Hennig, C. Asymmetric linear dimension reduction for classification. *Journal of Computational and Graphical Statistics*, 13(4):930–945, 2004.

Hennig, C. A method for visual cluster validation. In *Classification—the Ubiquitous Challenge*, pages 153–160. Springer, 2005.

Hennig, C. fpc: Flexible procedures for clustering. r package version 2.1-5, 2013.

Hennig, C. Clustering strategy and method selection. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 703–730. CRC Press, 2015a.

Hennig, C. What are the true clusters? *Pattern Recognition Letters*, 64:53–62, 2015b.

Hennig, C. Cluster validation by measurement of clustering characteristics relevant to the user. *arXiv preprint arXiv:1703.09282*, 2017.

Hennig, C. and Hausdorf, B. Design of dissimilarity measures: A new dissimilarity between species distribution areas. In *Data Science and Classification*, pages 29–37. Springer, 2006.

Hennig, C. and Liao, T. F. How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 62(3):309–369, 2013.

Hennig, C. and Meila, M. Cluster analysis: An overview. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 1–19. CRC Press, 2015.

Hill, I. Association football and statistical inference. *Applied Statistics*, pages 203–208, 1974.

Hitchcock, D. B. and Greenwood, M. C. Clustering functional data. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 265–288. CRC Press, 2015.

Hu, X. A data preprocessing algorithm for data mining applications. *Applied Mathematics Letters*, 16(6):889–895, 2003.

Hubert, L. and Arabie, P. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

Hubert, L. and Schultz, J. Quadratic assignment as a general data analysis strategy. *British journal of mathematical and statistical psychology*, 29(2):190–241, 1976.

Hwang, C.-L. and Masud, A. S. M. *Multiple objective decision making—methods and applications: a state-of-the-art survey*, volume 164. Springer Science & Business Media, 2012.

Impellizzeri, F., Marcora, S., Castagna, C., Reilly, T., Sassi, A., Iaia, F., Rampinini, E., et al. Physiological and performance effects of generic versus specific aerobic training in soccer players. *International Journal of Sports Medicine*, 27(6):483–492, 2006.

Inman, J. *Navigation and Nautical Astronomy, for the Use of British Seamen*. F. & J. Rivington, 1849.

Izenman, A. *Modern multivariate statistical techniques*, volume 1. Springer, 2008.

Jaccard, P. The distribution of the flora in the alpine zone. *New phytologist*, 11(2):37–50, 1912.

Jackson, D. A. Compositional data in community ecology: the paradigm or peril of proportions? *Ecology*, 78(3):929–940, 1997.

Jackson, J. E. *A user's guide to principal components*, volume 587. John Wiley & Sons, 2005.

Jardine, C., Jardine, N., and Sibson, R. The structure and construction of taxonomic hierarchies. *Mathematical Biosciences*, 1(2):173–179, 1967.

Jiang, D., Tang, C., and Zhang, A. Cluster analysis for gene expression data: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 16(11):1370–1386, 2004.

Johnson, S. C. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

Jolliffe, I. T. Discarding variables in a principal component analysis. i: Artificial data. *Applied statistics*, pages 160–173, 1972.

Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A. kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, 11(9):1–20, 2004. URL http://www.jstatsoft.org/v11/i09/.

Karlis, D. and Meligkosidou, L. Model based clustering for multivariate count data. In *18th International Workshop on Statistical Modelling*, page 211, 2003.

Karlis, D. and Ntzoufras, I. Bayesian modelling of football outcomes: using the skellam's distribution for the goal difference. *IMA Journal of Management Mathematics*, 20(2):133–145, 2009.

Kaufman, L. and Rousseeuw, P. J. *Finding groups in data: An introduction to cluster analysis*, volume 344. John Wiley & Sons, 1990.

Knorr-Held, L. Dynamic rating of sports teams. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 49(2):261–276, 2000.

Kosmidis, I. and Karlis, D. Model-based clustering using copulas with applications. *Statistics and Computing*, pages 1–21, 2015. ISSN 1573-1375. doi: 10.1007/s11222-015-9590-5. URL http://dx.doi.org/10.1007/s11222-015-9590-5.

Kruskal, J. B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964a.

Kruskal, J. B. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29(2): 115–129, 1964b.

Kulczynski, S. Die pflanzenassoziationen der pieninen. *Bulletin International de l'Academie Polonaise des Sciences et des Lettres, Classe des Sciences Mathematiques et Naturelles, B (Sciences Naturelles)*, II:57–203, 1927a.

Kulczynski, S. Zespoly roslin w pieninach. *Bulletin International de l'Academie Polonaise des Sciences et des Lettres, Classe des Sciences Mathematiques et Naturelles*, 1928(Suppl 2):57–203, 1927b.

Kullback, S. and Leibler, R. A. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

Lago-Fernández, L. F. and Corbacho, F. Normality-based validation for crisp clustering. *Pattern Recognition*, 43(3):782–795, 2010.

Latouche, P., Birmelé, E., and Ambroise, C. Overlapping stochastic block models. *arXiv preprint arXiv:0910.2098*, 2009.

Leisch, F. Resampling methods for exploring cluster stability. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 637–652. CRC Press, 2015.

Liao, T. W. Clustering of time series data—a survey. *Pattern recognition*, 38(11):1857–1874, 2005.

Likert, R. A technique for the measurement of attitudes. *Archives of psychology*, 1932.

Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J., and Wu, S. Understanding and enhancement of internal clustering validation measures. *IEEE transactions on cybernetics*, 43(3):982–994, 2013.

Lloyd, S. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2): 129–137, 1982.

Lovell, D., Müller, W., Taylor, J., Zwart, A., and Helliwell, C. Proportions, percentages, ppm: do the molecular biosciences treat compositional data right. In Pawlowsky-Glahn, V. and Buccianti, A., editors, *Compositional Data Analysis: Theory and Applications*, pages 193–207. John Wiley & Sons, 2011.

Lutz, D. *A cluster analysis of NBA players*. MIT Sloan Sports Analytics Conf., Boston, MA, 2012.

MacQueen, J. et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., and Hornik, K. *cluster: Cluster Analysis Basics and Extensions*, 2017. R package version 2.0.6 — For new features, see the 'Changelog' file (in the package source).

Maher, M. J. Modelling association football scores. *Statistica Neerlandica*, 36(3):109–118, 1982.

Martín, M. C. Performance of eight dissimilarity coefficients to cluster a compositional data set. In *Data Science, Classification, and Related Methods*, pages 162–169. Springer, 1998.

Martín-Fernández, J., Barceló-Vidal, C., Pawlowsky-Glahn, V., Buccianti, A., Nardi, G., and Potenza, R. Measures of difference for compositional data and hierarchical clustering methods. In *Proceedings of IAMG*, volume 98, pages 526–531, 1998.

Martín-Fernández, J. A., Barceló-Vidal, C., and Pawlowsky-Glahn, V. Dealing with zeros and missing values in compositional data sets using nonparametric imputation. *Mathematical Geology*, 35(3):253–278, 2003.

Martín-Fernández, J. A., Palarea-Albaladejo, J., and Olea, R. A. Dealing with zeros. In Pawlowsky-Glahn, V. and Buccianti, A., editors, *Compositional Data Analysis: Theory and Applications*, pages 43–58. John Wiley & Sons, 2011.

Maugis, C., Celeux, G., and Martin-Magniette, M.-L. Variable selection for clustering with gaussian mixture models. *Biometrics*, 65(3):701–709, 2009.

McHale, I. G. and Szczepański, Ł. A mixed effects model for identifying goal scoring ability of footballers. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 177(2): 397–417, 2014.

McHale, I. G., Scarf, P. A., and Folker, D. E. On the development of a soccer player performance rating system for the english premier league. *Interfaces*, 42(4):339–351, 2012.

McLachlan, G. and Basford, K. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.

McLachlan, G. and Peel, D. *Finite Mixture Models*. Wiley, 2000.

McLachlan, G. J. and Rahtnayake, S. I. Mixture models for standard p dimensional euclidean data. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 145–172. CRC Press, 2015.

Meila, M. Criteria for comparing clusterings. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 619–636. CRC Press, 2015a.

Meila, M. Spectral clustering. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 125–144. CRC Press, 2015b.

Meila, M. and Xu, L. Multiway cuts and spectral clustering. 2003.

Milligan, G. W. *Clustering validation: Results and implications for applied analyses*. World Scientific, Singapore, 1996.

Milligan, G. W. and Cooper, M. C. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, 1985.

Milligan, G. W. and Cooper, M. C. A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavioral Research*, 21(4):441–458, 1986.

Milligan, G. W. and Cooper, M. C. A study of standardization of variables in cluster analysis. *Journal of classification*, 5(2):181–204, 1988.

Mirkin, B. Quadratic error and k-means. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 33–54. CRC Press, 2015.

Mohr, M., Krustrup, P., and Bangsbo, J. Match performance of high-standard soccer players with special reference to development of fatigue. *Journal of sports sciences*, 21(7):519–528, 2003.

Moroney, M. Facts from figures. *Pelican Books*, (A236), 1954.

Mouselimis, L. Clusterr: Gaussian mixture models, k-means, mini-batch-kmeans and k-medoids clustering. *Journal of Statistical Software*, 2018. URL https://cran.r-project.org/web/packages/ClusterR/index.html.

Murphy, T. B. Model-based clustering for network data. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 337–357. CRC Press, 2015.

Murtagh, F. and Legendre, P. Ward's hierarchical agglomerative clustering method: Which algorithms implement ward's criterion? *Journal of Classification*, 31(3):274–295, 2014. ISSN 1432-1343. doi: 10.1007/s00357-014-9161-z. URL http://dx.doi.org/10.1007/s00357-014-9161-z.

Ng, A. Y., Jordan, M. I., and Weiss, Y. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

Nguyen, N. and Caruana, R. Consensus clusterings. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 607–612. IEEE, 2007.

Norušis, M. J. *IBM SPSS statistics 19 statistical procedures companion*. Prentice Hall, 2012.

Ogles, B. M. and Masters, K. S. A typology of marathon runners based on cluster analysis of motivations. *Journal of Sport Behavior*, 26(1):69, 2003.

Ostini, R. and Nering, M. L. *Polytomous item response theory models*. Number 144. Sage, 2006.

Palarea-Albaladejo, J. and Martín-Fernández, J. A modified em alr-algorithm for replacing rounded zeros in compositional data sets. *Computers & Geosciences*, 34(8):902–917, 2008.

Pawlowsky-Glahn, V., Egozcue, J. J., and Tolosana-Delgado, R. *Modeling and analysis of compositional data*. John Wiley & Sons, 2015.

Pearson, K. Mathematical contributions to the theory of evolution.–on a form of spurious correlation which may arise when indices are used in the measurement of organs. *Proceedings of the royal society of london*, 60(359-367):489–498, 1896.

Pearson, K. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

Pérez-Suárez, A., Martnez-Trinidad, J., Carrasco-Ochoa, J. A., and Medina-Pagola, J. E. A new overlapping clustering algorithm based on graph theory. *Advances in Artificial Intelligence*, 7629:61–72, 2012.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. URL http://www.R-project.org/. ISBN 3-900051-07-0.

Raftery, A. E. and Dean, N. Variable selection for model-based clustering. *Journal of the American Statistical Association*, 101(473):168–178, 2006.

Rampinini, E., Bishop, D., Marcora, S., Ferrari Bravo, D., Sassi, R., and Impellizzeri, F. Validity of simple field tests as indicators of match-related physical performance in top-level professional soccer players. *International journal of sports medicine*, 28(3):228, 2007.

Ramsay, J. O. Monotone regression splines in action. *Statistical science*, pages 425–441, 1988.

Rand, W. M. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

Rao, V. Dirichlet process mixtures and nonparametric bayesian approach. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 195–216. CRC Press, 2015.

Reep, C. and Benjamin, B. Skill and chance in association football. *Journal of the Royal Statistical Society. Series A (General)*, pages 581–585, 1968.

Rehder, S. and Zier, U. Letter to the editor: Comment on "logratio analysis and compositional distance" by j. aitchison, c. barceló-vidal, ja martín-fernández, and v. pawlowsky-glahn. *Mathematical Geology*, 33(7):845–848, 2001.

Ritter, G. *Robust cluster analysis and variable selection*. CRC Press, 2014.

Robinson, M. D., McCarthy, D. J., and Smyth, G. K. edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 2010.

Rocci, R., Gattone, S. A., and Vichi, M. A new dimension reduction method: Factor discriminant k-means. *Journal of classification*, 28(2):210–226, 2011.

Rogers, D. J. and Tanimoto, T. T. A computer program for classifying plants. *Science*, 132(3434): 1115–1118, 1960.

Romesburg, H. C. *Cluster analysis for researchers*. Lifetime Learning Publications, 1984.

Rue, H. and Salvesen, O. Prediction and retrospective analysis of soccer matches in a league. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 49(3):399–418, 2000.

Schumaker, L. *Spline functions: basic theory*. Cambridge University Press, 2007.

Scott, A. J. and Symons, M. J. Clustering methods based on likelihood ratio criteria. *Biometrics*, pages 387–397, 1971.

Scrucca, L., Fop, M., Murphy, T. B., and Raftery, A. E. mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, 8(1):205–233, 2016.

Shannon, C. E. A mathematical theory of communication. *The Bell System Technical Journal*, 27 (3):379–423, July 1948. ISSN 0005-8580. doi: 10.1002/j.1538-7305.1948.tb01338.x.

Shen, H., Cheng, X., Cai, K., and Hu, M.-B. Detect overlapping and hierarchical community structure in networks. *Physica A: Statistical Mechanics and its Applications*, 388(8):1706–1712, 2009.

Shepard, R. N. and Arabie, P. Additive clustering: Representation of similarities as combinations of discrete overlapping properties. *Psychological Review*, 86(2):87, 1979.

Shi, G. R. Multivariate data analysis in palaeoecology and palaeobiogeography—a review. *Palaeogeography, Palaeoclimatology, Palaeoecology*, 105(3):199–234, 1993.

Shi, J. and Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

Sneath, P. H., Sokal, R. R., et al. *Numerical taxonomy. The principles and practice of numerical classification.* W.H.Freeman & Co Ltd, 1973.

Sokal, R. R. A statistical method for evaluating systematic relationship. *University of Kansas science bulletin*, 28:1409–1438, 1958.

Sokal, R. R. and Rohlf, F. J. Taxonomic congruence in the leptopodomorpha re-examined. *Systematic Zoology*, 30(3):309–325, 1981.

Sørensen, T. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biol. Skr.*, 5:1–34, 1948.

Steinhaus, H. Sur la division des corps matriels en parties. *Bulletin of the Polish Academy of Sciences 4*, pages 801–804, 1957.

Steinley, D. K-medoids and other criteria for crisp clustering. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 55–66. CRC Press, 2015.

Stevens, S. S. On the theory of scales of measurement. *Science*, 103(2684):677–680, 1946. ISSN 0036-8075. doi: 10.1126/science.103.2684.677. URL http://science.sciencemag.org/content/103/2684/677.

Strehl, A. and Ghosh, J. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.

Templ, M., Filzmoser, P., and Reimann, C. Cluster analysis applied to regional geochemical data: problems and possibilities. *Applied Geochemistry*, 23(8):2198–2213, 2008.

Theus, M. and Urbanek, S. *Interactive graphics for data analysis: principles and examples*. CRC Press, 2008.

Tibshirani, R. and Walther, G. Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 14(3):511–528, 2005.

Titterington, D. M., Smith, A. F. M., and Makov, U. E. *Statistical analysis of finite mixture distributions*. John Wiley & Sons, New York, 1985.

Topchy, A., Jain, A. K., and Punch, W. A mixture model for clustering ensembles. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, pages 379–390. SIAM, 2004.

Torgerson, W. S. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, Dec 1952. ISSN 1860-0980. doi: 10.1007/BF02288916. URL https://doi.org/10.1007/BF02288916.

Torgerson, W. S. Theory and methods of scaling. *New York: J. Wiley*, 1958.

Tyler, D. E., Critchley, F., Dümbgen, L., and Oja, H. Invariant co-ordinate selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):549–592, 2009.

Van Mechelen, I., Hampton, J., Michalski, R. S., and Theuns, P. *Categories and concepts: Theoretical views and inductive data analysis*. Academic Press New York, 1993.

Vichi, M. Two-mode partitioning and multi-partitioning. In Hennig, C., Meila, M., Murtagh, F., and Rocci, R., editors, *Handbook of Cluster Analysis*, pages 519–544. CRC Press, 2015.

Vichi, M., Rocci, R., and Kiers, H. A. Simultaneous component and clustering models for three-way data: within and between approaches. *Journal of Classification*, 24(1):71–98, 2007.

Von Luxburg, U. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Walesiak, M., Dudek, A., and Dudek, M. A. clusterSim package, 2011. URL http://keii.ue.wroc.pl/clusterSim.

Wallace, D. L. Comment. *Journal of the American Statistical Association*, 78(383):569–576, 1983.

Walley, P. Inferences from multinomial data: learning about a bag of marbles. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 3–57, 1996.

Wang, C. J., Koh, K. T., and Chatzisarantis, N. An intra-individual analysis of players' perceived coaching behaviours, psychological needs, and achievement goals. *International Journal of Sports Science and Coaching*, 4(2):177–192, 2009.

Wang, H., Shan, H., and Banerjee, A. Bayesian cluster ensembles. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 4(1):54–70, 2011a.

Wang, J. Consistent selection of the number of clusters via crossvalidation. *Biometrika*, 97(4): 893–904, 2010.

Wang, P., Laskey, K. B., Domeniconi, C., and Jordan, M. I. Nonparametric bayesian co-clustering ensembles. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 331–342. SIAM, 2011b.

Ward Jr., J. H. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

Warton, D. I., Wright, S. T., and Wang, Y. Distance-based multivariate analyses confound location and dispersion effects. *Methods in Ecology and Evolution*, 3(1):89–101, 2012.

Weeks, D. G. and Bentler, P. Restricted multidimensional scaling models for asymmetric proximities. *Psychometrika*, 47(2):201–208, 1982.

Winsberg, S. and Soete, G. Multidimensional scaling with constrained dimensions: Conscal. *British Journal of Mathematical and Statistical Psychology*, 50(1):55–72, 1997.

Witten, D. M. Classification and clustering of sequencing data using a poisson model. *The Annals of Applied Statistics*, pages 2493–2518, 2011.

Witten, D. M. and Tibshirani, R. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 2012.

Xiong, H. and Li, Z. Clustering validation measures. In Aggarwal, C. C. and Reddy, C. K., editors, *Data clustering: algorithms and applications*, pages 572–602. CRC Press, 2013.

Xu, R. and Wunsch, D. *Clustering*, volume 10. John Wiley & Sons, 2008.

Yang, L., Yu, Z., Qian, J., and Liu, S. Overlapping community detection using weighted consensus clustering. *Pramana*, 87(4):58, 2016.

Yingying, L., Chiusano, S., and D'elia, V. Modeling athlete performance using clustering techniques. In *The Third International Symposium on Electronic Commerce and Security Workshops (ISECS 2010)*, page 169, 2010.

Zier, U. and Rehder, S. Grain-size analysis—a closed data problem. In *A. Buccianti, G. Nardi, and R. Potenza, Proceedings of IAMG98, The Fourth Annual Conference of the International Association for Mathematical Geology: De Frede, Naples*, pages 555–558, 1998.