An assessment of the application of cluster analysis techniques to the Johannesburg Stock Exchange

Robyn Tully

A dissertation submitted to the Department of Actuarial Science, Faculty of Commerce, at the University of Cape Town, in partial fulfilment of the requirements for the degree of Master of Philosophy.

February 17, 2014

Master of Philosophy specialising in Mathematical Finance,
University of Cape Town,
Cape Town

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the Degree of Master of Philosophy in the University of Cape Town. It has not been submitted before for any degree or examination in any other University.

February 17, 2014

Abstract

Cluster analysis is becoming an increasingly popular method in modern finance because of its ability to summarise large amounts of data and so help individual and institutional investors to make timeous and informed investment decisions. This is no less true for investors in smaller, emerging markets - such as the Johannesburg Stock Exchange - than it is for those in the larger global markets. This study examines the application of two clustering techniques to the Johannesburg Stock Exchange. First, the application of Salvador and Chan's (2003) L method stopping rule to a hierarchical clustering of time series return data was analysed as a method for determining the number of latent groups in the data set. Using Ward's method and the Euclidean distance function, this method appears to be able detect the correct number of clusters on the JSE. Second, the ability of three different clustering algorithms to generate consistent clusters and cluster members over time on the Johannesburg Stock Exchange was analysed. The variation of information was used to measure the consistency of cluster members through time. Hierarchical clustering using Ward's method and the Euclidean distance measure proved to produce the most consistent results, while the K-means algorithms generated the least consistent cluster members.

Keywords: cluster analysis, cluster validity, cluster consistency, hierarchical clustering, K-means, DBSCAN, time series, Johannesburg Stock Exchange, variation of information

Acknowledgements

To Dylan, Nicolas, Liddi, Kelly and Ralph, who surprised me with their kindness. Thank you for keeping me smiling.

To Nicole and Mark, for sharing these last seven years with me. We have lived together, travelled together, passed and failed together, and grown from geeky little kids into geeky big kids. I wouldn’t change a thing.

To Mum and Dad, for believing in all that I am and all that I can be. You are the start and the end of everything. Thank you.

And most of all to Francois, for giving me perspective.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Since the first algorithm was developed in the 1930s, cluster analysis has evolved to serve many different purposes in a wide variety of fields. One of its more recent applications is in the field of finance, where vast quantities of valuable data are generated by stock markets every day. Summarising this data in a meaningful way so that individuals and institutions can use it to make informed investment decisions has become one of the biggest challenges of modern finance. The ability of cluster analysis techniques to efficiently reduce the dimension of large data sets (Everitt, 1974) has made it an attractive method to researchers and financial market participants alike. Liao *et. al.* (2008), for example, used the K-means clustering algorithm to summarise and visualise the Taiwan stock market, and so constructed portfolios under different market conditions.

Dimension reduction is not the only reason that cluster analysis is a useful application in modern finance. Apartsin *et. al.* (2013) used cluster analysis techniques to recognise patterns in the behaviour of stocks across five of the world's largest stock markets. They found that investors in different national financial markets react differently, even when facing the same market conditions. Using K-means clustering, Manniste *et. al.* (2011) investigated reasons for the different behaviour of stocks during the 2007/2008 financial crisis. They observed that stocks which suffered the most through the turn-down were those with the highest ex ante P/E ratios and return on assets, while those which performed best tended to have low ex ante P/E ratios, high profit margins and a moderate return on assets.

Locally, methods in cluster analysis and other classification techniques have been successfully applied to the Johannesburg Stock Exchange (JSE) too. Hendricks *et. al*. (2014) monitored the intraday clustering behaviour of JSE stocks using a Master-Slave parallel genetic algorithm (PGA) with a Marsili and Giada log-likelihood function, and found that this ran significantly faster than a comparable serial genetic algorithm. Polakow and Gebbie (2008) used a singular-value decomposition and the Kaiser-Gutman stopping rule in their factor analysis of the JSE's Top-40 Index to estimate the true breadth of investment opportunities available to investors. This they found to be substantially lower than expected, and concluded this lack of diversification to be the result of market concentration, local currency volatility and exposure to the global commodity cycle.

The JSE is an interesting case to study in the context of world exchanges for three main reasons:

i. It is small; its market capitalisation makes up less than 1.5% of the total, global market capitalisation. This means that it generally exhibits lower liquidity and trade volumes than the large stock exchanges.
ii. It is dichotomous; the market is dominated by two distinct sectors (the resources sector, and financial and industrial sector), which respond to different drivers and have historically shown low levels of correlation.
iii. It is volatile. It is the preeminent stock exchange on the African continent, but its emerging market status makes it highly susceptible to global financial events and currency movements.

This combination of characteristics makes the JSE an interesting stock market to study. This paper therefore aims to determine how viable cluster analysis is as a tool for uncovering the structure inherent in financial time series of stocks listed on the JSE. To this end, two investigations were carried out. The first aimed to discover whether a cluster analysis technique could accurately determine the number of clusters in the JSE data set. The second aimed to determine how consistent the clusters and cluster members produced by various clustering techniques were through time.

A variety of clustering algorithms and methods were tested on the data set. One method from each of three well-known classes of clustering methods was selected: agglomerative clustering from the class of hierarchical clustering algorithms, K-means clustering from the class of partitional methods, and density based spatial clustering of applications with noise (DBSCAN) from the pool of density-based techniques. These clustering algorithms were coded from scratch and run in Microsoft Excel's Visual Basic for Applications.

It was decided to exclude a fuzzy clustering algorithm from the analysis since any grouping will not be useful unless each stock is ultimately assigned to a single cluster anyway. Although the concept is intuitively appealing, the true usefulness of fuzzy clustering is lost when these 'hard' categorisations are forced.

The investigation was conducted in two main stages. After some preliminary work was done on the data set, a stopping rule was applied to a hierarchical clustering to determine the optimal number of clusters. The output was then compared to results produced through alternative methods, as well as a prior knowledge of the data set in order to assess the results' validity.

Next, the consistency of the clusters and cluster members produced by each of the aforementioned clustering algorithms was assessed. The aim was to determine which method produced the most valid and robust results in the presence of the JSE's idiosyncrasies.

This thesis serves has a number of limitations. The intention is not to provide an exhaustive analysis of existing techniques on the data set, and the research makes no claims regarding the ideal clustering method for returns data from the JSE or other emerging market exchanges. Furthermore, the study does not analyse the entire universe of stocks on the JSE and can therefore not speculate on the true number of clusters underlying the exchange.

The paper begins in chapter 2 by providing a brief background of the three clustering techniques under consideration. The chapter discusses the various options available for implementing the algorithms and gives reasons for the choices that were made. Chapter 3 then provides a high level analysis of the data, as well as a description of the data preparation techniques which were applied. The methodology applied in discovering the number of clusters in the data set, is then outlined in chapter 4, before the results are presented and discussed. This is repeated for the cluster consistency investigation in chapter 5. Finally, chapter 6 discusses the suitability of cluster analysis as a tool for analysing JSE return data and the conclusions that can be drawn for future research of the South African stock market.

Chapter 2

Background of clustering methods

The term cluster analysis is used to describe a variety of mathematical techniques which attempt to classify objects within a data set into groups that are representative of the data's underlying structure (Romesburg, 1984). The classical approach, which is also known as unsupervised classification, derives these groups entirely from the structure of the data itself and makes no use of external or predefined information. Clusters are formed such that the objects in a cluster are similar to or related to one another by a chosen criterion, while at the same time being dissimilar or unrelated to the objects in other clusters (Spath, 1980).

The use of cluster analysis was first documented in 1932 in the field of anthropology (Driver and Kroeber, 1932). Since then, numerous clustering methods and algorithms have been developed in order to meet the needs of different users in different fields. Despite being the scrutiny of much research, however, there does not exist a consensus "best method" for solving all problems of a given type (Kogan, 2007), including financial applications.

It was therefore decided to review the consistency and robustness of three different clustering algorithms (a brief description of each can be found in Appendix A), and three different proximity measures. From there a number of decisions were made about the exact implementation of each clustering algorithm. This section reviews the literature and analysis which informed each of these decisions, beginning with a review of the proximity measures. It then considers the implementation of each algorithm in turn.

2.1 Proximity measures

All clustering algorithms relying on some measure of cluster proximity to determine which objects should be clustered together. Cluster proximity can be defined in terms of similarity of, or distance between objects (Frades and Matthiesen, 2010), and most algorithms can accept either.

Different proximity measures quantify different types of relationships in the data. In this study, the choice of measure defines the basis upon which two shares are considered to behave in the same way, and will therefore have a vital influence on the clustering results. According to Wittman (2002), "a time-series clustering will be valid if and only if the price fluctuations of stocks with a group are correlated, but the price fluctuations of stocks in different groups are uncorrelated or not as strongly correlated." This view suggests that the correlation between each pair of stocks is the most important feature to measure, and suggests that the following similarity measure may be the most appropriate for clustering stocks:

$$\textit{Pearson's Correlation}: \frac{\sum_{t=1}^{n}(x_{it}-\bar{x}_i)(x_{jt}-\bar{x}_j)}{\left[\sum_{t=1}^{n}(x_{it}-\bar{x}_i)^2 \sum_{t=1}^{n}(x_{jt}-\bar{x}_j)^2\right]^{1/2}} \quad (1)$$

where x_{it} and x_{ji} denote the returns of stocks i and j at time t respectively, and $\overline{x_i}$ and $\overline{x_j}$ are the average returns for stocks i and j over all time periods 1 to n (Frades and Matthiesen, 2010). Pearson's correlation coefficient quantifies the tendency for objects to increase or decrease together in a linear fashion. It was used by Da Costa *et. al.* (2005) in their presentation of a cluster analysis-based stock selection strategy. The measure tends to detect differences in the shape of two data objects rather than differences in magnitude, and is therefore very sensitive to outliers (Grabusts, 2011).

Another potential drawback of Pearson's correlation coefficient is that it assumes that the underlying data is approximately normally distributed, and may not give robust results for non-Gaussian data (Frades and Matthiesen, 2010). The monthly returns underlying this study are approximately normally distributed, but exhibit higher kurtosis and fatter tails than a normal distribution would suggest. Given these concerns, Pearson's correlation coefficient will be tested against two other popular proximity measures.

Euclidean Distance: $\sqrt{\sum_{t=1}^{n}(x_{it} - x_{jt})^2}$ (2)

Also known as L_2 distance, Euclidean distance has been the proximity measure of choice in the majority of prior research into cluster analysis of financial time series data (Wittman, 2002, Gavrilov *et al.*, 2000, Manniste *et al.*, 2011, Liao *et al.*, 2008) It calculates the geometric distance between two objects and is based only on magnitude, so tends to produce circular clusters. This measure can be a poor reflection of stocks which are far apart but highly correlated, since this correlation is not taken into account (Frades and Matthiesen, 2010). Another possible drawback of Euclidean distance is that it tends to give undue weight to outlying values (Cormack, 1971). Additionally, the Euclidean distance between stocks with similar betas or volatilities is usually small, and so it tends to cluster such stocks together (Tan, 2002).

Manhattan Distance: $\sum_{t=1}^{n}|x_{it} - x_{jt}|$ (3)

This measure, which is also known as the city-block metric, aims to minimise the sum of L_1 distances of an object to its cluster centroid (Tan *et al.*, 2005). It tends to dampen large differences in returns, and therefore gives greater weight to small differences (Tan, 2002). This measure was used by Craighead and Klemesrud (2003) in their derivation of a stock selection strategy based American stock return series.

2.2 Agglomerative hierarchical clustering

In order to implement the agglomerative hierarchical clustering framework presented in Appendix A1 it was necessary to choose a linkage function. Many such functions exist in the literature, but this study will focus on the four most widely cited methods presented in the appendix; namely the nearest neighbour, furthest neighbour, group average and Ward (1963) linkage functions.

Mathematically, the nearest neighbour method is the most tractable linkage technique, and is invariant under monotonic transformations of the proximity matrix (Everitt, 1974). The method's drawback is that it tends to produce elongated clusters. Additionally, to the extent that there are bellweather stocks, or lead-lag relationships amongst stocks, the nearest neighbour method is unsuitable for the analysis of stock clusters (Tan, 2002). The method was therefore omitted from the analysis.

The furthest neighbour, group average and Ward linkage functions tend to produce spherical clusters. Ward's method is generally very efficient, and is more likely to produce smaller clusters than the other two (Tan *et al.*, 2005). Overall and Magee (1992) found that Ward's method with Euclidean distance produced superior results for hierarchical clustering, but conceded that it favours equal-sized clusters which is not necessarily the case in the JSE data set. Since no clear evidence exists in the literature as to which of these methods is most suitable for clustering financial time series data, all three of these linkage functions were tested on the data set.

There are a number of drawbacks associated with hierarchical clustering. First, the method is expensive in terms of its computation and storage requirements because it produces a full nested structure of clusters with each iteration (Tan *et al.*, 2005). The method is therefore ill-suited to working with large or high-dimensional data sets.

A second critique of agglomerative hierarchical algorithms is that merge decisions are irreversible (Everitt, 1974). This is particularly problematic with noisy and high-dimensional data and means that, although the optimal decision is made at each step in the process, it is not guaranteed that the overall result will be an optimal clustering hierarchy.

The final drawback of the method is that it requires a subjective decision regarding which stage of the hierarchical clustering output is optimal (Everitt, 1979).

2.3 K-means clustering

Three decisions were required before the K-means methodology presented in Appendix A2 could be implemented. First, a centroid initialisation technique needed to be chosen. Second, a decision was needed regarding the treatment of centroids to which no clusters were assigned. Third, the ‘K’ parameter needed to be assigned a value.

A review of the K-means literature revealed countless possible techniques for selecting initial centroids. In their comparison of a number of methods, Pena *et. al.* (1999) found that the repeated random initialisation technique discussed in Appendix A2, as well as the method suggested by Kaufman and Rousseeuw (1990), generally produced superior clustering results. Since the repeated random initialisation method is more widely used, this is the technique that was implemented in the study.

The second decision that needed to be made before the K-means algorithm could be applied to the data was how to deal with centroids to which no objects were assigned. Since the aim of the K-means algorithm is to produce a clustering with the lowest possible Sum of Squared Errors (SSE), this was achieved by choosing the replacement centroid from the cluster with the highest SSE.

The third decision was arguably the most important decision of them all, because the value of K determines the number of clusters that will be produced by the algorithm. Since the algorithm will adhere to this value regardless of how appropriate it is for the data set under consideration, its accuracy is important. It was decided to set the value of K equal to the number of clusters suggested in the first part of the analysis: namely the application of a stopping rule to a hierarchical clustering algorithm. This process has the aim of uncovering the true number of clusters in the data set, and its output is therefore the best guess for the value of K.

The most serious drawback of the K-means algorithm is that the user is required to specify the number of clusters in advance (Frades and Matthiesen, 2010). Not only does this affect the quality of the clusters produced, but it also influences the method's rate of convergence and its ability to handle noise.

Another disadvantage of K-means is that the algorithm's results are very sensitive to the choice of initial centroids – particularly in smaller data sets (Mooi and Sarstedt, 2011). The robustness of the method is further diminished by the fact that the clustering results are sensitive to the order in which data points are chosen (Chen *et al.*, 2004).

2.4 DBSCAN

DBSCAN is a density-based clustering method which arose in an attempt to address the fact that neither the K-means nor agglomerative hierarchical algorithms can automatically determine the number of clusters in a data set (Ester *et al.*, 1996). Before running DBSCAN, the user is required to specify values for the algorithm's two parameters, ϵ and MinPts, which are defined in Appendix A3. Together these parameters describe the minimum requirements that must be met in order to form a cluster. Unlike the "K" parameter of K-means, however, determining reasonable values for ϵ and MinPts is not an intuitively simple task.

Ester *et. al.* (1996) proposed a simple heuristic for determining the appropriate values of the ϵ and MinPts parameters for 2-dimensional data. The heuristic is formulated so as to base the parameter values on the least dense cluster in the data set. This method was used as follows to determine the value of ϵ and MinPts for each of the three distance measures used in this study.

First, the distance from each stock to its 4th nearest neighbour (which will be called the 4-dist, for brevity) was calculated. Ester *et. al.* (1996) advocates the use of the 4th nearest neighbour for 2-dimensional data sets because values higher than this can become too computationally complex for large data sets. They assert that the results do not differ meaningfully from those which use the 4th nearest neighbour.

Second, the 4-dist values were sorted in descending order, and plotted to create what has been termed a "sorted 4-dist graph" (Ester *et al*., 1996). The sorted 4-dist graphs for each proximity measure are presented in Appendix B. Each graph was visually inspected for a "threshold" point, which is essentially the point at which the gradient begins to flatten out as one moves from left to right along the graph. The MinPts parameter was set equal to 4, while ϵ was set equal to the 4th nearest neighbour distance of the threshold point.

The suggested parameter values for each proximity measure are presented in Table 1. The threshold points for the Euclidean and Manhattan distance measures were immediately apparent from their sorted 4-dist graphs. On the sorted 4-dist graph for Pearson's correlation coefficient, however, there was no clearly discernable threshold point. This suggests that the density distribution of the data set is fairly uniform under this measure of proximity, and that the clustering results that it generates are unlikely to be useful.

Table 1: DBSCAN parameters suggested by Ester *et. al.*'s (1996) heuristic compared to those actually used in this analysis

	Suggested values		**Implemented values**	
	MinPts	ϵ	**MinPts**	ϵ
Euclidean	4	1.20	4	1.15
Manhattan	4	10.07	4	9.8
Pearson's correlation	4	Indeterminate	n/a	n/a

The suggested parameter values were then tested on the full data set. In the case of Pearson's correlation coefficient, a wide range of ϵ values were considered. All of them, however, yielded either a single, all-inclusive cluster or a single smaller cluster and many noise points. Neither of these are satisfactory outcomes, and the proximity measure was therefore deemed inappropriate for use with the DBSCAN algorithm.

The suggested ϵ values for the Euclidean and Manhattan distance measures generated more promising results, although both methods produced single-stock clusters. Although in certain instances stocks can be driven by idiosyncratic factors that would result in singleton clusters such as these, it would be an unlikely outcome for any of the stocks in this data set over an almost ten year period. These single-stock clusters were eliminated when the values of the ϵ parameters were lowered slightly relative to the suggested values. The implemented values of ϵ shown in Table 1 were found to produce more equally-sized clusters without significantly increasing the number of noise points.

Despite its attempts to improve upon the shortcomings of K-means and agglomerative hierarchical clustering, DBSCAN is not without its own drawbacks. Importantly, although the user is not required to specify the number of clusters in advance, the outcome of the algorithm is still parameter-dependent and the outcome is not always deterministic (Tan *et al.*, 2005). The need to select and adjust the highly sensitive values of ϵ and MinPts has a strong influence on the resulting number of clusters, as well as the method's ability to handle noisy data. In this vein, it is also important to note that, although DBSCAN copes well with noise where K-mean does not, it is ill-equipped to handle true outlying data points, or data sets of varying density (Vijendra, 2011).

Chapter 3

Data

For the purposes of this analysis, the 100 largest stocks on the JSE by market capitalisation as at February 2013 were selected. Those which had not been listed since July 2003 were then excluded, leaving a total of 86 stocks in the data set.

In a cluster analysis, feature selection is the process of identifying the most important attributes of each data point for inclusion in the analysis (Frades and Matthiesen, 2010). For this study, monthly total returns, adjusted for corporate actions, were collected for each of the stocks for the period July 2003 to February 2013. Monthly returns were used instead of month-end prices as a means of standardising the data. The resulting data set consisted of 116 monthly returns for 86 shares. The full list of shares, as well as their sector and industrial classifications can be found in Table C1 of Appendix C.

The use of monthly data, rather than weekly, daily or even intra-daily returns, is one of the limitations of this study. Although a lower sampling frequency reduces the dimensionality of the data set, it can also distort the shape of the time series (Fu, 2011). For example, Hendricks *et. al.* (2014) noticed that clear clustering patterns appear daily on the JSE at the times that the UK and US markets open. These, and many other nuances, are lost through the use of monthly returns.

3.1 Industrial composition of the share universe

The universe of stocks in the data set has representatives from all nine JSE sectors. Its composition, which is fairly representative of the JSE as a whole, is depicted in Figure 1. Stocks on the JSE are divided into these sectors based on the firm's core business activity, as measured by before tax profits (Van Rensburg, 2002). Since returns form the basis of these sectors, it is therefore reasonable to expect that stocks in the sample will cluster based on their classification.

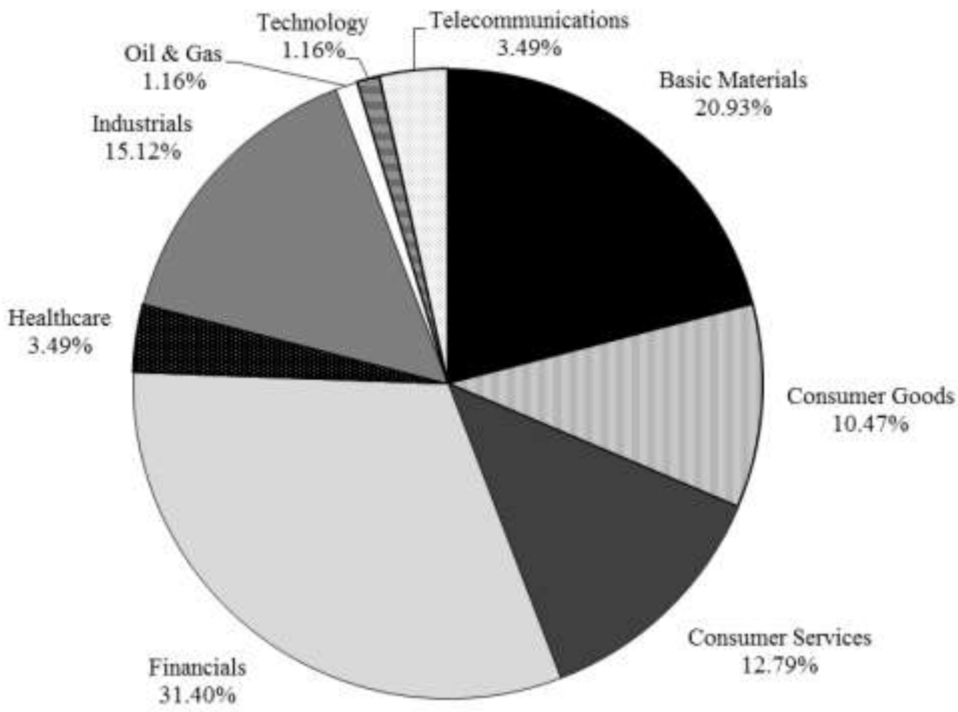

Figure 1: Sector composition of the data set by number of shares

In South Africa, however, the picture may not be quite this simple. The domestic equity market is often characterised by diverging behaviour between financial and industrial stocks and resource stocks (Hendricks *et al.*, 2014). This was observed by Page (1986), who found that the macroeconomic variables underlying the return generating process of JSE stocks can be divided into those that influence the resource sector, and those that influence the financial and industrial sector. The finding was reaffirmed by Van Rensburg (2002), who's factor analysis determined that the Financial and Industrial Index (CI12) and the Resources Index (CI11) were the best observable proxies for the first two factors extracted on the JSE.

This dichotomy in the South African market exists because the JSE All Share Index is a composite of these two contrasting sources of variations in returns. The core business of resource companies is the production and sale of commodities (Carroll and Rouseau, 1999). They generate the majority of their revenue abroad and their earnings are influenced by the Rand-dollar exchange rate and global economic growth (Van Rensburg, 2002). Financials and industrial firms, on the other hand, are more strongly influenced by domestic drivers. Individual securities on the exchange are influenced by either one of these factors, but seldom by both (Van Rensburg, 2000).

3.2 Treatment of outliers

Many clustering procedures are sensitive to the existence of outliers (Tan *et al.*, 2005). Algorithms that are unable to detect outlying data points can produce suboptimal cluster results which are not representative of the true structure underlying the data. The K-means and agglomerative hierarchical algorithms are examples of two such procedures. Neither was designed to be able to identify outliers in data sets, and as such they can produce distorted clustering results if outliers are not removed from the data set (Tan *et al.*, 2005).

K-means is sensitive to outliers on two fronts. If an outlying data point is selected as an initial centroid, it can prevent the algorithm from ever discovering the true cluster centres and distort the entire clustering output (Cordeiro de Amorim and Mirkin, 2010). Furthermore, outliers can unduly influence the location of centroids as the algorithm progresses, which can further distort the results.

It was therefore important to remove outliers from the JSE data set before the K-means and hierarchical procedures were applied. But this was important not only for the optimal functioning of the hierarchical and K-means algorithms. It was also important because it identified stocks whose behaviour is significantly different from the rest of the market. These stocks can then be analysed on their own so as to determine their unique drivers.

To this end, an approach similar to that used by Craighead and Klemesrud (2003) was employed. They ran the Partitioning Around Medoids algorithm (Kaufman and Rousseeuw, 1990) with Manhattan distance and five clusters on their financial data set, and assumed that any one-stock clusters were aberrations which should be removed. Since the true number of clusters in the JSE data set is unknown, a derivation of this procedure was used and instead a hierarchical clustering of the data set was performed. Hierarchical algorithms using the furthest neighbour and Ward (1963) linkage functions were run once for each of the chosen proximity measures, and the output was then examined for the existence of any single-stock clusters which persisted throughout the majority of the hierarchy.

The shares which persisted as single-stock clusters longest under each method are shown in Table 2. The table shows the hierarchical level, represented by the number of clusters at that level, to which each share existed as a single-stock cluster. For example, in the hierarchical clustering with Euclidean distance measure, PAM remained as a single-stock cluster until there were only three clusters left, after which it was incorporated into a cluster with other stocks. Only those single-stock clusters which still existed when there were 20 or fewer clusters in the hierarchy are shown. These are the single-stock clusters which differ most from the rest of the share universe, and which may distort the clustering results.

Each method generates similar results – a good sign that the stocks are true outliers which can be detected by various measures. PAM, EHS, PSG, ASR and DTC are the most outlying stocks according to both furthest neighbour methods. These same five stocks are found within the first six outliers of the Ward algorithm with Euclidean distance measure. The Ward method with Manhattan distance generates many more single-stock clusters than any of the other methods. The order of its results do not match those of any of the other methods, but PAM, EHS, ASR and DTC are single-stock clusters that remain into the last 20 levels of the hierarchy.

Table 2: Hierarchical level to which single-stock clusters remain under different linkage functions and proximity measures

	Ward linkage		Furthest neighbour linkage	
Number of clusters	Euclidean distance	Manhattan distance	Euclidean distance	Manhattan distance
1	-	-	-	-
2	-	PAM	-	-
3	-	HAR	PAM	PAM
4	-	EHS	-	-
5	PAM	NHM	-	EHS
6	EHS	GFI	EHS	-
7	-	LON	PSG	PSG
8	-	ANG	ASR	-
9	-	EXX	DTC	ASR
10	DTC, NHM	ASR	EXX	-
11	ASR	-	SAP	DTC
12	PSG	ACL	LON	ASR
13	EXX	ARI	-	LON, NHM
14	LON	AMS	-	EXX
15	SAP	DTC	NHM	-
16	ACL	IMP	ACL	-
17	-	AGL	-	-
18	-	AVI	-	TKG
19	HAR	BAW	-	-
20	GND	BIL	HAR	CML

There are no columns for Pearson's correlation coefficient in Table 2, because no single-stock clusters persisted to fewer than 25 clusters under this measure, for either of the linkage functions. According to this proximity measure no stock is so significantly different from the rest that it should be removed from the data set. This is unsurprising given the finding in section 2.4 regarding the relatively uniform distribution of stocks according to this proximity measure. This is likely due to the fact that most stocks in any particular market tend to move in the same direction over the longer term. The result is that every stock has some minimum level of correlation to every other stock, and there is less chance of outliers existing in the data set.

Based on these results, five stocks were removed from the database: PAM, EHS, PSG, ASR and DTC. The exclusion of DTC was expected because it is the only technology share in the data set and therefore subject to a unique set of drivers. Three of the stocks – PAM, EHS and ASR – are volatile stocks from the basic materials sector which showed particularly poor performance during the 2008 financial crisis. The emergence of PSG as a significantly outlying stock was somewhat surprising. Upon further inspection, however, it was discovered that it had the third highest annualised growth rate for the period under investigation, as well as a higher than average volatility. These factors, as well as that share's worse than average response to the financial crisis, likely contributed to its outlying status.

Although more stocks could have been removed on the basis of this analysis, it was decided to limit the number to five. It was reasoned that by removing at least five outliers, there were unlikely to be any single-stock clusters in the hierarchical and K-means outputs - assuming the number of clusters in the data set is close to the number of JSE sectors represented in the data set. Removing any more outliers, however, would have reduced the data set by too great a margin.

Going forwards it can be assumed that whenever the hierarchical or K-means algorithms are used, they are used on the data set with outliers removed. It is important to note, however, that outliers were not removed from the data set before the application of the DBSCAN procedure. DBSCAN is formulated so as to automatically detect outlying data points and label them as noise (Ester *et al.*, 1996). By leaving outliers in the data set, it was possible to analyse the DBSCAN output to see whether it correctly identified the outlying stocks.

Chapter 4

Determining the number of clusters in the data set

The first main step in this analysis was to determine the number of clusters in the JSE data set. This was done by applying a stopping rule to a hierarchical clustering of the data set in order to identify the optimal partitioning in the hierarchy.

In the context of this study, uncovering the number of clusters in the data set is interesting for two reasons. Firstly, it is used as a measure of the validity of the hierarchical clustering results. The clusters are expected to be broadly aligned to the sector classifications detailed section 3.1, so any significant deviations from this structure suggests an inappropriate choice of algorithm or clustering method. Secondly, the suggested number of clusters is used as an input to the K-means algorithm, and the result is therefore key to its successful implementation.

4.1 Methodology

One of the primary uses of cluster analysis is to infer the number and nature of distinct groups in a population (Atlas and Overall, 1994). Many partitional procedures, however, require the user to specify the number of clusters in advance. Even in hierarchical clustering the user is given no information about which level provides the best clustering solution. In situations where the user has no prior knowledge of the data set, this can negate the usefulness of cluster analysis.

In an attempt to address this problem, numerous studies have proposed rules for determining the number of clusters in a data set. When these rules are applied to hierarchical clustering procedures, they are referred to as stopping rules. Milligan and Cooper (1985) conducted perhaps the most extensive comparison of such procedures to date, examining the performance of 30 stopping rules in determining the number of clusters in an artificial data set with well-separated clusters. They found the rules proposed by Calinski and Harabasz (1974) and Duda and Hart (1973) to most reliably return the correct number of clusters in data sets containing distinct, non-overlapping clusters. Milligan and Cooper (1985) did, however, caution users that the performance of these rules may be data dependent.

Atlas and Overall (1994) subsequently compared Calinski and Harabasz' (1974) method to a split-sample replication stopping rule proposed by Overall and Magee (1992) under conditions of varying cluster numbers and varying degrees of overlap. The performance of Overall and Magee's rule was found to be superior in scenarios of increasing cluster overlap. Since the JSE data set is expected to exhibit overlapping clusters of various sizes, this stopping rule is likely to produce a more accurate assessment of its true number of clusters. This split-sample replication rule, however, is based on the assumption that independent sub-samples can be drawn from the underlying population. This assumption does not hold in a financial time series, whose successive observations cannot be assumed to be independent.

Some of the more recent methods for determining the number of clusters in a data set include those by Salvador and Chan (2003), Dudoit and Fridyland (2002) and Tibshirani *et. al.* (2001). Dudoit and Fridyland compared their prediction-based resampling method to Tibshirani *et. al.*'s gap statistic, and found it to be more robust and accurate. The drawback with Dudoit and Fridyland's method is that it requires the existence of both a test and a learning data set, which would require splitting the JSE data set in half. Since a single business cycle typically lasts more than five years, doing this would mean that neither data set is exposed to a full cycle, which could influence the relationships uncovered in between the stocks. Salvador and Chan's (2003) method does not require two data sets, and was shown to perform well in data sets of varying size, number of clusters, separation of clusters, density and number of outlier.

This study therefore employed the stopping rule proposed Salvador and Chan (2003), which they named the L method. The L method determines the number of clusters in the data set by finding the “knee”, or point of maximum curvature, in a graph with the number of clusters on the x-axis and the clustering evaluation metric on the y-axis. The method relies on the fact that usually the regions to the left and right of the knee on this evaluation graph are approximately linear – the area to the right comprising relatively homogenous clusters, and the area to the left made up of increasingly dissimilar data points. It is therefore possible to locate the knee by finding the pair of straight lines that best fit the evaluation graph.

Mathematically, this is equivalent to finding the pair of straight lines which minimise the total Root Mean Square Error (RMSE) when fitted to the evaluation graph. If the x-axis of the evaluation graph varies from 2 to b, and the left and right straight lines, L_c and R_c, are partitioned at data point c (such that L_c is fitted to data points $x = 2, 3 \dots c - 1$ and R_c is fitted to data points $x = c \dots b$), then the location of the knee, $\tilde{c}$, is calculated as follows:

$$\tilde{c} = argmin_c \, RMSE_c \,, \quad \text{where}$$

$$RMSE_c = \frac{c-1}{b-1} \times RMSE(L_c) + \frac{b-c}{b-1} \times RMSE(R_c) \qquad (4)$$

The L method produces the best results when the length of the left and right lines is reasonably similar. This will typically not be the case because the knee usually lies to the left of the evaluation graph. In the case where the right line is much longer than the left line, the algorithm will most likely locate a knee that is larger than the actual number of clusters in the data set. To accommodate this scenario, the algorithm “iteratively refines the knee by adjusting the focus region and reapplying the L method” (Salvador and Chan, 2003). This essentially means that the method is applied over a smaller and smaller portion of the evaluation graph (although the focus region is not allowed to be less than 20 data points), resulting in better fitting left and right lines each time. The iteration stops when the method returns the same value for the knee as then previous iteration.

The L method is a global method, and as such it is not sensitive to the existence of outliers. It is also able to find knees in the presence of jumps in the data, and is highly efficient because it requires only one run of a hierarchical clustering algorithm.

Salvador and Chan’s (2003) L method was run on the hierarchical clustering results from all permutations of the furthest neighbour, group average and Ward linkage functions and the three proximity measures. The application of the method to multiple hierarchical clustering algorithms allowed the method’s robustness to be assessed. The result of each algorithm was then compared, and the validity of the clusters members at the suggested number of clusters was evaluated.

4.2 Analysis of results

Salvador and Chan’s (2003) L method yielded a variety of results across the different linkage functions and proximity measures. The location of the knee for each different implementation of the hierarchical algorithm is shown in Table 3. The variety of results suggests that the L method does not work equally well in all situations.

The analysis of the share universe that was done in section 3.1 revealed that nine JSE sectors were originally represented in the data set. The technology sector, which consisted of a single share, was subsequently removed from the data set during the outlier analysis performed in section 3.2. It is therefore reasonable to assume that the true number of clusters in the data set is close to eight. This excludes 13, 18 or 38 from being the true number of clusters in the data set - a finding which is supported by the fact that these results contain many single-stock clusters, an undesirable outcome in a clustering of stocks.

Table 3: Number of clusters found by the L method (Salvador and Chan, 2003) for different combinations of linkage function and proximity measure

	Euclidean distance	Manhattan distance	Pearson's correlation coefficient
Ward	5	13	38
Furthest neighbour	4	4	4
Group average	6	5	18

Indeed, the L method does not appear to produce robust results when either Ward's method or Pearson's correlation coefficient similarity function is implemented as part of the hierarchical clustering algorithm. The reason is that neither of these methods produce an evaluation graph with a clear 'knee' - instead the graphs slope smoothly from right to left (see Appendix D for all the evaluation graphs forming part of this analysis). Outliers have been removed from the data set, so this is unlikely to be the cause of the problem. Instead, in their tests of the L method, Salvador and Chan (2003) found that the evaluation graphs failed to exhibit a prominent knee in the presence of overlapping clusters. This has already been identified as a weakness with Pearson's correlation coefficient, because all the stocks exhibit a minimum level of correlation with each other as a result of their tendency to move broadly together. It may also be the case with the Ward's method, which favours spherical, equally sized and well-separated clusters (Tan, 2002, Overall and Magee, 1992). The JSE data set's clusters, however, are unlikely to exhibit these qualities.

As an interesting aside, the use of the L method on the hierarchical algorithm implemented using group average and Pearson's correlation coefficient revealed its ability to find the knee even when it is located at a jump in the curve. See Table D3.3 in Appendix D for the evaluation curve depicting this.

The location of the knee is most stable when the furthest neighbour linkage function or Euclidean distance measure is used. The L method performs well because these methods produce evaluation graphs with distinct knees. In these cases the method suggests that the true number of clusters in the data set lies somewhere between four and six clusters – a result which is echoed by the group average linkage method with Manhattan distance. This is reasonably close to the predictions made above, and suggests that the stocks in a number of the JSE sectors exhibit similar behaviour. Given the dichotomous nature of the JSE and the expectation that financial and industrial stocks show similar growth patterns, this would not appear to be an unrealistic number of clusters in the data set.

In order to decide whether four, five or six is the true number of clusters in the data set, the clusters generated at the specified hierarchy under each method were examined. The group average linkage function produced poor clustering results, with both the Euclidean and Manhattan algorithms generating a number of single-stock clusters. Since outlying stocks have already been removed from the data set, a clustering result that is representative of the true underlying structure of the data would not be expected to contain single-stock clusters. The group average hierarchical clustering algorithms were therefore excluded from further analysis.

The Ward method's implementation, as well as all three furthest neighbour methods, produced one large cluster and a number of smaller clusters. Although more equally sized clusters may be preferable for some uses, the pervasiveness of this result strongly suggests that a large contingent of the stocks in the data set behave in a manner similar enough for them to exist in a single cluster.

All methods except that which used the furthest neighbour linkage function and Pearson's correlation coefficient identified a cluster comprising ANG, GFI and HAR. Recall from Table 2 in section 3.2 that these stocks were identified as exhibiting outlying behaviour, but were not deemed to be so significantly different as to be removed from the data set altogether. It is a positive result, however, that they are identified as being distinct from the rest of the stocks. The failure of the furthest neighbour with Pearson's correlation coefficient to identify this cohort is unsurprising given the fact that the Pearson's correlation coefficient did not identify any outliers in section 3.2. This limitation of the proximity measure is a clear drawback in its application to the JSE data set.

Further examination of the furthest neighbour and Manhattan distance method revealed that three of the four clusters identified were made up entirely of basic materials stocks, while the fourth cluster acted as a catch all cluster for the rest of the stocks in the data base. Although basic materials stocks are expected to be clustered separately from the other sectors, a more detailed clustering of the other sectors would be more useful for most purposes.

Of the two remaining implementations, the Ward method with Euclidean distance measure produced superior clustering results to the furthest neighbour method with Euclidean distance. The furthest neighbour results generated a cluster containing both consumer goods and basic materials stocks. This was not seen in any other small clusters across all the implementation techniques, and is therefore considered to be an anomalous result. The clusters produced by the Ward method, on the other hand, showed fairly strong separation of stocks between JSE sectors. Based on these results, it was therefore concluded that the true number of clusters in the data base was five.

Salvador and Chan (2003) note in their analysis of the L method that it is particularly sensitive to the presence of outliers. It might therefore be possible to improve upon the results obtained in this analysis by removing more outlying stocks from the data base. In order to obtain more detail about the differences between the non-basic materials stocks, one could also remove all basic materials stocks from the data set and run the L method on a hierarchical clustering of the remaining stocks.

Chapter 5

Testing the consistency of clusters and cluster members over time

The second step in the analysis was to test the consistency of clusters and cluster members generated by the different clustering algorithms when applied to different periods of the JSE data set. If a clustering algorithm is consistent, it will pick up on the longer term relationships between stocks and produce clusters which are relatively robust to the short term variability of individual stocks. Consistency is important if the user intends to update the clustering results over time.

5.1 Methodology

The data set was split into nine consecutive and non-overlapping 12 month periods, and one further eight month period. Each combination of clustering algorithm, proximity measure and, where applicable, linkage function was then run on each of these 10 time periods. The resulting clusters and cluster members were examined, and the consistency and robustness of the various methods were compared.

The hierarchical clustering algorithms were implemented as described in section 2.2. As per the results from section 4.2, the partitions containing five clusters were used in the analysis. It was decided to omit the group average linkage function from this phase of the investigation because, as demonstrated in section 4.2, this method produced poor clustering results when applied to the whole data set. It is not expected that if the method were applied to shorter time periods it would produce significantly better results, and the group average hierarchical clustering algorithm was therefore not included in the test of cluster consistency over time.

The K-means clustering algorithm was implemented as described in section 2.3. The "K" parameter was set equal to five. The algorithm was run 50 times on each time period, and the partition with the lowest SSE was selected as the best clustering result.

The implementation of DBSCAN, on the other hand, did not exactly follow the implementation described in section 2.4. The ϵ parameters that were determined based on the entire period did not produce meaningful results when applied to the shorter time periods examined in this section of the analysis. This is because the shorter the time period, the less variation each stock is likely to exhibit and the more similar the stocks appear according to the various proximity measures. The result is a smaller 4-dist for each stock. To account for this, the value of the ϵ parameters for the Euclidean and Manhattan proximity measures were reduced. Furthermore, as suggested in section 2.4, DBSCAN was not implemented using Pearson's correlation coefficient.

The heuristic described by Ester *et. al.* (1996) was again used to determine the correct values of the ϵ parameters. This time, separate sorted 4-dist graphs were generated for each time period, for both Euclidean and Manhattan distance measures. A threshold point and corresponding ϵ value were then identified on each graph. The final ϵ values were determined by averaging the ϵ values found in each of the 10 time periods. For the Euclidean proximity measure, the value of ϵ that was used to cluster stocks in each time period was determined to be 0.3. For the Manhattan distance measure, a value of 0.9 was used. In both instances, the value of the MinPts parameter remained four.

Once the clustering algorithms had been run on the 10 time periods, the consistency of the clustering results was analysed. The consistency of the clusters was judged on three factors:

i. The number of clusters generated in each time period.
ii. The size (i.e. number of stocks) of the clusters in each time period.
iii. The composition of the clusters in each time period.

Comparing cluster consistency based on the first two factors was a simple matter of counting. Comparison of the methods based on the third factor, however, required a more sophisticated measure.

There are two well-known classes of techniques for comparing partitions of the same data set. The first class is made up of methods which compare clusters by counting pairs. Numerous such methods exist (Wallace, 1983, Fowlkes and Mallows, 1983, Rand, 1971, Ben-Hur *et al*., 2002, Mirkin, 1996), and they are all based on counting the pairs of data points on which two different partitions do or do not agree (Meila, 2007). The second class of techniques compare clusters by set matching. These methods do not make any assumptions about how the partitions were generated, and are based only on set cardinality (Meila, 2007). Examples of set matching methods have been suggested by Larsen and Aore (1999) and Van Dongen (2000).

The drawback of both of these methods, however, is that they concentrate only on the matching pairs of data points, and ignore what happens to the unmatched parts of each cluster. The technique that was used to analyse the consistency of clusters in this study does not exhibit this weakness. The variation of information "measures the amount of information lost and gained in changing from clustering C to clustering C'" (Meila, 2007). It is based on the concepts of 'entropy' and 'mutual information'. The entropy associated with a particular partition attempts to quantify the uncertainty about which cluster each data point will be in. The entropy associated with a particular partitioning C is calculated as follows:

$$H(C) = -\sum_{k=1}^{K} P(k) log P(k) \qquad (5)$$

where K is the number of clusters in the partition and $P(k)$ is the probability any given data point will be in cluster C, assuming that each data point has an equal chance of being picked (Meila, 2007).

The mutual information between two clusterings C and C' can be thought of as the reduction in uncertainty about which cluster a data point will belong to in partition C, given that we know which cluster it belongs to in partition C' (Meila, 2007). When averaged over all data points, this reduction in uncertainty is equal to the mutual information, $I(C, C')$:

$$I(C, C') = \sum_{k=1}^{K} \sum_{k'=1}^{K'} P(k, k') log \frac{P(k,k')}{P(k)P(k')} \tag{6}$$

where $P(k, k')$ is the probability that a data point belongs to cluster C_k in clustering C and to cluster $C'_{k'}$ in clustering C' (Meila, 2007).

Given these two definitions, the variation of information for two partitions C and C' can be calculated as follows:

$$VI(C, C') = H(C) + H(C') - 2I(C, C') \tag{7}$$

For each clustering algorithm, the variation of information was calculated between the partitions generated in successive time periods. The idea is that if stocks exhibit persistently similar behaviour, they should be grouped together over successive time periods. Of course, this will not always be the case and so there will always be some inconsistency over time, but the measure nevertheless serves as a good means of comparison between the different algorithms.

5.2 Analysis of results

As explained in the previous section, it was decided that the partition containing five clusters would be used from the hierarchical algorithms. Similarly, the K-means algorithms were programmed to generate five clusters during each run. Therefore, by definition, these two methods perform perfectly on the first comparison criteria: the number of clusters generated in each time period.

The DBSCAN algorithm, on the other hand, automatically detects the number of clusters from the data set. It can therefore produce any number of clusters on each run, and as shown in Figure 2 the number generated in each time period did vary. The ability of the algorithm to automatically determine the number of clusters from the data – a feature which, in other circumstances, is considered a significant advantage of the method – can therefore be detrimental to the production of consistent clusters over time.

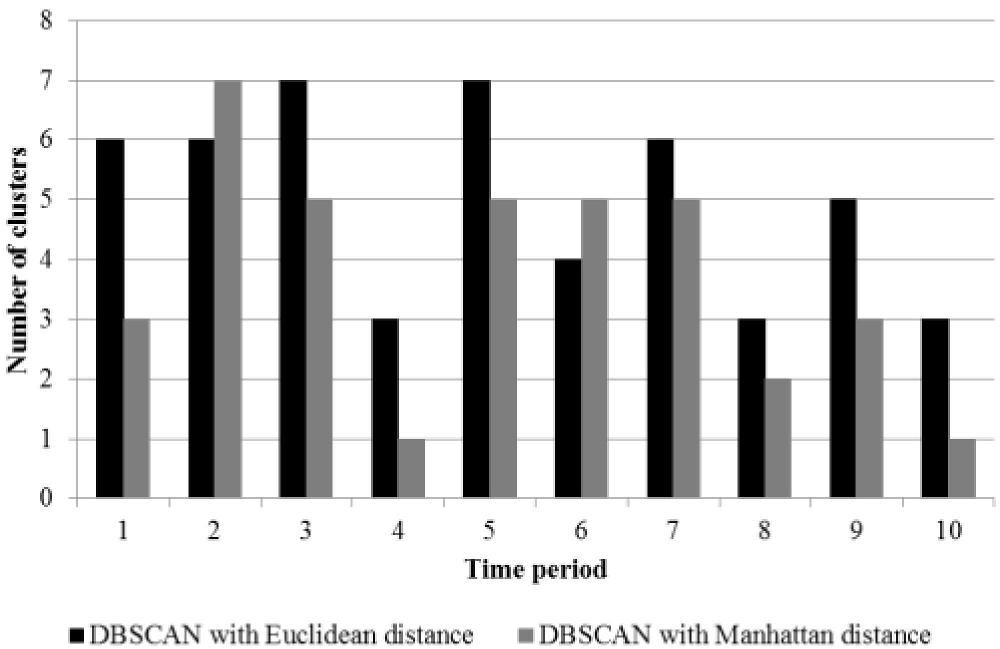

Figure 2: Number of clusters generated in each time period by two DBSCAN algorithms

The second criterion on which the consistency of the clustering algorithms was judged was their ability to generate clusters of consistent size over time. Consistency of cluster size does not refer to clusters being of equal size, but instead that the pattern of cluster sizes in one time period is repeated in successive periods. Furthermore, the consistency of cluster sizes ignores the composition of each cluster, so that two partitions can be deemed consistent based on size, even if the stocks in each cluster differ entirely.

On this basis, the K-means algorithms appear to produce the most consistent results, followed by the hierarchical and then the DBSCAN algorithms (see Appendix E for graphs comparing the size clusters in different time periods for the different methods). Of the hierarchical methods, Ward's method generally performed better than the furthest neighbour methods, and the Manhattan distance measure generally produced clusters of a more consistent size than the other two proximity measures. The differences, however, were not substantial.

Given that they generate varying numbers of clusters over time, it is not surprising that the DBSCAN algorithms perform poorly on this metric. A second factor that contributes to DBSCAN's inconsistent cluster sizes is the fact that it automatically determines the number of noise stocks in the data set. Noise stocks are essentially outliers, and they are not assigned to any cluster. This means that the size of the data set being clustered can change during each time period, which in turn affects the size of each cluster.

Figure 3 shows the number of stocks which were labelled as noise in each time period by the two DBSCAN algorithms. The number of noise stocks appears to be fairly consistent, except during time period six. This period runs from July 2008 to June 2009 and therefore encompasses the worst period of the financial crisis, during which significant losses were suffered by many JSE-listed stocks. This anomalous behaviour resulted in a large number of stocks being labelled as noise points. Although this is detrimental to the consistency of the clustering results over time, it does speak to the accuracy of DBSCAN's clustering results at a single point in time relative to the other two methods. In time period six, the

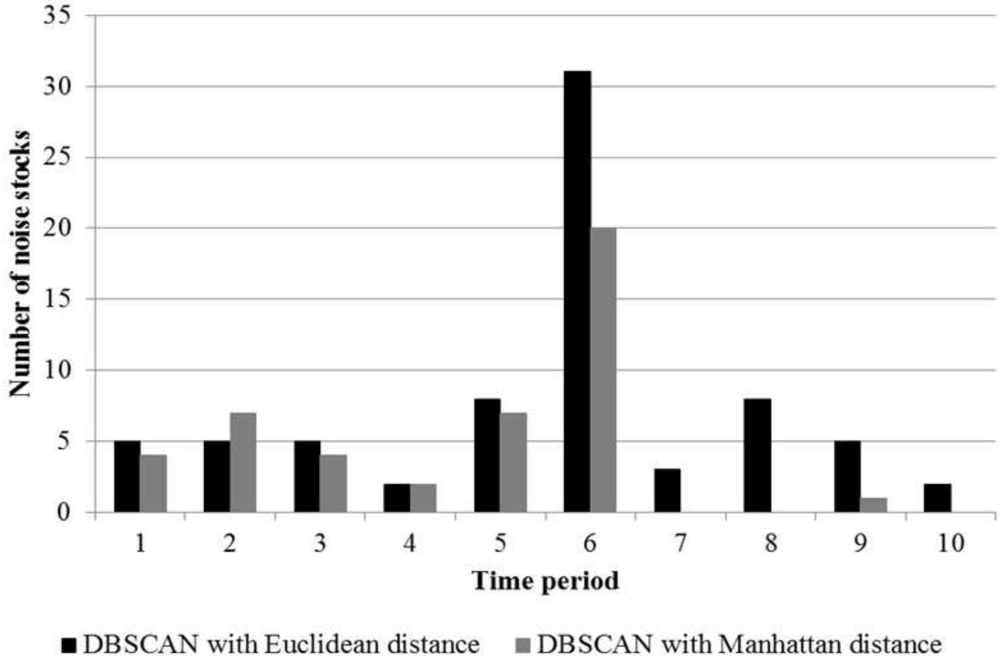

Figure 3: Number of stocks labelled as noise in each time period by two DBSCAN algorithms

hierarchical and K-means methods were required to create what can now be seen as essentially meaningless clusters from these outlying stocks.

The final criterion on which the consistency of the clustering algorithms was assessed was the consistency of cluster members over time, as measured by the variation of information (VI). Figure 4 graphs the average variation of information between the partitions generated in successive time periods by each clustering method (see Appendix F for the VI of each clustering algorithm for each pair of successive time periods). On this basis, Ward's hierarchical clustering with Manhattan distance measure clearly generated clusters with the most consistent members of over time. In this instance, however, the VI measure is misleading. The partitions that this algorithm produced were not useful - consisting typically of four small (often single-stock) clusters, and one large cluster comprising the remainder of the data set. This means that the algorithm essentially identified a few outlying stocks (which, it should be noted, were different in almost all time periods), and gave no further information about the clusters present in the rest of the data. The VI for this algorithm was therefore artificially low, and although its clusters members were consistent, Ward's hierarchical clustering with Manhattan distance would not be recommended as a method for producing meaningful clusters in a financial time series.

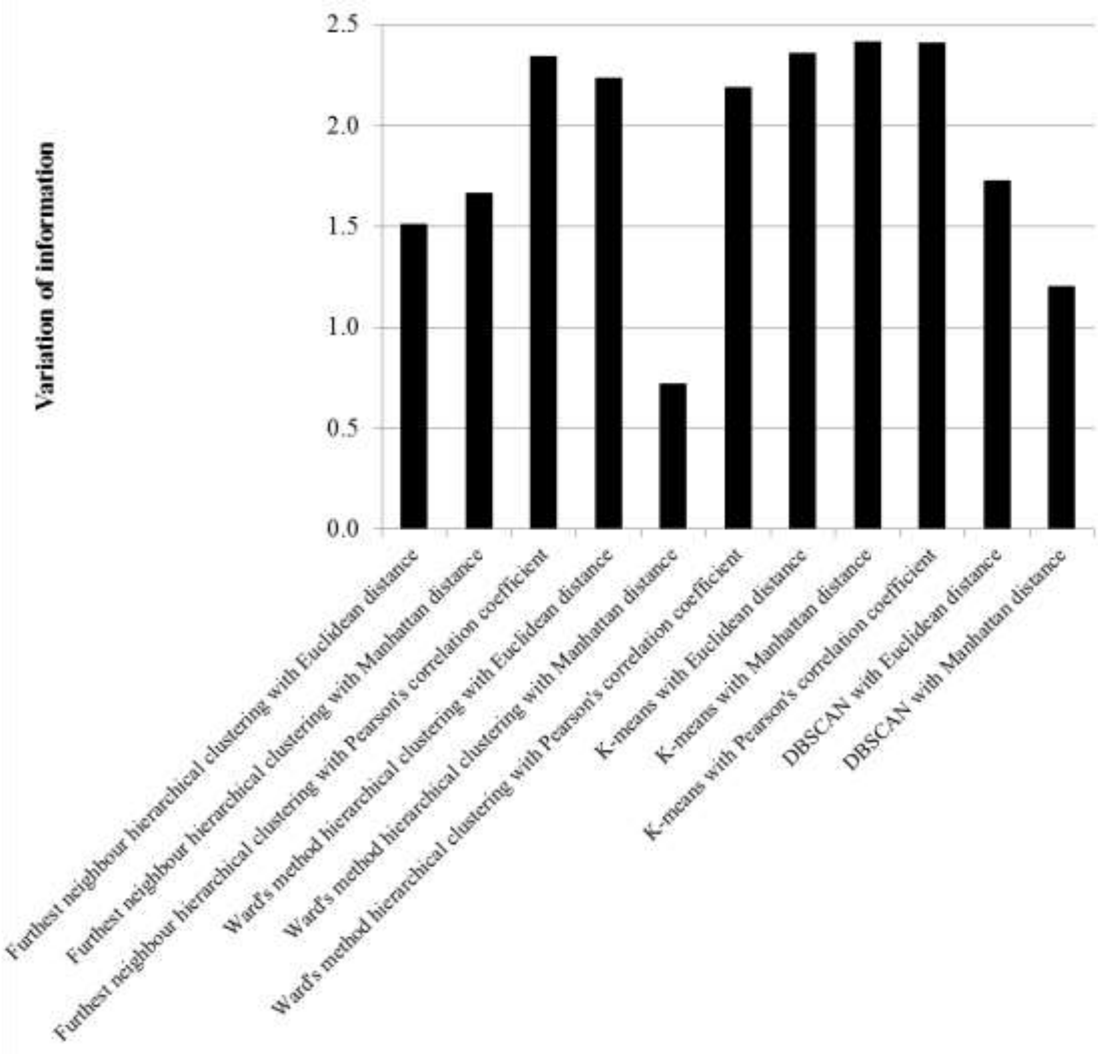

Figure 4: Average variation of information between partitions generated in successive time periods, for various clustering algorithms

The algorithm which produced the clusters with the second most consistent members was DBSCAN with Manhattan distance. This finding, however, may also be misleading because in two time periods this method generated only one cluster. Although this reduced the variation of information, it must be weighed up against the fact that a single cluster is in no way useful to a user looking to group the members of the data set. This, combined with the fact over the whole period there were determined to be five clusters in the data set, suggests that this method does not produce good clustering results. Indeed, the Manhattan method in general does not appear to cluster the data set accurately.

The method that produces the most consistent cluster members, as well as useful clustering results, therefore appears to be the furthest neighbour hierarchical algorithm with Euclidean distance. Across the board the K-means methods had the highest VI values. Thus, although these methods generate equal numbers of clusters of generally consistent sizes, their susceptibility to outliers means that the cluster members are not consistent from one period to the next.

There is scope for more analysis of this topic. A fairer reflection of the consistency of clustering algorithms over time may be produced if the time periods are allowed to overlap, or if the time periods are allowed to be longer. Additionally, the use of weekly data may give a more nuanced view of the similar behaviours between stocks. On the other hand, it may introduce unwanted variation in the data set, which can have a negative effect on certain clustering algorithms.

Chapter 6

Discussion and conclusions

The overarching objective of this paper was to analyse the application of two clustering techniques to the JSE. The first was the application of Salvador and Chan’s (2003) L method stopping rule to a hierarchical clustering of the largest and most frequently traded JSE stocks. The method performed best on the data set when Ward’s method and the Euclidean distance function were used, in which case five clusters were detected. This result is slightly lower than the eight JSE sectors represented in the data set (once outliers are removed), but this is not an unreasonable result in a dichotomous market where two groups of sectors dominate. Indeed, resource stocks (basic materials and oil and gas) were consistently clustered together, as well as separately from other sectors, and a similar pattern was observed for the financial and industrial stocks. These findings, which are consistent with those of previous studies, lend validity to the use of the L method on JSE data.

An implicit question asked in this first part of the study, was whether or not a latent structure did, in fact, exist in the data set. Although the L method cannot be used to reject the null hypothesis of no structure (i.e. only one cluster in the data set), the results generated by the DBSCAN algorithm support the L method’s findings. The average number of clusters generated in each time period by the DBSCAN algorithm with Euclidean distance (the Manhattan distance measure was found to produce suboptimal clustering results) was five. Since DBSCAN automatically detects the number of groups from the data set, this is strong evidence in support of an underlying structure in the JSE.

The second part of the study aimed to analyse the consistency of clusters and cluster members generated by three well-known and widely used clustering algorithms. In this regard, furthest neighbour hierarchical clustering using the Euclidean distance measure proved to produce the most useful and consistent clusters and cluster members. The analysis also suggested, however, that DBSCAN may produce the best clustering results at a given point in time, as a result of its ability to handle overlapping clusters, automatically determine the number of clusters from the data, as well as its ability to automatically label stocks as outliers.

Finally, it should be noted that this was not an exhaustive analysis of the use of cluster analysis on JSE data. Further research will be required using other clustering algorithms as well as other implementations of the clustering methods used here. Researchers should pay particular attention to the newer breed of fast, unsupervised algorithms that do not require parameter selection or adjustments in order to determine the number of clusters. In the financial world of increasingly large and noisy data sets, this is where the need most urgently lies.

Bibliography

APARTSIN, Y., MAYMON, Y., COHEN, Y. & SINGER, G. 2013. Nationality and risk attitude: Testing differences and similarities of investors' behavior in selected financial markets. *Global Finance Journal,* 24**,** 114-118.

ATLAS, R. S. & OVERALL, J. E. 1994. Comparative evaluation of two superior stopping rules for hierarchical cluster analysis. *Psychometrika,* 59**,** 581-591.

BEN-HUR, A., ELISSEEFF, A. & GUYON, I. A stability based method for discovering structure in clustered data. *In:* ALTMAN, R. B., DUNKER, A. K., HUNTER, L. & KLEIN, T. E., eds. Pacific Symposium on Biocomputing, 2002 Lihue, Hawaii. 6-17.

CALINSKI, R. B. & HARABASZ, J. 1974. A dendrite method for cluster analysis. *Communications in statistics,* 3**,** 1-27.

CARROLL, N. & ROUSEAU, R. 1999. What is in a Name? How the Reclassification of the JSE Sectors Will Affect You.: Deutsche Morgan Grenfell.

CHEN, J.-S., CHING, R. K. H. & LIN, Y.-S. 2004. An extended study of the K-means algorithm for data clustering and its applications. *The Journal of the Operational Research Society,* 55**,** 976-987.

CORDEIRO DE AMORIM, R. & MIRKIN, B. 2010. Minkowski metric, feature weighting and anomalous cluster initializing in K-Means clustering. *Pattern Recognition,* 45**,** 1061-1075.

CORMACK, R. M. 1971. A Review of Classification. *Journal of the Royal Statistical Society. Series A (General),* 134**,** 321-367.

CRAIGHEAD, S. & KLEMESRUD, B. 2003. Stock Selection Based on Cluster and Outlier Analysis. *In:* ROSENTHAL, J. & GILLAM, D. (eds.) *Mathematical Systems Theory in Biology, Communications, Computation, and Finance.* 1 ed. New York: Springer.

DA COSTA, N., CUNHA, J. & DA SILVA, S. 2005. Stock Selection Based on Cluster Analysis. *Economics Bulletin,* 13**,** 9.

DRIVER, H. E. & KROEBER, A. L. 1932. Quantitative expression of cultural relationships. *University of California Publications in American Archaeology & Ethnology,* 31**,** 211-256.

DUDA, R. O. & HART, P. E. 1973. *Pattern classification and scene analysis,* New York, Wiley.

DUDOIT, S. & FRIDLYAND, J. 2002. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology,* 3**,** RESEARCH0036-RESEARCH0036.

ESTER, M., KRIEGEL, H.-P., SANDER, J. & XU, X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *Second International Conference on Knowledge Discovery and Data Mining.* Menlo Park, CA: AAAI Press.

EVERITT, B. 1974. *Cluster Analysis,* London, Heinemann Educational Books Ltd.

EVERITT, B. S. 1979. Unresolved problems in cluster analysis. *Biometrics,* 35**,** 169-181.

FOWLKES, E. B. & MALLOWS, C. B. 1983. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association,* 78**,** 553-569.

FRADES, I. & MATTHIESEN, R. 2010. Overview on Techniques in Cluster Analysis. *Bioinformatics Methods in Clinical Research.* Humana Press.

FU, T.-C. 2011. A review on time series data mining. *Engineering Applications of Artificial Intelligence,* 24**,** 164-181.

GAVRILOV, M., ANGUELOV, D., INDYK, P. & MOTWANI, R. Mining the stock market: Which measure is best? Sixth international conference on knowledge discovery and data mining, 2000 Boston, Massachusetts. 487-496.

GEBBIE, T. & POLAKOW, D. 2008. How many independent bets are there?

GRABUSTS, P. 2011. Distance metrics selection validity in cluster analysis. *Scientific Journal of Riga Technical University,* 49**,** 72-77.

HENDRICKS, D., CIESLAKIEWICZ, D., WILCOX, D. & GEBBIE, T. 2014. An unsupervised genetic parallel cluster algorithm for graphics processing units.

KAUFMAN, L. & ROUSSEEUW, P. J. 1990. *Finding Groups in Data: An Introduction to Cluster Analysis,* New York, Jon Wiley & Sons, Inc.

KOGAN, J. 2007. *Introduction to Clustering Large and High-Dimensional Data,* New York, Cambridge University Press.

KRZANOWSKI, W. J. & LAI, Y. T. 1988. A criterion for determining the number of groups in a data set using sum-of-squares clustering. *Biometrics,* 44**,** 23-34.

LARSEN, B. & AONE, C. 1999. Fast and effective text mining using linear time document clustering. *Fifth ACM SIGKDD international conference on Conference on Knowledge Discovery and Data Mining.* San Diego: ACM Press.

LIAO, S.-H., HO, H.-H. & LIN, H.-W. 2008. Mining stock category association and cluster on Taiwan stock market. *Expert Systems with Applications,* 35**,** 19-29.

MACQUEEN, J. Some methods for classification and analysis of multivariate observations. *In:* LECAM, L. & NEYMAN, J., eds. 5th Berkley Symposium on Mathematics, Statistics and Probability, 1967 Berkley, USA. University of California Press, 281-297.

MANNISTE, M., HAZAK, A. & LISTRA, E. Typology of Eurpoean Listed Companies' Reactions to Global Credit Crunch: Cluster Alanysis of Share Price Performance. 3rd International Conference on Information and Financial Engineering, 2011 Shanghai. IACSIT Press, 565-569.

MEILA, M. 2007. Comparing clusterings - an information based distance. *Journal of multivariate analysis,* 98**,** 873-895.

MILLIGAN, G. W. & COOPER, M. C. 1985. An examination of procedures for determining the number of clusters in a data set. *Psychometrika,* 50**,** 159-179.

MIRKIN, B. 1996. *Mathematical classification and clustering,* Dordrecht, Kluwer Academic Press.

MOJENA, R. 1977. Hierarchical grouping methods and stopping rules: An evaluation. *The Computer Journal,* 20**,** 359-363.

MOOI, E. & SARSTEDT, M. 2011. Cluster Analysis. *A Concise Guide to Market Research: The Process, Data and Methods Using IBM SPSS Statistics.* 1st ed. Heidelberg: Springer.

NAGPAL, P. B. & MANN, P. A. 2011. Comparative study of density based clustering algorithms. *International Journal of Computer Applications,* 27**,** 44-47.

OVERALL, J. E. & MAGEE, K. N. 1992. Replication as a rule for determining the number of clusters in hierarchical cluster analysis. *Applied Psychological Measurement,* 16**,** 119-128.

PAGE, M. J. 1986. Empirical testing of the arbitrage pricing theory using data from the Johannesburg Stock Exchange. *South African Journal of Business Management,* 17**,** 78-81.

PENA, J. M., LOZANO, J. A. & LARRANAGA, P. 1999. An empirical comparison of four initilization methods for the K-means algorithm. *Pattern Recognition Letters,* 20**,** 1027-1040.

RAND, W. M. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association,* 66**,** 846-850.

ROMESBURG, H. C. 1984. *Cluster Analysis for Researchers,* Belmont, California, Lifetime Learning Publications.

ROTH, V., LANGE, T., BRAUN, M. & BUHMANN, J. 2002. A resampling approach to cluster validation. *In:* HARDLE, W. & RONZ, B. (eds.) *Compstat.* 1st ed. New York: Physica Verlag Heidelberg.

SALVADOR, S. & CHAN, P. 2003. Determining the Number of Clusters/Segments in Hierarchical Clustering/Segmentation Algorithms. Department of Computer Science, Florida Institute of Technology.

SMYTH, P. Clustering using Monte Carlo cross-validation. *In:* SIMOUDIS, E., HAN, J. & FAYYAD, U., eds. Second International Conference on Knowledge Discovery and Data Mining, 1996 Portland, Oregon. AAAI Press, 126-133.

SPATH, H. 1980. *Cluster Analysis Algorithms for Data Reduction and Classification of Objects,* Chichester, Ellis Horwood Limited.

TAN, M. 2002. Cluster Analysis of Stock Returns. New York: Apothem Capital Management.

TAN, P.-N., STEINBACH, M. & KUMAR, V. 2005. Cluster Analysis: Basic Concepts and Algorithms. *In:* STEINBACH, M. & KUMAR, V. (eds.) *Introduction to Data Mining.* Boston: Pearson Addison Wesley.

TIBSHIRANI, R. & WALTHER, G. 2005. Cluster Validation by Prediction Strength. *Journal of Computational & Graphical Statistics,* 14**,** 511-528.

TIBSHIRANI, R., WALTHER, G. & HASTIE, T. 2001. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology),* 63**,** 411.

VAN DONGEN, S. 2000. Performance criteria for graph clustering and Markov cluster experiments. Centrum voor Wiskunde en Informatica.

VAN RENSBURG, P. 2000. Macroeconomic variables and the cross-section of Johannesburg Stock Exchange returns. *South African Journal of Business Management,* 31**,** 31-43.

VAN RENSBURG, P. 2002. Market segmentation on the Johannesburg Stock Exchange II. *Journal of Studies in Economics and Econometrics,* 26**,** 83-100.

VIJENDRA, S. 2011. Efficient clustering for high dimensional data: Subspace based clustering and density based clustering. *Information Technology Journal,* 10**,** 1092-1105.

WALLACE, D. L. 1983. Comment. *Journal of the American Statistical Association,* 78**,** 569-576.

WARD, J. H. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association,* 58**,** 236-244.

WITTMAN, T. 2002. Time-series clustering and association analysis of financial data. University of Texas, Austin.

Appendix A

Description of clustering algorithms

A1. Agglomerative hierarchical clustering

As the name suggests, hierarchical clustering classifies each group into subgroups, and repeats the process at different levels to form a hierarchy of clusters (Cormack, 1971). All possible numbers of clusters are therefore contained in the output of a single run of the algorithm. The result is a nested set of clusters which can be organised into a tree and represented graphically as a dendrogram, an example of which is shown in Figure A1.1 below. Each node in the dendrogram represents a cluster, and is the union of its children (subclusters). The root of the tree is the cluster containing all of the original objects. (Tan *et al.*, 2005)

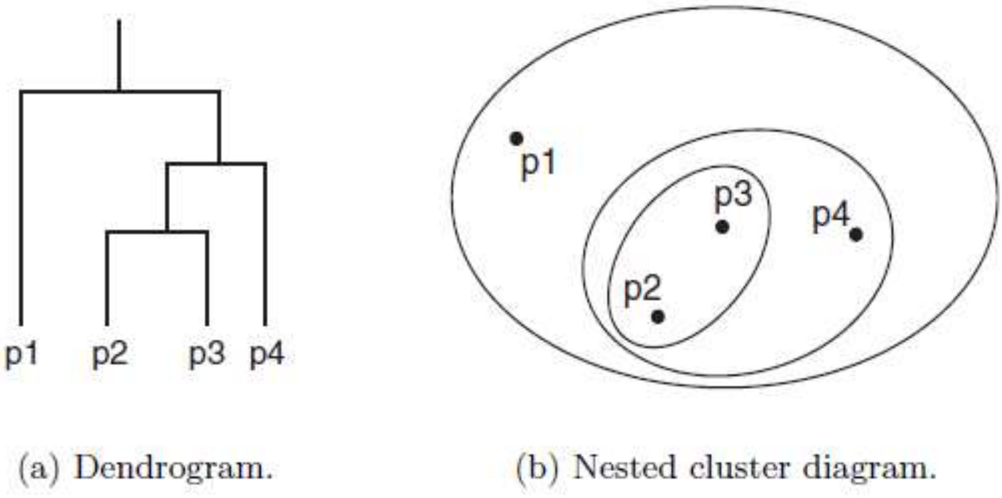

Figure A1.1: Two graphical representations of a hierarchical clustering of four points (Tan *et al.*, 2005)

In general, hierarchical clustering is best suited to data where a hierarchy can be assumed to exist naturally (Everitt, 1974). The method can still be used, however, when no such hierarchy is believed to exist. Numerous methods have been proposed for extracting the optimal number of clusters from a hierarchical clustering (Tibshirani *et al.*, 2001, Tibshirani and Walther, 2005, Krzanowski and Lai, 1988, Salvador and Chan, 2003, Mojena, 1977, Smyth, 1996, Atlas and Overall, 1994, Roth *et al.*, 2002, Dudoit and Fridlyand, 2002).

Hierarchical clustering can be subdivided into two basic approaches: agglomerative and divisive procedures. Agglomerative algorithms are computationally more efficient and are therefore used much more commonly than divisive methods (Spath, 1980).

Also known as bottom-up methods, agglomerative hierarchical algorithms start with each object as an individual cluster, and repeatedly merge the closest pair of clusters. The process continues until a single cluster, containing all the objects in the data set, remains (Salvador and Chan, 2003). The choice of which clusters to merge at each step is governed a linkage function, which uses the chosen proximity measure to define the proximity of two clusters. It is differences in the linkage function which give rise to different agglomerative methods (Everitt, 1974). Four linkage functions in common use are presented below.

The nearest neighbour (also known as single linkage) method defines the proximity of two clusters as the shortest distance between any member of one cluster and any member of the second cluster. Cluster proximity is defined in the opposite way under the furthest neighbour (also known as complete linkage) method - that is, the distance between two clusters' most remote pair of members. In the group average technique, the distance between two clusters is defined as the average distance between all pairs of members of two clusters (Frades and Matthiesen, 2010). Ward's method assumes that clusters are represented by their centroids, and at each step merges the two clusters which will result in the minimum increase in sum of squared errors (SSE) (Everitt, 1974).

Tan (2002) suggested the following basic algorithm for agglomerative hierarchical clustering, which can be applied to any of the aforementioned linkages:

1. Begin with a set S of n objects, and place the elements of S into singleton sets S_1, $S_2 \ldots S_n$
2. Compute the proximity matrix, if necessary
3. **Repeat**
4. Merge the two closest clusters (say S_i and S_j) as indicated by the proximity matrix. Remove S_i and S_j from S and replace them with $S_i U S_j$
5. Update the proximity matrix to reflect the similarities or distances between the new cluster and the original clusters
6. **Until** only one cluster remains

A2. K-means clustering

Developed in the 1967, the K-means algorithm (MacQueen, 1967) is the simplest and most commonly applied partitional clustering technique (Chen *et al.*, 2004). Partitional clustering methods divide data sets into K non-overlapping, mutually exclusive clusters such that each cluster contains at least one object, and each object belongs to at least one cluster (Kaufman and Rousseeuw, 1990). K-means differs from other partitional techniques in that it defines clusters in terms of their centroids.

The basic K-means algorithm can be summarised by the following steps (Frades and Matthiesen, 2010):

1. Specify the value of K
2. Select K data points as initial centroids
3. **Repeat**
4. Form K clusters by assigning each data point to a centroid, so as to minimise the within-cluster variation
5. Recompute each cluster's centroid
6. **Until** the centroids do not change and the clusters converge

The first step in the algorithm requires the user to specify the number of clusters to be produced by the output. Since K-means clustering is intended to be an unsupervised classification method, this represents an obvious weakness in the method.

A variety of techniques exist for the second step in the method. At the most basic level, K data points can be selected at random from the data set (Frades and Matthiesen, 2010). The K-means algorithm, however, is highly sensitive to the choice of initial centroids, and a random initialisation approach can therefore produce highly inconsistent results over repeated runs of the algorithm. A technique commonly used to address this problem is to perform multiple runs of the K-means procedure, each with a new set of randomly chosen initial centroids. The run which produces the clustering structure with the lowest Sum of Squared Errors (SSE) is then chosen as the final output (Tan *et al*., 2005).

The fourth step of the K-means algorithm assigns each data point to a centroid so as to minimise the within-cluster variation, which is represented by the SSE:

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist(c_i, x)^2 \qquad (8)$$

where $dist(c_i, x)$ is a proximity measure, and c_i is cluster centroid. At each repetition of the algorithm, an object is reassigned to another cluster if the move reduces the total SSE.

A problem that can occur at this stage is that empty clusters can be formed if no points are allocated to a centroid (Tan *et al.*, 2005). This will lead to a larger squared error than necessary. A replacement centroid therefore needs to be found. This is often done by choosing the point furthest away from the current centroid, or by choosing the replacement centroid from the cluster with the highest SSE.

The entire process is repeated until convergence is achieved, and neither cluster centroids not the cluster constituents change.

A3. DBSCAN

Density-based spatial clustering of applications with noise (DBSCAN) is one of the most commonly used density-based clustering algorithms. Density-based methods group the objects in a data set into clusters based on the local density conditions (Frades and Matthiesen, 2010). It is the definition of these density conditions which differentiates between most density-based algorithms.

DBSCAN relies on a centre-based notion of density in which the density of a particular data point is calculated by counting the number of objects within a specific radius of that point (Ester *et al.*, 1996). The method requires that the radius parameter, ϵ, be specified in advance by the user.

Before proceeding with the DBSCAN algorithm, it is necessary to define the terms core points, border points, noise, directly density reachable and density reachable. Core points lie inside dense regions of the data set. They are classified as such if the number of objects within the core point's ϵ region exceeds a particular threshold. This threshold, which is the minimum number of points required to form a cluster and is commonly referred to as MinPts, must also be specified in advance by the user. Border points are those points which fall within the neighbourhood of a core point, but do not themselves qualify as core points. It is possible for border points to fall within the neighbourhood of several core points. Noise points are all those objects which are neither a core points nor a border points (Nagpal and Mann, 2011). Figure A3.1 gives a visual depiction of core, border and noise points.

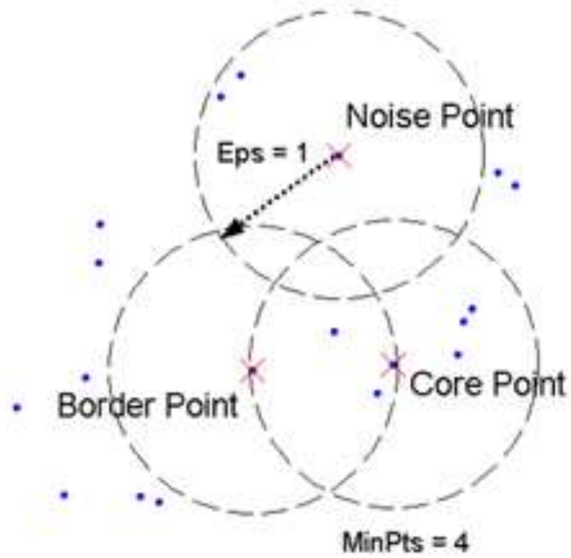

Figure A3.1: A graphical representation of core, border and noise points, as defined by the DBSCAN algorithm

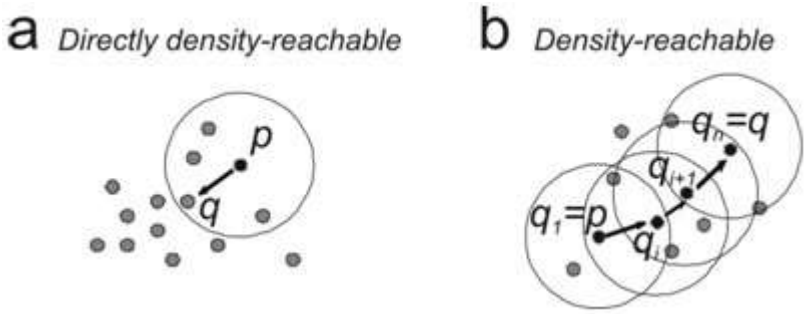

Figure A3.2: A graphical representation of (a) directly density-reachable points and (b) density reachable points

Continuing with the definitions, a point r is directly density reachable from a point s if it within the ϵ of s, and s is a core point. Thus border points are directly density reachable from core points, but not vice versa. A point r is density reachable from a point s if they are connected by a sequence of directly density reachable points (Nagpal and Mann, 2011). See Figure A3.2 for a visual example of points which are (a) directly density reachable, and (b) density reachable from one another.

Given the above definitions, a simplified version of the DBSCAN algorithm can be formulated as follows (Tan *et al*., 2005, Nagpal and Mann, 2011):

1. Specify values for ϵ and MinPts
2. Label all points as core, border or noise points
3. Eliminate noise points
4. **Repeat**
5. Arbitrarily select a point r
6. If r is a core point, identify all points which are density reachable from r and form a cluster
7. If r is a border point, do nothing
8. **Until** all data points have been processed

Essentially, DBSCAN forms clusters of objects with overlapping ϵ neighbourhoods (Vijendra, 2011).

Appendix B

Determining DBSCAN parameter values

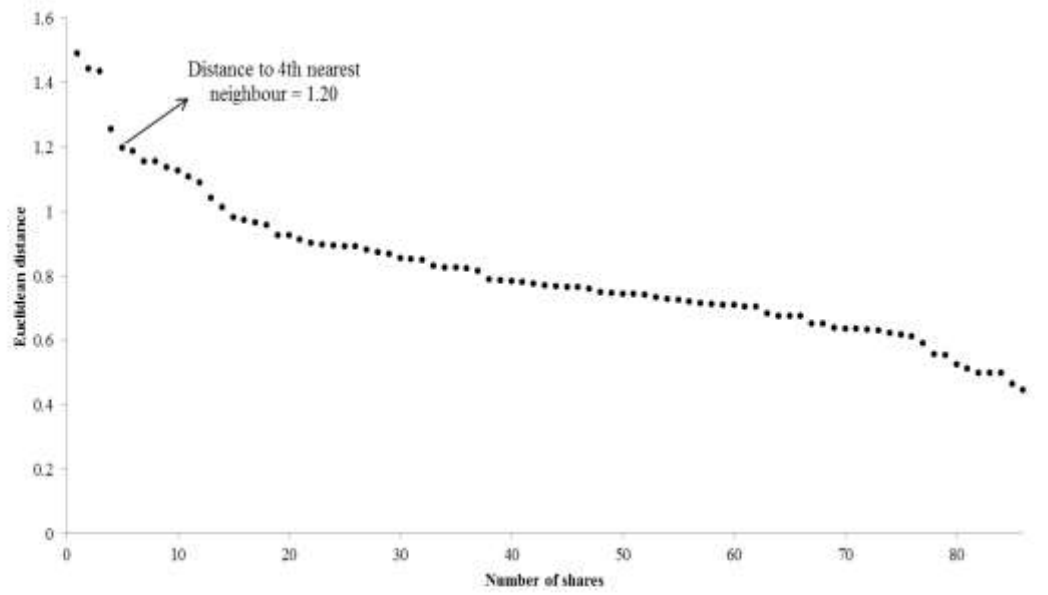

Figure B1.1: Euclidean sorted 4-dist graph

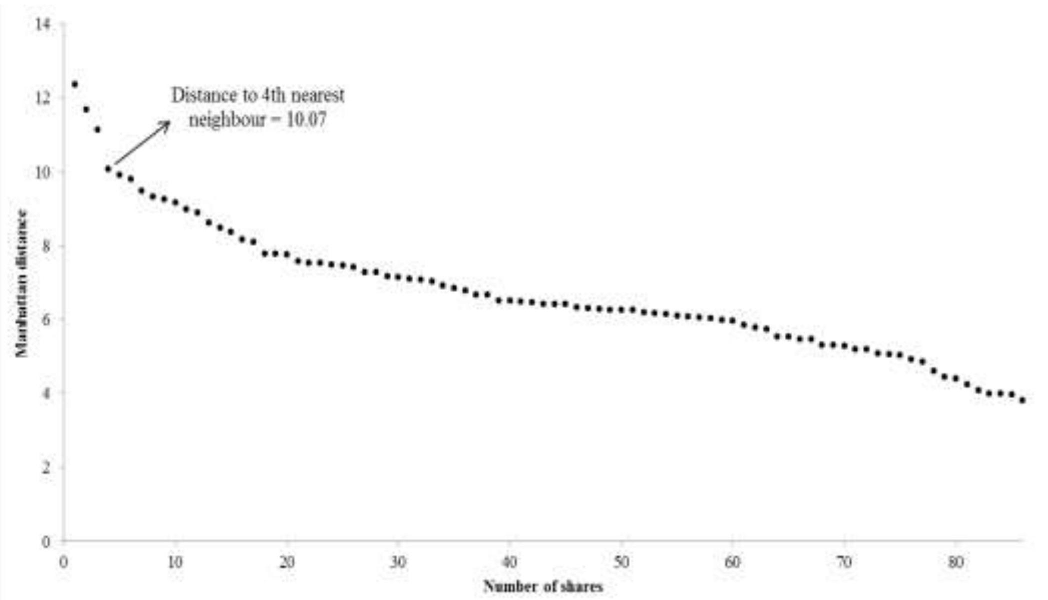

Figure B1.2: Manhattan sorted 4-dist graph

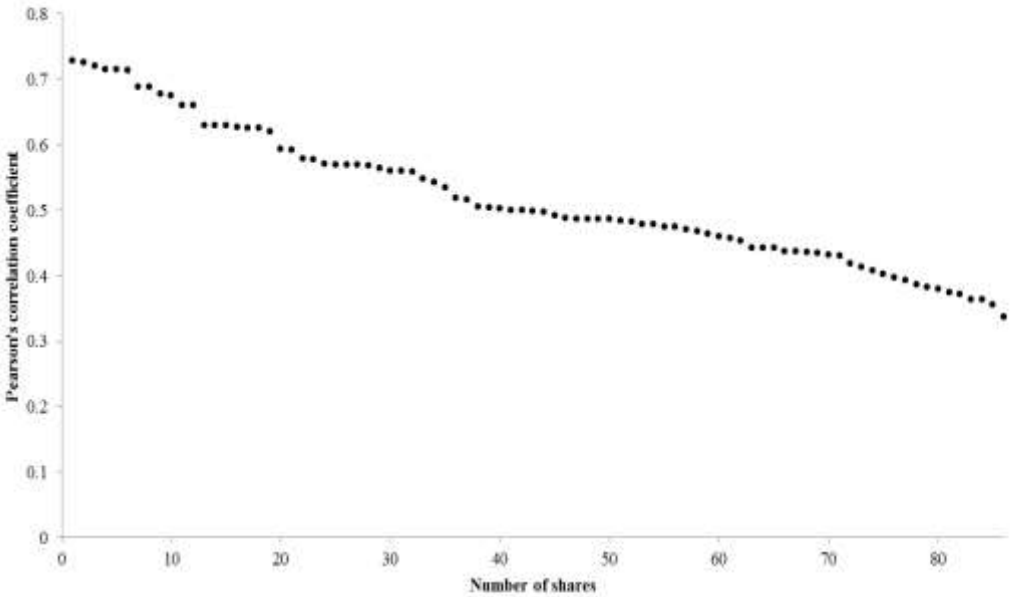

Figure B1.3: Pearson's correlation coefficient sorted 4-dist graph

Appendix C

Classification of shares in the data set

Table C1: Sector classification of shares included in the JSE data set

Ticker	Share Name	Sector
ABL	African Bank Investments Ltd	Financials
ACL	Arcelor Mittal South Africa Ltd	Basic Materials
ACP	Acucap Properties Limited	Financials
AEG	Aveng	Industrials
AFE	AECI	Basic Materials
AFX	African Oxygen Ltd	Basic Materials
AGL	Anglo American PLC	Basic Materials
ALT	Allied Technologies	Telecommunications
AMS	Anglo Platinum Ltd	Basic Materials
ANG	Anglogold Ashanti Ltd	Basic Materials
APN	Aspen Pharmacare Holdings Ltd	Health Care
ARI	African Rainbow Minerals Ltd	Basic Materials
ARL	Astral Foods Ltd	Consumer Goods
ASA	Absa Group	Financials
ASR	Assore Limited	Basic Materials
ATN	Allied Electronics Corporation Ltd	Industrials
ATNP	Allied Electronics Corp Part Prf	Industrials
AVI	AngloVaal Industries ORD	Consumer Goods
BAW	Barloworld	Industrials
BIL	BHP Billiton	Basic Materials
BVT	Bidvest Group	Industrials
CFR	Compagnie Financiere Richemont SA	Consumer Goods
CLS	CLICKS GROUP LTD	Consumer Services

CML	Coronation	Financials
CPI	Capitec Bank Hldgs Ltd	Financials
CPL	Capital Property Fund	Financials
CSO	Capital Shopping Centres Group PLC	Financials
DSY	Discovery Holdings	Financials
DTC	Datatec	Technology
EHS	Evraz Highveld Steel and Van	Basic Materials
EXX	Exxaro Resources	Basic Materials
FPT	Fountainhead Property Trust	Financials
FSR	Firstrand Limited	Financials
GFI	Gold Fields	Basic Materials
GND	Grindrod	Industrials
GRT	Growthpoint Prop Ltd	Financials
HAR	Harmony	Basic Materials
HYP	Hyprop Investments Ltd	Financials
ILV	Illovo Sugar	Consumer Goods
IMP	Impala Platinum Hlds	Basic Materials
INL	Investec Ltd	Financials
INP	Investec PLC	Financials
IPL	Imperial Holdings	Industrials
JDG	JD Group	Consumer Services
LBH	Liberty Holdings Limited Ord	Financials
LON	Lonmin PLC	Basic Materials
MDC	Medi-Clinicrp	Health Care
MMI	MMI Holdings Ltd	Financials
MPC	Mr Price Group	Consumer Services
MSM	Massmart Holdings	Consumer Services
MTN	MTN Group	Telecommunications
MUR	Murray & Roberts	Industrials

NED	Nedbank Group.	Financials
NHM	Northam Platinum	Basic Materials
NPK	Nampak	Industrials
NPN	Naspers	Consumer Services
NTC	Netcare	Health Care
OML	Old Mutual	Financials
PAM	Palabora Mining	Basic Materials
PIK	Pick N Pay Stores	Consumer Services
PPC	Pretoria Portland Cement	Industrials
PSG	PSG Group	Financials
RBW	Rainbow Chicken	Consumer Goods
RDF	Redefine Properties LTD	Financials
REM	Remgro	Industrials
RES	Resilient Prop Inc Fd	Financials
RLO	Reunert	Industrials
RMH	RMB Holdings	Financials
SAB	SABMiller	Consumer Goods
SAC	SA Corporate Real Estate Fund	Financials
SAP	Sappi	Basic Materials
SBK	Standard Bank Group	Financials
SHF	Steinhoff International Holdings	Consumer Goods
SHP	Shoprite	Consumer Services
SLM	Sanlam	Financials
SNT	Santam	Financials
SOL	Sasol	Oil & Gas
SUI	Sun International Ltd	Consumer Services
SYC	Sycom Property Fund	Financials
TBS	Tiger Brands	Consumer Goods
TFG	The Foshini Group Ltd	Consumer Services

TKG	Telkom	Telecommunications
TON	Tongaat Hulett	Consumer Goods
TRU	Truworths International	Consumer Services
WBO	Wilson Bayly Holmes-Ovcon	Industrials
WHL	Woolworths Holdings	Consumer Services

Appendix D

L method evaluation graphs

D1. Ward's method (1963)

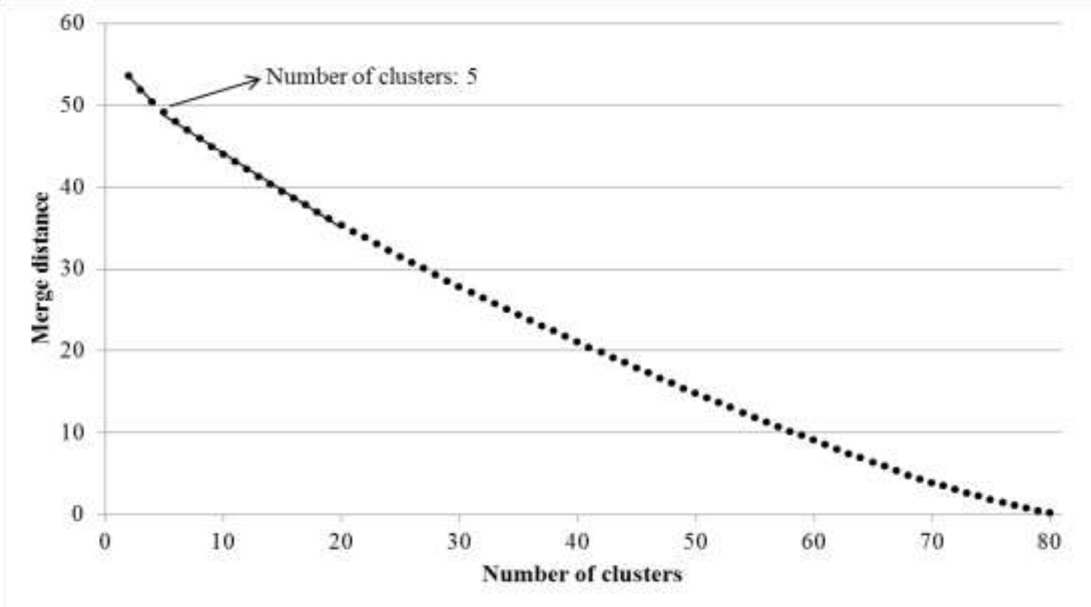

Table D1.1: L method evaluation graph for a hierarchical clustering using Ward's linkage and Euclidean distance measure

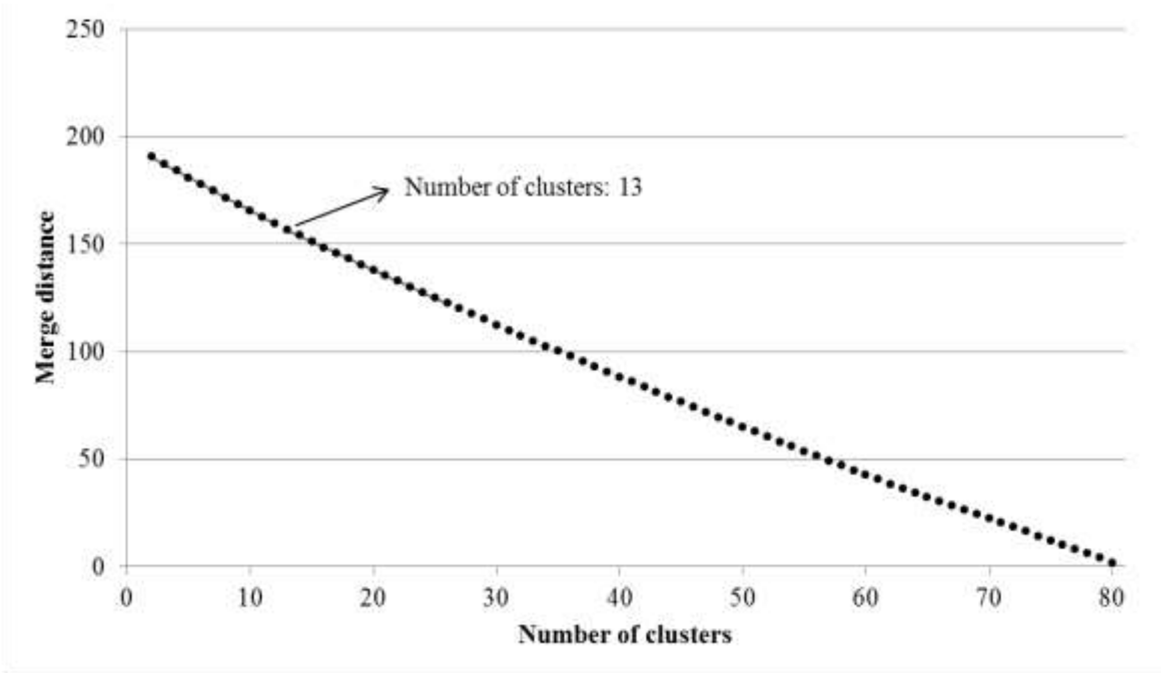

Table D1.2: L method evaluation graph for a hierarchical clustering using Ward's linkage and Manhattan distance measure

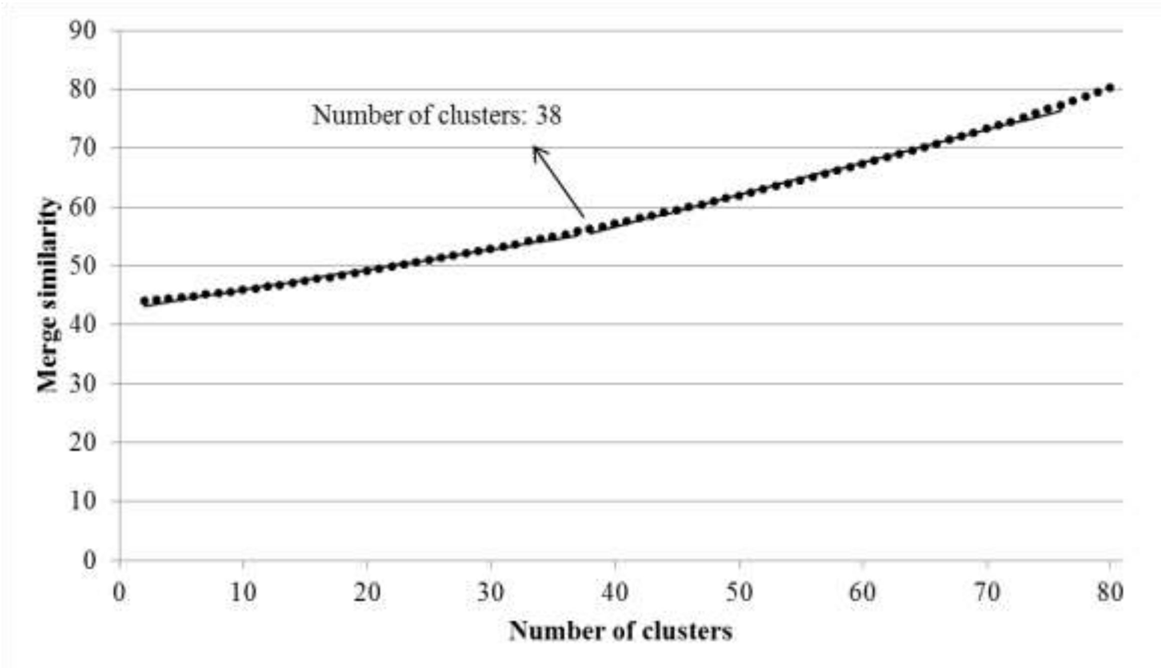

Table D1.3: L method evaluation graph for a hierarchical clustering using Ward's linkage and Pearson's correlation coefficient

D2. Furthest neighbour

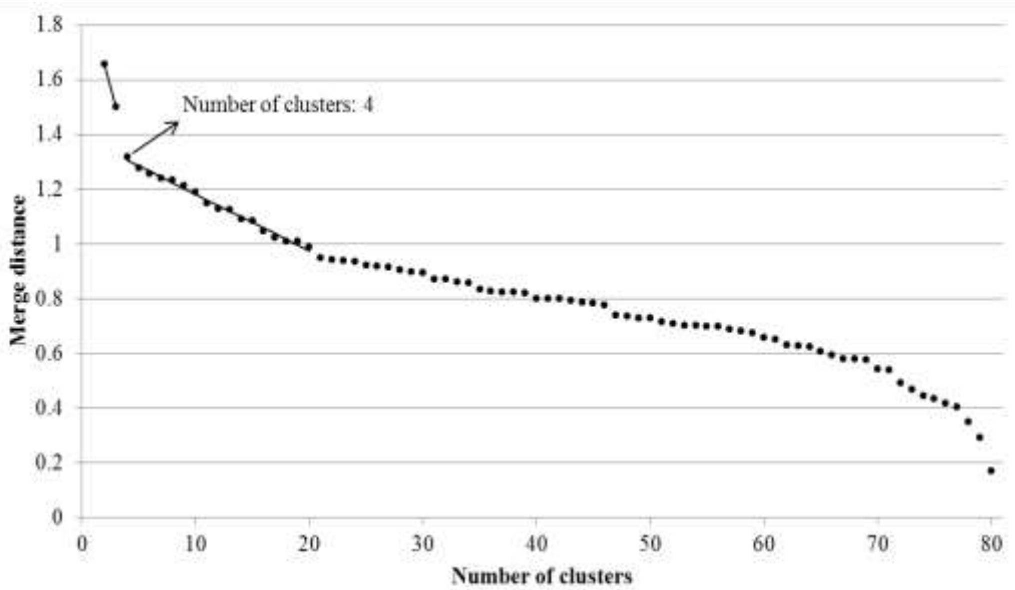

Table D2.1: L method evaluation graph for a hierarchical clustering using furthest neighbour linkage and Euclidean distance measure

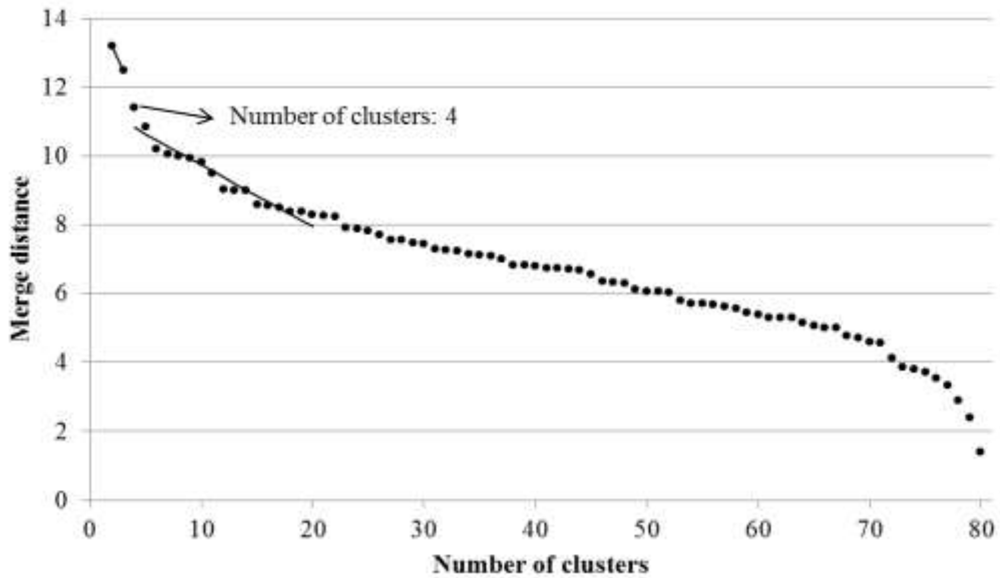

Table D2.3: L method evaluation graph for a hierarchical clustering using furthest neighbour linkage and Manhattan distance measure

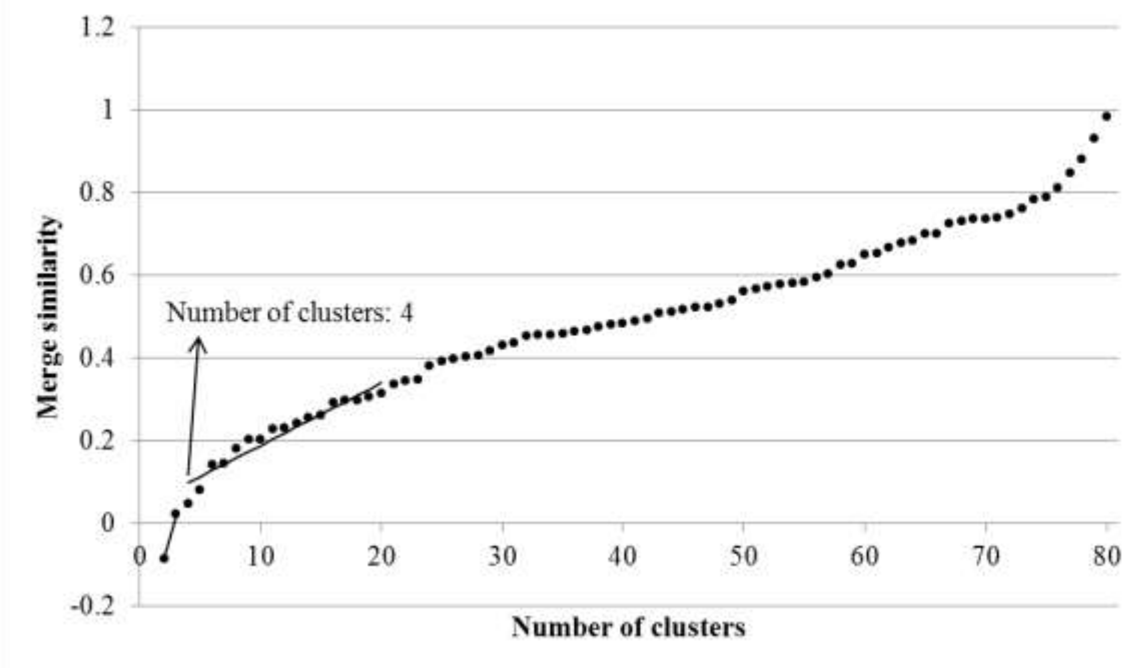

Table D2.3: L method evaluation graph for a hierarchical clustering using furthest neighbour linkage and Pearson's correlation coefficient

D3. Group average

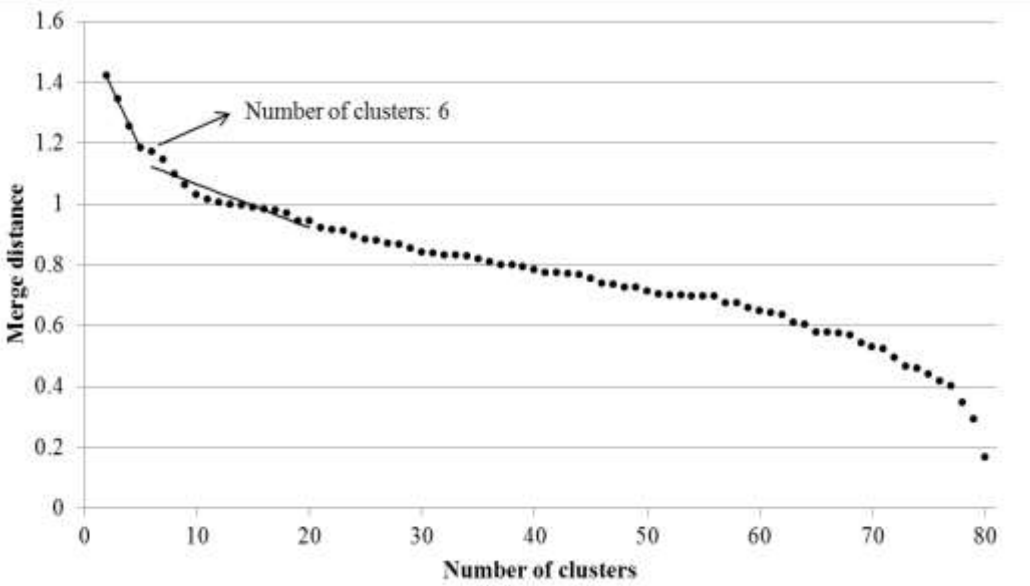

Table D3.1: L method evaluation graph for a hierarchical clustering using group average linkage and Euclidean distance measure

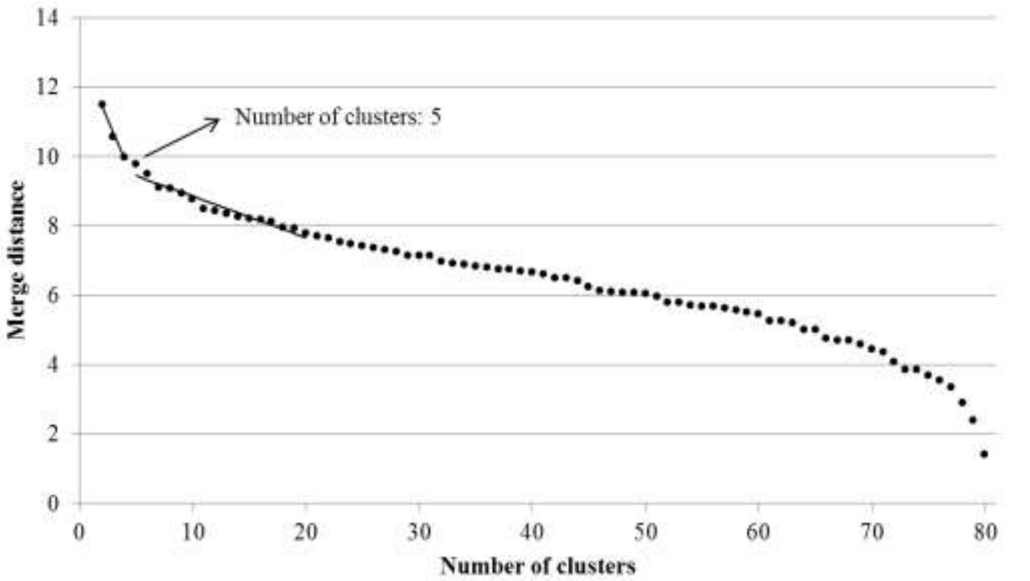

Table D3.2: L method evaluation graph for a hierarchical clustering using group average linkage and Manhattan distance measure

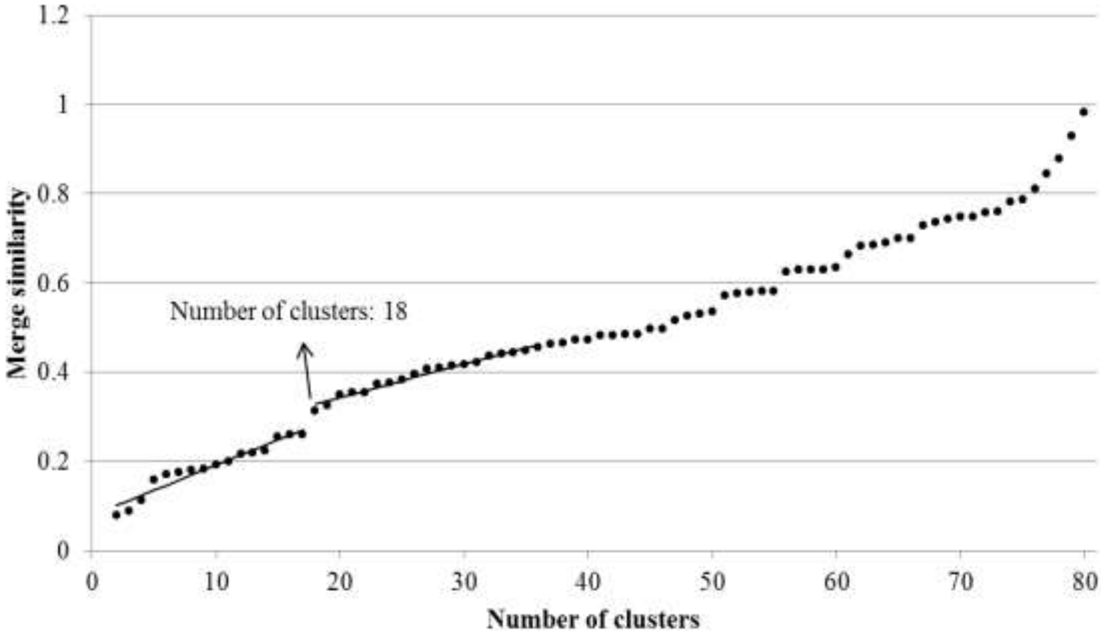

Table D3.3: L method evaluation graph for a hierarchical clustering using group average linkage and Pearson's correlation coefficient

Appendix E

Consistency of cluster size over time

E1. Hierarchical clustering with Ward’s method (1963)

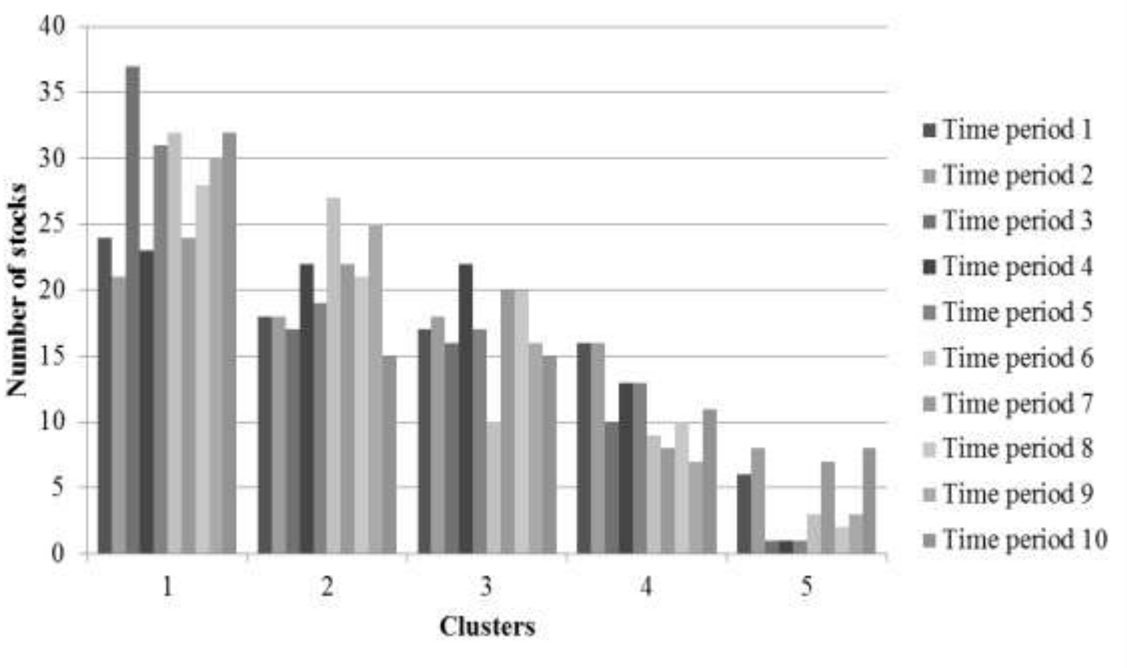

Figure E1.1: Size of clusters generated in each time period by a hierarchical algorithm using Ward's method and Euclidean distance

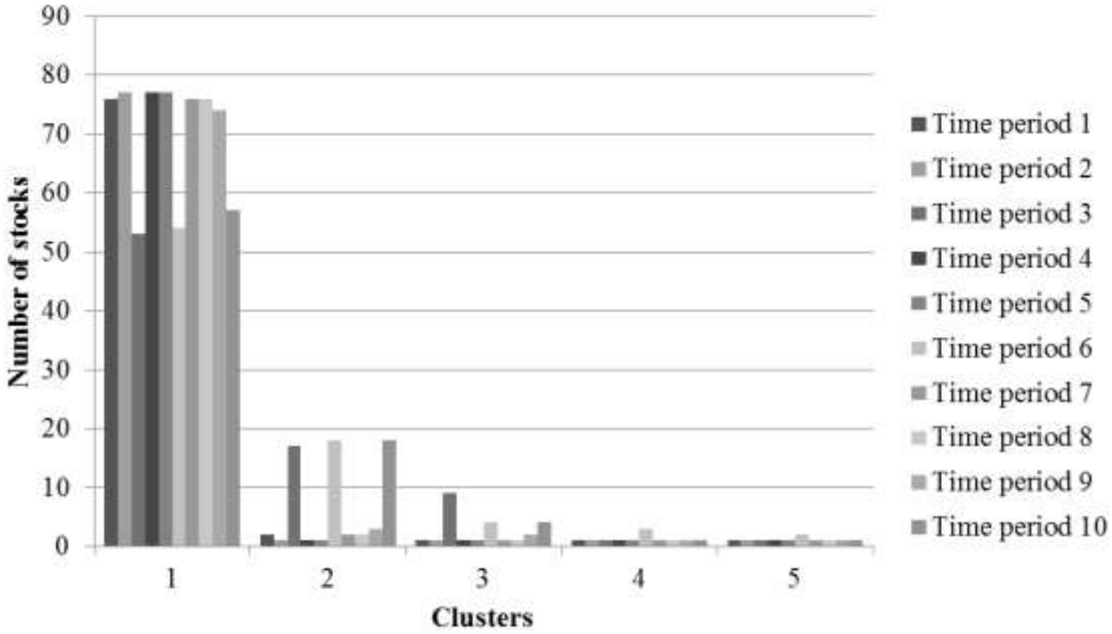

Figure E1.2: Size of clusters generated in each time period by a hierarchical algorithm using Ward's method and Manhattan distance

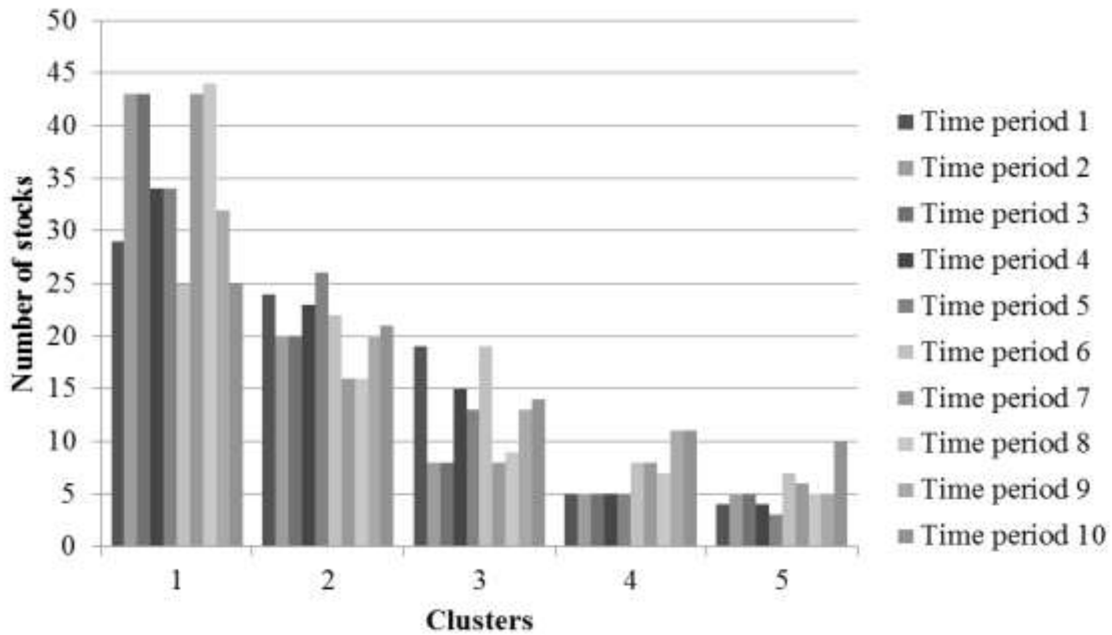

Figure E1.3: Size of clusters generated in each time period by a hierarchical algorithm using Ward's method and Pearson's correlation coefficient

E2. Furthest neighbour hierarchical clustering

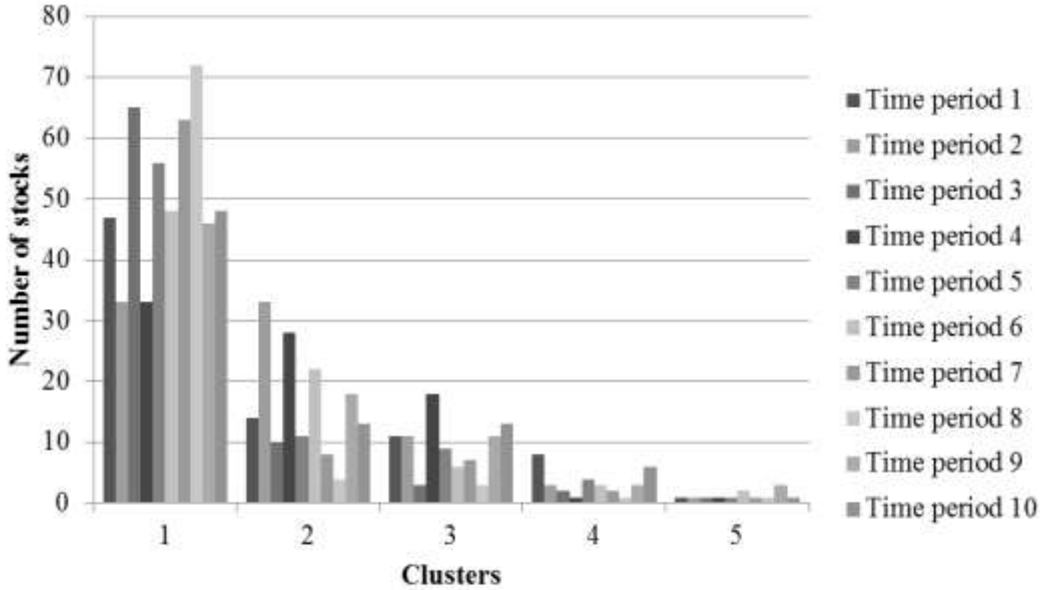

Figure E2.1: Size of clusters generated in each time period by a hierarchical algorithm using the furthest neighbour linkage function and Euclidean distance

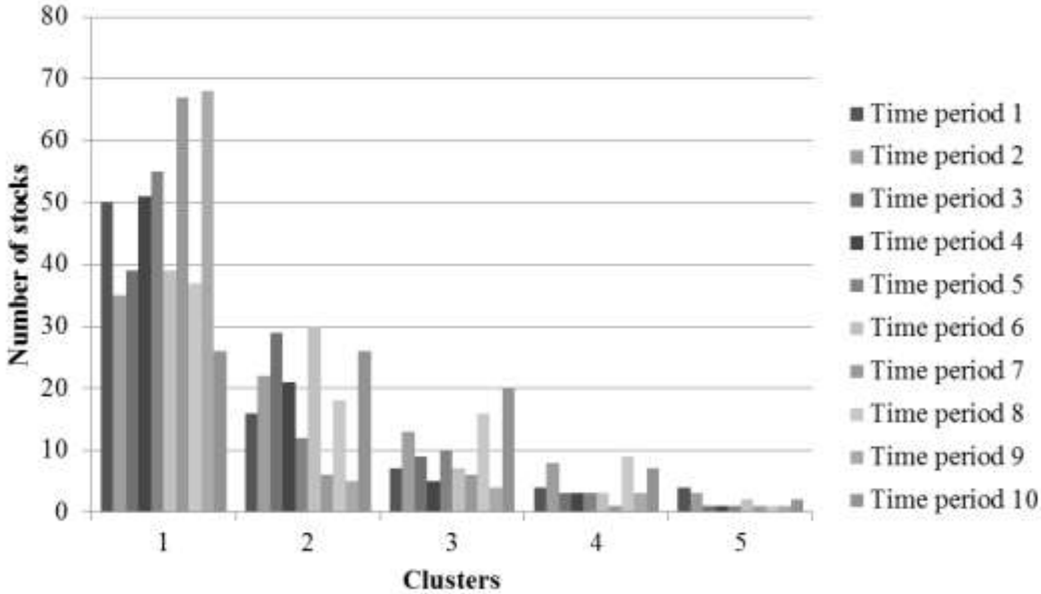

Figure E2.2: Size of clusters generated in each time period by a hierarchical algorithm using the furthest neighbour linkage function and Manhattan distance

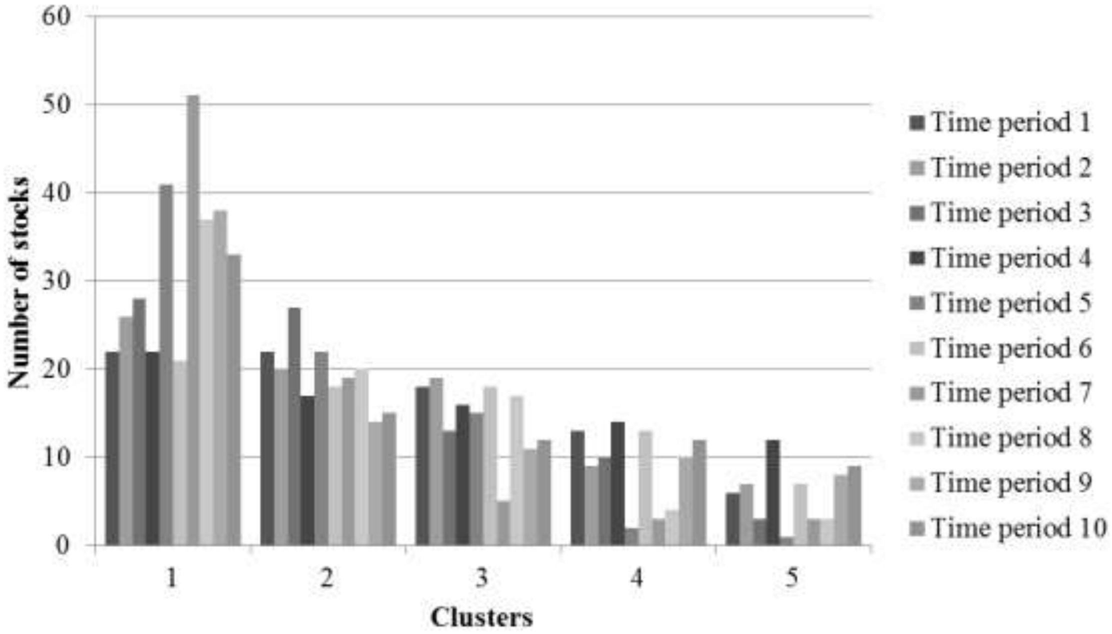

Figure E3.3: Size of clusters generated in each time period by a hierarchical algorithm using Ward's method and Pearson’s correlation coefficient

E3. K-means clustering

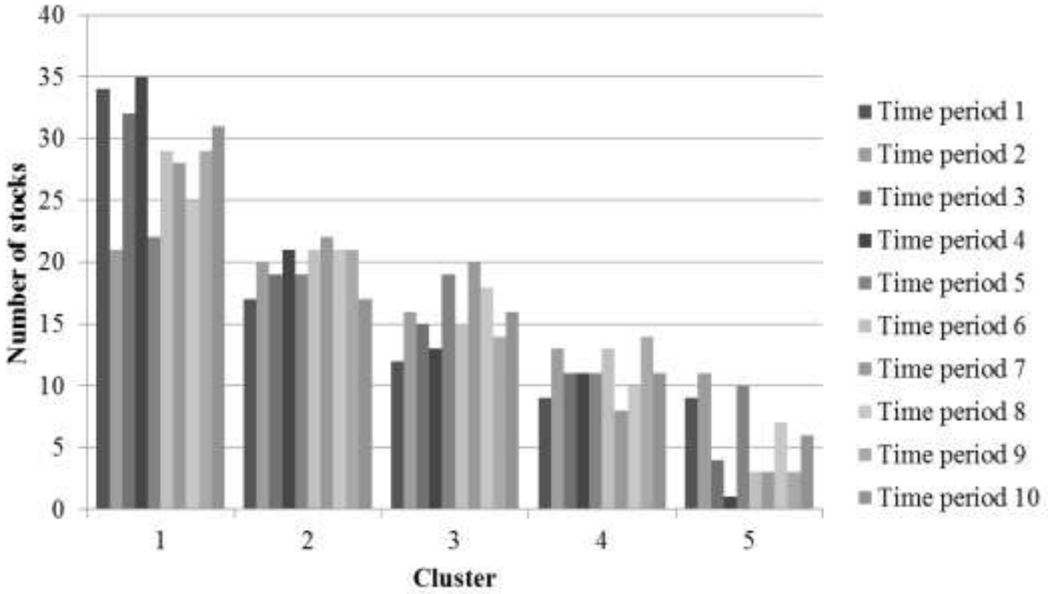

Figure E3.1: Size of clusters generated in each time period be the K-means algorithm with Euclidean distance

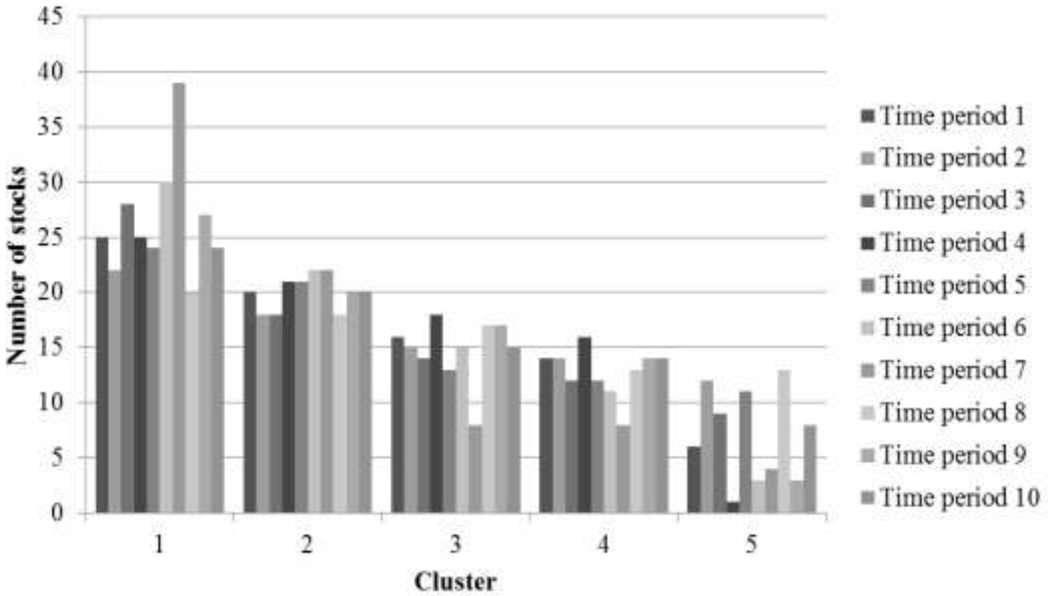

Figure E3.2: Size of clusters generated in each time period be the K-means algorithm with Manhattan distance

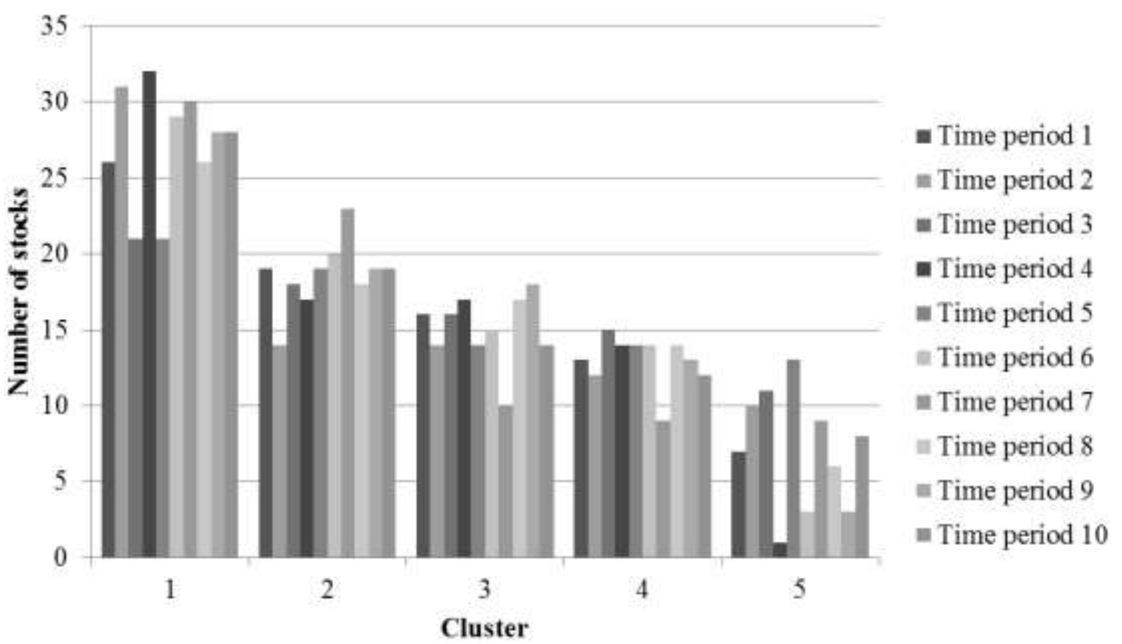

Figure E3.3: Size of clusters generated in each time period be the K-means algorithm with Pearson's correlation coefficient

E4. DBSCAN clustering

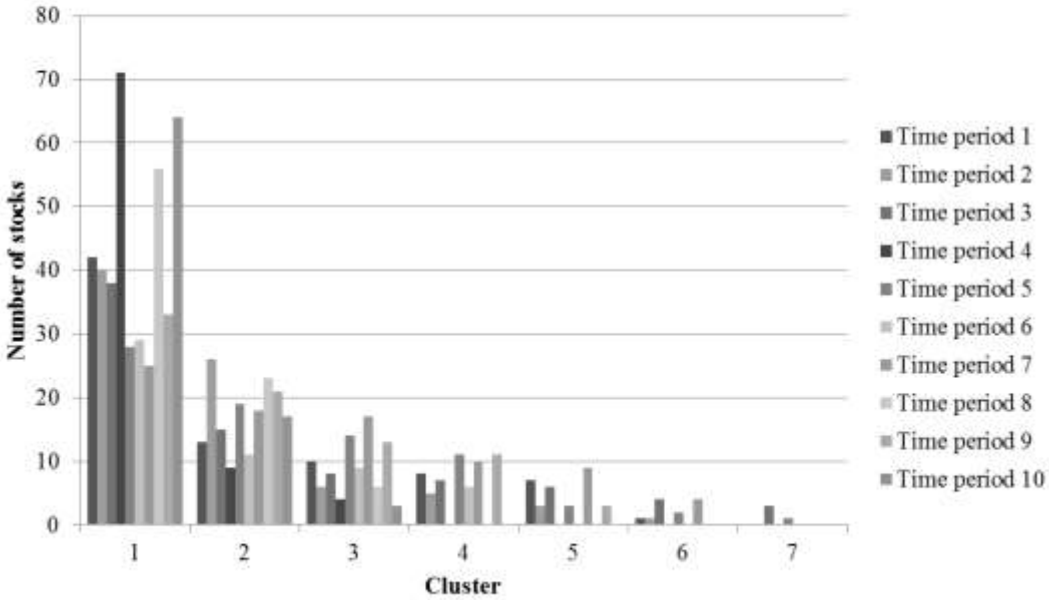

Figure E4.1: Size of clusters generated in each time period by the DBSCAN algorithm with Euclidean distance

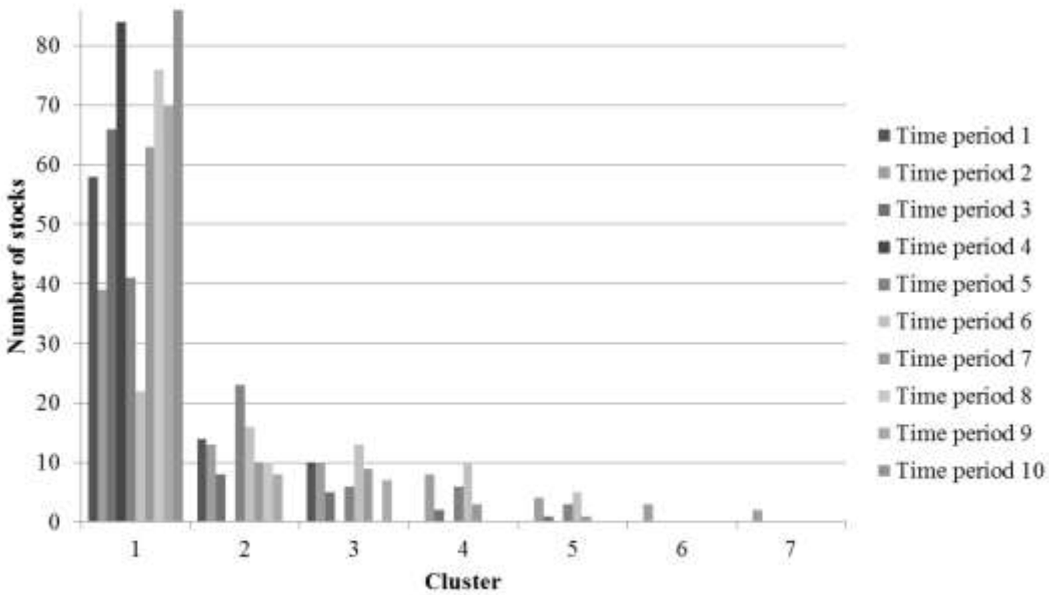

Figure E4.2: Size of clusters generated in each time period by the DBSCAN algorithm with Manhattan distance

Appendix F

Consistency of cluster members over time

Table F1: Variation of information between clusters generated in successive time periods by each clustering algorithm

Variation of information between time periods:	Furthest neighbour hierarchical clustering with Euclidean distance	Furthest neighbour hierarchical clustering with Manhattan distance	Furthest neighbour hierarchical clustering with Pearson's correlation coefficient	Ward's method hierarchical clustering with Euclidean distance	Ward's method hierarchical clustering with Manhattan distance	Ward's method hierarchical clustering with Pearson's correlation coefficient	K-means with Euclidean distance	K-means with Manhattan distance	K-means with Pearson's correlation coefficient	DBSCAN with Euclidean distance	DBSCAN with Manhattan distance
1 and 2	1.9854	1.7746	2.5760	2.5297	0.5726	2.4750	2.5677	2.4401	2.4213	1.7987	1.7921
2 and 3	1.3187	1.9747	2.2705	2.0167	0.9697	0.0000	2.1805	2.2928	2.2753	2.1775	1.4498
3 and 4	1.6974	1.8778	2.5658	2.2535	1.0586	2.1582	2.3504	2.7187	2.6166	1.6802	0.6722
4 and 5	1.8711	1.7298	2.3759	2.4983	0.2632	2.5321	2.6296	2.6990	2.6370	1.6029	1.1715
5 and 6	1.4707	1.4559	2.2909	1.6905	1.0495	2.4462	2.3081	2.0356	2.0477	1.1228	1.7502
6 and 7	1.4487	1.3754	2.1297	2.2895	0.8974	2.6203	2.1994	2.2518	2.3982	1.2529	1.2869
7 and 8	0.9401	1.6019	1.9311	2.4833	0.3592	2.3340	2.3773	2.4608	2.5416	2.2642	1.2301
8 and 9	1.2310	1.5852	2.2966	2.1047	0.3990	2.4891	2.2990	2.3354	2.2794	1.9867	0.9008
9 and 10	1.6703	1.6074	2.6871	2.2722	0.9352	2.6783	2.3391	2.5212	2.4776	1.6522	0.5927