Multiple-objective Clustering Analysis

A Dissertation Presented

by

Tingjun Ruan

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Applied Mathematics and Statistics

(Statistics)

Stony Brook University

May 2016

Stony Brook University

The Graduate School

Tingjun Ruan

We, the dissertation committee for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation.

Wei Zhu – Dissertation Advisor
Deputy Chair, Professor
Department of Applied Mathematics and Statistics

Xuefeng Wang – Chairperson of Defense
Assistant Professor
Department of Preventive Medicine, and Applied Mathematics and Statistics

Song Wu – Member of Defense
Assistant Professor
Department of Applied Mathematics and Statistics

Annie Laurie W. Shroyer – Outside Member of Defense
Professor of Surgery, Internal Medicine, and Preventive Medicine

This dissertation is accepted by the Graduate School

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

Multiple-objective Clustering Analysis

by

Tingjun Ruan

Doctor of Philosophy

in

Applied Mathematics and Statistics

(Statistics)

Stony Brook University

2016

Cluster analysis is an important tool for unsupervised learning. It is commonly used for pattern recognition and dimension reduction. Traditional clustering algorithms include hierarchical clustering and k-means clustering, as well as model based approach such as the group trajectory analysis. A major draw-back of the traditional clustering analysis is that it considers only a single objective (dissimilarity measurement) whilst in reality, one usually holds several criteria for the classification. Therefore, in this thesis, we strive to develop novel multiple-objective clustering methods – with a focus on the more approachable dual-objective ones.

This thesis consists of two parts. In the first part, we introduce the framework of multiple-objective clustering methods. We then introduce the Biclusering analysis method – an existing dual-objective clustering analysis classifying data matrix on the rows and columns simultaneously. Biclustering has been used in gene expression analysis to identify interpretable biological patterns involving a subset of genes and a subset of conditions. Our novel contribution lies in generalizing and extending the objective function used in biclustering in the form of compound clustering, where it is a linear combination of the objective functions with respect to the rows and columns. We also compared the generalized biclustering to the original biclustering algorithm using a microarrary gene expression data set, and a simulation study. Both demonstrated that overall, the generalized biclustering is better than the original biclustering algorithm.

Subsequently, we try to apply both the dual-objective bi-clustering algorithms as well as the classic clustering algorithms to understanding the stock market movements. We attempted to detect the patterns in the bear and the bull stock markets using the biclustering and the generalized biclustering techniques. The pros and cons of the dual-objective clustering in a time domain application are therefore summarized. Subsequently, we used the classic clustering method to identify historical stock market periods resembling the current market in an effort to infer the trend of our current market – especially whether we are approaching a recession or not. We conclude the thesis by performing analysis of intraday pattern of high frequency trading data at the aggregation level of one minute and five minute using stocks traded on NYSE using a model-based clustering approach.

Table of Contents

List of Figures

List of Tables

Acknowledgments

In completion of this work, first I would like to thank my advisor, Prof. Wei Zhu for her dedicated support and help along the way during the past five years' graduate study at Stony Brook University. This thesis would not have been completed without her encouragement and academic support. I am particularly grateful and lucky to have her as my advisor.

I would also like to thank my parents, my grandparents, and my dearest friends for their love, support, and encouragement.

My deepest gratitude also goes to my wonderful committee – Professor Ann Laurie Shroyer, Professor Xuefeng Wang, and Professor Song Wu, for their time, patience, support and insightful comments and revisions that have made this thesis a better work. Thank you!

Chapter 1

1.1 Introduction

Machine learning can be mainly categorized into two categories, supervised learning and unsupervised learning. In supervised learning, the goal is to use inputs to predict the value of the outcome measure. In unsupervised learning, there is no response/outcome variable, the goal is to group similar data together and find hidden patterns in the data. Difference between supervised learning and unsupervised learning is the presence of the outcome variable. Supervised learning has an outcome measure to guide the learning process; while in unsupervised learning, we only have data but no outcome measure (Hastie et al, 2009). The dimension of the data is sometimes very high which is mitigated by the fact that those inputs represent all of the variables under consideration (Hastie et al, 2009). Unsupervised learning algorithms include Principal Component Analysis (PCA), Self Organizing Maps (SOM), Clustering Analysis, Non-negative Matrix Factorization, Independent Component Analysis (ICA), and Multidimensional Scaling and etc. (Hastie et al, 2009).

Clustering analysis is a very important tool for unsupervised learning which is essentially about finding hidden data patterns and discovering groups in data (Everitt et al, 2011). Clustering analysis is widely used in many applications, pattern recognition, image processing, market research, customer segmentation, as well as the analysis of gene expression data. Clustering as pattern recognition and dimension reduction are commonly used. On top of traditional clustering analysis which only considers single objective to cluster data, novel Multiple-objective clustering considers multiple objective functions. For example, MOCK (Handl and Knowles, 2007) clusters data based on both compactness and connectivity. Compound Clustering (Zhang, 2011) integrated multiple data sources, and biclustering takes consideration into both rows and columns and find local patterns called biclusters.

Financial time series have been documented to embrace the characteristics of asymmetry, mean reversion, fat tail and volatility, among which volatility clustering has interested many researchers. Volatility models are developed to model volatility clustering feature. In recent years, increased automation has reduced the role for traditional human market makers and led to the rise of high frequency trading (Brogaard et al, 2014). It was first documented by Wood et al. (1985) and Harris (1986) that average intraday return volatility exhibit distinct U shape over the trading day. Varies models are developed to model this phenomenon, for example Flexible Fourier method in Andersen and Bollerslev (1997) or incorporating seasonality into the GARCH model in Bollerslev and Ghysels (1996) and most recently the Multiplicative Component GARCH by Engle (2012).

1.2 Thesis structure

This thesis is organized as follows. Chapter 2 and Chapter 3 discuss Cluster Analysis and the Multiple-objective Clustering analysis methods. We provide literature review of cluster analysis, its general procedure, commonly used similarity/dissimilarity measurements and clustering algorithms, notably k-means and hierarchical clustering. Group-based trajectory analysis as a model based clustering is introduced in chapter two, followed by an analysis of trajectory analysis on Dow Jones Industrial stock prices. We also discussed different ways of determine the optimal number of clusters and review the application of cluster analysis. Having covered cluster analysis which considers single objective (similarity/dissimilarity measurement), we introduce the framework of multiple-objective clustering analysis methods taking consideration into multiple objectives, the Multiple-objective Clustering with Automatic k determination (MOCK) and the novel Comstrained and Compound Clustering, as well as biclustering/co-clustering. Different algorithms of different objective functions of biclustering are discussed. In Chapter 4, we extended and generalized the objective function used in Cheng and Church (2000) algorithm, representing it in the form of compound clustering where it is a linear combination of different objective functions with respect to rows and columns of the matrix. Analysis on a microarray gene expression dataset as well as simulation study is performed. We conclude chapter 4 by performing biclustering analysis on financial stock data to detect the patterns of bear and bull stock market and thus predict and infer our current market type, which is in line with what we found in the analysis in Chapter 5.

The second part of the thesis work is discussed in Chapter 5 and Chapter 6. Chapter 5 discussed the identification of historical periods where stock prices exhibit similar pattern resembles current market pattern to infer the potential market trend and whether the market is going for recession/depression, and then we analyze all the sectors to see which sectors are mostly likely to be heavily impacted by the impending recession/depression. We conclude the thesis by detection of intraday pattern of high frequency trading data in Chapter 6. Literature review of financial time series and its characteristics is provided. We analyze the real stock high frequency trading data at the aggregation level of one minute and five minute using stocks traded on New York Stock Exchange (NYSE) to study the intraday pattern. We conclude the chapter by applying Multiplicative Component GARCH model to 1-minute return stock data and detect similar pattern which builds up the result. Finally, discussion and future work is laid out in Chapter 7.

Chapter 2

Clustering Analysis

Clustering is an unsupervised learning approach. It groups objects into groups of similar objects. Each group, called cluster, consists of objects similar with each other and dissimilar to objects of other groups. Clustering algorithms is most commonly categorized into two types, hierarchical clustering (Ward, 1963; Johnson, 1967) and partitional clustering (Steinhaus, 1957; Macqueen, 1967). Partitional clustering, notable k means clustering algorithm, splits data points into k partition, where each partition represents a cluster. Hierarchical clustering is a technique of clustering which divide the similar data points by constructing a hierarchy of clusters. Clustering is very useful in pattern recognition, grouping, machine learning, data mining, image segmentation and pattern classification, especially when there is little prior knowledge available about the data, clustering analysis is particularly appropriate to explore the hidden interrelationship among the data points (Jain et al., 1999).

2.1 General Steps of Clustering

Typical clustering steps includes the following (Jain and Dubes, 1988):

1) Pattern representation (feature extraction and feature selection)
2) Define pattern proximity measure
3) Clustering or grouping

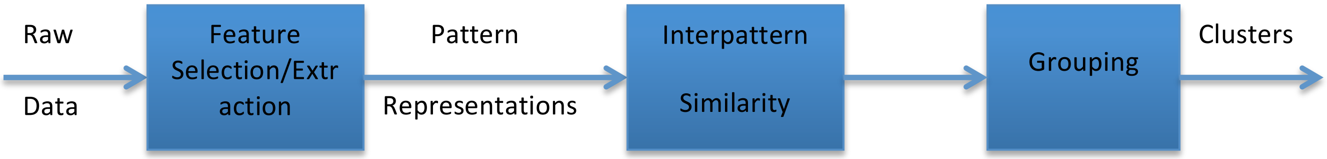


Figure 2.1 General Clustering Stage (Jain et al., 1999)

2.2 Similarity Measures

Similarity is fundamental to define clusters. Measure of the similarity between two patterns drawn from the same feature space is essential to most clustering algorithms. The distance measures should be chosen accordingly due to the variety of the feature types and scales. It is more common to calculate the dissimilarity defined on feature space (Jain et al., 1999). In this section, we will review the most well-known distance measures for continuous variables. The summary of the distance measures are shown in Table 2.1 below:

Table 2.1 Common Similarity/Dissimilarity Measurements for Continuous Variables

Measure	Distance Metric	Comment
Euclidean Distance	$d_{Euc}(x_i, x_j) = \sqrt{\sum_{k=1}^{m}(x_{ik} - x_{jk})^{2}}$	The most commonly used metric, special case of Minkowski Distance at p=2
Manhattan Distance	$d_{Man}(x_i, x_j) = \sum_{k=1}^{m}\lvert x_{ik} - x_{jk}\rvert$	Special case of Minkowski Distance p=1

Chebyshev Distance	$d_{Cheb}(x_i, x_j) = \lim_{p \to \infty} d_{Min}$ $= \max_{k} \lvert x_{ik} - x_{jk} \rvert$	Special case of Minkowski Distance p=∞
Minkowski Distance	$d_{Min}(x_i, x_j) = \left(\sum_{k=1}^{m} \lvert x_{ik} - x_{jk} \rvert^{\frac{1}{p}} \right)^{p}$	Features with large values or variance tend to dominate others
Mahalanobis Distance	$d_{Mah}(x_i, x_j) = (\boldsymbol{X_i} - \boldsymbol{X_j})\Sigma^{-1}(\boldsymbol{X_i} - \boldsymbol{X_j})^{T}$	$\Sigma = cov(\boldsymbol{X_i}, \boldsymbol{X_j})$ covariance matrix
Correlation-based Distance	$d_{corr}(x_i, x_j) = \frac{1 - corr(\boldsymbol{X_i}, \boldsymbol{X_j})}{2}$	Derived from correlation coefficient

Euclidean distance is the most commonly used distance for continuous features (Per-Erik, 1980). The Euclidean distance has is commonly used to evaluate the similarity in two or three-dimensional space and it works very well when the data has compact or isolated clusters (Mao and Jain, 1996). Minkowski distance is the generalized metric distance. When p=2, the distance becomes the Euclidean distance. When p=1, the distance city becomes the city block or Manhattan distance. Chebyshev distance is a special case of Minkowski distance when p goes to ∞. The distance can be used for both ordinal and quantitative variables (Grabusts, 2012). The drawback of using the Minkowski distance directly is the tendency of largest-scaled feature to dominate others. Solution is to include normalization of the continuous features for other weighting schemes (Jain et al., 1999). The regularized Mahalanobis distance was used in Mao and Jain (1996) to extract hyperellipsoidal clusters.

It is problematic to compute distance between patterns with features being non-continuous, that is to say, in the case when we have mixed data types. Practitioners, however, especially those in machine learning field where mixed data types are very common, have developed similarity measurements for heterogeneous type patters (Jain et al., 1999). For example, Wilson and Martinez (1997) proposed a combination of modified Minkowski metric for continuous features and a distance based on counts for nominal attributes. Diday and Simon (1976); Ichino and Yaguchi (1994) developed some other metrics for computing the similarity between patterns represented by both quantitative as well as qualitative features (Jain et al., 1999).

2.3 Hierarchical Clustering

Hierarchical clustering procedure is characterized by tree like structure and is one of the most widely used clustering approaches. As early as the 1970s, it was held that about 75% of all published work on clustering employed hierarchical algorithms (Blashfield and Aldenderfer, 1978). It can be categorized into agglomerative clustering and divisive clustering. Agglomerative hierarchical clustering techniques are by far the most common. Clusters are consecutively formed from objects starting with each objective as an individual cluster then sequentially merged two clusters that have smallest dissimilarity as measured by linkage, until there is only one cluster. A hierarchical clustering procedure is often displayed using dendrogram, which is a convenient graphic to display a hierarchical sequence of clustering assignments and cluster-subcluster relationship. Hierarchical information is very useful in many applications (Handl and Knowles, 2007). As Duda and Hart (2001) pointed out: “ hierarchical clustering permeates classificatory activities in science". Different clusters are generated by cutting the dendrogram at different levels. An example of dendrogram is shown in the following:

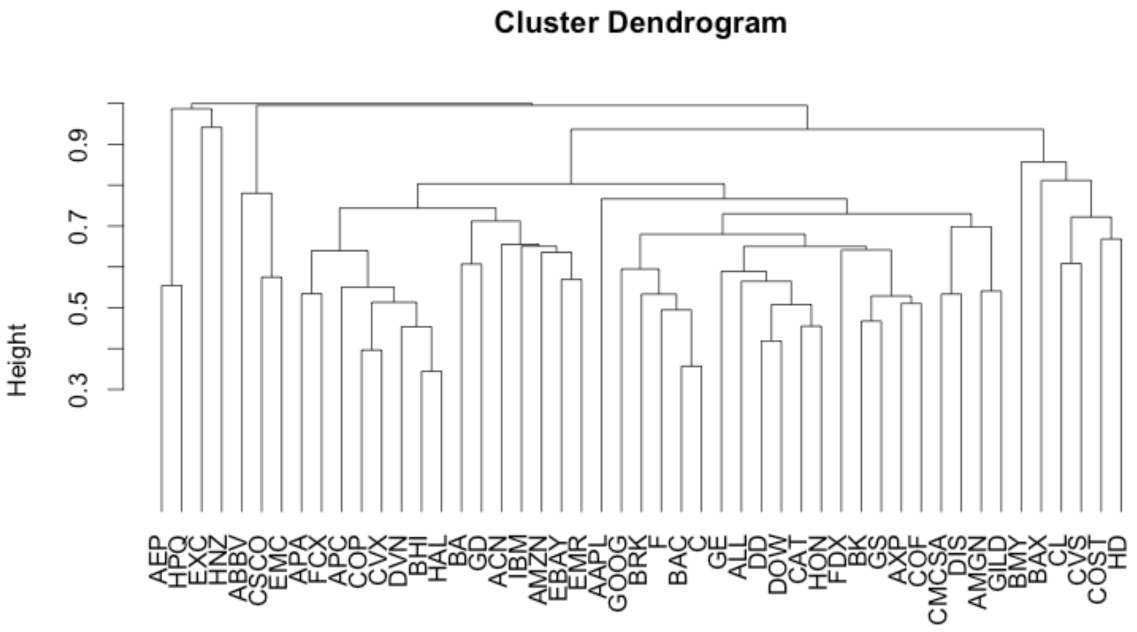


Figure 2.2 Clustering Dendrogram using 49 stocks from S&P100

The general procedure of hierarchical clustering can be summarized in the following steps:

1) Select a measure of similarity/dissimilarity
2) Select a clustering algorithm/linkage criterion, merge two clusters into one based linkage criterion selected until there is only one cluster left

There are a lot of measures to define the similarity/dissimilarity between any two objects. The most common dissimilarity measures are summarized above. They are expressed by means of distance matrix where the non-diagonal elements express distances between pairs of objects and zeros on the diagonal. By choose a dissimilarity measure, the distance between objects is determined. Then linkage criteria should be chosen to define the distance between clusters. The major linkage algorithms include Single linkage (Sneath and Sokal, 1973), complete linkage (King, 1967) and average linkage (Ward, 1963; Murtagh, 1984):

1) Single linkage (nearest neighbor): the dissimilarity between two clusters is the smallest dissimilarity between any two objects in the two clusters.
2) Complete linkage (furthest neighbor or compact): the dissimilarity between two clusters is the largest dissimilarity between any two objects in the two clusters.
3) Average linkage: the dissimilarity between two clusters is the average dissimilarity over all the objects in two clusters.
4) Ward linkage (minimum variance): Combine objects with minimum within-cluster variance.

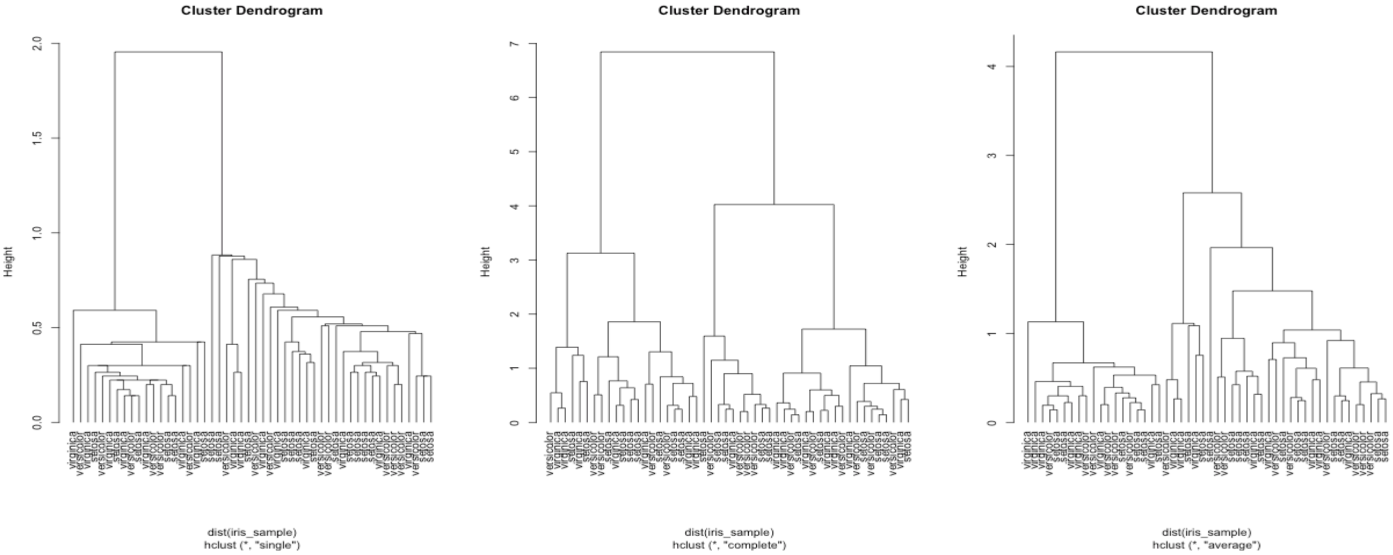


Figure 2.3 Dendrograms of Iris data (random of 40 obs) using different linkages

Different linkage algorithm yields different clustering result on the same dataset, as each has its specific properties, as we can tell from Figure 2.3 above. The choice of linkage used significantly affects clustering algorithms as different linkage criteria reflect different connectedness and closeness. Single linkage and complete linkage algorithms are among those the most popular and commonly used ones. Complete linkage tends to create more compact clusters (Baeza-Yates, 1992); while single linkage tends to create clusters that are straggly or elongated suffering from a chaining effect (Nagy, 1968). However, complete linkage algorithms produce more useful hierarchies in many applications than single linkage algorithms (Jain and Dubes, 1988; Jain et al., 1999).

Hierarchical clustering groups data in a hierarchical tree structure according to the proximity matrix. The result of hierarchical clustering is usually displayed in a dendrogram which is a very convenient representation of the data struture. The final clustering result is obtained by cutting the dendrogram at different levels which provides very informative descriptions.

Hierarchical clustering methods can be traced back to early 1960s and 1970s is one of the most important clustering techniques addressed in many works. Some major surveys of clustering covering hierarchical clustering include Gorden (1981), March (1983), Jain and Dubes (1988), Jain et al., (1999), and Xu and Wunsch (2005).

2.4 K-means Clustering

Another important clustering procedure is partitioning method, notably K-means is the most important clustering technique. The most intuitive and frequently used criterion for partitional clustering techniques is the square error criterion, which works well with isolated and compact clusters (Jain et al., 1999). K-means is the simplest and most commonly used clustering algorithms employing a squared error criterion (McQueen, 1967). Unlike hierarchical clustering which is based on proximity measurements, k-means is based on within-cluster variance as a measure to for homogenous clusters. Specially, clusters are formed so that with-in cluster variance is minimized. Therefore, we do not need to calculate the similarity/dissimilarity measurements upfront of the analysis.

K-means is intended for situations in which all variables are of the quantitative type, and squared Euclidean distance is chosen as the dissimilarity measure (Hastie et al., 2009). The clustering process starts by random initial partition, which is user specified number of clusters. The objects are then successively reassigned to clusters to minimize the within-cluster variation which is the squared distance from each observation to the center of the clusters. The center of the cluster is then updated based on the objects assigned to the cluster. The process is repeated until the center of the cluster remains the same, or there is no reassignment of any objects to other clusters that can reduce the squared error significantly. The sum of squared error is defined as:

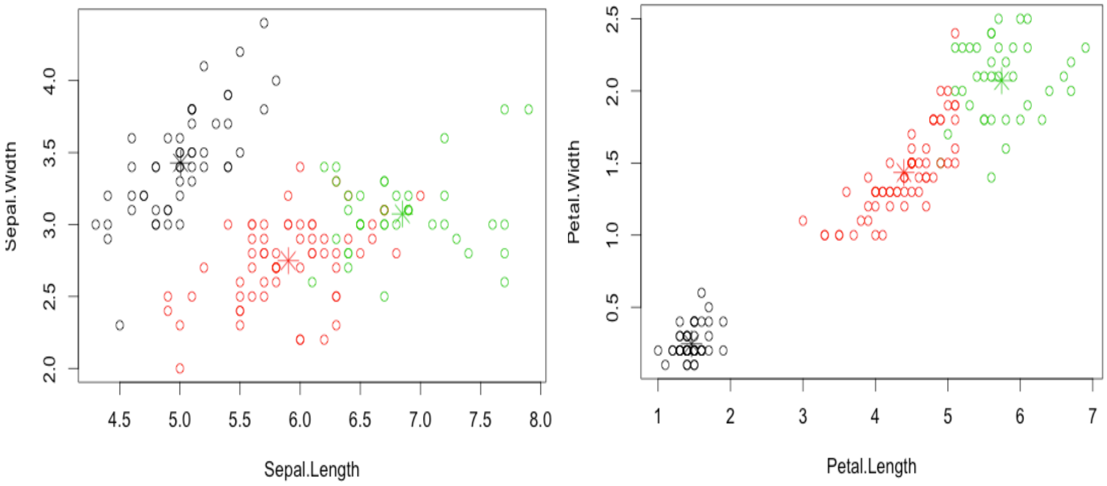
$$SSE = \sum_{j=1}^{K} \sum_{i=1}^{n_j} \left\| \boldsymbol{x}_i^j - \boldsymbol{c}_j \right\|^2$$

where $\boldsymbol{x}_i^j$ is the i^{th} objects belonging to the j^{th} cluster and $\boldsymbol{c}_j$ is the centroid of the j^{th} cluster. Figure 2.4 shows k-means clustering using the famous Fisher's Iris data.

K-means algorithm is very popular because it's easy to implement, and it complexity is $O(n)$, where n is the number of objects (Jain et al., 1999). Generally, k-means is less affected by outliers and it can be applied to very large dataset, as it is less computationally demanding than hierarchical clustering. One problem associated with k-means clustering, however, is that it requires the pre-specification of the initial partitions which makes k-means less versatile than hierarchical clustering. It is often employed by researchers to perform hierarchical clustering first to determine the number of clusters and then apply k-means clustering which provides some clue to find the initial clusters.

Anderberg (1973) has documented several variants of k-means clustering to select a good initial partition so that the algorithm is more likely to find the global minimum value (Jain et al., 1999). ISODATA (Ball and Hall, 1965) employs the technique of merging and splitting clusters based on a pre-specified threshold of the cluster variance. Diday (1973) and Symon (1977) proposed dynamic clustering algorithm to select a different criterion function by permitting representations other than centroid for each cluster.

Figure 2.4 is the plots of applying k-means to famous Fisher's Iris data. We specify k=3.



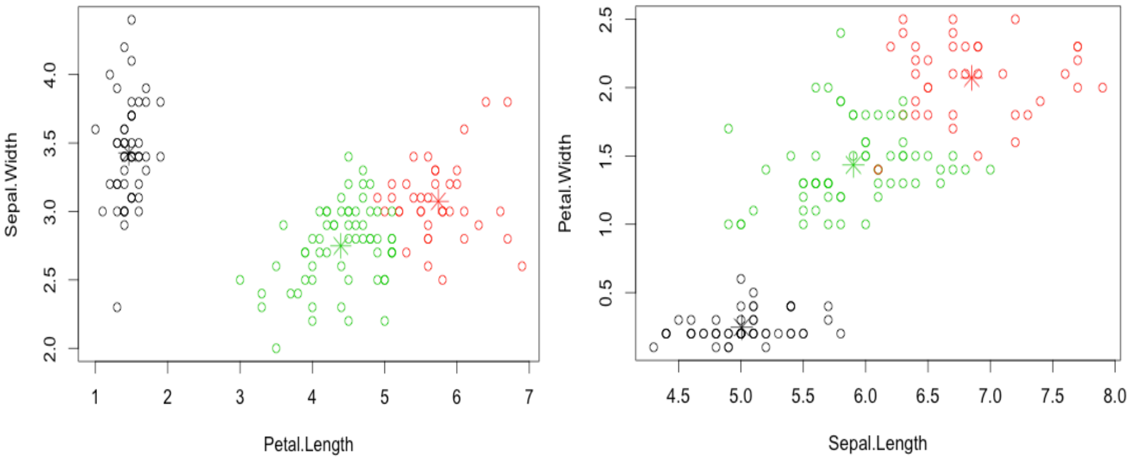


Figure 2.4 K-means clustering using Fisher's Iris data

2.5 Model-based Clustering Method – Group-based Trajectory Analysis

The modeling of longitudinal developing trajectories is very popular in psychology, sociology and criminology (Fergusson and el al., 1996; Loeber and LeBlanc 1990; Moffitt 1993; Patterson 1996; Patterson and et al., 1989; Nagin and Jones 2001). Group-based trajectory models are designed to identify clusters of individuals following similar progressions of some behavior or outcome over age or time, or developing trajectories (Nagin and Jones, 2007), which is especially useful in identifying meaningful different subgroups overtime. It is a model based clustering method. A polynomial relationship is used to model the link between time/age and model's parameters. There is a SAS statistical modeling procedure PROC TRAJ developed by Nagin and Jones (2001) for estimating the developmental trajectories. The procedure is based on a semiparametric, group based modeling, specifically, it is a mixture of probability distributions. Proc traj procedure assumes that every subject in a group follows the same trajectory. Proc traj provides the option of modeling three different distributions and different data types. Zeroinflated Poisson model for count data when there are more zeros than under Poisson assumption; Censored Normal model (CNORM) for continuous data and logistic model (LOGIT) for binary data (Jones and Nagin, 2000). The Bayesian Information Criterion (BIC) score is used to select the best model, it can also be used to identify the number of groups within the population. The best model is one with highest BIC score. The software allows for the specification of the polynomial relationship between age and model's parameters of up to a fourth order polynomial in age (Nagin and Jones, 2001). The underlying statistical theories are reported in detail in Nagin and Jones (2001, 2007).

2.6 Application of PROC TRAJ on Financial Time Series Data

We use daily close price data on the Dow Jones Industrial Index, over the period of a month from 1 May 2013 to 28 May 2013 to illustrate trajectory analysis. Dow Jones Industrial Index is comprised of 30 industrial companies' stocks, and it is one of the stock index that represent the US stock market. The sample consists of 30 stocks, 19 trading days during the 1 month period. Trajectory analysis was performed to estimate the number of stock price patterns in the collected sample. PROC TRAJ plug-in in SAS developed by Nagin and Jones (2001) is used to calculate the result. Normal distribution (CNORM) model was specified in the estimation because the stock price data is continuous variable. BIC score was used to select the number of trajectory groups. Cubic polynomial relationship is specified to model the link between time variable and model's parameter.

Table 2.2 BIC score of trajectory groups from 1 to 5 of the Dow Jones Industrial stocks price

Group	BIC (N=30)	AIC (N=30)
1	-2870.35	-2866.84
2	-2673.28	-2666.27
3	-2363.22	-2352.71
4	-2251.96	-2237.95
5	-2136.74	-2119.23

Table 2.6 shows the BIC scores of the trajectory groups from 1 to 5. We select the number of trajectory groups with smallest absolute value of BIC (i.e., largest BIC score). Based on the table, five-group trajectory model is favored. Figure 2.5 is the estimated trajectory patterns of the stock prices, the percentage of each trajectory is greater than 5%, which means that the trajectory patterns are all robust.

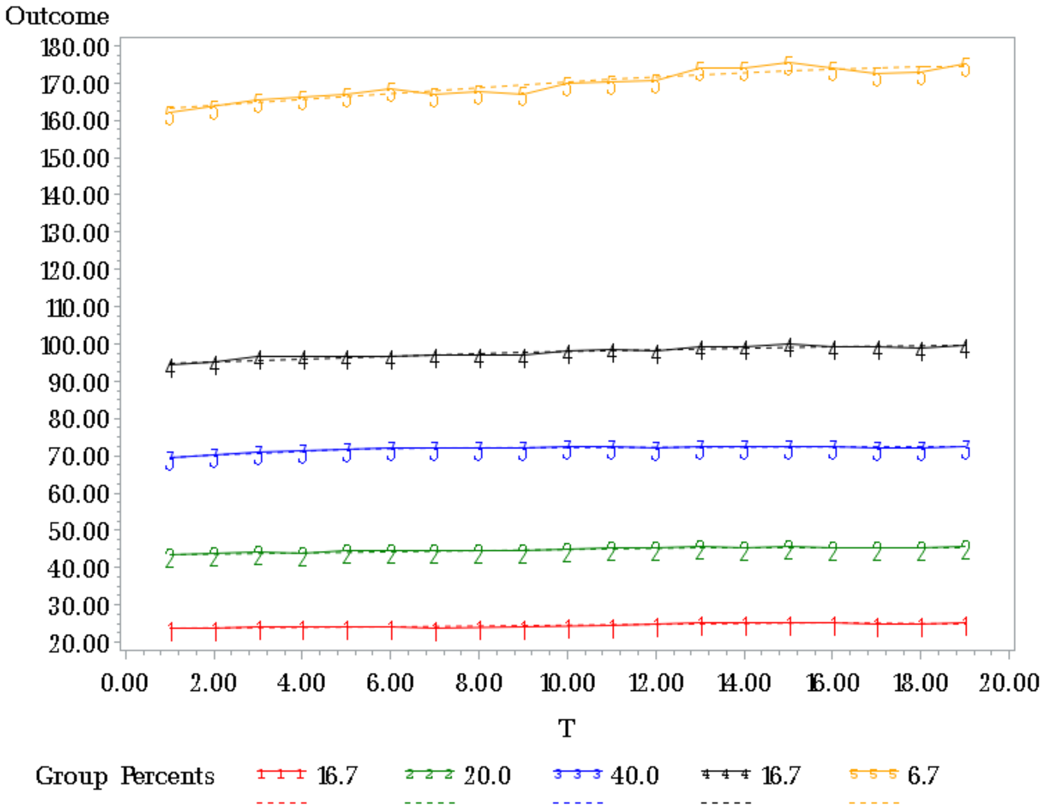


Figure 2.5 Estimated trajectory patterns of Dows Jones Industrial stock prices (five trajectory groups) – (CNORM model: Censored Normal Distribution)

2.7 Determination of Number of Clusters

Clustering analysis is an important tool for unsupervised learning, to find groups in data without the help of response variables. The estimation of optimal cluster numbers is always a major challenge (Tibshirani, 2001). There are no completely satisfactory methods for determining the number of population clusters for any type of cluster analysis (Everitt, 1979; Hartigan, 1985; Bock, 1985). Many methods have been proposed for estimating the number of clusters. Milligan and Cooper (1985) give a very comprehensive summary on methods for estimating the number of clusters. Gordon (1999) also summarized many methods for estimating the number of clusters, where he divides the approaches into global and local methods. Curvas et al. (2000) proposed method which relies on high dimensional density estimate.

In this section, we will give a summary of existing methods for estimating the number of clusters, we will also introduce the "Complete linkage" R^2 proposed by Zhang (2011) which he used to determine the number of clusters in his analysis of gene expression data.

R^2 was introduced to be considered (Sarle, 1996). The larger the R^2 , the better the clusters. Zhang (2011) adopted the "complete linkage" hierarchical clustering method for his analysis of the data, defining the distance between clusters by maximum of distances between any two components of the clusters. He then proposed the "Complete Linkage R^2" (Zhang, 2011) defined as follow:

$$R^2 = 1 - \frac{SSE}{SST}$$

$$SST = \sum (X - \bar{X})^2 \Rightarrow SST \approx nD^2$$

$$SSE_j = \sum \left(X_i - C_j\right)^2 \Rightarrow SSE_j \approx n_j D_j^2$$

where SSE_j is for each cluster j, n_j is the number of objects in each cluster j, and D_j is the maximum distance within cluster j. That's defined, the "Complete Linkage R^2" is:

$$R^2 = 1 - \frac{\sum_j n_j D_j^2}{nD^2}$$

Kaufman and Rousseeuw (1990) proposed the largest average silhouette statistic. Denote a_i as the average distance to other objects in its cluster for observation i, and b_i be the average distance to points in the nearest cluster besides its own is defined by the cluster minimizing this average distance (Tibshirani, 2001). The silhouette statistic is defined by

$$s(i) = \frac{b_i - a_i}{\max\,(a_i, b_i)}$$

Kaufman and Rousseeuw (1990) proposed that the optimal number of clusters k is chosen to maximize the average $s(i)$ over the data set, which is equally to maximize:

$$\frac{1}{N}\sum_{i=1}^{N} s(i)$$

Tibshirani (2001) proposed an approach using Gap statistic. The idea is to standardize the graph of $\log(W_k)$ by comparing it with its expectation under an appropriate reference distribution of data without any clustering structure. The estimate of the number of clusters is k for which $\log(W_k)$ falls the farthest below the reference curve. Gap statistic is defined as

$$Gap_n(k) = E_n^*(\log(W_k)) - \log\,(W_k)$$

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r$$

$$D_r = \sum\nolimits_{i,i' \in C_r} d_{ii'}$$

where W_k is the sum of pairwise distances for all objectives in cluster r, E_n^* denotes expectation under a sample of size n from the reference distribution. The optimal k would be the value maximizing $Gap_n(k)$.

Milligan and Cooper (1985) and Cooper Milligan (1988) carried out a comprehensive simulation comparing 30 different methods, and Calinski and Harabasz index (1974) performs best among the global methods. Krzanowski and Lai (1985) proposed the quantity $W_k k^{\frac{2}{p}}$ as a criterion for choosing the number of clusters and Marriott (1971) followed the proposal by using the determinant rather than trace of the within sum of square matrix. Hartigan (1975) proposed a statistic and its idea is to start from k=1 and add a cluster as long as the statistic is sufficiently large.

The number of clusters can also be found by resampling approach, that is, to choose k according to the similarity of clusters results on randomly produced or sampled data. Resampling can be interpreted as using many randomly produced copies of data for assessing statistical properties of a method in question (Mirkin, 2011). Minaei-Bidgoli, Topchy and Punch (2004), Dudoit, Fridlyand (2002), McLachlanm Khan (2004) and Bel Mufti, Bertrand, Moubarki (2005) have tried to find the number of clusters based on resampling approaches.

2.8 Applications of Clustering Analysis

Clustering analysis techniques as one of the most important unsupervised technique are used frequently in many areas: biology, botany, medicine, psychology, geography, marketing, image processing, psychiatry, etc (Brian et al., 2011). We list some of the applications of clustering analysis in those areas.

In market research, dividing variables into homogeneous groups is always very important step in data analytics due to huge amount of customer information data. Green et al (1967) use cluster analysis to classify the cities into a small number of groups among which are very similar to each other on the basis of 14 variables including city size, newspaper circulation and etc. due to economic restrictions at early time. Chakrapani (2004) use clustering analysis to identify people with a lifestyle most a associated with buying sports cars to make market campaign strategy.

In bioinformatics and genetics, DNA microarrays (Cortese, 2000) are a revolutionary breakthrough in molecular biology that has the ability to simultaneously study thousands of genes. After genome sequencing, DNA microarray analysis has become the most widely used functional genomics approach in the bioinformatics field. Cluster analysis can be applied to identify genes with similar patterns of expression, and can help to find hidden information how gene expression is affected by various disease and which genes are mostly likely to cause specific disease (Brian et al., 2011). Clustering analysis on microarray data is presently by far the most used method for gene expression analysis which provides a strategy to extract meaningful information from the express profile (Naghieh and Peng, 2009). For example, Eisen et al. (1998) use clustering analysis of genome wide expression data to identify cancer subtypes associated with survival. Kerr and Churchill (2001) investigate and make statistical inferences using clustering applied to gene expression data.

In weather classification, huge amount of data are collected on weather worldwide. Clustering analysis provides insights into climatological and environmental trends that have both scientific and practical significance (Brian et al., 2011). Average linkage was used to group data into days with similar weather conditions (Huth et al., 1993). Littmann (2000) uses clustering analysis on the daily occurrences of several surface pressures for weather in the Mediterranean basin. Liu and George (2005) applied fuzzy k-means clustering method to account for the spatiotemporal nature of weather data in the South Central USA (Brian et al., 2011).

In Psychiatry, clustering analysis techniques have been used a lot to refine or even redefine current diagnostic categories, much of the work has involved depressed patients (Brian et al., 2011). Pilowsky et al. (1969) clustered patients based on their responses to a depression questionnaire together with other information such as mental state, sex age and length of illness. Clustering analysis has also been used to find classification of individuals who attempt suicide. Clustering methods, Ward's methods were applied to the suicide attempters together with pool of other variables to get a classification of three groups (Paykel and Rassaby, 1978). Kurtz et al. (1987) and Ellis et al. (1996) also investigate the use of clustering analysis (average linkage clustering) on suicidal psychotic patients. Further more, clustering analysis is also used to best classify eating disorders, Hay et al. (1996) applied Ward's method of clustering to investigate the problem.

In hedge fund, cluster analysis has been a popular tool among money managers to group investments. Clustering analysis can reveal hidden patterns that provide insights to help problem solving (Hartigan, 1975). Clustering is applied to classify various hedge funds based on the fund returns (Miceli and Susinno, 2003; 2004). They argue that clustering can be used to predict potential style drifts, conduct peer group analysis and identify benchmarks for groups of founds.

They also argue that cluster analysis makes things easier to interpret than large correlation matrices, and it is very helpful during portfolio construction. Martin (2001) concludes that there is significant heterogeneity in individual fund returns within clustering by analyzing monthly returns for hedge funds. Cluster analysis is more successful than the ZCM/Hedge classification in categorizing manager return histories by clustering managers based on asset class, style of hedge fund, incentive fee, risk level, and liquidity (Das, 2003). Fuzzy clustering was used to illustrate the degree of misclassification that exists in the industry-accepted investment-style classifications (Gibson and Gyger, 2007).

Chapter 3

Multiple-objective Clustering Analysis

3.1 Existing Multiple-objective Clustering Approach

In this chapter, we will be discussing multiobjective clustering analysis methods. First part we provide the existing multiobjective clustering methods from literature review, specifically framework of MOCK algorithm (Multiobjective Clustering with automatic k-determination) proposed by Handl and Knowles (2007), which is based on Multiobjective Evolutionary Algorithm (MOEA) obtained via the framework of Pareto optimality. We will introduce the theoretical advantage behind it. In the second part, we will introduce the framework of a novel multiobjective clustering analysis approach proposed by Zhang (2011), the Compound Clustering and Constrained Clustering.

There are many approaches for clustering analysis such as k-means and hierarchical clustering which have been detailed described in Chapter 2, Section 2.3 and Section 2.4. The main problem in clustering is to find the best partitions among k clusters and determine the optimal number of clusters. K-means is better at find partitions but users have to give the initial k while hierarchical clustering is better at determining k but worst at finding partitions. Both of the methods use a single objective, the objective of k-means is compactness and objective of hierarchical clustering is connectivity of similar data. MOCK optimizes two complementary objectives, considering both the cluster compactness and connectedness. It has been reported that MOCK shows better performance than k-means and hierarchical clustering and other evolutionary clustering algorithms (Matake et al., 2007).

In order to develop a clustering algorithm that simultaneously considers several complementary aspects of clustering quality, MOCK embraces the framework of Pareto optimization, Specifically, employ a multiobjective evolutionary algorithm (MOEA) to optimize several clustering objectives, and to obtain a set of trade-off solutions, which represent a good approximation to the Pareto front, that is a set of partitionings that Pareto optimal with respect to the objectives optimized. Compared with single clustering algorithm, multiobjective algorithm will always find a solution as good or better than those of single objective algorithms (Handl and Knowles, 2007). There are many other multiobjective optimization algorithms, such as MOGA (Fonseca and Fleming, 1993), VIENNA (Handl and Knowles, 2004), VEGA (Schaffer, 1984), Niched Pareto GA (Horn et al., 1994), SPEA (Zitzler and Thiele, 1998), NSGA (Srinivas and Deb, 1994), and classic PESA-II (Corne et al., 2001) which form the basis of MOCK.

The classical ways of tackling multiple-objective optimization problems is fairly straightforward as reported by Deb (1999), to convert multiple objectives into a single objective problem. The conversion methods existed are: weighted sum approach, ϵ-perturbation method, Tchybeshev method, min-max method, goal programming method, and etc (Chankong and Haimes, 1883; Miettinen, 1999; Sen and Yang, 1998; Deb, 1999). The conversion to a single-objective optimization problem is usually subjective to the parameter settings chosen by the user. Additionally, only one solution can be found in one simulation run as usually a classical optimization is used for single-objective optimization problem. Therefore, in order to find multiple pareto solutions, the chosen optimization algorithms is need to run for many times. Classical methods have been found to be sensitive to convexity and continuity of the pareto-optimal region (Deb, 1999).

3.2 Multiple-objective Clustering with automatic k-determination (MOCK)

MOCK (Handl and Knowles, 2007) is a multiple-objective clustering algorithm that use MOEA to optimize two complementary clustering objectives based on compactness and connectivity with automatics k determination scheme. In the initialization and clustering phase, it adopts its initialization using graph-based minimum spanning tree (MST) and two objective functions. MST-based initialization is based on two different objectives to obtain good initial spread of solutions and a close approximation of Pareto front (Handl and Knowles, 2007). After optimizing two objective function through generic operations – crossover, mutation, and selection, Pareto solutions with a different k is generated, which are a set of different tradeoffs between the two objectives over a range of different cluster numbers (Matake et al., 2007). In the k determination phase based on Gap statistics (Tibshirani, 2001), it is able to determine the final solution from the Pareto solutions and find the optimal k. MOCK analyzes the output from the clustering phase, the tradeoffs and compare it to the tradeoffs obtained under an appropriate null reference distribution, such as uniform distribution. The estimate of the optimal number of clusters is the one falls the farthest away from the reference curve.

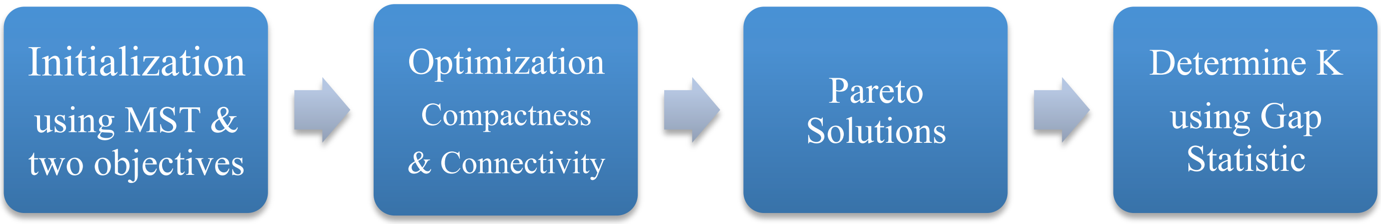

Initialization using MST & two objectives
Optimization Compactness & Connectivity
Pareto Solutions
Determine K using Gap Statistic

Figure 3.1 The general procedure of MOCK algorithm

Two Objective Functions

The two complementary objectives of MOCK selected are connectivity and compactness which are two criteria that reflect different aspects of good clustering solutions. Compactness is based on overall deviation and connectivity is based connectedness of clusters. To express the compactness of clusters, overall deviation is used which is defined as:

$$Dev(C) = \sum_{C_k \in C} \sum_{i \in C_k} \delta(i, \mu_k)$$

where C is the set of all clusters, μ_k is the centroid of cluster C_k, and δ is the distance function in which Euclidean distance is used in MOCK. Overall distance is minimized in order to make the clusters more compact. Minimizing the overall deviation increases the number of clusters.

Connectivity intends to group similar data into the same cluster. To express the connectivity of clusters, a measure of connectivity is used, which evaluates the degree to which neighboring data points have been placed in the same cluster.

$$Conn(C) = \sum_{i=1}^{N} (\sum_{j=1}^{L} x_{i,nn_{ij}})$$

where $x_{r,s} = \begin{cases} \frac{1}{j}, & if \nexists C_k : r \in C_k \wedge s \in C_k \\ 0, & otherwise \end{cases}$

Connectivity is minimized. When all the similar data are group into one cluster, connectivity becomes zero; when similar data are grouped into different clusters, connectivity increases by $\frac{1}{j}$. Minimizing the two complementary objectives gives the efficient solutions with different k, therefore MOCK returns a range of k minimizing the two objectives compactness and connectivity.

Initialization using Minimum Spanning Tree (MST)

The initialization is motivated by the fact that different single-objective clustering algorithms tend to perform well, or find good approximations in different regions of Pareto front (Handl and Knowles, 2007). In other words, algorithms based on connectivity tend to generate good approximations in the regions of Pareto front where connectivity is low, and algorithms based on compactness generate good approximation in the regions of Pareto front where overall deviation is low. Therefore, MOCK use an initialization based on two different single-objective algorithms to get a good spread of initial solutions. Solutions performing well under connectivity are generated using Minimum Spanning Tree (MST). MST is the shortest tree where the total cost of all edges is the minimum. In MST, similar data are connected and dissimilar data are not (Matake et al, 2007). On the other hand, solutions performing well under compactness are generated using k-means algorithms. Taking consideration of two objectives create good spread of initial solutions of pareto front close to the optimal solutions. MOCK uses Prim's algorithm (Prim, 1957) to create MST.

K-determination using Gap Statistic

MOCK carries over the concept of Gap Statistic (Tibshirani et al, 2001) to determine the number of clusters in the dataset. Gap statistic is based on the expectation that most suitable number of clusters shows a significant "knee" when plotting the performance plot. MOCK extends this concept and apply it to the case of two objectives. Having two objectives, overall deviation decreases with an increasing k, and an increasing connectivity. Define $R = \frac{\delta_{dev}}{\delta_{conn}}$, when k gradually increases, the significant change in R stands for an appropriate number of clusters, which is seen as “knee” (Handl and Knowles, 2007). MOCK also uses the random distribution (reference curve) from Gap statistics (Tibshirani et al, 2001). The purpose of this random distribution is getting estimates of the values of connectivity and overall deviation that can be expected for unstructured data. The estimate of optimal number of clusters would be the value of k for which the performance curve falls the farthest below the reference curve.

3.3 Compound and Constrained Clustering

We introduced two recently proposed novel multiple-objective cluster analysis methods (Zhang, 2011), the compound cluster analysis and the constrained cluster analysis. They cluster data using multiple data sources and similarity measures.

3.3.1 Motivation

Tradition cluster analysis consists of single objective and single distance metrics generated from the dataset. In reality, however, we may get data from multiple data sources describing the data from different views. All the information should be taken into consideration to better represent the data and let us better understand the objective of interest. A very good example in biology is the knowledge based clustering in biology. In biology, it is very common that we have continuous microarray data to describe the gene expression profile and also biology network database to describe the gene functions. Motivated by the needs in biology to consider more than just the gene expression data to cluster similar genes, Integrating all those information from difference data sources, such as the source of gene functions, knowledge based clustering will give us superior performance in grouping genes (Zhang, 2011).

Here we introduce the two recently proposed multiple-objective cluster analysis methods by Zhang (2011), the compound clustering and the constrained clustering, to cluster data by integrating multiple data sources and multiple dissimilarity measures. The general procedure will be given, and a dual-objective case (n=2) will be illustrated.

3.3.2 General framework of compound clustering and constrained clustering

Zhang (2011) proposed two novel multiple-objective cluster analysis methods. Suppose we have n datasets with n distance/dissimilarity measurements $\mathbf{D_1}, \dots, \boldsymbol{D_n}$

Compound Clustering

Clusters are obtained by minimizing the overall distance **D** as weighted average of the individual distance measurements. The overall distance can be defined as:

$$\mathbf{D} = \boldsymbol{\lambda_1 D_1 + \lambda_2 D_2 + \cdots + \lambda_n D_n, where \sum_{i=1}^{n} \lambda_i = 1}$$

Constrained Clustering

Constrained Clustering is an n-step algorithm, in which step we minimize $\mathbf{D_i}$ under the constraint that $\mathbf{D_j} \leq \boldsymbol{d_j, j = 1, 2, \dots, i - 1}$.

Algorithm for Constrained Clustering

1. Perform clustering analysis based on $\mathbf{D_1}$ on all objects

2. For object from $\mathbf{i = 2\ to\ n}$,

Perform clustering analysis based on $\mathbf{D_i}$ on each cluster generated from step i

And minimize $\mathbf{D_i}$

Dual-objective Case (n=2)

Compound clustering is to minimize the overall distance $\mathbf{D}$ with parameter $\boldsymbol{\lambda}$ as follow:

$$\mathbf{D} = \boldsymbol{\lambda}\mathbf{D_1} + (\mathbf{1} - \boldsymbol{\lambda})\boldsymbol{D_2}$$

Constrained clustering is a two-step approach to minimize $\mathbf{D_2}$ subject to the constraint that $\mathbf{D_1} \leq \boldsymbol{d_1}$. In other words, firstly cluster analysis based on $\mathbf{D_1}$ is performed. Secondly, in each cluster generated from step one, cluster analysis is performed based on $\mathbf{D_2}$ to determine the final cluster results. Compound clustering and constrained clustering are not equivalent with each other in term of clustering regions.

Compound Clustering

$$\mathbf{D}(\mathbf{X_1}, \mathbf{X_2}) \leq \mathbf{d} \Leftrightarrow \mathbf{X_1}\ \boldsymbol{and}\ \boldsymbol{X_2}\ \boldsymbol{are\ clustered}$$

Constrained Clustering

$$\mathbf{D_1}(\mathbf{X_1}, \mathbf{X_2}) \leq \boldsymbol{c_1}\ \&\ \boldsymbol{D_2}(\boldsymbol{X_1}, \boldsymbol{X_2}) \leq \boldsymbol{c_2} \Leftrightarrow \mathbf{X_1}\ \boldsymbol{and}\ \boldsymbol{X_2}\ \boldsymbol{are\ clustered}$$

The figure below shows the comparison of clustering regions of compound clustering and constrained clustering region:

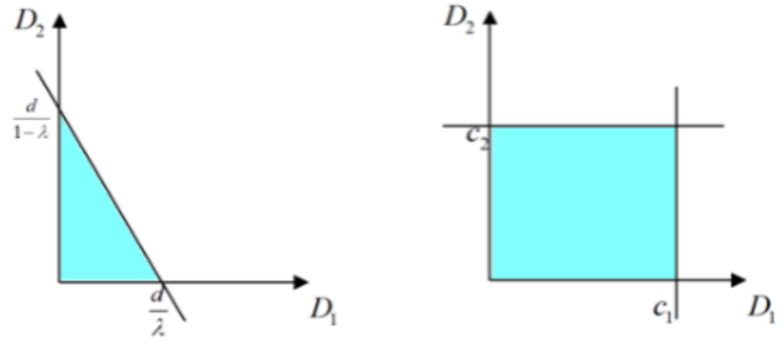


Figure 3.2 Clustering regions of compound clustering and constrained clustering (Zhang, 2011)

From clustering regions, it is obvious that the two clustering methods are not equivalent. It can be easily generalized to the case when $\mathbf{n} > \mathbf{2}$.

Zhang (2011) performed compound clustering on microarray data from Cold Spring Harbor Laboratory (CSHL) to illustrate the method. He used two functional distance and two statistic to determine the appropriate parameter (λ, k) for compound clustering, specifically, he used correlation base distance as the gene expression distance D_1, and two candidates for measuring the functional distance D_2, the Hamming/Euclidean distance and the Kappa statistic based distance together with two approaches, his newly proposed “Complete Linkage R^2” and the largest silhouette to determine the parameters. He performed the compound clustering using the hierarchical clustering algorithm. By comparison, he concluded that Euclidean distance as functional distance together with his newly proposed “Complete Linkage R^2” is the best combination for compound clustering to generate the most meaningful clusters with enriched biological functions. He also compared the his compound clustering incorporating biological information to the traditional single-objective hierarchical clustering method using only gene expression data, and found that compound clustering with Euclidean distance as functional distance yields more biological clusters.

3.4 Biclustering Analysis

3.4.1 Motivation

Clustering as an unsupervised learning technique clusters objects with the same attributes or functions. Traditional clustering algorithm clusters the data based either rows or the columns which sometimes is very difficult to extract local patterns including subsets of rows and subsets of columns while biclustering takes consideration of both rows and columns. It clusters data along the rows and columns simultaneously. The goal of biclustering is to find statistically significant sub-matrices, called biclusters. In other words, it searches for interpretable biological structure in gene expression microarray data.

Biclustering was first used by Cheng and Church (2000) in gene expression data analysis. Tradition clustering identifies groups of genes/conditions that show similar activity under all the set of conditions and all the set of genes under analysis while biclustering identifies groups of genes with similar/coherent expression patterns under a specific subset of the conditions. In biology, clustering gene similarities reflected by their activity across all conditions is that all genes in the cluster share the exact same functions, therefore all effected by the same conditions. However, many genes could have more than one function which means that a group of genes displays similar expression behavior across some of the conditions (those related to the shared functions) and displays different expression behavior in some conditions relating to functions not shared among all the gens in the group (Sharan, 2006). Genes with similar expression patterns are likely to be regulated by the same factors and therefore may share function. Analyzing gene expression profiles from different biological conditions and identifying joint patterns of gene expression among them, many researchers have characterized transcriptional programs and assigned putative function to thousands of genes (Spellman et al., 1998; Hughes et al., 2000; Gasch et al., 2001; Tanay et al., 2004). Influential papers including Eisen, Spellman, Brown and Bostein (1998) apply clustering methods to identify groups co-regulated genes from microarray data (Lazzeroni and Owen, 2000). Standard clustering analysis is oversimplified to detect the underlying patterns and biclustering is considered more appropriate for gene expression analysis. Biclustering performs clustering along both the genes and conditions (expression) simultaneously to find a joint behavior of some genes under some conditions while clustering can be applied to either rows (genes) or columns (conditions) of data matrix separately. Figure 3.3 below illustrates the difference between clustering and biclustering. Clusters of genes alone would be $(\{G_2, G_3\}, C)$, highlighted in orange. Clusters of conditions alone would be $(G, \{C_2, C_3, C_4\})$, highlighted in blue. Biclusters of both rows (genes) and columns (conditions) can be represented as $(\{G_2, G_3\}, \{C_2, C_3, C_4\})$, which is highlighted in green.

	C_1	**C_2**	**C_3**	**C_4**	**C_5**
G_1	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
G_2	A_{21}	A_{22}	A_{23}	A_{24}	A_{25}
G_3	A_{31}	A_{32}	A_{33}	A_{34}	A_{35}
G_4	A_{41}	A_{42}	A_{43}	A_{44}	A_{45}

Figure 3.3 Clustering vs Biclustering

3.4.2 Biclustering Types

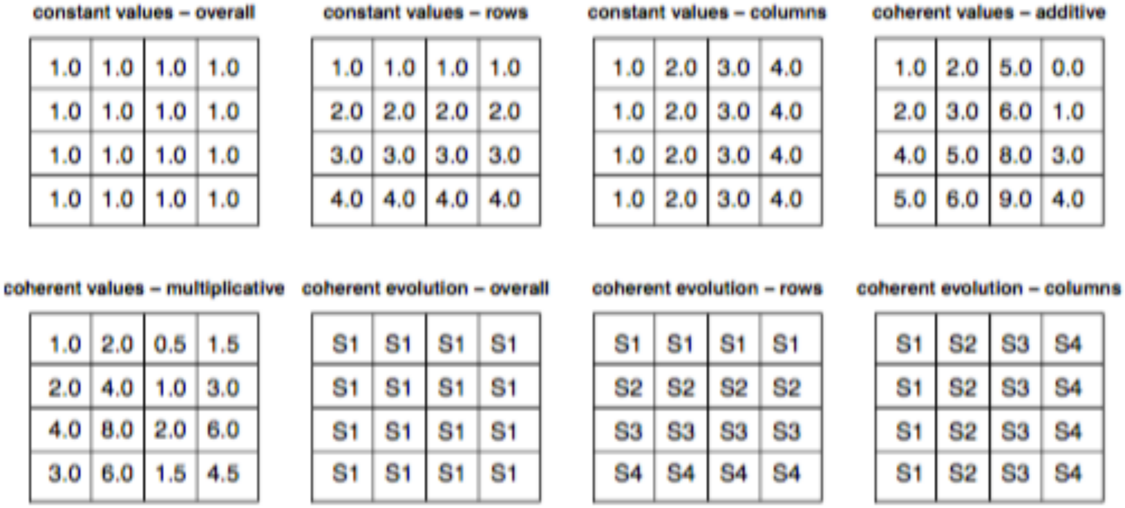
Just like traditional clustering, there many different ways of calculating the similarity within a bicluster, and many different algorithms have been development by researchers based this. Madeira and Oliveria (2004) reported that biclusters can be generally grouped into four types:

1) Biclusters with constant values

2) Biclusters with constant values on rows and columns

3) Biclusters with coherent values

4) Biclusters with coherent evolutions



constant values – overall

1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0

constant values – rows

1.0	1.0	1.0	1.0
2.0	2.0	2.0	2.0
3.0	3.0	3.0	3.0
4.0	4.0	4.0	4.0

constant values – columns

1.0	2.0	3.0	4.0
1.0	2.0	3.0	4.0
1.0	2.0	3.0	4.0
1.0	2.0	3.0	4.0

coherent values – additive

1.0	2.0	5.0	0.0
2.0	3.0	6.0	1.0
4.0	5.0	8.0	3.0
5.0	6.0	9.0	4.0

coherent values – multiplicative

1.0	2.0	0.5	1.5
2.0	4.0	1.0	3.0
4.0	8.0	2.0	6.0
3.0	6.0	1.5	4.5

coherent evolution – overall

S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1

coherent evolution – rows

S1	S1	S1	S1
S2	S2	S2	S2
S3	S3	S3	S3
S4	S4	S4	S4

coherent evolution – columns

S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4

Figure 3.4 Example of different bicluster types (Madeira and Oliveria, 2004)

Figure 3.4 explains the typical examples for those types corresponding to the four types reported in Madeira and Oliveria (2004). The last three aim to find bicluster with coherent evolutions, which means that the algorithms are not looking for exact numeric value of the matrix elements, instead it is looking for subsets of columns with coherent behaviors.

3.4.3 Algorithms

Hartigan's Direct Clustering (1976)

Biclustering was originally introduced by Hartigan in 1972. It is not generalized until 2000 when Cheng and Church proposed a biclustering algorithm based on variance and apply to gene expression data. Hartigan's direct clustering begins with the entire data in a single block and then at each stage finds the row or column split of every block into two pieces, choosing the one that produces largest reduction in the sum of square, the splitting is continued till the reduction of SSQ is less than a given threshold. Equivalently, it is minimizing the SSQ (i.e., minimum variance).

Objective function is the sum of squares:

$$SSQ = \sum_{k} \sum_{i,j \in B_k} \left(a_{ij} - b_k\right)^2$$

$$SSQ = \sum_{i \epsilon I, j \epsilon J} \left(a_{ij} - a_{IJ}\right)^2$$

where b_k is the average value in the bicluster B_k, a_{ij} is the element value in the cluster. The clustering technique was introduced in way in which the model for a single cluster relates a cluster of variables to a cluster of cases. Variables and cases are thus clustered simultaneously.

Cheng and Church (2000)

Cheng and Church (2000) were the first to introduce biclustering to gene expression analysis. The algorithm searches for constant values, rows or columns, where they define a score for each candidate bicluster. The objective function – the mean square residue score (MSRS) is defined as:

$$H(I,J) = \frac{1}{|I||J|}\sum_{i\in I, j\in J} R^2(a_{ij}) = \frac{1}{|I||J|}\sum_{i\in I, j\in J}\left(a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}\right)^2$$

where a_{Ij}, a_{iJ}, and a_{IJ} are defined as

$$a_{Ij} = \frac{1}{I}\sum_{i=1}^{I} a_{ij}$$

$$a_{iJ} = \frac{1}{J}\sum_{j=1}^{J} a_{ij}$$

$$a_{IJ} = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J} a_{ij}}{IJ}$$

They are the row and column means and the mean in the submatrix (I,J). A submatrix A_{IJ} is called a $\delta - bicluster$ if $H(I,J) \leq \delta$ for some $\delta \geq 0$.

Cheng and Church assume that genes conditions pairs in a good has constant expression level, and row column effects. After removing row column, bicluster averages, the residual should be as small as possible (Tanay et al., 2004). The subset is called a cluster if the score is below a level δ.

Row variance is defined as

$$Var(I,J) = Var(Row) = \frac{1}{|J|}\sum_{j\epsilon J}\left(a_{ij} - a_{Ij}\right)^2$$

Column variance is defined as

$$Var(I,J) = Var(Column) = \frac{1}{|I|}\sum_{i\epsilon I}\left(a_{ij} - a_{iJ}\right)^2$$

Representation of Objective Function into Linear Combination of Different Objective Functions

The objective function:

$$H(I,J) = \frac{1}{|I||J|}\sum_{i\in I, j\epsilon J}\left(a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}\right)^2$$

$$= \frac{1}{|I||J|}\sum_{i\epsilon I, j\epsilon J}\left[\left(a_{ij} - a_{iJ}\right) + \left(a_{ij} - a_{Ij}\right) - a_{ij} + a_{IJ}\right]^2$$

$$= \frac{1}{|I||J|}\sum_{i\epsilon I, j\epsilon J}\left[\left(a_{ij} - a_{iJ}\right) + \left(a_{ij} - a_{Ij}\right) - \left(a_{ij} - a_{IJ}\right)\right]^2$$

$$= \frac{1}{|I||J|}\sum_{i\epsilon I, j\epsilon J}\left(a_{ij} - a_{Ij}\right)^2 + \frac{1}{|I||J|}\sum_{i\epsilon I, j\epsilon J}\left(a_{ij} - a_{iJ}\right)^2 + \frac{1}{|I||J|}\sum_{i\epsilon I, j\epsilon J}\left(a_{ij} - a_{IJ}\right)^2$$

$$= \frac{1}{|I|}\sum_{i\in I}\left[\frac{1}{|J|}\sum_{j\epsilon J}\left(a_{ij} - a_{Ij}\right)^2\right] + \frac{1}{|J|}\sum_{j}\left[\frac{1}{|I|}\sum_{i\epsilon I}\left(a_{ij} - a_{iJ}\right)^2\right] + \frac{1}{|I||J|}\sum_{i\epsilon I, j\epsilon J}\left(a_{ij} - a_{IJ}\right)^2$$

$$= \frac{1}{|I|}\sum_{i} Var(Row) + \frac{1}{|J|}\sum_{j} Var(Column) + \frac{1}{|I||J|}\sum_{i\epsilon I, j\epsilon J}\left(a_{ij} - a_{IJ}\right)^2$$

$$= MVAR(Row) + MVAR(Col) + VAR(overall)$$

It can be represented as the linear combination of three objective functions, which equals variance of the set of all elements in the bicluster, plus the mean row variance and the mean column variance. It is the effort representing the objective function in Cheng and Church (2000) in the form of compound clustering, where it's a linear combination of three objective functions. Zhang’s compound clustering (2011) incorporated more than single data source, so it’s linear objective function involving more than one objective functions from more than one data matrix, while here we only consider one data matrix, but perform clustering based on rows and columns simultaneously. So it is the linear combination of objective functions involving only one data matrix (in terms of rows/or columns).

From the objective function in Cheng and Church, where it can be represented as the linear combination of three objective functions, the third objective function, the variance of all elements in the bicluster, or the sum of squares, is actually the objective function in Hartigan’s. In other words, Cheng and Church is a generalization of Hartigan’s direct clustering, where it considers the row effect, column effect and overall effect, while only sum of squares was taken into consideration in Hartigan’s.

3.4.4 Generation of Cheng and Church Algorithm

Cheng and Church define the objective score function as

$$H(I,J) = \frac{1}{|I||J|} \sum_{i\in I, j\epsilon J} \left(a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}\right)^2$$

which proved can be equally written as

$$H(I,J) = \frac{1}{|I||J|}\sum_{i\epsilon I, j\epsilon J}\left(a_{ij} - a_{Ij}\right)^2 + \frac{1}{|I||J|}\sum_{i\epsilon I, j\epsilon J}\left(a_{ij} - a_{iJ}\right)^2 + \frac{1}{|I||J|}\sum_{i\epsilon I, j\epsilon J}\left(a_{ij} - a_{IJ}\right)^2$$

$$= MVAR(Row) + MVAR(Col) + VAR(overall)$$

where the parameters are fixed.

We generalize the objective function in Cheng and Church (2000) by representing the objective function in the form of compound clustering with the objective function in terms of rows and objective function in terms of columns as below:

$$H(I,J)_{general} = \lambda * \frac{1}{|I||J|}\sum_{i\epsilon I, j\epsilon J}\left(a_{ij} - a_{Ij}\right)^2 + (1-\lambda) * \frac{1}{|I||J|}\sum_{i\epsilon I, j\epsilon J}\left(a_{ij} - a_{iJ}\right)^2$$

$$= \lambda * MVAR(Row) + (1-\lambda) * MVAR(Col)$$

where the parameters λ is not fixed.

The goal is to search for the submatrices with generalized objective function $H(I,J)_{general} \leq \delta$.

3.4.5 Other Algorithms

Many other used algorithm such as Plaid model developed by Lazzeroni and Owen (2002) for analysis of gene expression data. This algorithm model data matrix to a sum of layers, the model is fitted through Ordinary Least Square (OLS) to minimize the error.

$$a_{ij} = \mu_0 + \sum_{k=1}^{K}\theta_{ijk}\rho_{ik}k_{jk} + \varepsilon_{ij}$$

$$a_{ij} = \left(\mu_0 + \alpha_{i0} + \beta_{j0}\right) + \sum_{k=1}^{K}\left(\mu_k + \alpha_{ik} + \beta_{jk}\right)\rho_{ik}k_{jk} + \varepsilon_{ij}$$

μ_0 corresponds to the global effect layer

θ_{ijk} models the effect of layer k

μ, α, βrepresent mean, row and column effect. ρ_{ik} $(or\ k_{jk})$ identifies whether a row or a column is member of the layer respectively, it equals 1 when object i(or j) belongs to layer k, 0 otherwise.

Parameters are estimated by minimizing sum of squared residual

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(a_{ij} - \theta_{ij0} - \sum_{k=1}^{K}\theta_{ijk}\rho_{ik}k_{jk}\right)^{2}$$

$$= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(a_{ij} - (\mu_0 + \alpha_{i0} + \beta_{j0}) - \sum_{k=1}^{K}(\mu_k + \alpha_{ik} + \beta_{jk})\rho_{ik}k_{jk}\right)^{2}$$

Murali and Kasif (2003) developed algorithm Xmotifs to find biclusters with coherent evolutions and etc. There are many other algorithms developed for gene expression data analysis, Tanay et al. (2004) gives a very detailed summary on those algorithms. A summary table of algorithms with objective functions is summarized below:

Table 3.1 Summary of biclustering algorithms with objective functions

Algorithm	Objective Function	Comment
Direct Clustering **Hartigan (1976)**	$SSQ_k = \sum_{i\epsilon I, j\epsilon J} \left(a_{ij} - a_{IJ}\right)_k^{\;2}$	Used in Hartigan's direct clustering, algorithm spits the original data matrix into a set of submatrices minimizing the variance
Cheng and Church **(2000)**	$H(I,J)_k = \frac{1}{\lvert I\rvert \lvert J\rvert} \sum_{i\in I, j\epsilon J} \left(a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}\right)_k^{\;2}$	First algorithm applied on gene expression data, distance-based algorithm, can be represented as the summation of mean row variance, mean column variance and overall variance of submatrix.
Plaid Model **Lazzeroni and Owen** **(2002)**	$Q_k = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(a_{ij} - \theta_{ij0} - \sum_{k=1}^{K}\theta_{ijk}\rho_{ik}k_{jk}\right)_k^{2}$	Value of an element in the data matrix is assumed as sum of layers. Data matrix is described as linear function of layers corresponding to its biclusters: $a_{ij} = \sum_{k=0}^{K}\theta_{ijk}\rho_{ik}k_{jk}$ OLS is used to estimate the parameters.

3.6 Application of Biclustering on Microarray Gene Expression Data

Microarray yeast data

We use the microarray dataset Yeast data Prelic et al. (2006) used for biclustering technique to illustrate the algorithms. The microarray dataset is a subsample of the Saccharomyces Cerevisiae organism (Yeast), which contains 419 rows (Genes) and 70 columns (Conditions/Expression levels).

We demonstrate the biclustering algorithms by Cheng and Church and plaid models using the gene expression data set. There are build up software packages for all those algorithms developed in R, the one we used here is called “Biclust” (Kaiser and et al., 2015) which comes with a lot of different algorithms. The heatmap of the first bicluster is shown below, we can also see the local heatmap with the first bicluster only with its rows (gene names) and columns (conditions) shown in the bottom right. Since Cheng and Church algorithm searches constant rows or columns, the heatmap of CC algoriothm of the first bicluster shows the same color compared to the plaid model result.

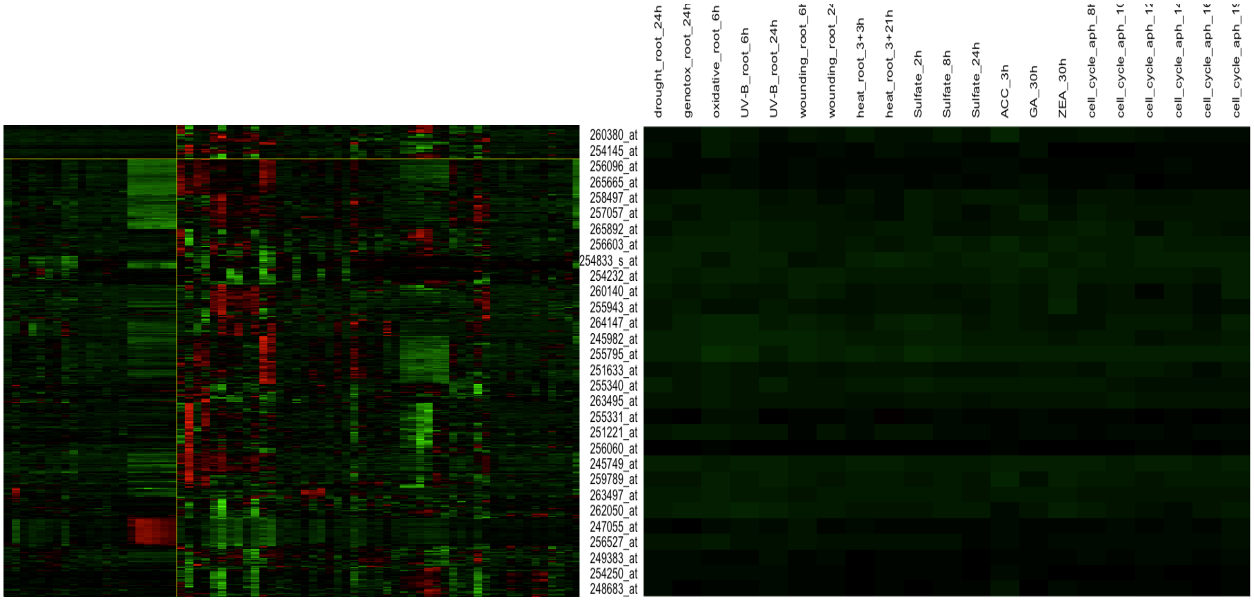


Figure 3.5 Left: heatmap of first bicluster performing CC algorithm Right: local heatmap of first bicluster performing CC algorithm

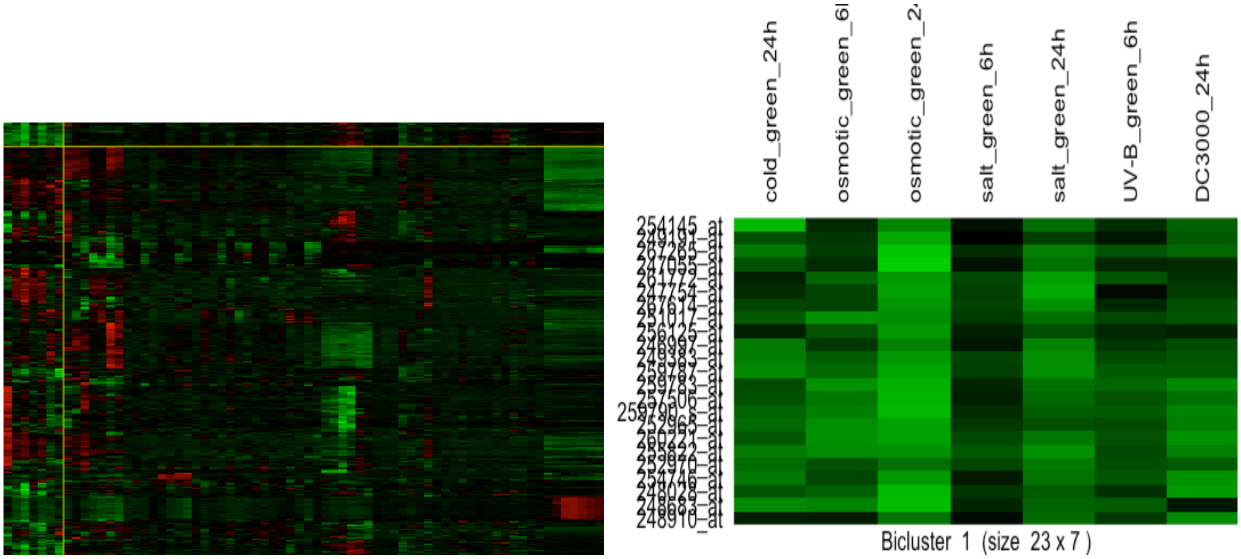


Figure 3.6 Left: heatmap of first bicluster performing plaid model Right: local heatmap of first bicluster performing plaid model

Setting the threshold $\delta = 0.01$, Cheng and Church algorithm searches for submatrices with mean square residual score $H(I, J) \leq \delta$, Plaid model models data matrix with a sum of layers, and the model is fitted through minimization of the error as described before.

10 biclusters patterns are found in result of performing plaid models, and 35 bicluster patterns are found using Cheng and Church algorithm. We only show the first two biclusters in the summary table 3.2. Table 3.2 is a summary table of first two biclustering results performing Cheng and Church algorithm and plaid model algorithm. We list the total number of biclusters found performing the two algorithms as well as the result on the first bicluster found.

Table 3.2 Summary of biclustering results of CC algorithm and plaid model algorithm

	Cheng & Church	Plaid Model
Total number of biclusters found	35	10
BC1	30 rows 21 columns	23 rows 7 columns
BC2	25 rows 14 columns	29 rows 8 columns

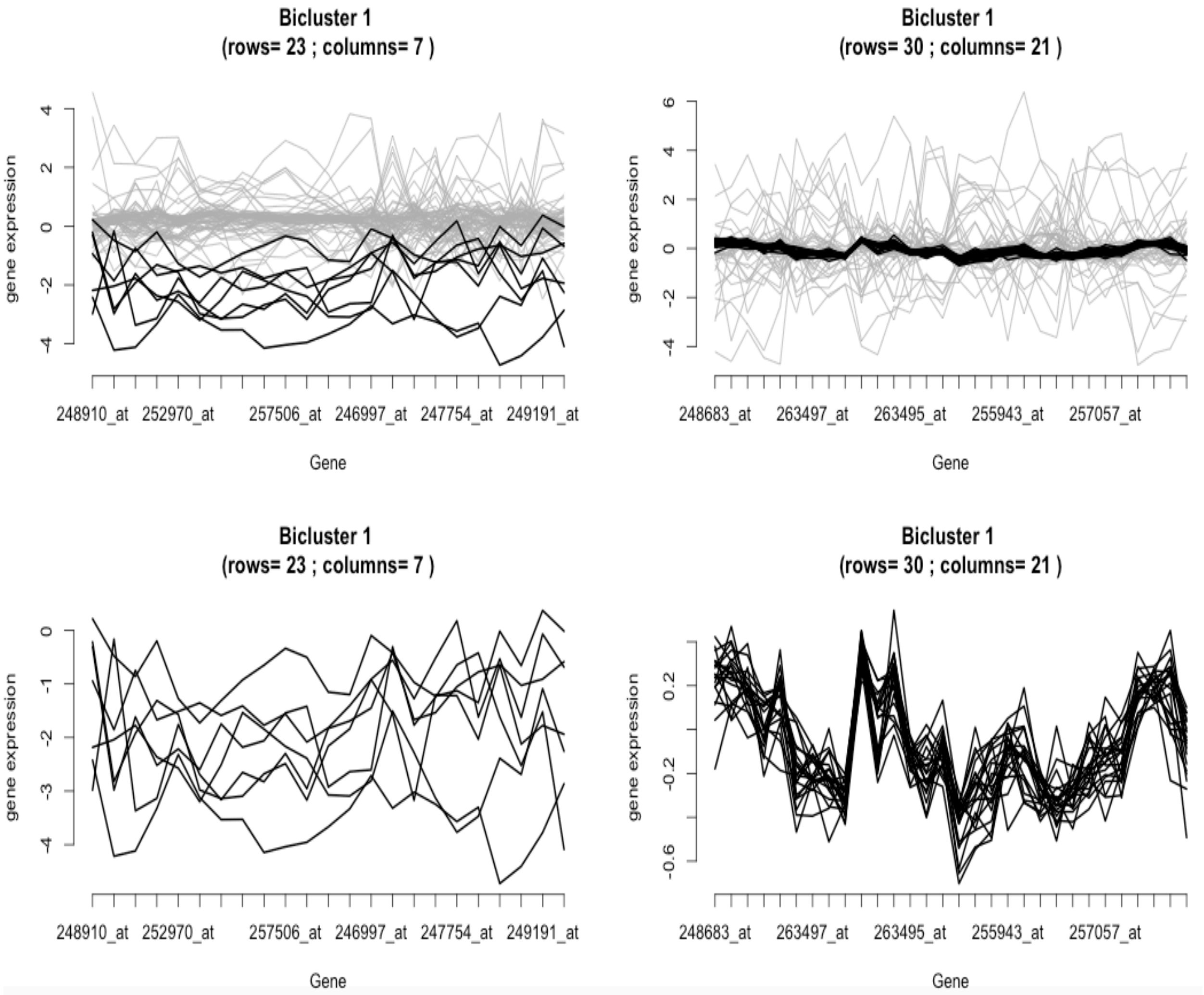


Figure 3.7 (a) Left two: profile plots - expression levels of conditions across their genes in the first bicluster of plaid models. Right two: profile plots – expression levels of conditions across their genes in the first bicluster of Cheng & Church algorithm.

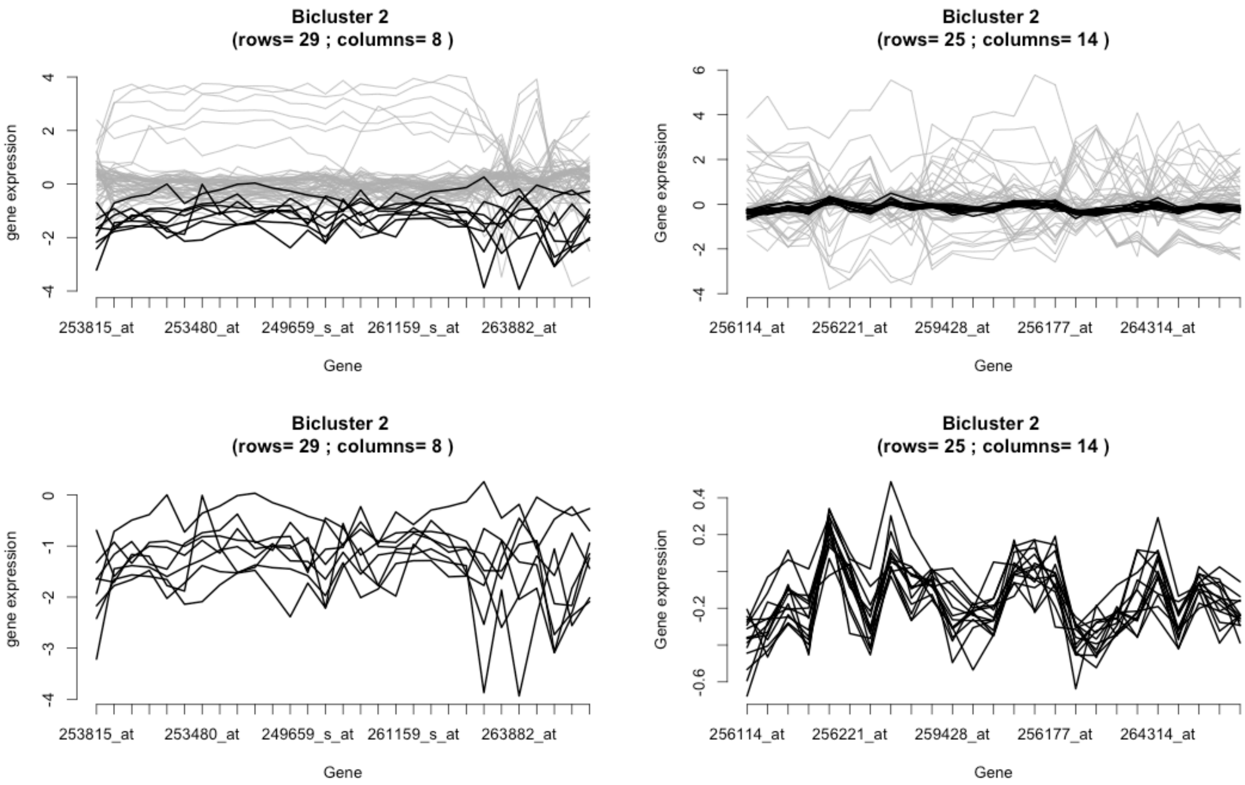


Figure 3.7 (b) Left two: profile plots - expression levels of conditions across their genes in bicluster two of plaid models. Right two: profile plots – expression levels of conditions across their genes in the bicluster two of Cheng & Church algorithm.

Note: Dark lines are the profile of the corresponding biclusters, grey lines are the remaining of the data

The profile plot (i.e., the parallelCoordinates plot), shows the expression levels of conditions across genes. The left two plots of Figure 3.7(a) and Figure 3.7(b) are the profile plots of first bicluster and second bicluster in the result of performing plaid models, and right two plots of Figure 3.6 are the profile plots of first bicluster and second bicluster in the result of performing Cheng and Church algorithm. Both Figure 3.7(a) and 3.7(b) shows that the plaid model identifies genes conditions patterns with expression levels close to zero. Cheng and

Church algorithm identifies bicluster patterns with smaller variance than the plaid model, clearly observed in the bottom row of zoomed in version which makes sense because Cheng and Church algorithm searches for biclusters with constant value. Table 3.3 is a summary table of the results performing CC algorithm and Plaid model, we only list the first two biclusters.

Chapter 4

Proposed Multiple-objective Clustering Approach

– Generalized Compound Biclustering Algorithm

We generalized the original Cheng and Church algorithm, representing the objective function in the form of compound clustering where it can be represented as the linear combination of different objective functions with regards to rows and columns, and compare it to the original CC algorithm on the gene expression dataset and the simulated dataset.

4.1 Existing Method with objective function

Cheng and Church (2000)

Cheng and Church (2000) were the first to introduce biclustering to gene expression analysis. The algorithm searches for constant values, rows or columns, where they define a score for each candidate bicluster. The objective function – the mean square residue score (MSRS) is defined as:

Objective Function:

$$H(I,J) = \frac{1}{|I||J|}\sum_{i\in I, j\in J} R^2(a_{ij}) = \frac{1}{|I||J|}\sum_{i\in I, j\in J}\left(a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}\right)^2$$

where a_{Ij}, a_{iJ}, and a_{IJ} are defined as

$$a_{Ij} = \frac{1}{I}\sum_{i=1}^{I} a_{ij}$$

$$a_{iJ} = \frac{1}{J}\sum_{j=1}^{J} a_{ij}$$

$$a_{IJ} = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J} a_{ij}}{IJ}$$

They are the row and column means and the mean in the submatrix (I,J). A submatrix A_{IJ} is called a $\delta - bicluster$ if $H(I,J) \leq \delta$ for some $\delta \geq 0$.

Cheng and Church assume that genes conditions pairs in a good has constant expression level, and row column effects. After removing row column, bicluster averages, the residual should be as small as possible (Tanay et al., 2004). The subset is called a cluster if the score is below a level δ.

4.2 Representation of the objective function in the form of compound clustering

We write and prove the objective function in Cheng and Church's algorithm written as below:

$$H(I,J) = \frac{1}{|I||J|}\sum_{i\in I,j\in J}\left(a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}\right)^2$$

$$= \frac{1}{|I||J|}\sum_{i\epsilon I,j\epsilon J}\left[(a_{ij} - a_{iJ}) + \left(a_{ij} - a_{Ij}\right) - a_{ij} + a_{IJ}\right]^2$$

$$= \frac{1}{|I||J|}\sum_{i\epsilon I,j\epsilon J}\left[(a_{ij} - a_{iJ}) + \left(a_{ij} - a_{Ij}\right) - (a_{ij} - a_{IJ})\right]^2$$

$$= \frac{1}{|I||J|}\sum_{i\epsilon I,j\epsilon J}\left(a_{ij} - a_{Ij}\right)^2 + \frac{1}{|I||J|}\sum_{i\epsilon I,j\epsilon J}\left(a_{ij} - a_{iJ}\right)^2 + \frac{1}{|I||J|}\sum_{i\epsilon I,j\epsilon J}\left(a_{ij} - a_{IJ}\right)^2$$

$$= \frac{1}{|I|}\sum_{i \in I}[\frac{1}{|J|}\sum_{j \epsilon J}\left(a_{ij} - a_{Ij}\right)^2] + \frac{1}{|J|}\sum_{j}[\frac{1}{|I|}\sum_{i \epsilon I}\left(a_{ij} - a_{iJ}\right)^2] + \frac{1}{|I||J|}\sum_{i \epsilon I, j \epsilon J}\left(a_{ij} - a_{IJ}\right)^2$$

$$= \frac{1}{|I|}\sum_{i} Var(Row) + \frac{1}{|J|}\sum_{j} Var(Column) + \frac{1}{|I||J|}\sum_{i \epsilon I, j \epsilon J}\left(a_{ij} - a_{IJ}\right)^2$$

$$= MVAR(Row) + MVAR(Col) + VAR(overall)$$

where row variance is defined as

$$Var(I,J) = Var(Row) = \frac{1}{|J|}\sum_{j \epsilon J}\left(a_{ij} - a_{Ij}\right)^2$$

Column variance is defined as

$$Var(I,J) = Var(Column) = \frac{1}{|I|}\sum_{i \epsilon I}\left(a_{ij} - a_{iJ}\right)^2$$

4.3 Generalization of Cheng and Church Algorithm

We generalize the objective function in Cheng and Church (2000) by representing the objective function in the form of compound clustering with the objective function in terms of rows and objective function in terms of columns as below:

$$H(I,J)_{general} = \lambda * \frac{1}{|I||J|}\sum_{i \epsilon I, j \epsilon J}\left(a_{ij} - a_{Ij}\right)^2 + (1 - \lambda) * \frac{1}{|I||J|}\sum_{i \epsilon I, j \epsilon J}\left(a_{ij} - a_{iJ}\right)^2$$

$$= \lambda * MVAR(Row) + (1 - \lambda) * MVAR(Col)$$

where the parameters λ is not fixed.

The goal is to search for the submatrices with generalized objective function $H(I,J)_{general} \leq \delta$.

4.4 Relationship between the original algorithm and the generalized compound algorithm

Objective function in Cheng and Church:

$$H(I,J) = \frac{1}{|I||J|} \sum_{i \in I, j \epsilon J} \left(a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}\right)^2$$

which after derivation, we can write the objective function in the following intuitive form:

$$\frac{1}{|I||J|} \sum_{i \epsilon I, j \epsilon J} \left(a_{ij} - a_{Ij}\right)^2 + \frac{1}{|I||J|} \sum_{i \epsilon I, j \epsilon J} \left(a_{ij} - a_{iJ}\right)^2 + \frac{1}{|I||J|} \sum_{i \epsilon I, j \epsilon J} \left(a_{ij} - a_{IJ}\right)^2$$

$$= MVAR(Row) + MVAR(Col) + VAR(overall)$$

The first part is the mean row variance, the second part is mean column variance, and the third part is the overall variance of the bicluster matrix.

Where

$$Var(Row) = \frac{1}{|J|} \sum_{j \epsilon J} \left(a_{ij} - a_{Ij}\right)^2$$

$$Var(Column) = \frac{1}{|I|} \sum_{i \epsilon I} \left(a_{ij} - a_{iJ}\right)^2$$

Cheng and Church algorithm adds up the mean row variance, mean column variance and the overall variance with fixed parameter 1 in front of each part.

Objective in the generalized algorithm:

We extend the Cheng and Church algorithm by introducing a parameter λ and express the objective function in the form of compound clustering where it's linear relationship of different objective functions in term of rows and column in the form of below:

$$H(I,J)_{general} = \lambda * \frac{1}{|I||J|} \sum_{i\epsilon I, j\epsilon J} \left(a_{ij} - a_{Ij}\right)^2 + (1-\lambda) * \frac{1}{|I||J|} \sum_{i\epsilon I, j\epsilon J} \left(a_{ij} - a_{iJ}\right)^2$$

$$= \lambda * MVAR(Row) + (1-\lambda) * MVAR(Col)$$

where it's linear combination of mean row variance and mean column variance with flexible parameter λ.

4.5 Evaluation Measurements

4.5.1 Evaluation of single bicluster Measures

Variance (VAR)

Hartigan used bicluster variance as a coherence measure, where the goal is to minimize the sum of bicluster variances

$$VAR(I,J) = \sum_{i\in I, j\in J} \left(a_{ij} - a_{IJ}\right)_k^{\ 2}$$

The smaller the variance is, the more coherent the bicluster is, and better its quality.

Mean Squared Residue (MSR)

Cheng and Church defined the mean square residue score as below, and he used the MSR to access the quality of biclusters. Cho et al. also used the residue as the measure to evaluate the homogeneity of a bicluster.

$$H(I,J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} r\left(a_{ij}\right)_k^{\ 2} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} \left(a_{ij} - a_{Ij} - a_{iJ} + a_{IJ}\right)_k^{\ 2}$$

The lower the mean squared residue, the stronger the coherence exhibited by the bicluster, and better its quality.

Constance and coherence Variance

The variance is defined by (Sebastian Kaiser, 2011) building on the work by Madeira and Oliveira (2004). The Constance variance of rows returns the corresponding variance of rows as the average sum of Euclidean distances between all rows of the bicluster denoted as x:

$$VAR = \frac{1}{nrow(x) * (nrow(x) - 1)} \sum_{i=1}^{nrow(x)} \sum_{j=1}^{nrow(x)} \left(\frac{1}{ncol(x)} \sum_{k=1}^{ncol(x)} (x[i,k] - x[j,k])^2 \right)$$

Similarly, the Constance variance of columns returns the corresponding variance of columns as the average sum of Euclidean distances between all columns of the biclusters, and the Constance variance of the bicluster returns the weighted mean of row and column calculation. The variance here is a coherence measurement, it measures how much coherent a bicluster is. The lower the value is, the more coherent or constant the bicluster is.

4.5.2 Evaluations of the overall quality of biclustering with k biclusters

Overall Variance

Hartigan (1976) introduced the partition-based algorithm called direct biclustering, and used variance to evaluate the quality of each bicluster. The quality of the resulting biclustering with K biclusters is evaluated by the overall variance of the K biclusters:

$$VAR(I,J)_K = \sum_{k=1}^{K} \sum_{i \in I, j \in J} \left(a_{ij} - a_{IJ}\right)^2$$

Average residue

Yang et al. (2003) used the average residue to access the overall quality of a biclustering with K biclusters

$$\frac{1}{K}\sum_{k=1}^{K} H(I,J)_k$$

$$= \frac{1}{K}\sum_{k=1}^{K} \frac{1}{|I||J|} \sum_{i \in I, j \in J} \left(a_{ij} - a_{Ij} - a_{iJ} + a_{IJ}\right)_k^{\,2}$$

Total Squared Residue

Cho et al. (2004) used the mean squared residue score defined by Cheng and Church (2000) to evaluate the homogeneity of a bicluster, and used the total squared residue which is the sum of the squared residues of each bicluster to evaluate the overall quality of biclustering with K biclusters.

$$\sum_{I,J} H(I,J)$$

$$= \sum_{I,J} \frac{1}{|I||J|} \sum_{i \in I, j \in J} \left(a_{ij} - a_{Ij} - a_{iJ} + a_{IJ}\right)^2$$

4.6 Comparison on Gene Expression Data Set

4.6.1 Data

The microarray dataset is a subsample of the Saccharomyces Cerevisiae organism (Yeast), which contains 419 rows (Genes) and 70 columns (Conditions/Expression levels).

4.6.2 Determination of the parameter $\boldsymbol{\lambda}$

The key is to determine the parameter λ in the compound clustering. The generalized algorithm searches for the submatrices that satisfied $H(I,J)_{general} \leq \delta$. We set the threshold δ to be 0.02 and 0.05 and searches for the best combination of λ using the evaluation criteria discussed above. Once we find the best combination of the number of clusters and the parameters, we will compare the generalized algorithm to the original CC algorithm in both scenarios to see the performances. Table 1 and table 2 are the measurements of biclustering results when δ is 0.05 and 0.02. Figure 1 and figure 2 are the corresponding efficiency plots of the scenarios.

Table 4.1 Measurements of biclustering result when $\boldsymbol{\delta = 0.05}$

$\boldsymbol{\lambda}$	Number of Biclusters	Overall Variance	Total Squared Residue	Average Residue
0.1	16	0.93973337	0.59434673	0.037146671
0.2	15	0.86389596	0.5314821	0.03543214
0.3	17	1.05989824	0.56961205	0.033506591
0.4	18	1.10170686	0.61168484	0.033982491
0.5	16	0.97374735	0.57314383	0.035821489
0.6	14	0.91044969	0.4701282	0.033580586
0.7	16	1.12046176	0.502107165	0.031381698
0.8	13	1.12213149	0.45602933	0.035079179
0.9	12	1.4115522	0.44326254	0.036938545

Table 4.2 Measurements of biclustering result when $\boldsymbol{\delta = 0.02}$

$\boldsymbol{\lambda}$	Number of Biclusters	Overall Variance	Total Squared Residue	Average Residue
0.1	38	0.99646542	0.512536523	0.013487803
0.2	37	0.9209575	0.471540078	0.012744326
0.3	39	0.94096289	0.510466647	0.013088888
0.4	38	0.92478016	0.474831769	0.012495573

0.5	40	0.98449448	0.50764566	0.012691142
0.6	39	1.01786132	0.474742634	0.012172888
0.7	37	1.10057987	0.441356481	0.011928554
0.8	37	1.29950443	0.460428119	0.012444003
0.9	31	1.49734192	0.3777463	0.012185365

4.6.3 Efficiency plots

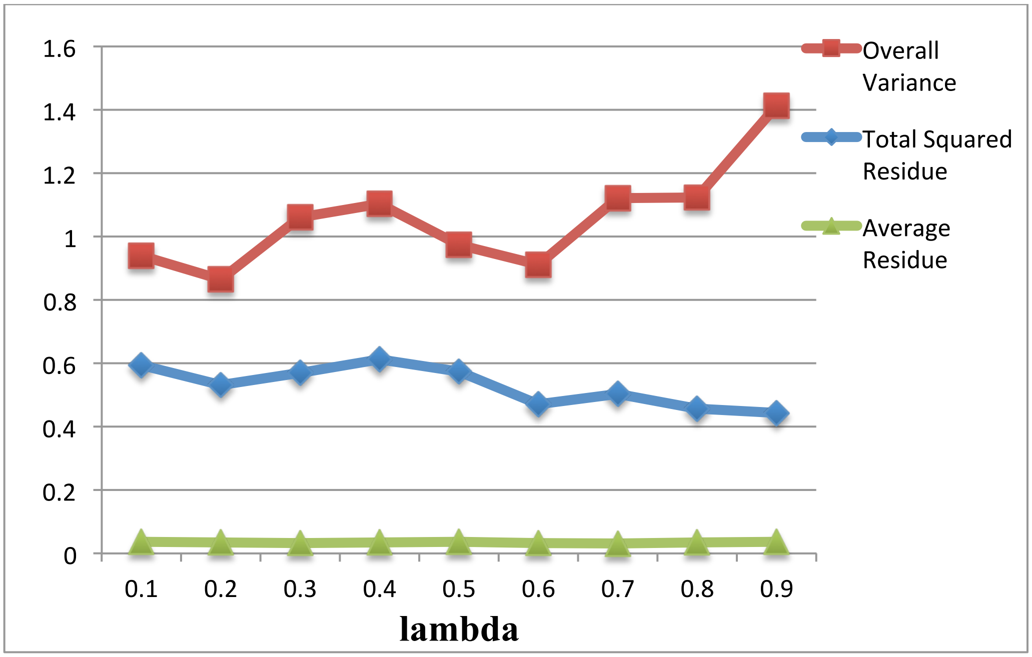


Figure 4.1 Efficiency plot of biclustering result when $\delta = 0.05$

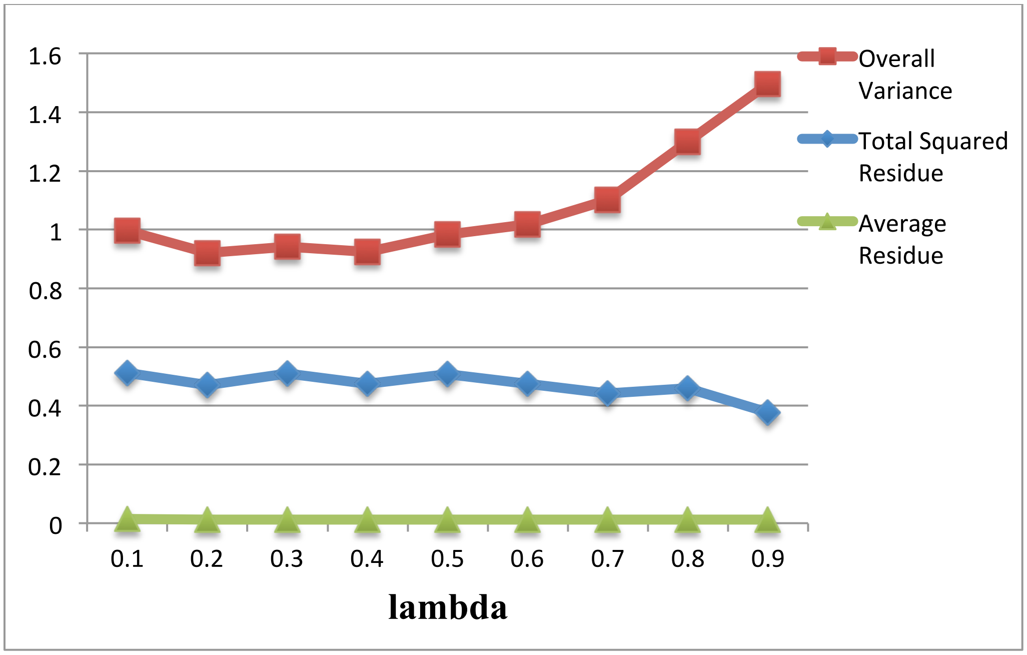


Figure 4.2 Efficiency plot of biclustering result when $\boldsymbol{\delta = 0.02}$

4.6.3 Results

Table 4.3 Summary of the biclustering results from the generalized CC algorithm

$\boldsymbol{\lambda}$	**Total Number of Biclusters**	**Objective Function** **Criterion Threshold** $\boldsymbol{H(I,J) \leq \delta}$
0.2	*37*	$\delta = 0.02$
0.4	38	
0.2	15	$\delta = 0.05$
0.6	*14*	

Table 3.4 is the summary of results performing the generalized CC algorithm represented in the form of compound clustering. In order to find the best combination of parameter λ and number of biclusters, we decided to set the threshold δ to be 0.02 and 0.05 respectively and

searches for the biclusters with our generalized objective function to satisfy $H(I,J)_{general} \leq \delta$. We decide the number of bicluster number and the parameter λ combination based on the evaluation criteria we found above. As we all know, a larger cluster number gives the less informative and meaningful result. Therefore, once we set δ to be 0.02 and 0.05 and we need to find λ that yields the smallest number of biclusters. Based on the results, we decided to choose $\lambda = 0.2$ with a corresponding bicluster number 37 when δ is 0.02, and $\lambda = 0.6$ with a corresponding bicluster number 14 when δ is 0.05. We will compare the generalized algorithm to the original CC algorithm in both scenarios when the threshold δ is set to be 0.02 and 0.05.

Table 4.4 Comparison performances on single biclusters of the generalized CC algorithm to the Original CC algorithm ($\boldsymbol{\delta = 0.02}$)

Algorithm	Total Number of Bicluster	Bicluster #	Standard Deviation	Coherence Variance	Mean Square Residue
CC ($\delta = 0.02$)	20	1	0.261819766	0.8166902	0.01977323
		2	0.23719277	0.6728342	0.01999681
		3	0.303272485	0.7636795	0.01993836
		4	0.217695085	0.5523057	0.01997881
		5	0.251900893	0.5519045	0.01951768
Generalized CC ($\delta = 0.02$, $\lambda = 0.2$)	37	1	0.150432144	0.4424418	0.013821774
		2	0.147961414	0.4193504	0.016383526
		3	0.148267731	0.3704083	0.015464489
		4	0.166964487	0.3637941	0.014766713
		5	0.151528446	0.3609276	0.014394898

Table 4.5 Comparison performances on single biclusters of the generalized CC algorithm to the Original CC algorithm ($\delta = 0.05$)

Algorithm	Total Number of Bicluster	Bicluster #	Standard Deviation	Coherence Variance	Mean Square Residue
CC ($\delta = 0.05$)	10	1	0.311913193	1.3139724	0.0497105
		2	0.264815577	1.0753153	0.04955709
		3	0.377244788	1.0987554	0.04965742
		4	0.381239636	1.1234724	0.04989543
		5	0.310587588	0.8215469	0.0474674
Generalized CC ($\delta = 0.05$, $\lambda = 0.6$)	14	1	0.246740937	1.0131388	0.04200287
		2	0.268741214	0.8524074	0.03202411
		3	0.248631897	0.7375823	0.03954227
		4	0.259087823	0.6830453	0.03312338
		5	0.275722197	0.6709196	0.02952475

Table 4.6 Comparison of the overall quality of the resulting biclustering of CC algorithm and generalized CC algorithm of K biclusters

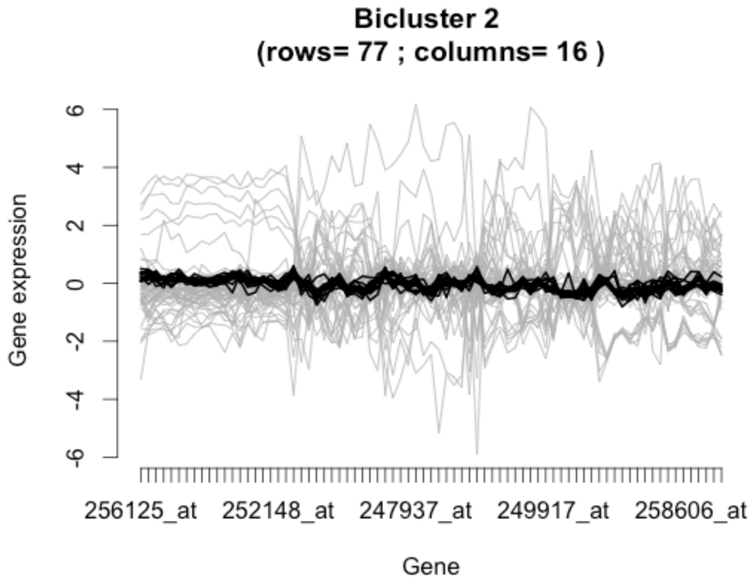
$\boldsymbol{\delta}$	Algorithm	Overall Variance (*Hartigan, 1972*)	Total Squared Residue (*Cho et al., 2004)*	Average Residue (*Yang et al., 2002*)
0.02	CC	2.03504914	0.38032461	0.019016231
	Generalized CC	0.9209575	0.471540078	0.012744326
0.05	CC	1.12123431	0.48121408	0.048121408
	Generalized CC	0.91044969	0.4701282	0.033580586

Table 4.4 and table 4.5 are the biclustering results of the two algorithm comparing on single biclusters using three evalution criteria, and table 4.6 are the biclustering results of the two algorithm comparing on the overall performance using the three overall biclustering quality measurements. The results show that generalized CC algorithm overall performs better than the original CC algorithm in both scenarios when the threshold δ is set to be 0.02 and 0.05.

In conclusion, we compare the two methods using three measurements, variance Hartigan (1976) used in his direct clustering, the mean square residue and the constant and coherence variance defined by Sebastian (2011) to evaluate the single bicluster. Table 4.4 and table 4.5 are the evaluation results of the three measures comparing the two methods. The result shows that the generalized CC algorithm has better results than the original CC algorithm overall evaluating single biclusters in both scenarios when the threshold δ is 0.02 and 0.05.

We used the overall variance, the average residue and the total squared residue to evaluate the overall quality of resulting biclustering with K biclusters (where K is the total number of biclusters found) shown in table 4.6. Table 4.6 shows that the generalized has better result (smaller in all three evaluations) than the original CC in both scenarios when the threshold δ is set to be 0.02 and 0.05.

Now we plot the profile plot to see the patterns those algorithms generate:



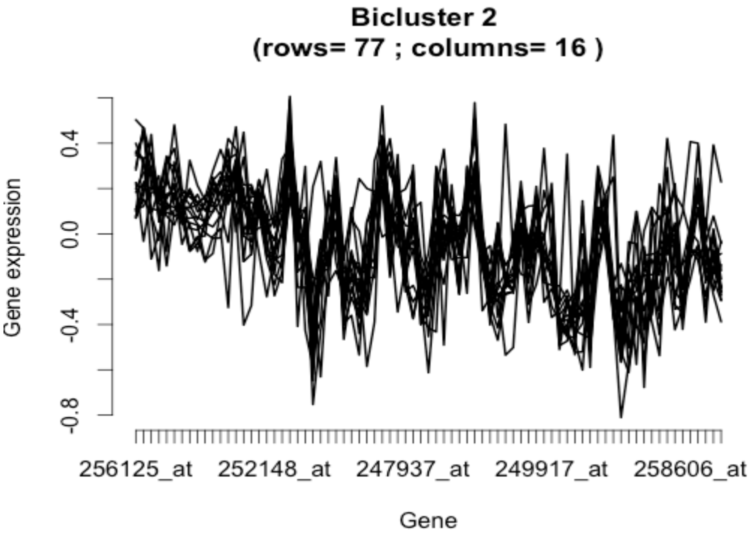


Figure 4.3 Profile plot of CC algorithm by Cheng and Church (2000) (expression levels of conditions across their genes)

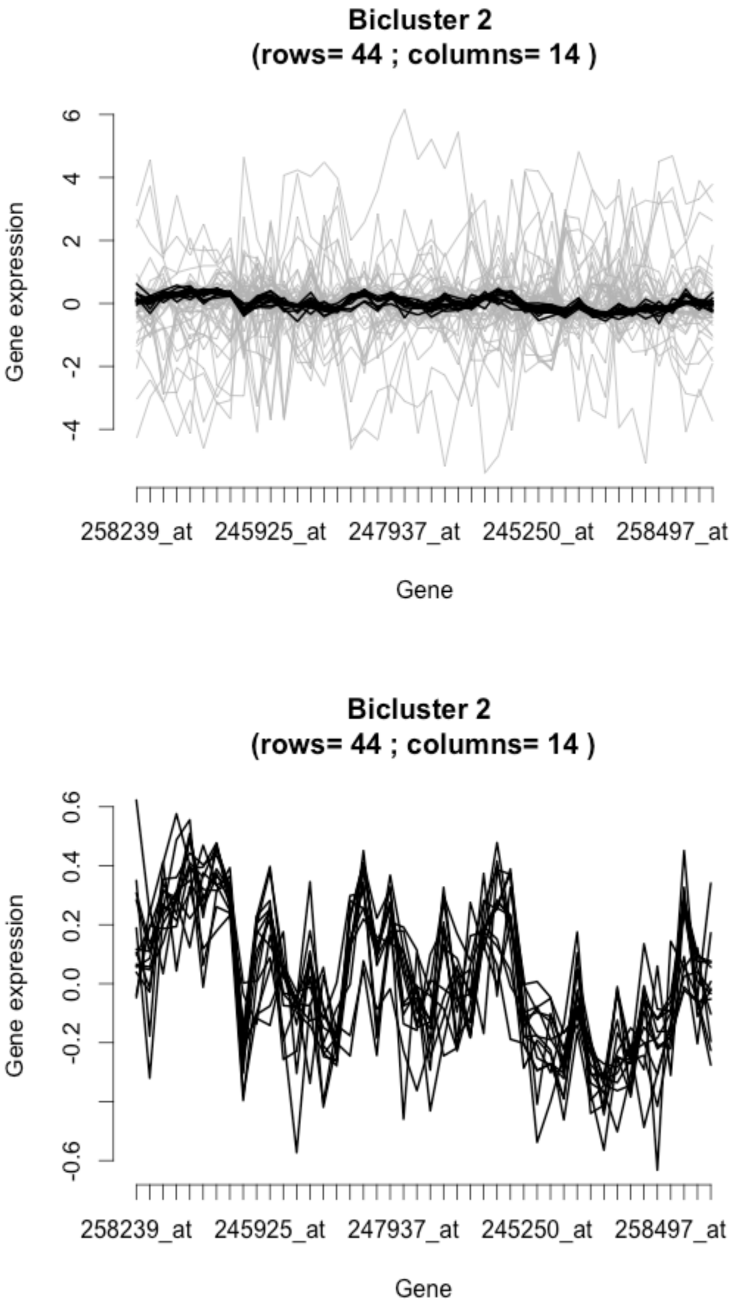


Figure 4.4 Profile plot of the novel generalized CC algorithm (expression levels of conditions across their genes)

Heatmap

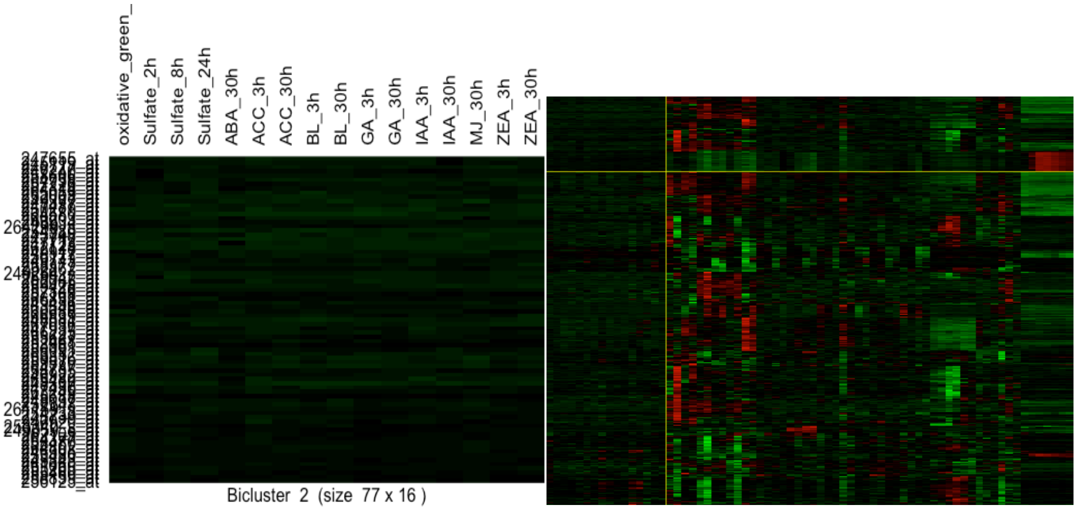


Figure 4.5 Local and global heatmap of original CC algorithm

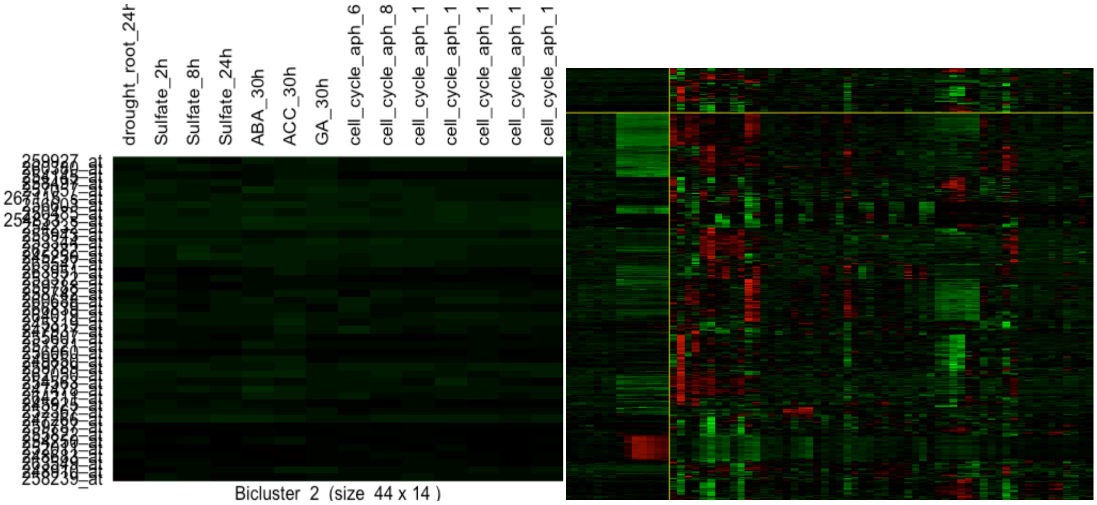


Figure 4.6 Local and global heatmap of generalized CC algorithm

From the heatmaps of the three algorithms, we can see that the original and the generalized CC algorithms have the bicluster with same color block than the Plaid model.

4.7 Comparison of the generalized CC and the original CC algorithm on the simulated data

4.7.1 Data

The data matrix consists of 400 observations, 20 rows and 20 columns randomly generated by binomial distribution with parameters $n = 50$ and $p = 0.4$. Data information is summarized below. We perform both the original CC algorithm and the generalized CC algorithm on the data and the result is shown and summarized in Table 4.7.

Table 4.7 Summary of the simulated data

	Rows	Col	Observation	Distribution
Simulation	20	20	400	Binom (50, 0.4)

4.7.2 Result

Table 4.8 Summary of the biclustering result of the original and generalized CC algorithm

Algorithm	Parameter	Objective Function	Total Number of Bicluster
CC	$\delta = 1.5$	$H(I,J) \leq \delta$	5
Generalized CC	$\delta = 1.5$ $\lambda = 0.3$	$H(I,J)_{general} \leq \delta$	5

Table 4.8 is the summary result of the generalized CC algorithm and the original CC algorithm using the simulated dataset. The threshold is set to be 1.5, and we chose the parameter λ is in the generalized algorithm to be 0.3 using the same method discussed.

Table 4.9 Comparison performances on single bicluster of the generalized CC algorithm to the original CC algorithm on the simulated data set

Algorithm	Total Number of Bicluster	Bicluster #	Standard Deviation	Coherence Variance	Mean Square Residue
CC	5	1	1.961859	2.678275	0.9755556
		2	2.018316	2.766359	1.3024
		3	2.107773	2.802292	1.303819
		4	2.740438	3.364539	1.345
		5	3.00458	2.928289	1.4625
Generalized CC	5	1	1.266557144	1.948807	1.13194444
		2	1.321398502	1.682047	0.53515625
		3	1.452368755	1.881989	0.765625
		4	1.802775638	1.846407	0.40277778
		5	1.384437431	1.343398	0.05555556

Table 4.10 Comparison of the overall quality of the resulting biclustering of CC algorithm and generalized CC algorithm of the K biclusters (K=5) on the simulated data set

Algorithm	Overall Variance $\sum_{k=1}^{K} Var(I,J)_k$ (*Hartigan, 1972*)	Total Squared Residue $\sum_{I,J} H(I,J)$ (*Cho et al., 2004*)	Average Residue $\frac{1}{K}\sum_{k=1}^{K} H(I,J)_k$ (*Yang et al., 2002*)
CC	28.9027	6.389275	1.277855
Generalized CC	10.626303	2.89105903	0.578211806

4.7.3 Profile Plot

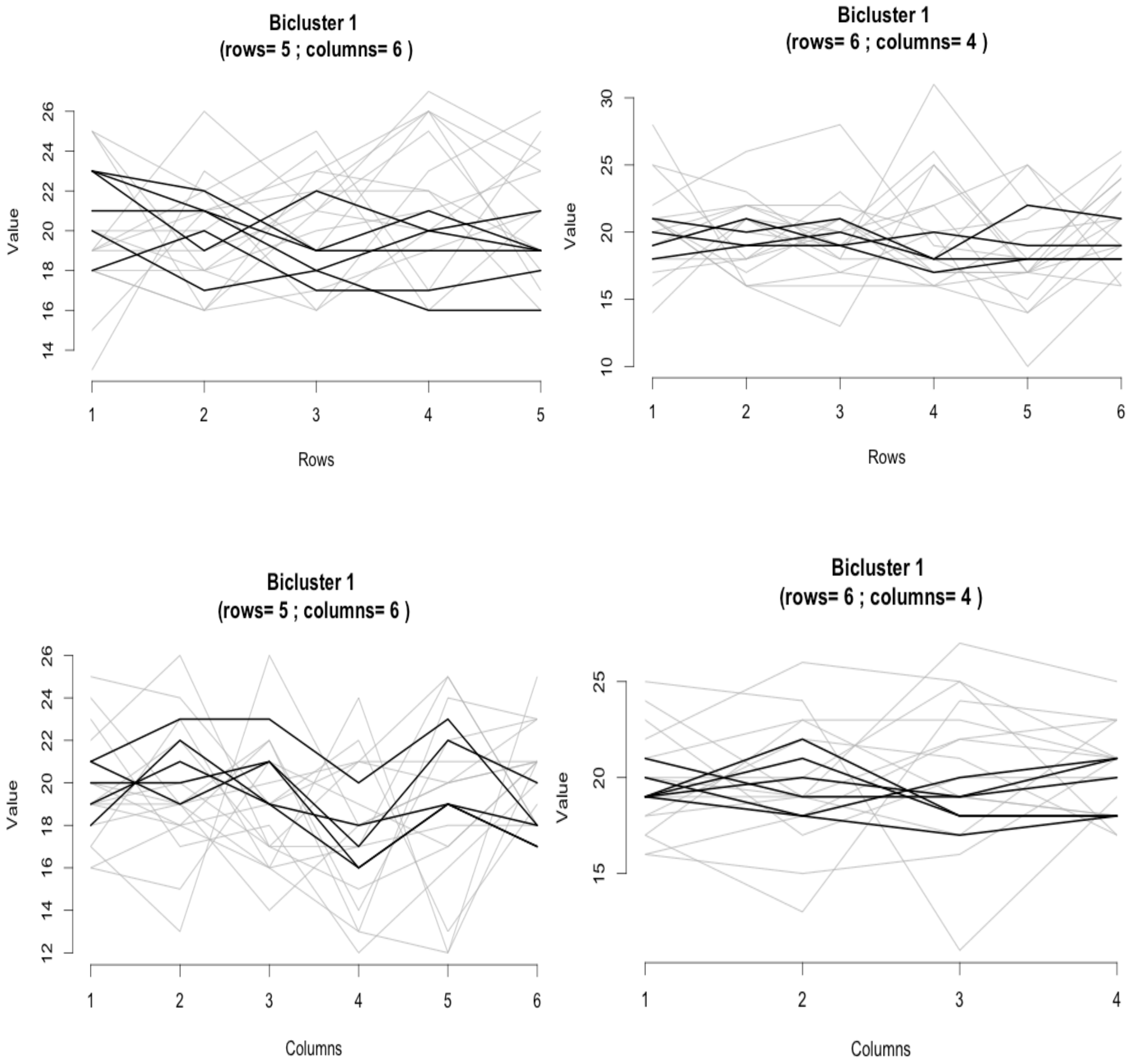


Figure 4.7 Profile plots of biclusters of CC algorithm and generalized CC algorithm across rows and columns on simulated dataset

Table 3.6 summarized the result of performing the original CC algorithm and the generalized CC algorithm using the simulated dataset setting the threshold δ to be 1.5. We use the same method to determine the parameter λ. The two algorithms generate same number of biclusters. The result shows that the biclusters generated by the generalized CC algorithm is more coherent than the biclusters generated by the original CC algorithm which is consistent with the result we got from the microarray dataset.

4.8 Stock Market Pattern Detection using Biclustering Technique

Biclustering technique has been popularly and widely applied to microarray gene expression data to explore gene and expression level combinations and search for interpretable biological patterns. We now apply biclustering technique on financial data trying to find patterns using financial stock data. In biology, we have microarray gene expression data matrix consists of genes as rows and its expression levels (conditions) as columns. For financial data, most commonly, the data matrix consists of stocks as row and time as column. We apply biclustering technique on financial log return data clustering the stock on both time and stock price to detect the patterns of stocks that have same pattern over a subset of time points, which gives us more information about the financial market.

4.8 Pattern Detection of Bear and Bull Stock Market using Biclustering Technique

4.8.1 Data

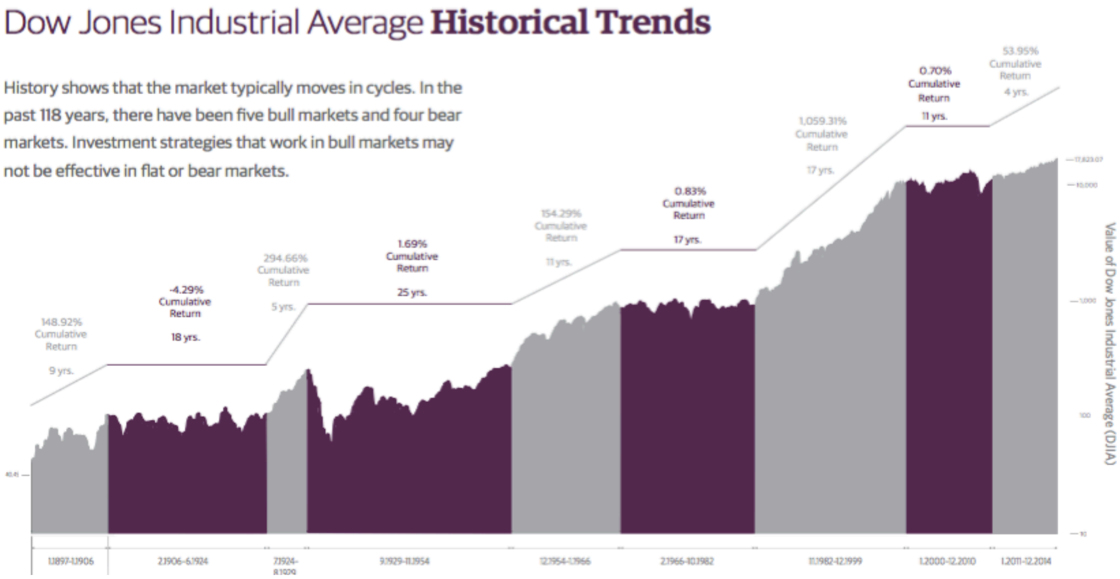
We cluster the all the historical log return of daily close price of Dow Jones Industrial Index Average (DJIA) index available from yahoo finance to analyze the pattern of bear and bull market using biclustering. Dow Jones Industrial Index is comprised of 30 industrial companies' stock market representing about fifth of the total value of the US stock market (Engle and Patton, 2000). We chose Dow Jones Industrial index because it represents large and well-known U.S.

companies, and it also covers all industrials with exception of transportation and utilities. We cluster the data based on day level.

The earliest date available traces back to 3 Jan 2007, and it is documented that we were going through bear market from Jan 2000 to Dec 2010 (132 months/ 11 years) and we were going through bull market from Jan 2011 to Dec 2014 (48 months/ 4 years) (Figure 4.9). So our data ranges from 3 Jan 2007 to 31 Dec 2014 (8 years/ 96 months) as the earliest data we can get from Yahoo finance is from 3 Jan 2007. This period we chose for our analysis include both bear market period and bull market period. Our data contains 30 stocks (Dow Jones Industrial index has 30 components) over 2013 trading days (3 Jan 2007 to 31 Dec 2014), detailed data description is summarized in table 4.12 below. Log return is used in the analysis instead of stock closing price.

Table 4.11 Dow Jones Industrial Average Index (DJI) top components

Company	Ticker	Industry	Weight (%)
Goldman Sachs Group Inc	GS	Financials	6.72
3M Co	MMM	Industrials	6.28
Home Depot Inc	HD	Consumer Services	5.23
Intl Business Machines Corp	IBM	Technology	5.19
McDonald’s Corp	MCD	Consumer Services	5.15
Boeing Co	BA	Industrials	5.00
Unitedhealth Group Co	UNH	Health Care	4.79
Travelers Cos Inc	TRV	Financials	4.45
Johnson & Johnson	JNJ	Health Care	4.34
Apple Inc.	AAPL	Technology	4.05


Dow Jones Industrial Average **Historical Trends**

History shows that the market typically moves in cycles. In the past 118 years, there have been five bull markets and four bear markets. Investment strategies that work in bull markets may not be effective in flat or bear markets.

Figure 4.8 Dow Jones Industrial Average Historical Trends

Source: Graph created by Guggenheim Investments using data from dowjones.com

Table 4.12 Data Summary

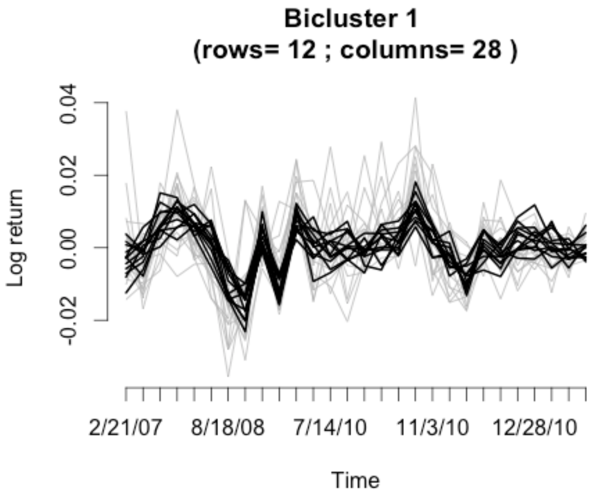
Symbol	Time	Trading Days	Type
DJI (30 stocks)	2007/1/3-2010/12/31	1007	Bear Market
	2011/1/3-2014/12/31	1006	Bull Market

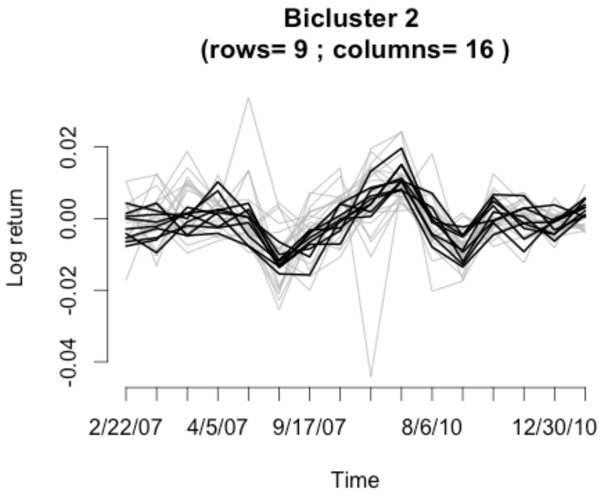
4.8.2 Analysis

Biclustering analysis on stock data enables us to find a subset of stocks that exhibit the same price pattern over a subset of disjoint time points, which gives us more information about the stock market. The analysis consists of two parts. First, we performed biclustering analysis on bear market ranging from 2007-01-03 to 2010-12-31 and bull market ranging from 2011-01-03 to 2014-12-31 separately and compare the patterns of both market from the local biclustering pattern as well as the biclustering distribution plots. Secondly, we performed biclutering analysis on the entire time series ranging from 2007-01-03 to 2014-12-31 and compare the distribution of bear market and bull market. Finally, we perform biclustering analysis on our current market to detect our current market pattern and therefore infer current/future market type. We selected two time periods to analyze current market pattern: 2015-01-05 to 2016-02-23 (Now) and 2015-05-21 (Market most recent peak) to 2016-02-23 (Now). We detect current market pattern and distribution on both periods to infer our current market type.

Clustering Bear Market and Bull Market Separately

Bear market local biclustering patterns from 2007-01-03 to 2010-12-31





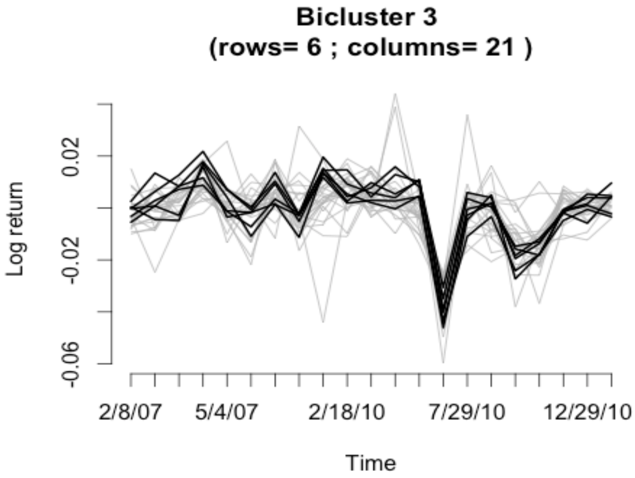


Figure 4.9 Biclustering local pattern of bear market

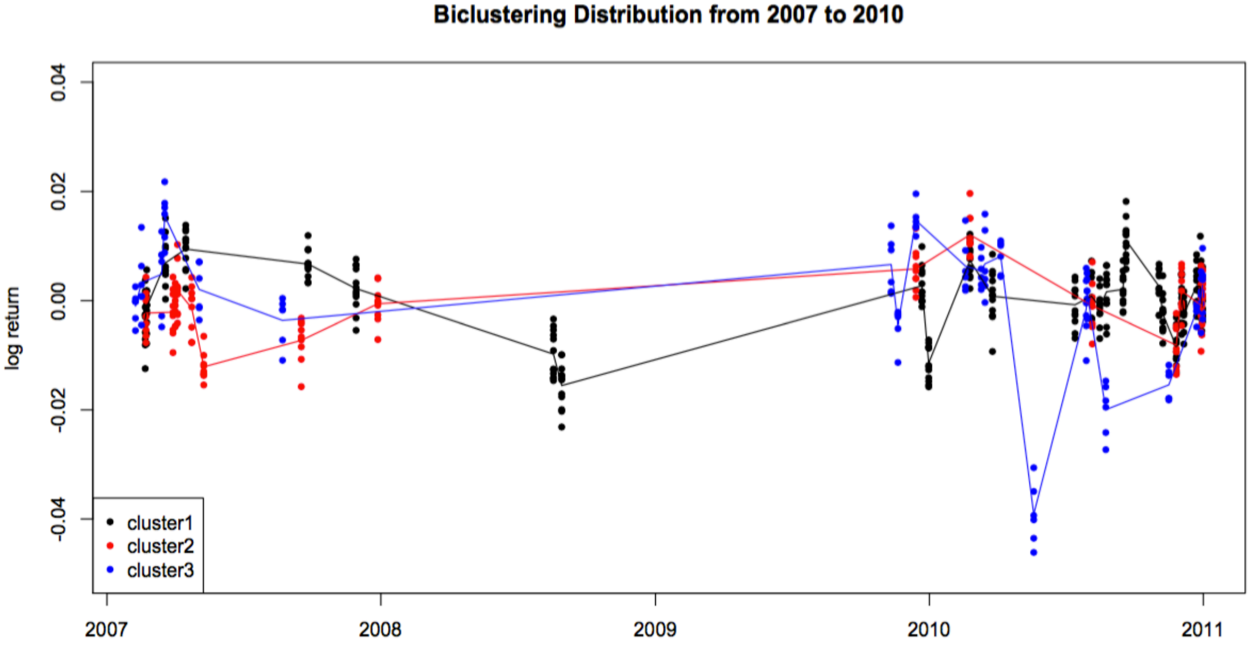


Figure 4.10 Biclustering Distribution of Bear Market from 2007-01-03 to 2010-12-31

Table 4.13 Summary of biclustering result on bear market

Bear Market 2007-01-03 to 2010-12-31			
Bicluster #	**Trading Days**	**Stocks**	**Ticker**
1	28 days	Chevron	CVX
		IBM	IBM
		Johnson & Johnson	JNJ
		Coca-Cola	KO
		McDonald's	MCD
		3M	MMM
		Merck	MRK
		Procter & Gamble	PG
		United Technologies	UTX
		Verizon	VZ
		Wal-mart	WMT
		Exxon Mobil	XOM
2	16 days	Boeing	BA
		Disney	DIS
		General Electric	GE
		Goldman Sachs	GS
		Home Depot	HD
		Intel	INTC
		Microsoft	MSFT
		Pfizer	PFE
		Travelers	TRV
3	21 days	Apple	AAPL
		American Express	AXP

		Caterpillar	CAT
		E I du Pont de Nemours and Co	DD
		JPMorgan Chase	JPM
		UnitedHealth	UNH

Bull Market Local Biclustering Patterns from 2011-01-03 to 2014-12-31

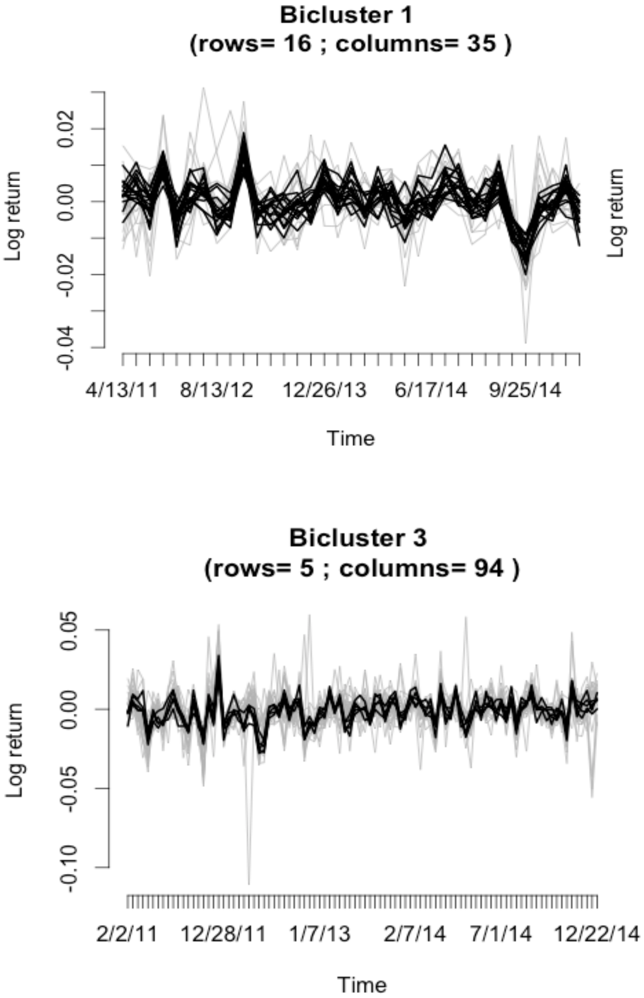


Figure 4.11 Biclustering local pattern of bull market

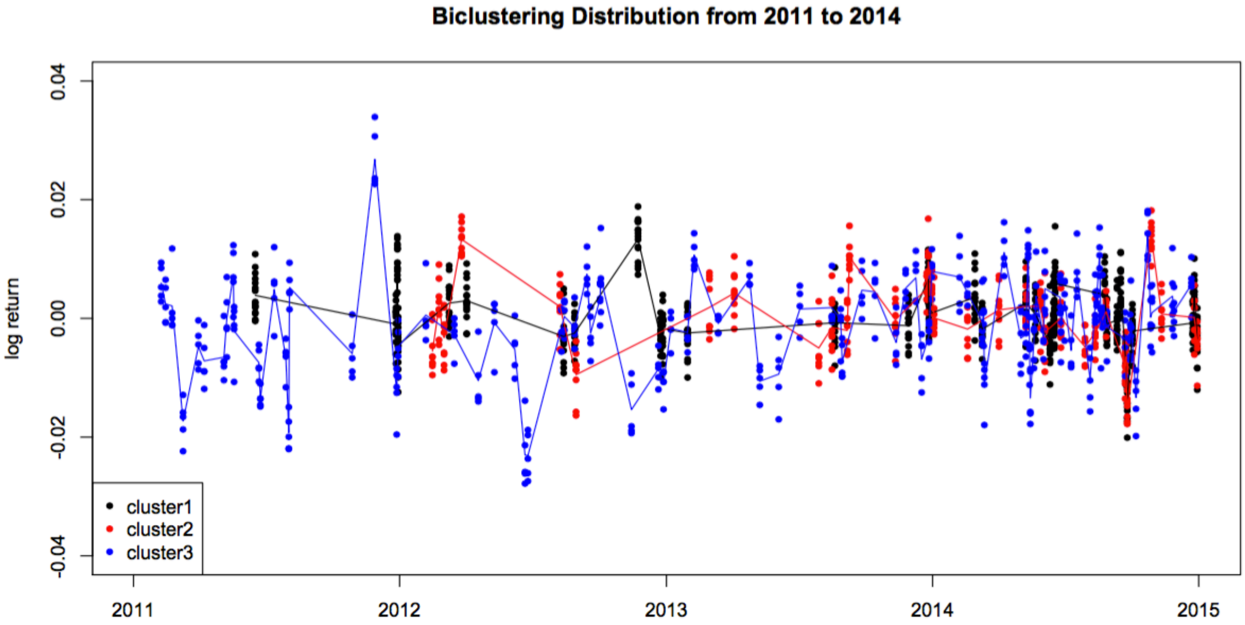


Figure 4.12 Biclustering Distribution of Bull Market from 2011-01-03 to 2014-12-31

Table 4.14 Summary of biclustering result on bull market

Bull Market 2011-01-03 to 2014-12-31			
Bicluster #	**Trading Days**	**Stocks**	**Ticker**
1	35 days	American Express	AXP
		Caterpillar	CAT
		E I du Pont de Nemours and Co	DD
		Disney	DIS
		IBM	IBM
		Johnson & Johnson	JNJ
		Coca-Cola	KO
		McDonald’s	MCD

		3M	MMM
		Merck	MRK
		Pfizer	PFE
		Procter & Gamble	PG
		Travelers	TRV
		United Technology	UTX
		Verizon	VZ
		Wal-mart	WMT
2	34 days	Boeing	BA
		Cisco	CSCO
		Chevron	CVX
		General Electric	GE
		Home Depot	HD
		Intel	INTC
		Nike	NKE
		Visa	V
		Exxon Mobil	XOM
3	94 days	Apple	AAPL
		Goldman Sachs	GS
		JPMorgan Chase	JPM
		Microsoft	MSFT
		UnitedHealth	UNH

Clustering the entire time series from 2007 to 2014

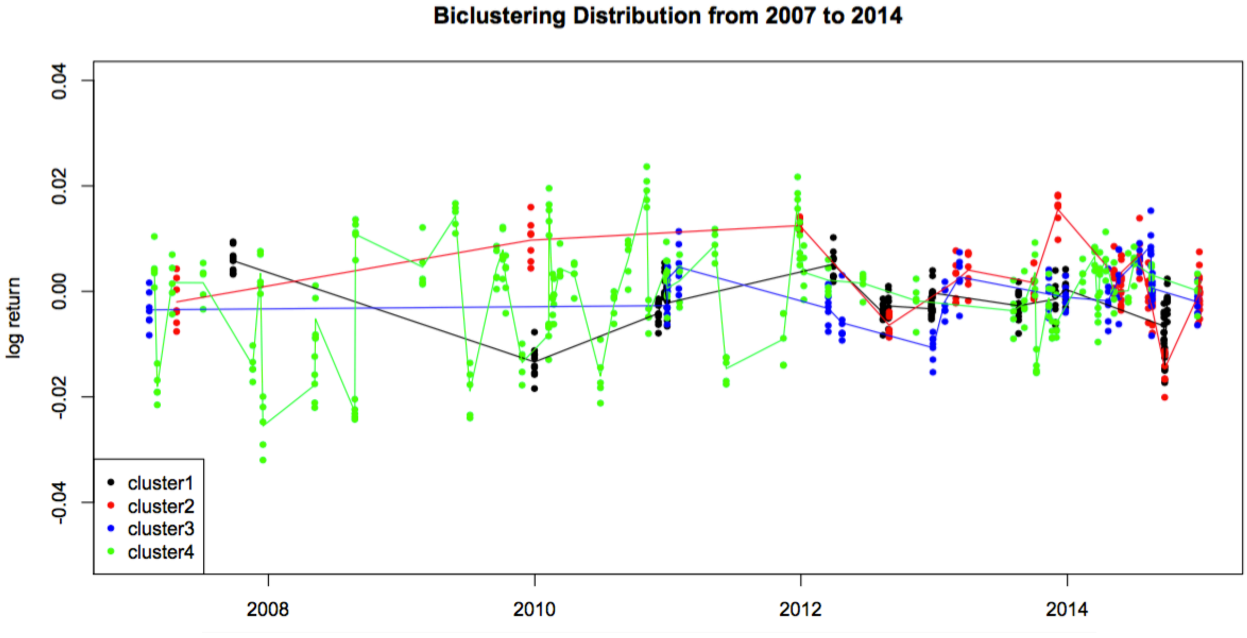


Figure 4.13 Biclustering distribution on the entire time series from 2007-01-03 to 2014-12-31

Current Market Type Analysis

Table 4.15 Summary of current market data

Symbol	Start	End	Trading Days	Type
DJIA	2015-01-05	2016-02-23 (Now)	286	Current
	2015-05-21 (Market most recent peak)	2016-02-23 (Now)	191	Current

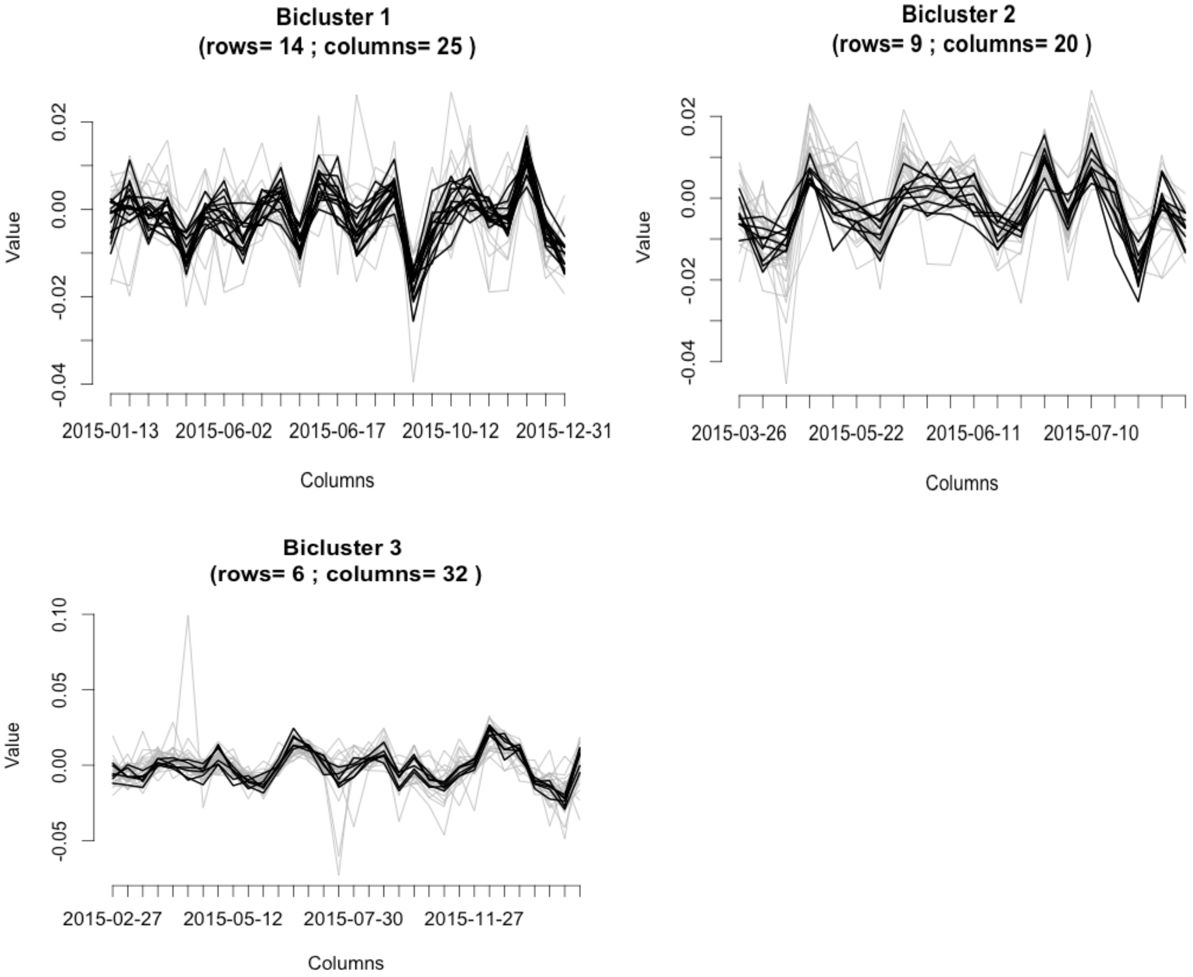


Figure 4.14 Biclustering local patterns of current market from 2015-01-05 to 2016-02-23 (Now)

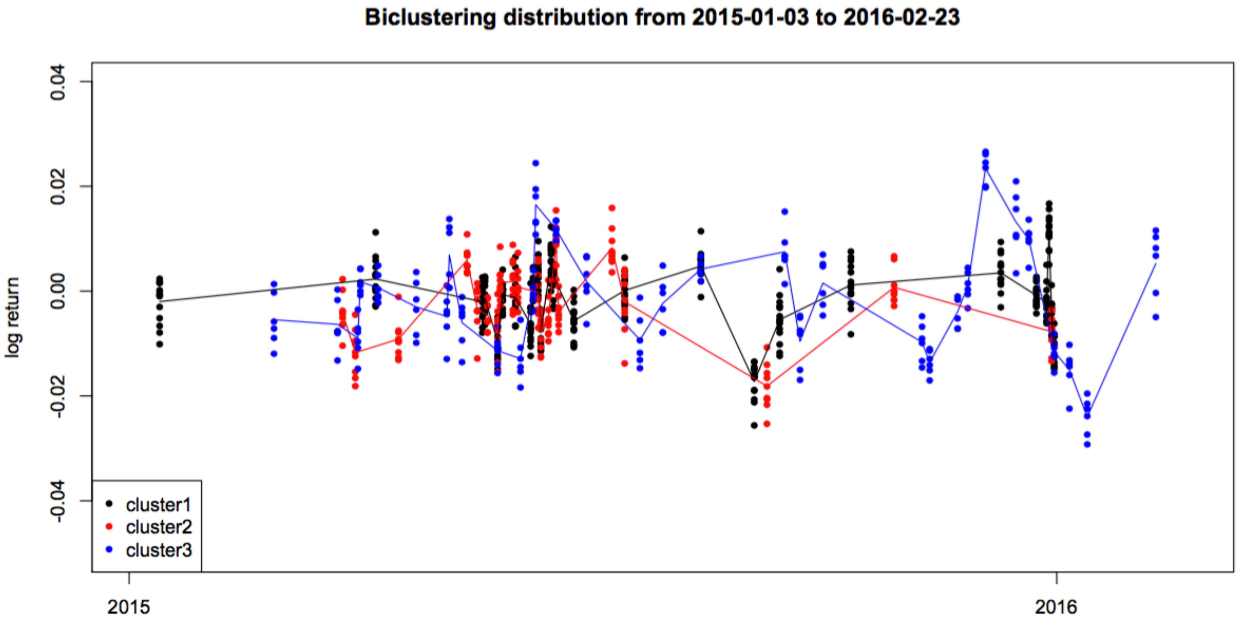


Figure 4.14 Biclustering distribution of current market from 2015-Now

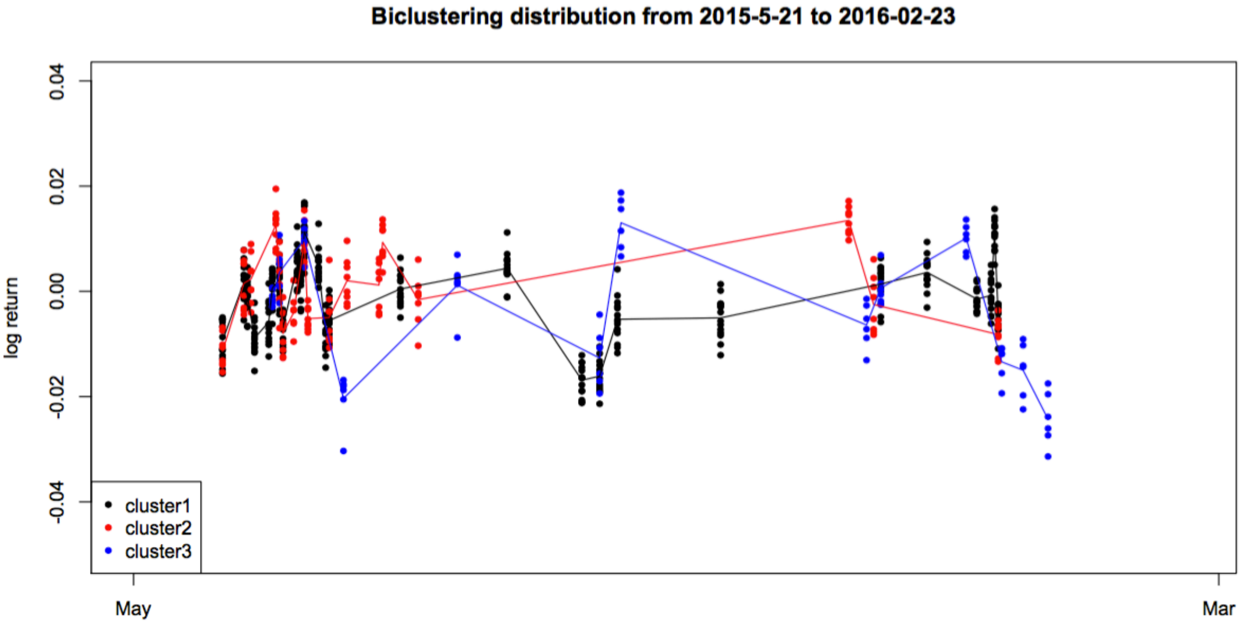


Figure 4.15 Biclustering distribution of current market from 2015-05-21 (market most recent peak) to 2016-02-23 (Now)

Table 4.16 Summary of biclustering results on current market

Bicluster #	Current Market	
	2015-01-05 to 2016-02-23	2015-05-21 to 2016-02-23
1	AXP	GE
	CSCO	HD
	GS	IBM
	HD	JNJ
	IBM	JPM
	JNJ	KO
	JPM	MCD
	KO	MMM
	MMM	MSFT
	MSFT	PFE
	PFE	TRV
	TRV	UTX
	UTX	V
	WMT	WMT
2	CAT	AXP
	CVX	BA
	DD	CSCO
	DIS	CVX
	GE	DIS
	MCD	GS
	PG	PG
	VZ	XOM
	XOM	

3	BA	AAPL
	INTC	DD
	MRK	INTC
	NKE	NKE
	UNH	UNH
	V	VZ

Interpretation

We performed two analysis, ran the biclustering analysis on bear and bull markets separately and ran the biclustering analysis on the entire time series ignoring bear and bull, compare the distribution of bear and bull market. Finally, we infer our current market type by detecting the patterns of current market.

From the results of both biclustering analysis of bear and bull market separately and on the entire time series, the same conclusion can be made. From the local biclustering patterns as well as the distribution plots, the pattern in bear market captures some all time lows with high volatility and very little return. The pattern in bull market captures some all time highs with small volatility, returns are very stable, swaging around 0. We then ran the biclustering analysis on our current market, we chose two time periods, from Jan 05 2015 to Feb 23 2016 and from market's most recent peak May 21 2015 to Feb 23 2016 (Now), we see that market was very stable at the beginning of 2015, and then pattern captures some deep lows around -0.04 in the second half of 2015 till now with increasing volatility, we infer that our current market has shown signs of bear market.

Figure 4.14 and figure 4.15 are the distribution plots of current market on both periods. Both distribution plots show that Cluster 3 in blue seems to be worst impacted, which means that we should avoid those stocks in cluster 3 in the portfolio of investment, e.g., INTL, AAPL, and etc. Table 4.13 and table 4.14 are the summary of biclustering results of the bear market and the bull market respectively, it listed stocks in each bicluster, and we find that in each bicluster, most of the stocks are the top components of Dow Jones Industrial Index listed in table 10. Table 4.16 summarized the biclustering result on our current market.

4.9 Other Application of Biclustering in Finance

The prediction of stock market has always been a challenge for many researchers because the market is highly complicated and dynamic. In the past two decades, we have gone through two big financial crises which have brought us the attention that it is necessary to study the stock market and find the patterns. As Mark Twain often quoted: “History does not repeat itself, but it does often rhythm”. Technical analysis is one kind of method summarizing the market and forecasting the future trend by analyzing the historical stock prices and trading volume of stock utilizing financial technical indicators. Analyzing the stock market using technical analysis with financial indicators is very important to uncover the hidden patterns and help traders to make important trading decisions (i.e., buy, sell or no action). Traders may be interested in a small number of technical indicators which provide the most useful prediction to the market to help make trading decisions Biclustering can be used to find local patterns in the historical data where different patterns contain a subset of most important technical indicators (Xue et al., 2015). Biclustering uncovers the local coherent patterns in stock data through finding the subset of indicators which have similar behaviors in a subset of turning points (trading days) to help the analysis of stock market.

Huang et al. (2015) and Xue et al. (2015) propose the novel use of biclustering method to discover local trading patterns containing a subset of technical indicators from historical financial time series which are the first attempts to use biclustering on financial time series data instead of gene expression data. The patterns found can be transformed into trading rules. Xue et al. (2015) developed Biclustering-based Intelligent System (BIC) and applied it to find patterns and use those patterns in the short term prediction of stock price. Huang et al. (2015) proposed method biclustering algorithm and the K nearest neighbor (BIC-K-NN) in which they classified the trading patterns found by biclustering into three trading actions (sell, buy, and no-action signals) with respect to the maximum support and K nearest neighborhood (K-NN) method is applied to classification of trading day in the testing period. We will give brief introduction how they use biclustering to uncover the local trading patterns containing a subset of technical indicators.

Chapter 5

Finding Historical Periods Resembles Current Stock Market Pattern

This chapter we first identified historical time periods where stock prices exhibit similar behavior resembles current market pattern and thus we infer the potential market trend and whether the market is going for recession/depression. In the second part of the analysis, we zoom into each financial sector to see which sectors will be heavily impacted by the impending recession/depression.

5.1 Data

In the analysis, we analyzed the S&P 500 index over the last 3 months (90 trading days) from 14 October 2015 to 23 February 2016, and we want to identify historical none overlapping 3 month periods where S&P 500 index exhibit similar behavior to our current market pattern. We selected S&P 500 index because it is considered as one of the best representations of the U.S. stock market and U.S. economy. The data information is summarized below in table 5.1.

Table 5.1 Summary of data

Data	Length	Start Date	End Date	Duration/ Trading days
Historical	10 years	17-Feb-06	23-Feb-16	2520
Current	3 month	14-Oct-15	23-Feb-16	90

5.2 Measure of Similarity

There is a wide selection of similarity measurements. We chose both the most commonly used Euclidean distance and a correlation-based distance Pearson correlation distance to measure the similarity of two time series, the current time series (query time series) to any historical time series (reference series).

Euclidean Distance

$$d_{Euc}(x_i, x_j) = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^{2}}$$

the smaller the distance, the more similar the two time series are.

Pearson Correlation Distance

$$d_{corr}\left(x_i, x_j\right) = \frac{1}{2}(1 - corr(x, y))$$

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

the closer to 0 the correlation distance, the more similar the two time series are.

Monotone relationship between Pearson correlation distance and Euclidean distance

Proof:

Assume that x and y are normalized time series

Pearson Correlation:

$$r = \frac{Cov(x, y)}{\delta_x \delta_y} = \frac{E(x - Ex)(Y - Ey)}{\delta_x \delta_y} = E(xy) = \frac{\sum xy}{n}$$

$$\Rightarrow r = \frac{\sum xy}{n}$$

Euclidean Distance:

$$d = \sqrt{\sum (x_i - y_i)^2}$$

$$= \sqrt{\sum x_i^2 + \sum y_i^2 - 2\sum x_i y_i}$$

$$= \sqrt{2n - 2\sum x_i y_i}$$

$$\Rightarrow d^2 = 2n - 2rn$$

$$\Rightarrow 1 - r = \frac{d^2}{2n}$$

$$\Rightarrow \frac{1}{2}\frac{d^2}{2n} = \frac{1}{2}(1 - r)$$

$$\Rightarrow D_{corr} = \frac{1}{4n} D_{euc}^2$$

where on the left of the equation is Pearson correlation distance and on the right of the equation if the Euclidean distance. The two measurements have a monotone relationship with each other.

5.3 Results

Based on the result, we selected the top 7 matches of historical periods measured by both the Pearson correlation distance and the Euclidean distance. Both similarity measurements generate the same results as they are in a monotone relationship with each other.

Table 5.2 Summary of similar historical time periods to current market pattern

Match #	**Start Date**	**End Date**	**Pearson Correlation-based distance**	**Euclidean Distance**	**Type**
1	2-Jun-15	7-Oct-15	0.04223565	3.877614	Current
2	4-Apr-08	11-Aug-08	0.04283016	3.904809	Bear Market
3	13-May-11	20-Sep-11	0.07475694	5.158824	Bull Market
4	3-Mar-10	9-Jul-10	0.075816995	5.195272	Bear Market
5	21-Nov-08	2-Apr-09	0.08489055	5.497366	Bear Market
6	14-Feb-12	21-Jun-12	0.092341355	5.733544	Bull Market
7	7-May-07	12-Sep-07	0.096376205	5.857468	Bear Market
Current	**14-Oct-15**	**23-Feb-16**			

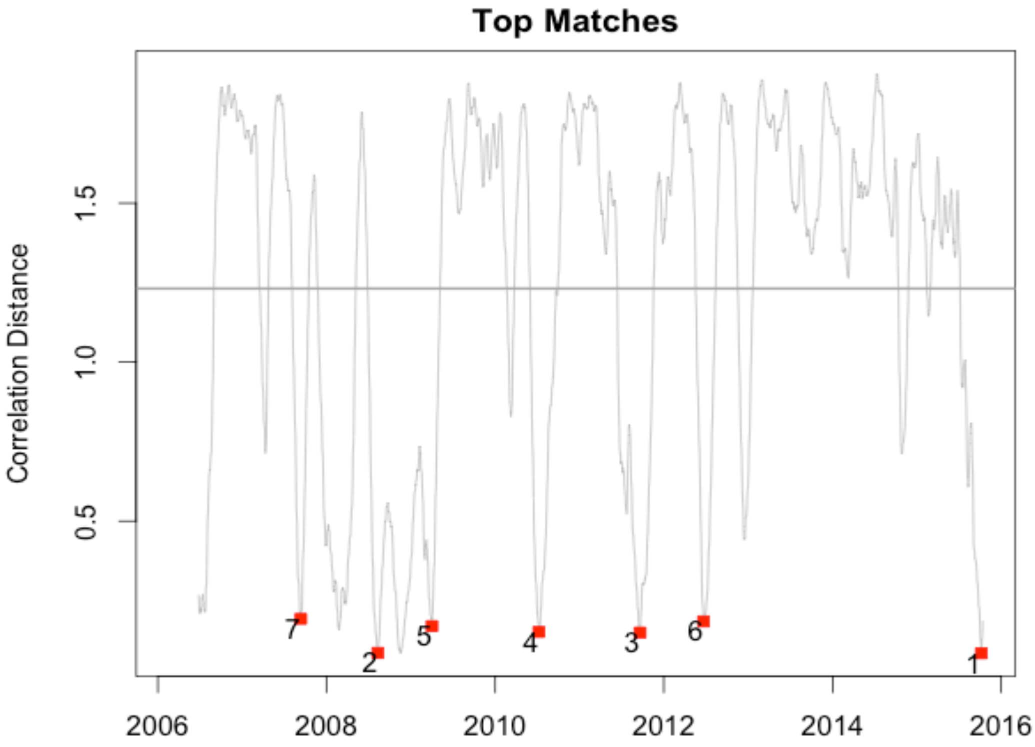


Figure 5.1 Pearson correlation distance top match chart

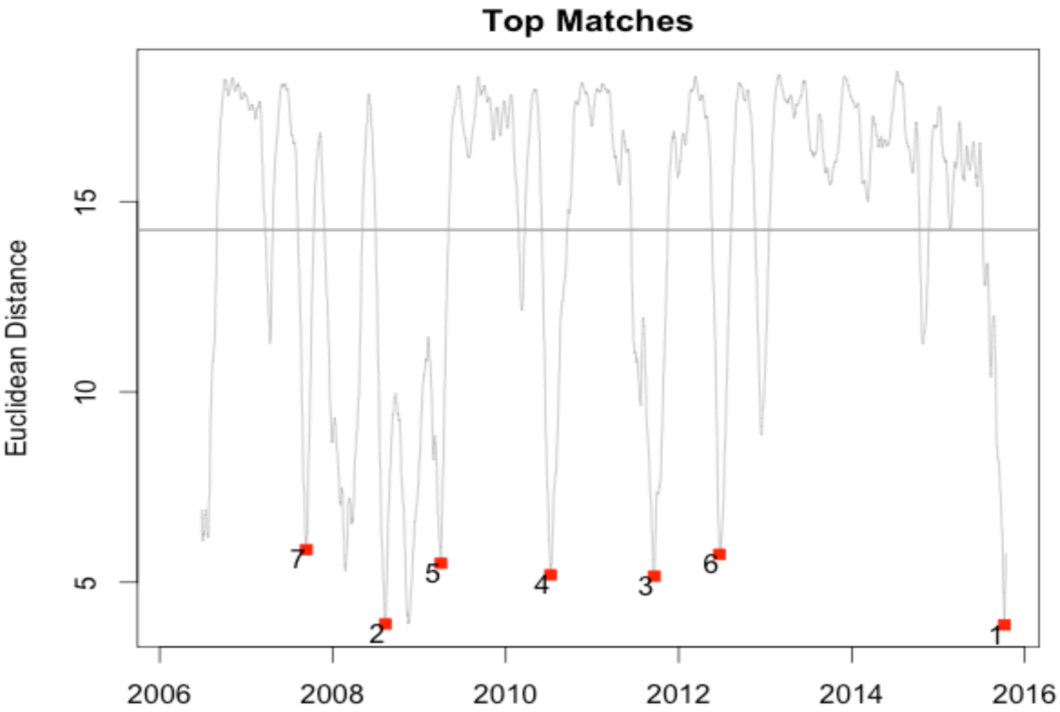


Figure 5.2 Euclidean distance top match chart

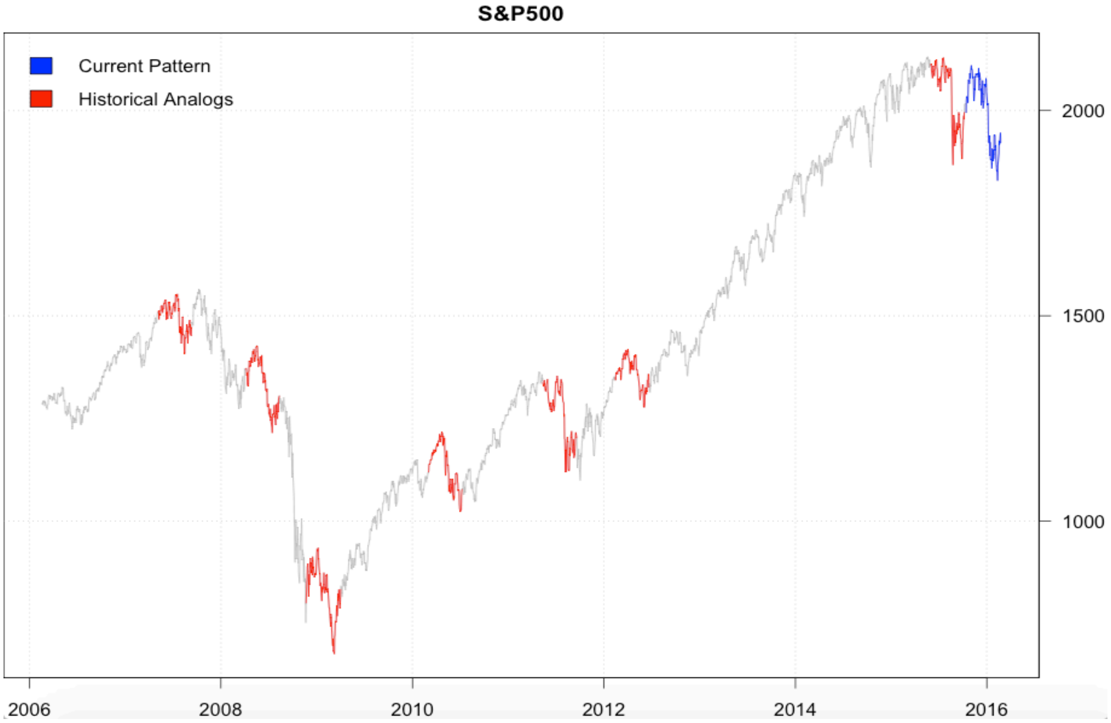


Figure 5.3 Historical time periods similar to current market pattern

Note: Blue – current market pattern

Red – identified historical periods

Next we analyze S&P 500 index over the past 30 years from February 27 1986 to current (February 23 2016). Same similarity measurements are used as summarized in table 5.5. We selected the top 15 most matched historical periods as summarized in table 5.4.

Table 5.3 Summary of data

Data	Length	Start Date	End Date	Duration (trading days)
Historical	30 years	27-Feb-1986	23-Feb-16	7560
Current	3 month	14-Oct-15	23-Feb-16	90

Results

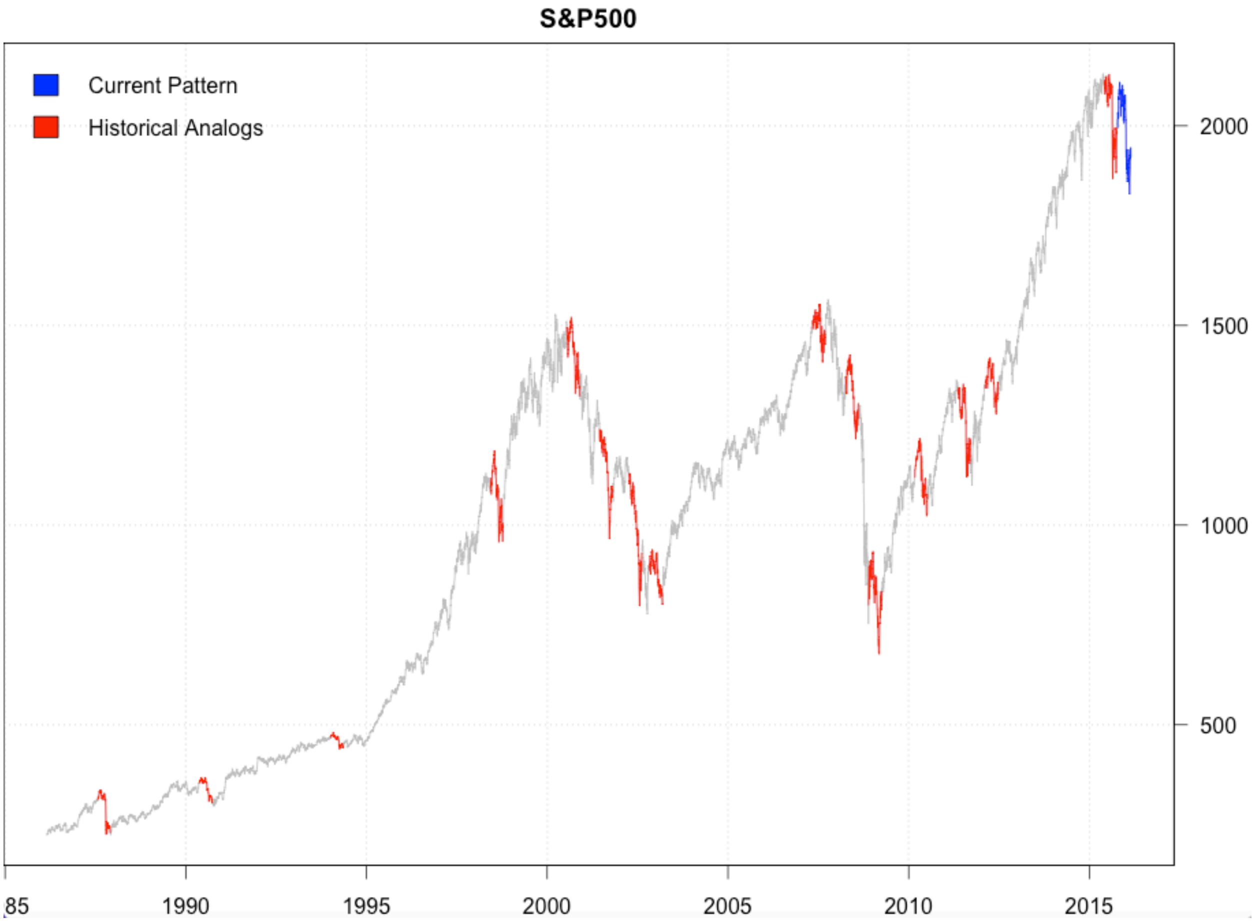


Figure 5.4 15 top match historical time periods similar to current market pattern (Past 30 years)

Figure 5.4 is the 15 top-matched historical time periods similar to current market pattern over the past 30 years during which we went through two big financial crises. The time window covers recent three financial crashes in 1987, 2000 and 2008, so it should be enough to identify our current and potential market trend.

Table 5.4 Summary of top 15 top match similar historical periods

Match #	Start Date	End Date	Type
1	27-Jul-87	1-Dec-87	Bear Market
2	2-Jun-15	7-Oct-15	Current
3	4-Apr-08	11-Aug-08	Bear Market
4	18-May-90	25-Sep-90	Bull Market
5	13-Jun-01	24-Oct-01	Bear Market
6	6-Jan-94	16-May-94	Bull Market
7	10-Apr-02	15-Aug-02	Bear Market
8	9-Jun-98	14-Oct-98	Bull Market
9	31-Oct-02	12-Mar-03	Bear Market
10	13-May-11	20-Sep-11	Bull Market
11	3-Mar-10	9-Jul-10	Bear Market
12	21-Nov-08	2-Apr-09	Bear Market
13	14-Feb-12	21-Jun-12	Bull Market
14	7-May-07	12-Sep-07	Bear Market
15	19-Jul-00	22-Nov-00	Bear Market
Current	**14-Oct-15**	**23-Feb-16**	

Table 5.5 Similarity measurements of 15 top match historical periods to current

Match #	Euclidean Distance	Pearson Correlation Distance
1	3.468788	0.033799135
2	3.877614	0.04223565
3	3.904809	0.04283016
4	4.322032	0.0524718
5	4.409774	0.054623885
6	4.4628	0.055945455
7	4.838296	0.06575593
8	4.903195	0.06753179
9	5.068073	0.072149915
10	5.158824	0.07475694
11	5.195272	0.075816995
12	5.497366	0.08489055
13	5.733544	0.092341355
14	5.857468	0.096376205
15	5.898328	0.097725495

5.4 Historical Analogs VS current market pattern

From the 15 top matched historical periods we found above, the following 8 historical periods plotted in red in the figure 5.5 below worth paying attention to because those are when we were having bear markets and also those are when the 2000 and 2008 financial crushes happened. The blue plot is our current latest 3-month stock price pattern.

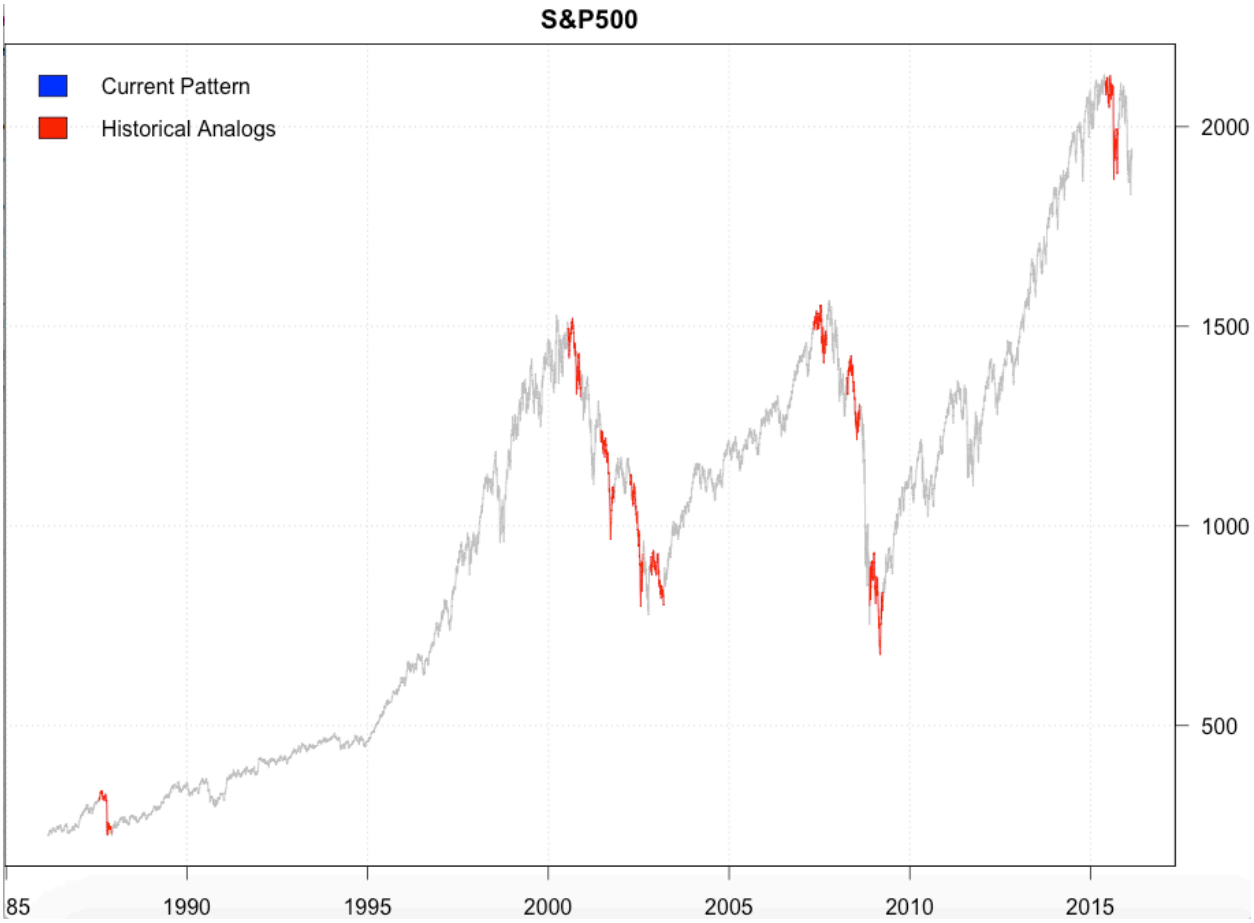


Figure 5.5 S&P500 Historical analogs VS current pattern over past 30 years from Feb 27 1986 to Feb 23 2016

Table 5.6 Historical Analog 1987

March #	Start	End	Type	Peak
1	27-Jul-87	01-Dec-87	Bear Market	Aug 25 87

Table 5.7 Historical Analog 2000-2003

March #	Start	End	Type	Peak
5	13-Jun-01	24-Oct-01	Bear Market	Mar 24 00

7	10-Apr-02	15-Aug-02	Bear Market	
9	31-Oct-02	12-Mar-03	Bear Market	
15	19-Jul-00	22-Nov-00	Bear Market	

Table 5.8 Historical Analog 2007-2009

March #	Start	End	Type	Peak
3	04-Apr-08	11-Aug-08	Bear Market	Oct 09 07
12	21-Nov-08	02-Apr-09	Bear Market	
14	07-May-07	12-Sep-07	Bear Market	

Table 5.9 Current Market Pattern

March #	Start	End	Peak
2	02-Jun-15	07-Oct-15	May 21 15
Current	14-Oct-15	23-Feb-16	

From figure 5.5, table 5.6, 5.7, 5.8, and 5.9 the results show that history does repeat itself. We experienced the two big financial crushes, the 2000 recession from Mar 11 2000 to October 9 2002, and the 2008 global financial crisis starting from 2007 to 2009. Figure 5.5 shows that our current pattern very much similar to the past two historical analogs. The two historical analogs share the characteristics of the market reached its peak and then dropping drastically from its peak to bottom, and there are several drop downs, each dropping down is lower than the previous one. Figure 5.5 also shows that our current market is reaching its latest peak on May 21 2015, and it already followed by two small drops happening in the second half of 2015 and early 2016 in table 5.10, while the most recent drop off (last 3 month) is lower than the previous one in

2015. We conclude that our current market has slight shown some similarity of the past historical analogs if the market continues with the dropping offs without any signs of going for its new high.

5.5 Financial Sectors likely to be heavily impacted by the impending recession

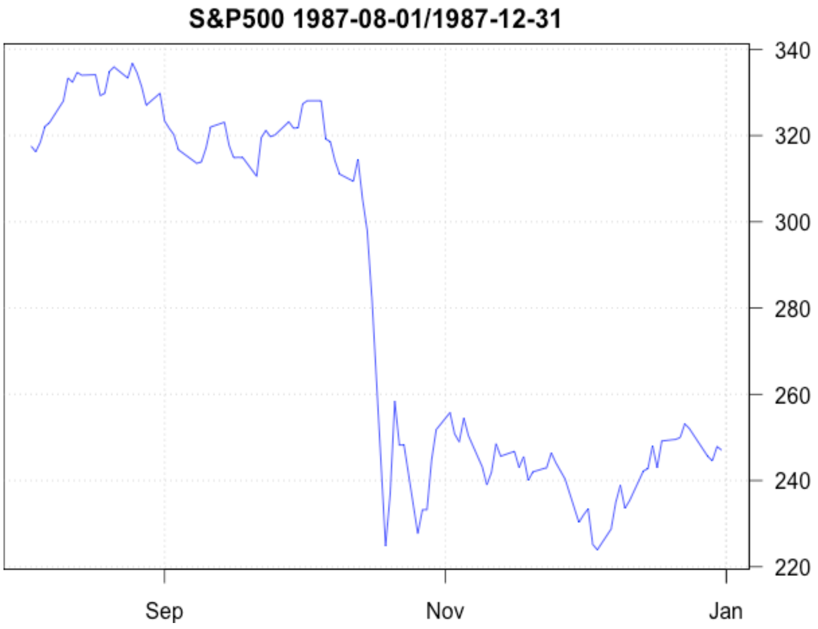
In terms of sectors, we analyze all the sectors of S&P500 index during the recent two large recessions/sell-offs as well as the current market sell-off (latest peak to now). We then represent the results and infer which sectors are most likely to be heavily impacted by the impending recession/depression.

Background of recent two recessions/depressions

Table 5.10 Summary of recent three financial crises

Depression /Recession	Occurrence	Peak Date	Trough Date	Peak to Trough / Duration (Trading days)	Type
Black Monday 1987	Oct 19 87	25 Aug 87	Dec 04 87	-34% / 72	Bear Market
Great Recession	Mar 11 00- Oct 9 02	24 Mar 00	Oct 09 02	-49% / 638	Bear Market
Global financial crisis	2007-2009	09 Oct 07	09 Mar 09	-57% / 356	Bear Market

From table 5.10 above, we see that the 2008 global financial crisis is the worst; the market declined 57% over 356 trading days.



S&P 500 declined 21% on October 19 1987 to 224.84, the largest one-day percentage drop in history.

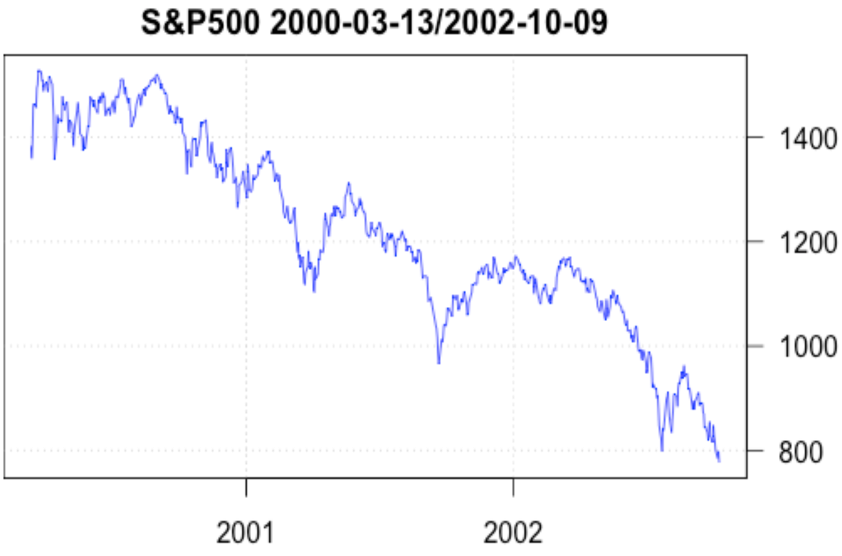


Figure 5.6 S&P 500 time plot during 2007 great recession

The S&P500 declined 49% (Peak to Trough) from its peak 1527 in March 2000 to its bottom 777 in October 2002.

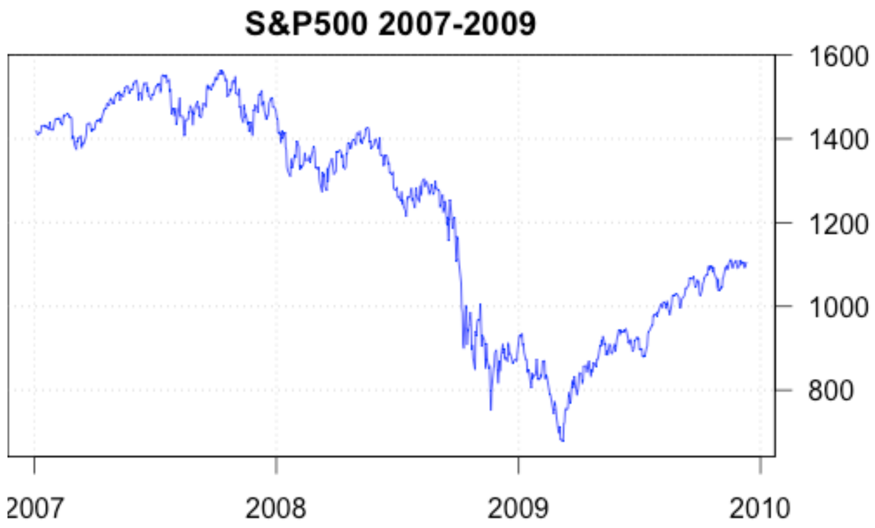


Figure 5.7 S&P 500 time plot during 2008 global financial crisis

The S&P500 declined 57% (Peak to Trough) from its peak 1576 in October 2007 to its bottom 676 in March 2009.

Financial Sectors likely be heavily impacted by impending recession

We analyze SPY index (SPDR S&P 500) whose primary benchmark is S&P500 index and it seeks to provide investment results that correspond to the price and yield performance of the S&P500 index, and tracks the S&P 500 index. We looked at each sector and infer which sectors are likely to be heavily impacted by the impending recession/depression. Table below is the top sectors of SPY as of December 31 2015, and we will look at all the sectors in the analysis below.

Table 5.11 Top sectors of SPY (as of 12/31/2015)

Top Sectors	Ticker Symbol	Weight (%)
Technology	XLK	20.68
Financials	XLF	16.48
Health Care	XLV	15.14
Consumer Discretionary	XLY	12.89
Industrials	XLI	10.06
Consumer Staples	XLP	10.05
Energy	XLE	6.48
Utilities	XLU	2.97
Materials	XLB	2.76
Telecommunication Services	XTL	2.43

Table 5.12

Past two recessions/Sell-Offs

Index	Duration (Trading days)	Peak-To-Trough
S&P 500	638	-49%
S&P 500	256	-57%

Table 5.13

Current Sell-Off

	Duration (Trading days)	Peak-To-Trough	Latest Peak	Now
S&P 500	191	-14%	21-May-2015	23-Feb-2016

Table 5.14

S&P500 ***Sector stack up during 2000 Great Recession***

Energy	Mat.	Ind.	C. Disc	C. Staples	HlthCare	Fins	Tech	Telco	Utilities
-43%	-32%	-45%	-32%	-34%	-30%	-38%	-82%	*	-57%

Note: (*) no data available online

Table 5.15

S&P500 ***Sector stack up during 2008 Global Financial Crisis***

Energy	Mat.	Ind.	C. Disc	C. Staples	HlthCare	Fins	Tech	Telco	Utilities
-58%	-61%	-63%	-58%	-34%	-41%	-83%	-54%	-51%	-49%

Table 5.16

S&P500 *Recent two large Sell-Offs* ***Median***

Energy	Mat.	Ind.	C. Disc	C. Staples	HlthCare	Fins	Tech	Telco	Utilities
-51%	-47%	-54%	-45%	-34%	-36%	-61%	-68%	51%	-53%

Table 5.17

S&P500 *Current Sell-Off Peak-to-Now*

Energy	Mat.	Ind.	C. Disc	C. Staples	HlthCare	Fins	Tech	Telco	Utilities
-36%	-27%	-16%	-16%	-11%	-18%	-23%	-15%	23%	12%

Table 5.17 shows that Energy and Materials sector have already began to show some decline towards the sell-offs as highlighted in green, that may be caused by a continued decline in commodity prices and slowing global growth while Industrials sector has not started to show and obvious decline. Table 5.16 is the performance of each sector on average (median of the recent two recessions/sell-offs). Looking at table 5.16 and table 5.17, we see that Consumer Discretionary, Financials, Technology and Utilities sector haven't fallen as much as they had historically in the recent two big recessions which may potentially impacted by the impending recession/depression.

Chapter 6

Intraday Pattern of High Frequency Data

In this chapter, we will be discussing patterns of financial time series data including the financial stock data and high frequency trading data. Financial time series data is characterized by many facts, among which volatility clustering is one of the most important characteristics that interested many researcher and many volatility models are developed to capture this property of financial time series data. Volatilities do cluster, that is small change in price tends to cluster together and big changes in price tend to cluster together. We use data on Dow Jones Industrial Average (DJIA) to illustrate the volatility clustering property. GARCH type models and its variants are able to capture volatility clustering property. Second part, we will discuss the intraday pattern of high frequency data. Over the last few decades, high frequency trading (HFT) has gained its great usage and popularity in the financial markets. Joe Ratterman, CEO of BATS exchange, commented that "Nearly all equity trading in the US today is automated in some fashion and can exhibit characteristics that fall under the umbrella label of high frequency trading." Given the popular rise of the high frequency trading and motivated by the characteristics of financial time series, we will discuss the intraday pattern of high frequency data. High frequency stock trading data traded on New York Stock Exchange (NYSE) is used to analyze the intraday pattern.

6.1 Characteristics of Financial Time Series

Many characteristics about financial time series have been discussed in a lot of studies. In this section, we will summarize some of the common facts of financial time series volatility behavior based on different documented studies. Empirically documented facts of financial time series are the following:

Fat/heavy Tails

The probability distributions of time series returns often exhibit fat tail feature, which is also known as excess kurtosis or leptokurtic. If returns are fat tailed, the probability of having extreme events, such as very high or very low returns is higher than if it’s normally distributed. Heavy tails implies that extreme values are more frequently than expect if the time series is normally distributed. Also, the distribution of the time series usually has narrower and higher peaks. Most volatility models such as GARCH model take into account this characteristic of the time series being fat tailed. This is true whether the underlying shocks are Gaussian or are themselves not Gaussian but fat tailed such as t distribution and so on.

Volatility Clustering

Financial time series usually exhibit a characteristic known as volatility clustering, also called persistence. Specifically, large changes tend to follow large changes and small changes tend to follow by small changes as noted by Mandelbrot (1963) This behavior of financial time series has been constantly reported in other studies too such as Bailie et al (1996), Chou (1988) and Schwert (1989) (Engle and Patton, 2000). The volatility is more likely to be high at time t if it was also high at time t-1. In other words, a shock at time t-1 increases not only the variance at time t-1 but also the variance at time t. Volatilities cluster in time, in which periods of low volatility are followed by periods of low volatility, and periods of high volatility are followed by periods of high volatility. We will find evidence of volatility clustering by plotting Dow Jones

Industrial Index (Figure 6.2). The GARCH type of models capture this effect very well, they also precisely specify how volatility at time t depends on past volatility.

Mean reversion

Volatility clustering implies that volatility sometimes is large and sometimes small. Thus, a period of high volatility will eventually return to a more normal volatility level and similarly, a period of low volatility will be followed by a high volatility period (Engle and Patton, 2000). Mean reverting implies that financial time series have their own normal level of volatility, and volatility tends to return to that level in the long run. For example, volatility not only spikes during financial crisis, but it eventually drops back to approximately as before the crisis. Statistical speaking, volatility is often stationary over time.

Asymmetry

Asymmetric implies that volatility of financial time series tend to react differently on big price increase or big price drop. Generally, bad news generate greater volatility than good news, and this phenomenon is also called leverage effect, or sometimes risk premium effect. The former story implies that the increase in risk was believed to come from the increased leverage induced by a negative shock. The latter story implies that news of increasing volatility reduces the demand for a stock because of risk aversion. The consequent decline in stock value is followed by the increased volatility as forecast by the news (Engle and Patton, 2000). Black (1976), Christie (1982), Nelson (1991), Glosten et al (1993) and Engle and Ng (1993) all find the evidence of volatility being negatively related to equity returns (Engle and Patton, 2000).

Among those properties of financial time series, *volatility clustering* is the most popular topics interested many researchers. There exist volatility clusters, that is to say volatility may be high for certain time periods and low for other periods (Tsay, 2002). To illustrate the characteristics of volatility clustering of financial time series, we plot the daily close price and log return of Dow Jones Industrial Index from 23 August 2000 to 21 August 2015 with total number of 3772 observations. The Dow Jones Industrial Index is comprised of 30 industrial companies' stock market representing about fifth of the total value of the US stock market (Engle and Patton, 2000). Figure 1 is the daily close price of the Dow Jones Industrial Index in the last 15 years and Figure 6.2 is the log return of the Dow Jones Industrial Index over the same period (in the last 15 years). Figure 6.2 shows the evidence of volatility clustering, that is, we see that large changes tend to be followed by large changes and small changes tend to be followed by small changes. Volatilities are clustered together.

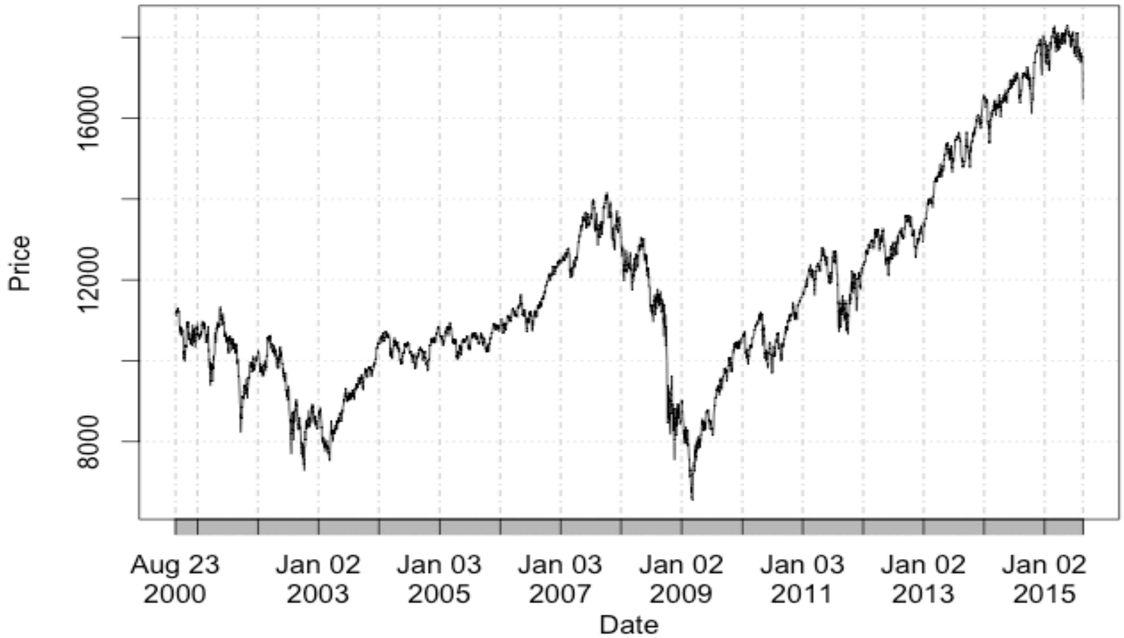


Figure 6.1 The Dow Jones Industrial Index from 23 August 2000 to 21 August 2015.

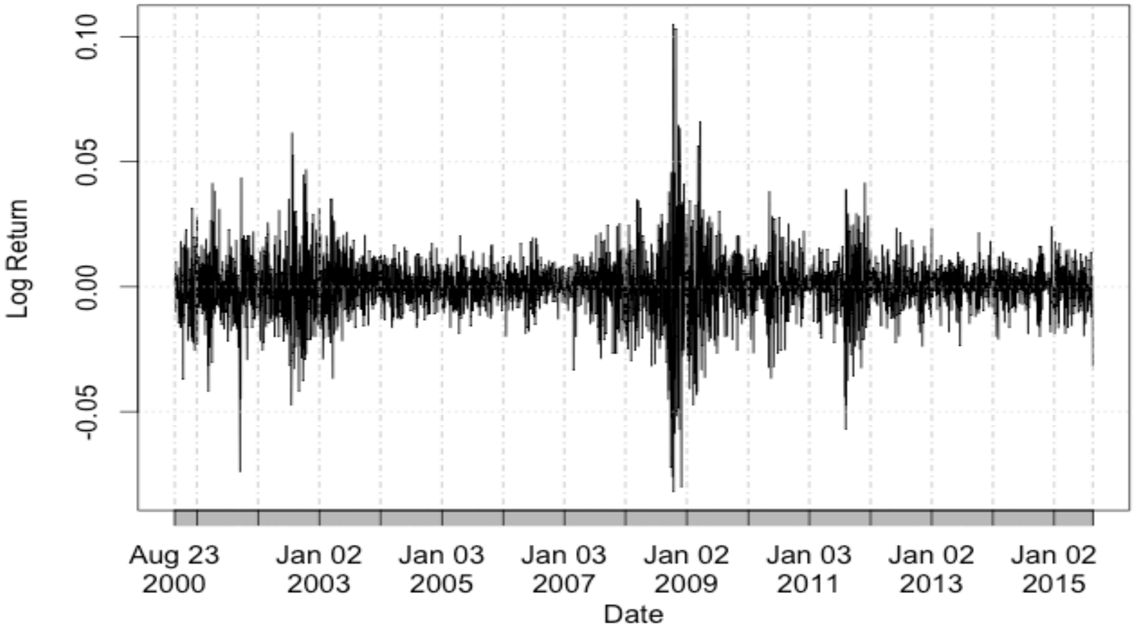


Figure 6.2 Log Returns on the Dow Jones Industrial Index from 23 August 2000 to 21 August 2015 showing volatility property

Those characteristics of volatility play a very important role in volatility model development. Some volatility models were proposed to correct the weakness of the existing models because the existing ones may not be able to capture certain volatility characteristics. GARCH model captures both the fat tail phenomenon and volatility clustering commonly associated with financial time series, it primarily is to model volatility clustering phenomenon. Many proposed volatility models assumed that the conditional volatility is affected symmetrically by positive and negative shocks. GARCH(1,1) (Engle, 1982; Bollerslev, 1986) is commonly used in vast literature due to its simplicity. The key feature is its mean reversion imposed by the restriction of $\alpha + \beta < 1$ and its symmetry, that is the magnitude of past returns, not the sign of the past returns influences future volatility (Brownlees et al, 2011). However, the sign of the innovation may also influence the volatility except for the magnitude (Engle and Patton, 2000). Normal GARCH model is not able to capture this. However, the GARCH model can be easily augmented to capture this feature. Models account for asymmetry are Exponential GARCH (EGARCH) of Nelson (1991), the Threshold GARCH (TGARCH) (Glosten et al, 1993; Zakoian, 1994), also known as GJT-GARCH followed the work of Glosten, Jagannathan, Runkle (1993) motivated by EGARCH. Other models accounting for asymmetry are the nonlinear GARCH (NGARCH) proposed by Engle (1990) and asymmetric power ARCH (APARCH) by Ding et al (1993). Many studies were carried out to compare the performance of those models. The simplest asymmetric GARCH model, the threshold GARCH model of Glosten et al (1993) is claimed to be often the best forecaster (Brownlees et al, 2011).

6.2 Intraday Pattern of High Frequency Data

6.2.1 Motivation

Financial time series often exhibit volatility clustering property, that is large changes in price tend to cluster together, and small changes in price tend together. It was first documented by Mandelbert (1963): " large changes tend to be followed by larges, of either sign, and small changes tend to be followed by small changes." Motivated by the volatility clustering characteristic of financial time series, we wonder what is the intraday pattern of high frequency data. This chapter includes two parts, first part we will introduce the characteristics of financial time series, its asymmetry, mean reversion, volatility clustering and fat tail. Second part, we will introduce intraday pattern of high frequency data. Trading data is illustrated to detect notably intraday pattern. Standard time series models of volatility – GARCH family models have been proven inadequate to model high frequency data where the intraday return dynamics is neglected (Andersen and Bollerslev, 1997). Thus Engle's (2012) Multiplicative Component GARCH model is introduced when applying to high frequency data. Finally, we conclude the chapter by applying Citigroup one-minute return data to the Multiplicative Component GARCH model, where we observe the same intraday volatility pattern.

6.2.2 Intraday/ Diurnal U-shape Pattern

It is widely documented that return volatility varies systematically over the trading day, and this pattern is highly correlated with the intraday variation of trading volume and bid-ask price (Anderson and Bollerslev, 1997). The empirical evidence on the average intraday stock returns dates back to Wood et al. (1985) and Harris (1986), they documented the existence of a distinct U-shaped pattern in return volatility over the trading day (Anderson and Bollerslev, 1997). Volatility is high at the open and close of trading day and low in the middle of the day, that's why the intraday trading activity is documented as U-shape. There are diurnal U-shaped patterns in intraday trading activity. Trading volume tends to be very high soon after the market opening (9:30 am in local time), volatility calms down during lunch hours and high again before the market closes (4pm in local time). These qualitative features are present at the aggregation level of one second and one minute (Ito, 2013). We will see later in the example that U-shape intraday patterns also exist at the aggregation level of five minute. Ito (2013) explains that extreme movements in trading activity in the first hour of trading day may be caused by news transmitted over night. Trading activity slows down towards in the middle of the day (lunch time) when the overnight information is processed, it picks up again in the afternoon as traders rebalance their positions before market closes.

6.2.3 Analysis of High Frequency Trading Data

We perform the analysis using LOBSTER (Limit Order Book System – The Efficient Reconstructor) data. LOBSTER is an online limit order book data tool designed with the goal of providing researchers with limit order book data. LOBSTER provides “message” and “orderbook” file for each active trading day of a selected ticker. Orderbook file: The 'orderbook' file contains the evolution of the limit order book up to the requested number of levels. Message file: The ’message’ file contains indicators for the type of event causing an update of the limit order book in the requested price range. All events are timestamped to seconds.

We analyze the trading volume (in the number of shares) and number of executions of Amazon (AMZN) and Apple (AAPL) stock traded on New York Stock Exchange (NYSE) on 22 June 2012. Trade volume is a measure of intensity of trading activity. They are a variety of volume measures used in the literature including the number of shares traded, dollar volume, number of transactions, turnovers (shares traded divided by shares outstanding), and dollar turnover (Ito, 2013). We choose two measures to analyze the trading activity, trading volume as measured by the number of shares traded and number of executions.

As mentioned, the raw data set is in tick-format and consists of the record of every trade in the sequence of occurrence. The tick-data is irregularly spaced and have multiple transactions in one second. The tick-data is aggregated over equally spaced time-intervals. There are total number of 269,747 observations and 6 variables (Time, Type, Order ID, Size, Price, Direction) in Message file. There are 269,747 observations and 40 variables (Ask Price 1, Ask Size 1, Bid Price 1, Bid Price 2,…, in 10 levels) in Order Book file. Here, we aggregate the raw data in tick format by different intervals, 1 minute time interval and 5 minute time interval and describe the notably intraday patters.

The left panel of Figure 6.3 shows the number of executions of Amazon (AMZN) at the aggregation interval of one minute during the market open hours of New York Stock Exchange (NYSE) (9:30 am to 4pm in local time) on 22 June 2012. The right panel of Figure 6.3 shows the trading volume of AMAZON at the aggregation interval of one minute during the market open hours on the same date. At the aggregation interval of five minute, there are 78 observations per trading day. Likewise, Figure 6.4 shows the number of executions and trading volume of at the aggregation interval of one minute. At the aggregation interval of one minute, there are 390 observations per trading day. Figure 6.3 and Figure 6.4 are the number of executions and trading volume of Apple (AAPL) at the aggregation interval of five minute and one minute respectively.

Figure 6.3-6.6 shows executions and trading volume of Amazon and Apple. We see extreme movements in the beginning of the day and end of the day, and the trading activity slows down during lunch hour, which exhibit a U-shape of trading activity. We see notably intraday pattern from the trading data we have, which coincide with what has been documented the U-shape in the intraday pattern, and this builds up the result. Figure 6.7-6.8 shows an intraday evolution of depth plot at three levels. We clearly observe more volatility soon after the market begins (9:30am, official NYSE trading hour) and calms down during lunch hour and becomes more volatile before closure (4pm, official NYSE trading hour), showing the same intraday pattern.

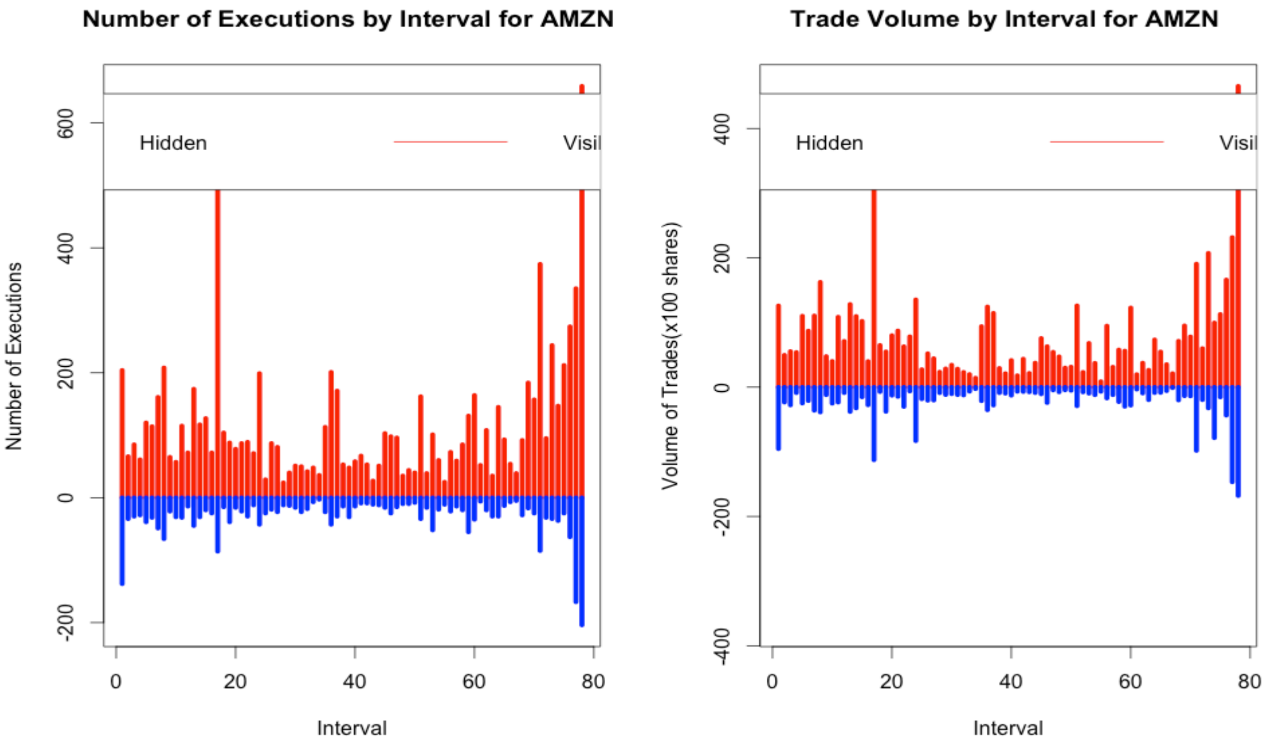


Figure 6.3 Intraday Number of Executions and Trade Volume (5-min interval) for Amazon

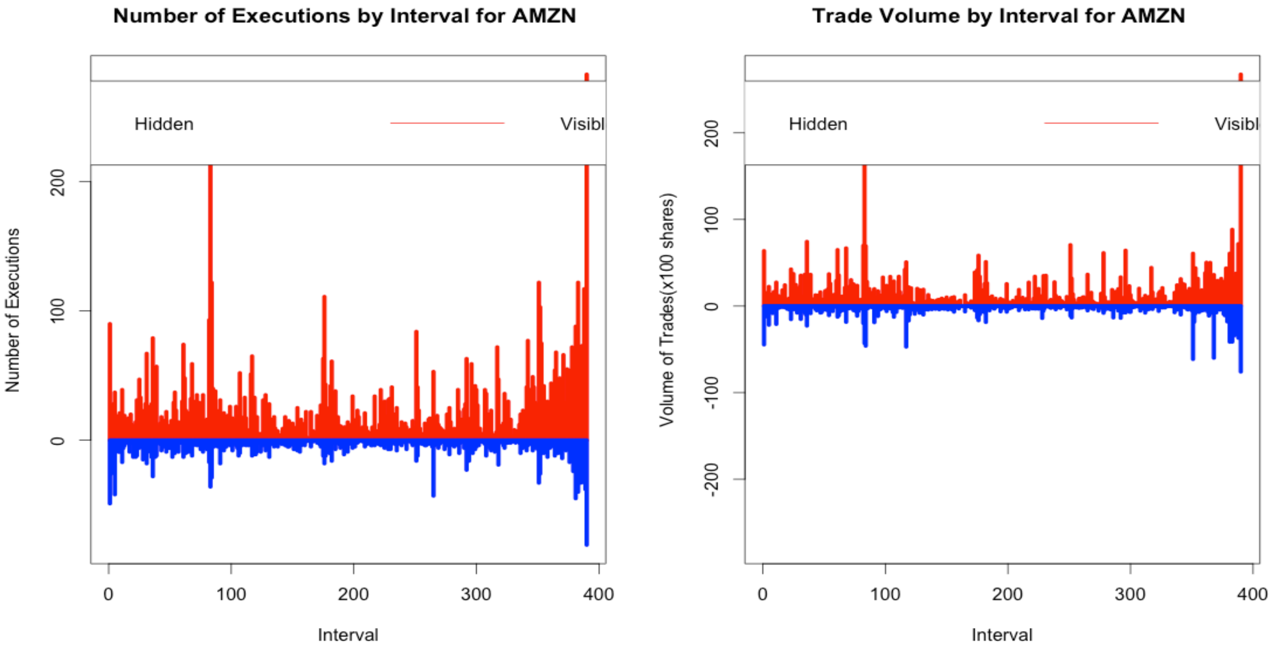


Figure 6.4 Intraday Number of Executions and Trade Volume (1-min interval) for Amazon

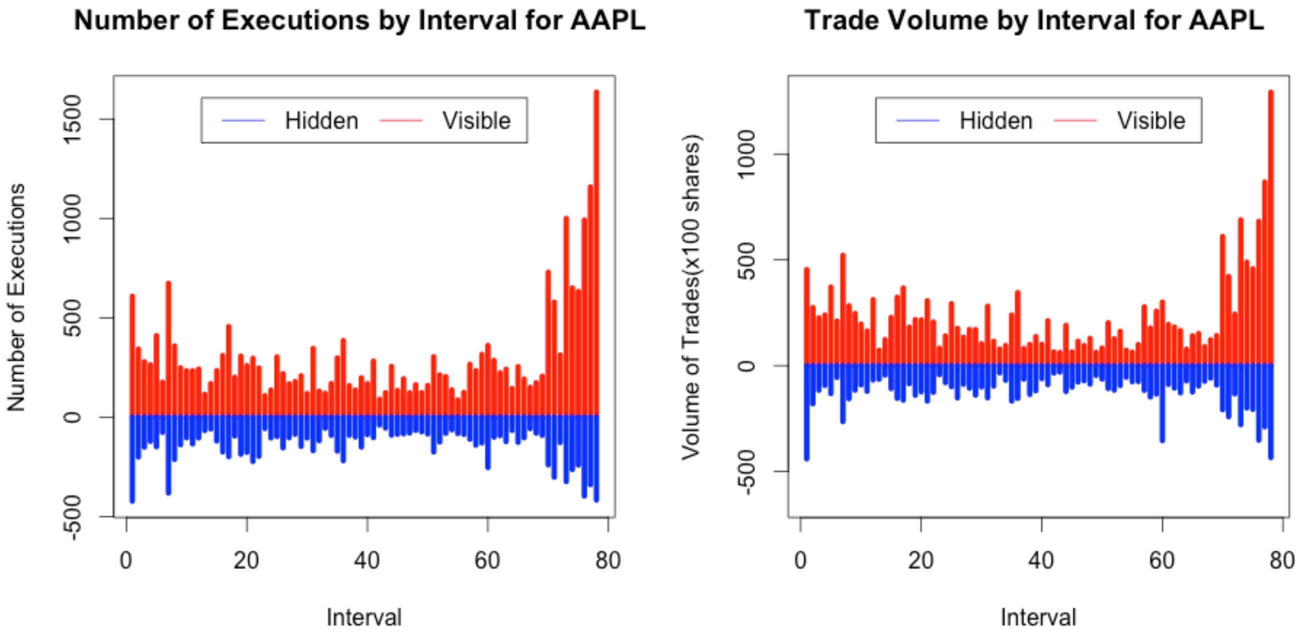


Figure 6.5 Intraday Number of Executions and Trade Volume (5-min interval) for Apple

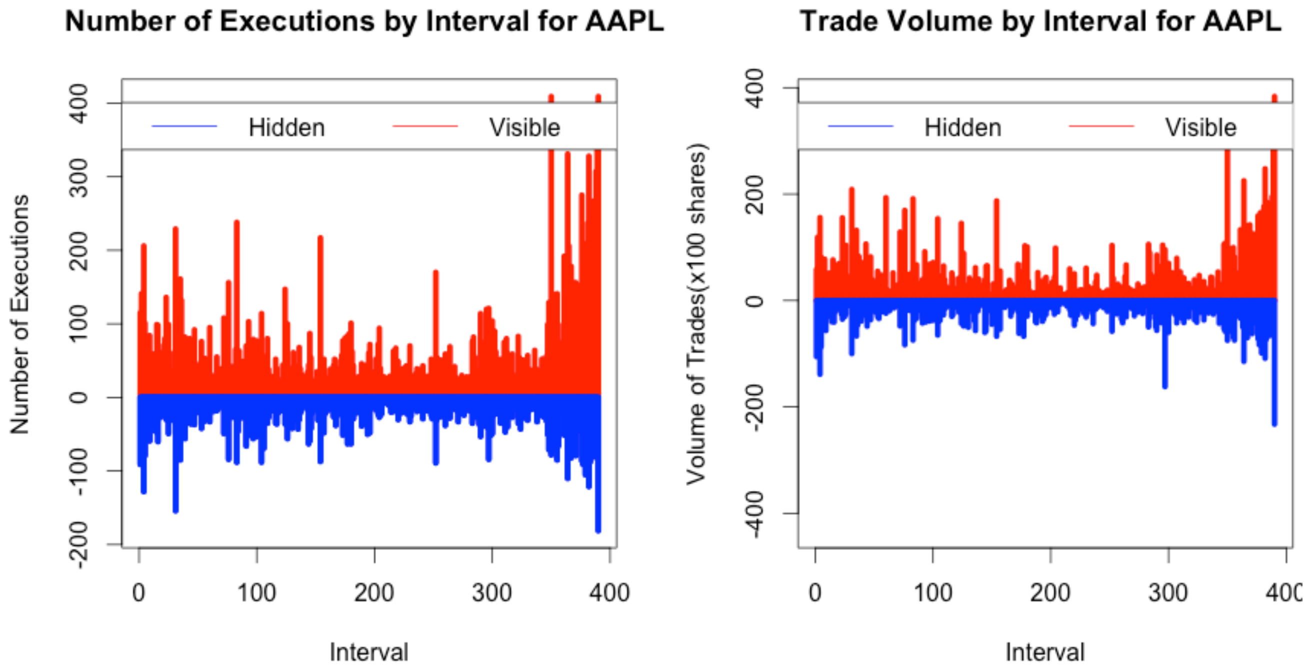


Figure 6.6 Intraday Number of Executions and Trade Volume (1-min interval) for Apple

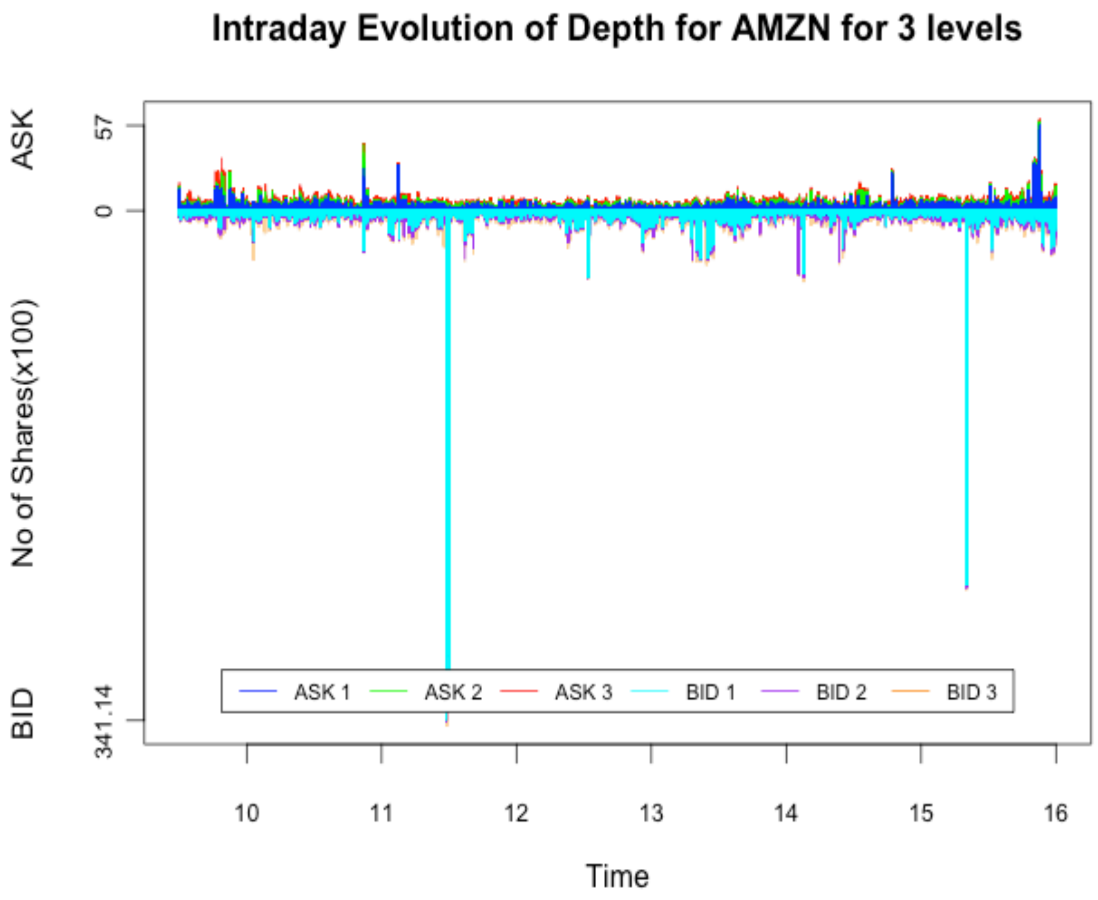


Figure 6.7 Intraday Evolution of Depth for Amazon (3 levels) from 9:30am-4pm, official NYSE trading hours.

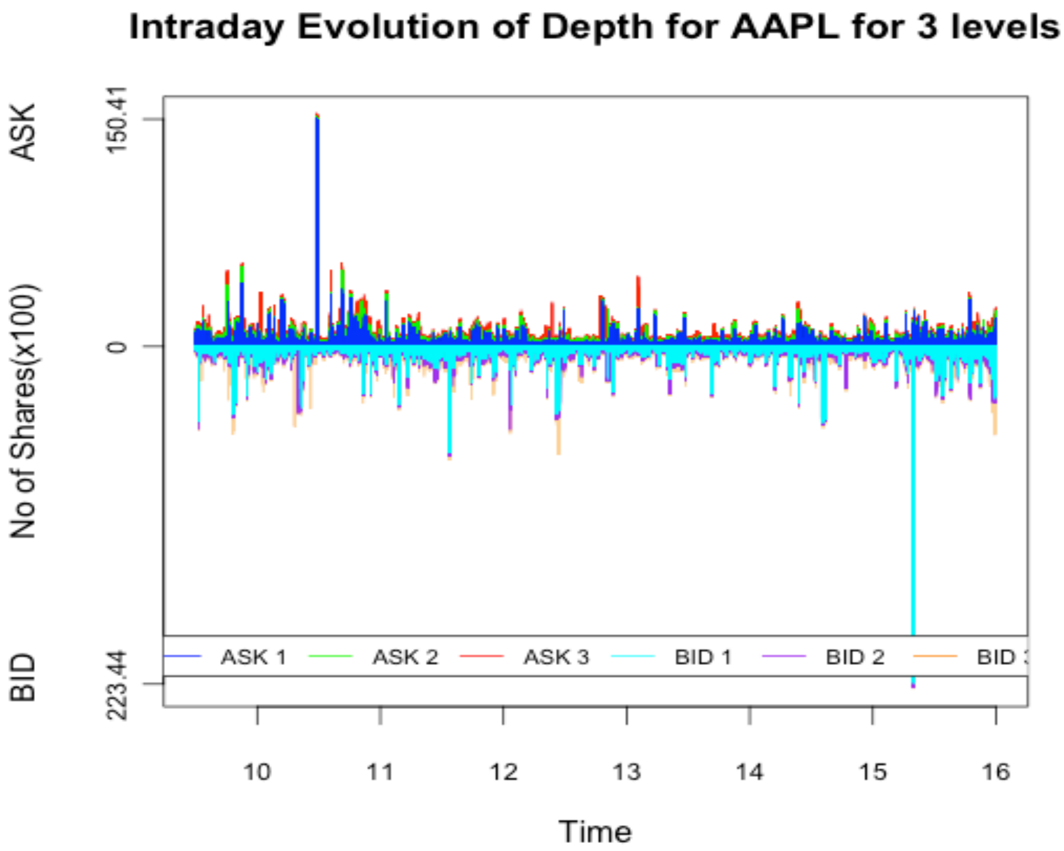


Figure 6.8 Intraday Evolution of Depth for Apple (3 levels) from 9:30am-4pm, official NYSE trading hours.

Figure 6.7 and Figure 6.8 shows the Evolution of Depth plots for Amazon and Apple stock from 9:30am – 4pm on 22 June 2012. The plots plot the level up to three (Ask price 1, Ask price 2, Ask price 3, Bid price 1, Bid price 2, Bid price 3) giving that the orderbook file contains up to 10 levels. From the Evolution of Depth plots for both stock, we can see significant big movements in trading activity in the beginning and end of the day, and quiet movements in the middle of day, which consistent with the pattern detected before.

6.3 Multiplicative Component GARCH Model

The key problem with using GARCH models for intraday data is the intraday pattern which can be considered as the seasonality effect (Ghalanos, 2015). The U-shaped intraday pattern is observed at the aggregation level of one minute and five minute – big movements in the beginning and end of the day and the movements slow down in the middle of the day. Ghalanos (2015) reported that for regular sampled intervals (1-min, 5-min), a number of models have tried to either "de-seasonalize" the residuals and then fit the GARCH model, for example, Flexible Fourier method in Andersen and Bollerslev (1997) or incorporating seasonality into the GARCH model in Bollerslev and Ghysels (1996).

5.3.1 Background

Many papers have reported intraday returns related work including Andersen and Bollerslev (1997, 1998), Giot (2005), Dacorogna et al (2001), Muller, Dacorogna, and Pictet (1996), Engle and Gallo (2006) (Singh et al, 2013). Engle and Sokalska (2012) which is the most recent work developing Multiplicative Component GARCH model based on Andersen and Bollerslev (1997, 1998). Among those, Andersen and Bollerslev (1997, 1998) is one of the most commonly cited studies, where they propose models for 5-minute returns on Deutschemark-dollar exchange rate and the S&P500 index. Andersen and Bollerslev (1997) build a multiplicative model for daily and diurnal volatility. They add the additional component taking account of macroeconomic announcements (Engle and Sokalska, 2012). Engle and Sokalska (2012) argue that the intra-daily volatility component in Andersen and Bollerslev's models are all deterministic. They then propose the multiplicative component GARCH model based on

Andersen and Bollerslev's model incorporating both the deterministic volatility component and stochastic volatility component. They apply the model on a comprehensive sample consisting of 10-minute returns on more than 2500 US equities. Conventional GARCH approaches were argued to be inadequate (Engle and Sokalska, 2012). The intraday return dynamics is neglected in standard time series models of volatility which have been proven inadequate when applied to high frequency data (Andersen and Bollerslev, 1997).

5.3.2 The Model

Engle and Sokalska (2012) propose a new way of modeling and forecasting intraday returns where the components in the model include both deterministic and stochastic. They compose the volatility of high frequency returns into multiplicative components, which can be easily interpreted and estimated. The conditional variance is expressed as the product of daily diurnal and stochastic intraday volatility components (Engle and Sokalska, 2012). In this section, we will give brief introduction to Engle and Sokalska's (2012) Multiplicative Component GARCH model and conclude the chapter by applying the model to 1-minute return of Citigroup to capture the intraday volatility pattern.

Consider the continuously compounded return $\boldsymbol{r_{t,i}}$, where $\boldsymbol{t(t = 1, \ldots, T)}$ denotes the day and $\boldsymbol{i(0, \ldots, N)}$ denotes the regularly spaced time interval at which returns are calculated. The current period is $\{\boldsymbol{t, i}\}$. Engle's (2012) multiplicative component GARCH model for high frequency data models the conditional variance as the multiplicative product of daily, diurnal and stochastic intraday volatility components. Intraday return process can be described as:

$$\boldsymbol{r_{t,i} = u_{t,i} + \varepsilon_{t,i}}$$

$$\boldsymbol{\varepsilon_{t,i} = (\sigma_t S_i q_{t,i}) z_{t,i}}$$

where,

$\boldsymbol{\sigma_t}$ is the daily variance (exogenously determined forecast) component

$\boldsymbol{S_i}$ is the diurnal variance component in each regularly spaced interval $\boldsymbol{i}$

$\boldsymbol{q_{t,i}}$ is the stochastic intraday variance component

$\boldsymbol{z_{t,i}}$ is the i.i.d (0,1) standardized innovation (white noise).

The daily volatility component can be estimated in different ways (Engle, 2012). Engle and Sokalska (2012) use commercially available volatility forecasts produced daily for each company in their analysis. Andersen and Bollerslev (1997, 1998) estimate the daily variance component from daily GARCH model. It can also be estimated based on daily realized variance (Engle, 2001; Engle and Gallo, 2006). Practically, it can be generated from a daily GARCH model. The diurnal (seasonal) part can be estimated as the variance of intraday returns in each regularly spaced interval.

$$\boldsymbol{S_i = \frac{1}{T}\sum_{t=1}^{T} \frac{\varepsilon_{t,i}^2}{\sigma^2}}$$

Dividing the residuals by the diurnal and daily volatility gives the normalized residuals

$$\boldsymbol{\bar{\varepsilon}_{t,i} = \varepsilon_{t,i}/(S_i \sigma_t)}$$

The model is estimated in two stages. First, normalizing the returns by daily and diurnal volatility components. Second stage, model the residual volatility as a GARCH(1,1) process.

$$q_{t,i}^2 = \omega + \alpha \bar{\varepsilon}_{t,i-1}^2 + \beta q_{t,i-1}^2$$

6.3.3 Data and Results

We analyze Citigroup one-minute return data between 2 May and 29 May 2013 (span over a month) which includes 19 trading days and no public holiday. We get log return of the dataset and we then apply the multiplicative component GARCH model introduced earlier to our data.

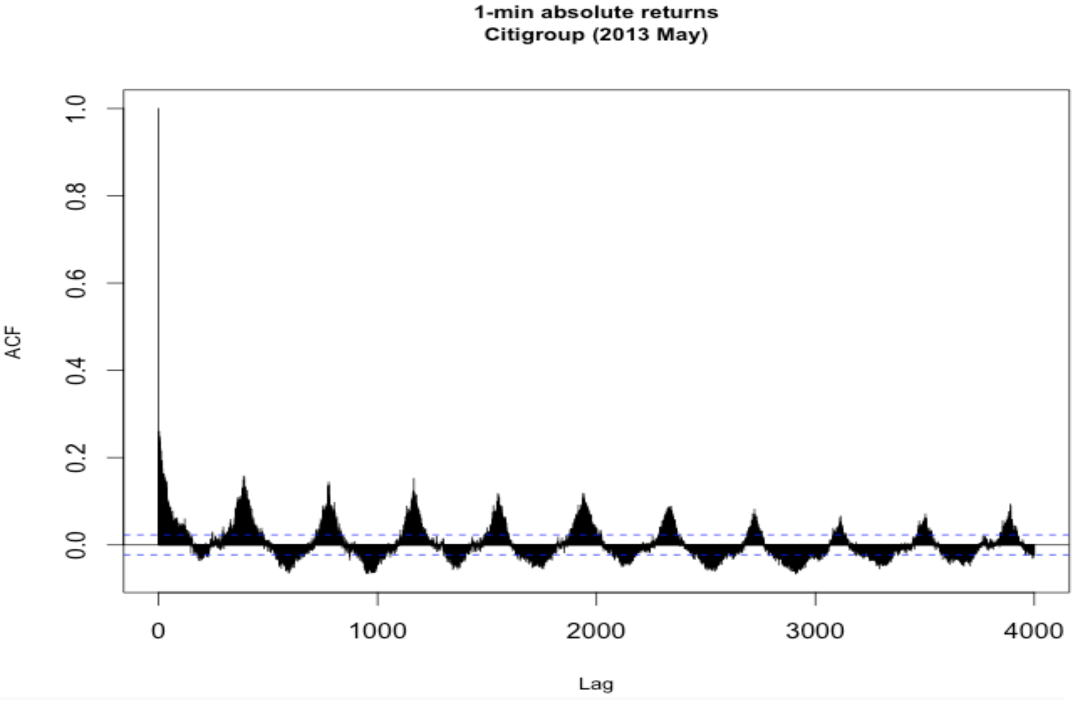


Figure 6.9 ACF plot of 1-min absolute return of Citigroup in May 2013

From Figure 6.9, we can see the regular pattern is very obvious, repeating approximately every 390 periods (minutes) which is one trading business day. The volatility is high at the open and close of the trading day and low in the middle of the day. The pattern is absolutely consistent with what have been documented the U-shape intraday pattern. From this figure, we can also tell why conventional GARCH model is not suitable to model high frequency data because general GARCH type models can only handle ACF exponentially decay, not the pattern we see in the plot.

From Figure 6.10, we can tell that the diurnal intraday volatility and the total composite volatility for the 1-min Citigroup return intraday data also exhibit U-shape pattern, specifically, extreme movements in the beginning and end of the day and the movements slow down in the middle of the day, which builds up the result.

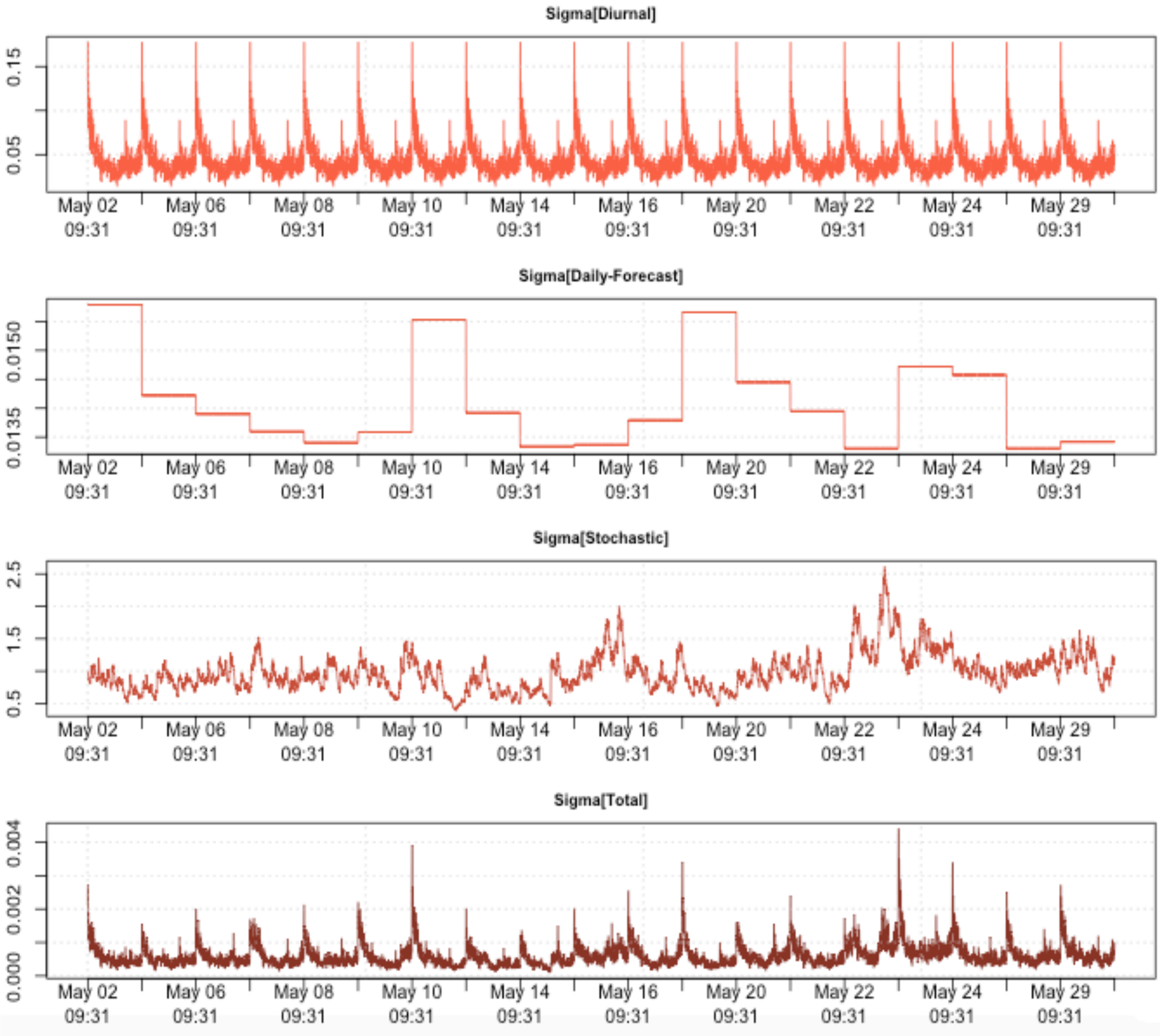


Figure 6.10 Volatility Components for Citigroup (1 min returns) Estimation Period May 2013. *Top panel*: the square root of diurnal variance component. *Second panel*: The square root of daily variance component. *Third panel*: The square root of intraday variance component. *Fourth panel*: The square root of the total variance component being the product of the up three variance component.

Chapter 7

Discussion and Future Work

7.1 Discussion

Clustering analysis is very popular unsupervised learning tool. It is vast used for dimension reduction and pattern recognition. Traditional clustering analysis only considers single objective or single dissimilarity measurements. In this thesis, the framework of multiple-objective clustering framework is discussed. We include three multiple-objective clustering methods. Multiple-objective clustering with automatic k-determination (MOCK) which considers both compactness and connectivity of cluster. Novel constrained and compound clustering proposed by Zhang (2011) incorporating multiple data source and dissimilarity measurements which gives a more comprehensive representation of the data. In gene expression analysis, we can incorporate both the gene expression data source and gene function data source. Biclustering analysis is also discussed. It is multiple-objective clustering analysis in a way that allows simultaneous clustering of the rows and columns of the data matrix while tradition clustering allows clustering either on the rows or on the columns. It is often used to analyze gene expression data and discover the interpretable biological patterns involving a subset of genes (rows) and subset of columns (conditions), which was first introduced to gene expression analysis by Cheng and Church (2000). Biclustering has been used in gene expression analysis and many researchers have developed many algorithms (Cheng and Church, 2000; Barkow et al., 2006; Turner et al., 2005; Kluger et al., 2003; Murali and Kasif, 2003) in analyzing gene expression data. The generalization of biclustering algorithm to represent it in the form of compound clustering with respect to rows and columns is discussed, where the paramester in the objective function is flexible instead of fixed. We applied biclustering technique on financial stock data to detect and compare the patterns and distributions of bear and bull market, and thus infer our current market type.

In the second part of the thesis, we identified the historical periods where stock price exhibit similar pattern resembles current market pattern. In the analysis, we see that history does rhyme and repeat itself. We experienced recent two large financial crisis, the 2000 great recession and the 2008 global financial crisis which is by far the worst in history. Analyzing the patterns of historical periods and then infer whether the market is going for recession/depression again is very meaningful. We them zoomed into each sector and looked at the data to see which sectors are mostly likely to be heavily impacted by the impending recession/depression.

The thesis is concluded with the analysis of intraday pattern of high frequency data. Financial time series data exhibit volatility clustering phenomenon, that is big changes in price tend to cluster together and small changes in price tend to cluster together. Real high frequency trading stock data traded on New York Stock Exchange (NYSE) was analyzed to study the intraday pattern. Notable pattern (U-shape) is detected at the aggregation level of one minute and five minute - significant big movements in trading activity in the beginning and end of the day, and quiet movements in the middle of day. The U shape intraday pattern was first documented by Wood et al. (1985) and Harris (1986) that average intraday return volatility exhibit distinct U shape over the trading day. My analysis is consistent with their findings. Standard GARCH models are not suitable for high frequency data. Engle (2012) proposed a Multiplicative Component GARCH model which can be applied to high frequency trading data. We applied a one-minute stock data to this model and same U shape pattern is detected which builds up the result.

7.2 Future Work

Multiple-resolution clustering analysis of financial time series can be considered as multiple-objective clustering with respect to clustering time series data at different time scale/resolution. For example, we want to cluster stocks that are show similar hourly, weekly, and monthly patterns which keeps both local and global information. Many works have made the effort to take consideration of multiple-resolution of financial time series data, such as Megalooikonomou et al. (2005), Fu et al. (2001), Vlachos et al. (2003) and Li and Kuo (2008). Also, in Chapter 5, identification of historical periods that resembles current market pattern to infer where the market is going for recession or depression and then analyze which sectors are likely to be heavily impacted by the impending recession. We performed the analysis using S&P 500 and focused on US stock market, further analysis cross country using indexes other than S&P 500, e.g., MSCI ACWI (ACWI) for global index, MSCI Europe (IEUR) for Europe stock market, FTSE (^FTSE) for London stock market, TOPIX for Tokyo stock market, and MSCI EM for emerging markets can also be included to see the impaction of recession on those markets outside US.

Reference

1. ANDERBERG, MR. (1973) Cluster Analysis for Applications. Academic Press, Inc., New York, NY.
2. Andersen, TG and Bollerslev, T. (1997) Intraday periodicity and volatility persistence in financial markets. Journal of Empirical Finance, 4(2), 115–158.
3. Andersen, TG and Bollerslev, T. (1998) Deutsche mark dollar volatility: Intraday activity patterns, macroeconomic announcements, and longer run dependencies. Journal of Finance, 53(1), 219-265.
4. Andersen, TG and Bollerslev, T. (1998) Forecasting financial market volatility: sample frequency vis-‘a- vis forecast horizon. Journal of Empirical Finance, 6.
5. BAEZA-YATES, RA. (1992) Introduction to data structures and algorithms related to informa- tion retrieval. In Information Retrieval: Data Structures and Algorithms, W. B. Frakes and R. Baeza-Yates, Eds. Prentice- Hall, Inc., Upper Saddle River, NJ, 13–27.
6. Baillie RT, Bollerslev, T and Mikkelsen, HO. (1996) Fractionally integrated generalized autoregressive conditional heteroskedasticity J. Econometrics 74 3–30
7. BALL, GH AND HALL, DJ. (1965) ISODATA, a novel method of data analysis and classification. Tech. Rep.. Stanford University, Stanford, CA.
8. Barkow, S., S. Bleuler, A. Prelic, P. Zimmermann, and E. Zitzler. (2006) Bicat: a biclustering analysis toolbox. Bioinformatics 22, 1282–1283.
9. Bel Mufti G, Bertrand P, El Moubarki L. (2005) Determining the number of groups from measures of cluster stability, In: Proceedings of International Symposium on Applied Stochastic Models and Data Analysis, 404-412.

10. Black, F. (1976) Studies of stock market volatility changes Proc. 1976 Meetings of the American Statistical Association, Business and Economic Statistics Section pp 177–81
11. Blashfield RK and Aldenderfer MS. (1978) The literature on cluster analysis Multivariate Behavioral Research 1978, 13: 271–295.
12. Bock, HH. (1985) On Some Significance Tests in Cluster Analysis. Journal of Classification, 2, 77–108.
13. Bollerlsev T, Chou, RY and Kroner, KF. (1992) ARCH modeling in finance: a review of the theory and empirical evidence J. Econometrics 52 5–59
14. Bollerslev, T and Ghysels, E. (1996) Periodic autoregressive conditional heteroscedasticity. Journal of Business & Economic Statistics, 14(2), 139–151.
15. Bollerslev, T. (1986) Generalized Autoregressive Conditional Heteroskedasticity. Journal of Econometrics.
16. Bollerslev, T. (1986) Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics 31, 307–327.
17. Brogaard, J, Hendershott, T amd Riordan, R. (2010) High Frequency Trading and Price Discovery. Forth-coming, Review of Financial Studies 27(8), 2267-2306.
18. Brownless, CT, Engle, R and Kelly, BT. (2010) A practical guide to volatility forecasting through calm and strom. Technical report, Department of Finance, NYU.
19. Calinski, RB and Harabasz, J. (1974) A dendrite method for cluster analysis. Communs Statist., 3, 1-27.
20. Chakrapani, C. (2004) Statistics in Market Research. Arnold, London.
21. Chankong, V and Haimes, YY. (1983) Multiobjective decision making theory and methodology. New York:North-Holland.

22. Cheng Y, Church GM. (2000) Biclustering of expression data. ISMB. 2000;8:93–103.

23. Cheng, Y and Church, GM. (2000) Biclustering of expression data. In Proc. Int. Conf. Intell. Syst. Mol. Biol., vol. 8. San Diego, CA, USA, 2000, pp. 93–103.

24. Cho, H, Dhillon IS, Guan Y and Sra S. (2004) "Minimum Sum-Squared Residue Cococlustering of Gene Expression Data," Proc. Fourth SIAM Int'l Conf. Data Mining, 2004.

25. Christie, AA. (1982) The stochastic behavior of common stock variances: value, leverage and interest rate effects J. Financial Economics 10 407–32

26. Cooper, MC and Milligan, GW. (1988) The Effect of Error on Determining the Number of Clusters. In Proceedings of the International Workshop on Data Analysis, Decision Support, and Expert Knowledge Representation in Marketing and Related Areas of Research, 319–328. Berlin: Springer-Verlag.

27. Corne, DW, Jerram, NR, Knowles, JD, Oates, MJ. (2001) PESA-II: Region-based selection in evolutionary multiobjective optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001), pp. 283–290. Morgan Kaufmann, San Francisco (2001).

28. Cortese, JD. (2000) The array of today: biomolecule arrays become the 21st century test tube. The Scientist, 14, 25.

29. Cuevas, A, Febrero, M and Fraiman, R. (2000) Estimating the number of clusters. Can. J. Statist., 28, 367-382.

30. Dacorogna, M, Gencay. R, Muller U, Olsen, R and Pictet, O. (2001) An introduction to high-frequency finance: Academic Press.

31. Das, N. (2003) Hedge Fund Classification Using K-Means Clustering Method. Ninth International Conference on Computing in Economics and Finance, University of Washington, Seattle.

32. Deb, K. (2003) Multi-objective evolutionary algorithms: Introducing bias among pareto-optimal solutions. In: Ghosh A, Tsutsui S, editors. Advances in Evolutionary Computing: Theory and Applications. London: Springer-Verlag. pp. 263–292.

33. DIDAY, E. AND SIMON, JC. (1976) Clustering analysis. In Digital Pattern Recognition, K. S. Fu, Ed. Springer-Verlag, Secaucus, NJ, 47–94.

34. Ding, Z, Engle, R and Granger, C. (1993) A long memory property of stock market returns and a new model. Journal of Empirical Finance 1(1), 83–106.

35. Dudoit S, Fridlyand J. (2002) A prediction-based resampling method for estimating the number of clusters in a dataset, Genome Biology, 3(7): research0036.1-0036.21.

36. Eisen, MB, Spellman, PT, Brown, PO and Botstein, PO. (1998) Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences, USA, 95, 14863–14868.

37. Eisen, MB, Spellman, PT, Brown, PO, and Bostein, D. (1998) Cluster analysis and display of genome-wide expression patterns. Preceedings of the national academy of science 95(25), 14863-14868.

38. Ellis, TE, Rudd, MD, Harsan Rayab, M and Wehrly, TE. (1996) Cluster analysis of McMI scores of suicidal psychiatric patients: four personality profiles. Journal of Clinical Psychology, 52, 411–422.

39. Engle, RF and Gallo, GM. (2006) A multiple indicators model for volatility using intra-daily data. Journal of Econometrics, 131(1-2), 3-27.

40. Engle, RF and Ng, V. (1993) Measuring and testing the impact of news on volatility J. Finance 48 1749–78

41. Engle, RF and Patton, A. (2001) What Good is a Volatility Model?. Quantiative Finance V1N2, pp237-245

42. Engle, RF and Sokalska, ME. (2012) Forecasting intraday volatility in the US equity market. Multiplicative component GARCH. Journal of Financial Econometrics, 10(1), 54–83.

43. Engle, RF. (1982) Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica 50(4), 987–1007.

44. Engle, RF. (1982) Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of UK Inflation. Econometrica, pp987-1008

45. Engle, RF. (1990) Discussion: stock market volatility and the crash of '87. Review of Financial Studies 3, 103–106.

46. Engle, RF. (2009) Anticipating Correlations: A New Paradigm for Risk Management. Princeton University Press.

47. Everitt, BS, Landau, S, Leese, M and Stahl, D. (2011) Cluster Analysis. John Wiley & Sons, Chichester, 5th edition.

48. Everitt, BS. (1979) Unresolved Problems in Cluster Analysis. Biometrics, 35, 169–181.

49. Fonseca, CM and Fleming, PJ. (1993) Genetic algorithms for multiobjective optimization: Formu- lation, discussion, and generalization. Proceedings of the Fifth International Conference on Genetic Algorithms. 416–423.

50. Fu, TC, Chung, FL, Ng, V and Luk, R. (2001) Pattern discovery for stock time series using self-organizing maps. In Workshop on temporal data mining, 7th international conference on knowledge discovery and data mining (pp. 27-37). New York: ACM Press.

51. Gaschetal, AP. (2001) Genomic expression responses to DNA-damaging agents and the regulatory role of the yeast ATR homolog mec1p. Mol. Biol. Cell, 12(10):2987–3003.

52. Ghalanos, A. (2013, March 20). High Frequency GARCH: The multiplicative component GARCH (mcsGARCH) model [Blog post] Retrieved from

53. Ghalanos, A. (2015) Introduction to the rugarch package (Version 1.3-1), August 16, http://cran.r-project.org/web/packages/rugarch/vignettes/Introduction_to_the_rugarch_package.pdf , sourced by web.

54. Gibson, R. and Gyger, S. (2007), The Style Consistency of Hedge Funds. European Financial Management, 13: 287–308. doi: 10.1111/j.1468-036X.2006.00355.xDuda, RO, Hart, PE and Stork, DG. (2001) Pattern Classification, 2nd ed. New York: Wiley, 2001.

55. Giot, P. (2005) Market risk models for intraday data. The European Journal of Finance, 11(4), 309-324.

56. Giot, P. (2005) Time transformations, intraday data and volatility models. Journal of Computational Finance, 4, 31–62.

57. Glosten, LR, Jagannathan, R and Runkle, DE. (1993) On the Relation between the Expected Value and the Volatility of the Nominal Excess Returns on Stocks, Journal of Finance, 48(5), 1779-1801.

58. Goodall, DW. (1954) Objective methods for the classification of vegetation. III. An essay in the use of factor analysis. Austral. J. Bot. 1:39-63

59. Gordon, A. (1996) Null models in cluster validation. In from data to Knowledge (eds W. Gaul and D. Pfeifer), pp. 32-44. New York: Springer.

60. Gordon, AD. (1981) Classification, Chapman and Hall, London.

61. Gordon, AD. (1999) Classification (2nd edition). Chapman and Hall/CRC Press, London.

62. Grabusts, P. (2011) The choice of matrics for clustering algorithms. 8th International Scientific and Practical conference, vol-2,2011.

63. Green, PE, Frank, RE and Robinson, PJ. (1967) Cluster analysis in test market selection. Management Science, 13, 387–400.

64. Handl, J and Knowles, J. (2004) Evolutionary Multiobjective Clustering, in Xin Yao et al. (editors), Parallel Problem Solving from Nature (PPSN VIII), pp. 1081–1091, Springer-Verlag, Lecture Notes in Computer Science, Vol. 3242, Berlin, September 2004.

65. Handl, J and Knowles, J. (2004) Multiobjective clustering with automatic determination of the number of clusters, Technical Report No. TR-COMPSYSBIO-2004-02, UMIST, Department of Chemistry, August 2004.

66. Handl, J and Knowles, J. (2005) Exploiting the Trade-Off — the Benefits of Multiple Objectives in Data Clustering.

67. Handl, J, and Knowles, J. (2007) An Evolutionary Approach to Multiobjective Clustering. IEEE Transactions on Evolutionary Computation, 11, 56-76.

68. Harrion, M. (June 22, 2014). PCA and K-means Clustering of Delta Aircraft [Blog post] Retrieved from

69. Harris, L. (1986) A transaction data study of weekly and intradaily patterns in stock returns. Journal of Financial Economics 16, 99-117.

70. Hartigan, J. (1975) Clustering Algorithms. New York: Wiley.

71. Hartigan, JA. (1972) Direct Clustering of a Data Matrix. Journal of the American Statistical Association, 6, 123–129.

72. Hartigan, JA. (1975) Clustering Algorithms. New York, NY. John Wiley & Sons.

73. Hartigan, JA. (1985) Statistical Theory in Clustering. Journal of Classification, 2, 63–76.

74. Hastie, T, Tibshirani, R and Friedman, J. (2009) The elements of statistical learning: data mining, inference and prediction, Spinger.

75. Hay, PJ, Fairburn, CG and Doll, HA. (1996) The classification of bulimic eating disorders: a community-based cluster analysis study. Psychological Medicine, 26, 801–812.

76. HOTELLING, H. (1933) Analysis of a Complex of Statistical Variables into Principal Components. Journal of Educational Psychology, 24(6 & 7), 417–441 & 498–520.

77. http://unstarched.net/2013/03/20/high-frequency-garch-the-multiplicative-component-garch-mcsgarch-model/, sourced by web.

78. http://www.everydayanalytics.ca/2014/06/pca-and-k-means-clustering-of-delta-aircraft.html, sourced by web.

79. Huang, Q, Wang T, Tao, D and Li, X. (2015) Biclustering Learning of Trading Rules. IEEE T. Cybernetics 45(10): 2287-2298.

80. Huang, Q, Wang T, Tao, D and Li, X. (2015) Biclustering Learning of Trading Rules. IEEE T. Cybernetics 45(10): 2287-2298.

81. Hughes, JD, Estep, PE, Tavazoie, S and Church, GM. (2000) Computational identification of cis- regulatory elements associated with groups of functionally related

genes in Saccharomyces Cerevisiae. J. Mol. Biol., 296:1205–1214, 2000. http://atlas.med.harvard.edu/.

82. Huth, R, Nemesova, I and Klimperova, N. (1993) Weather categorization based on the average linkage clustering technique: an application to European mid-latitudes. International Journal of Climatology, 13, 817–835.
83. ICHINO, M. AND YAGUCHI, H. (1994) Generalized Minkowski metrics for mixed feature-type data analysis. IEEE Trans. Syst. Man Cy- bern. 24, 698–708.
84. Ito, Ryoko. (2013) Modeling dynamic diurnal patterns in high frequency financial data. Cambridge Working Papers in Economics 1315, Faculty of Economics, University of Cambridge.
85. Jain, AK, and Dubes, R. (1988) Algorithms for Clustering Data, Prentice-Hall, Inc.
86. Jain, AK, Murty, MN, and Flynn, PJ. (1999) Data Clustering: A Review. ACM Comput. Surv., 31, 264-323.
87. Johnson, SC. (1967) Hierarchical Clustering Schemes. Psychometrika, 32, 241-254.
88. Jones BL, Nagin DS. (2007) Advances in group-based trajectory modeling and an SAS procedure for estimating them. Sociological Methods and Research 2007;35:542-571.
89. Jones, BL, Nagin, DS and Roeder, K. (2001) A SAS procedure based on mixture models for estimating developmental trajectories. Sociological Methods & Research, 29, 374-393.
90. Kaiser, S and Friedrich L. (2008) A toolbox for bicluster analysis in R. In Compstat 2008: Proceedings in Computational Statistics. edited by Paula Brito. Heidelberg, Germany: Physica Verlag. Pp 201-208.

91. Kaiser, S, Santamaria, R, Khamiakova, T, Sill, M, Theron, R, Quintales, L and Friedrich L. (2015). biclust: BiCluster Algorithms. http://cran.R-project.org/package=biclust.

92. Kaiser, S. (2011) Biclustering: Methods, Software, and Application. PhD thesis Ludwing-Maximilians-University Munich, Department of Statistics.

93. Kaufman, L and Rousseeuw, P. (1990) Finding Groups in Data: an Introduction to Cluster Analysis. New York: Wiley.

94. Kerr, MK and Churchill, GA (2001) Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments. Proceedings of the National Academy of the USA, 98, 8961–8965.

95. King, B. (1967) Step-wise clustering procedures. Journal of the American Statistical Association. 69: 86–101.

96. Kluger, Y., R. Basri, J. T. Chang, and M. Gerstein (2003). Spectral biclus- tering of microarray data: Coclustering genes and conditions. Genome Re- search 13, 703–716.

97. Krzanowski, WJ and Lai, YT. (1985) A criterion for determining the number of groups in a data set using sum of squares clustering. Biometrics, 44, 23-34.

98. Kurtz, A, Moller, HJ, Bavidl, G and et al. (1987) Classification of parasuicide by cluster analysis. British Journal of Psychiatry, 150, 520–525.

99. L. Lazzeroni and A. Owen. Plaid models for gene expression data. Statistica Sinica, 12:61– 86, 2002.

100. Lazzeroni L, Owen A. (2000) Plaid models for gene expression data. Statistica Sinica 12:61-86.

101. Lin, J, Vlachos, M, Keogh, E and Gunopulos, D. (2003) Multi-Resolution K-Means Clustering of Time Series and Application to Images, Workshop on Multimedia

Data Mining, the 4th SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington D.C.

102. Littmann, T. (2000) An empirical classification of weather types in the Mediterranean Basin and their interrelation with rainfall. Theoretical and Applied Climatology, 66, 161–171.

103. Liu, S and George, R. (2005) Mining Weather Data using Fuzzy Cluster Analysis. Springer, Berlin.

104. MacQueen, J. (1967) Some Methods for Classification and Analysis of Multivariate Observations, in Proc. 5th Berkeley Symp. Mathematical Statist. Probability, pp. 281- 297.

105. Madeira SC, Oliveira AL. (2004). "Biclustering Algorithms for Biological Data Analysis: A Survey". IEEE Transactions on Computational Biology and Bioinformatics 1 (1): 24–45. doi:10.1109/TCBB.2004.2. PMID 17048406.

106. Madeira, SC and Oliveira, AL. (2004) Biclustering algorithms for biological data analysis: A survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics 1(1), 24–45.

107. Mandelbrot, BB. (1963) The variation of certain speculative prices, Journal of Business, XXXVI (1963), pp. 392–417.

108. MAO, J. AND JAIN, AK. (1996) A self-organizing network for hyperellipsoidal clustering (HEC). IEEE Trans. Neural Netw. 7, 16–29.

109. March, ST. (1983) Techniques for structuring database records. ACM Computing Surveys, 15: 45–79.

110. Marriott, FHC. (1971) Practical problems in a method of cluster analysis. Biometrics, 27, 501-514.

111. Martin, G. (2001) Making Sense of Hedge Fund Returns: What Matters and What Doesn’t, Derivatives Strategy. working paper. Isenberg School of Management, University of Massachusetts, Amherst.

112. Matake, N, Hiroyasu, T, Miki, M, and Senda, T. (2007) Multiobjective Clustering with Automatic K-Determination for Large-Scale Data. Proceedings of the 9th annual conference on Genetic and evolutionary computation, 861-868.

113. McLachlan GJ, Khan N. (2004) On a resampling approach for tests on the number of clusters with mixture model-based clustering of tissue samples, Journal of Multivariate Analysis, 90: 90-1005.

114. Megalooikonomou, V, Wang, Q, Li, G and Faloutsos, C. (2005) A multiresolution sym- bolic representation of time series. In Proc. of the International Conference on Data Engineering, 668–679.

115. Meglen, RR. (1992) Examining large databases: a chemometric approach using principal component analysis. Marine Chemistry 39, 217–237.

116. Miceli, MA and Susinno, G. (2003) Using Trees to Grow Money. Risk. S11–S12.

117. Miettinen, K. (1999). Nonlinear multiobjective optimization, Boston: Kluwer.

118. Milligan, GW and Cooper, MC. (1985) An examination of procedures for determining the number of clusters in a dataset. Psychometrika, 50, 159-179.

119. Minaei-Bidgoli B, Topchy A, Punch WF. (2004) A comparison of resampling methods for clustering ensembles, International conference on Machine Learning; Models, Technologies and Application (MLMTA04), Las Vegas, Nevada, pp. 939-945.

120. Mirkin B. (2011) Choosing the number of clusters. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1(3), 252–260.

121. Muller, U, Dacorogna, M and Pictet, O. (1996) Heavy tails in high-frequency financial data: Olsen preprint.

122. Murali, T. and S. Kasif (2003). Extracting conserved gene expression motifs from gene expression data. Pacific Symposium on Biocomputing 8, 77–88.

123. MURTAGH, F. (1984) A survey of recent advances in hierarchical clustering algorithms which use cluster centers. Comput. J. 26, 354 –359.

124. Naghieh, E and Peng, Y. (2009) Microarray Gene Expression Data Mining: Clustering Analysis Review", Aug 20, 2009.

125. NAGY, G. (1968) State of the art in pattern rec- ognition. Proc. IEEE 56, 836–862.

126. Nelson, DB. (1991) Conditional heteroscedasticity in asset returns: a new approach Econometrica 59 347–70

127. Paykel, ES and Rassaby, E. (1978) Classification of suicide attempters by cluster analysis. British Journal of Psychiatry, 133, 42–52.

128. PEARSON, K. (1901) On Lines and Planes of Closest Fit to Systems of Points in Space. Philosophical Magazine, Series 6, 2(11), 559–572.

129. Per-Erik, D. (1980) Euclidean Distance Mapping. Computer Graphics and Image Processing, 14, 227-248.

130. Pilowsky, I, Levine, S and Boulton, DM. (1969) The classification of depression by numerical taxonomy. British Journal of Psychiatry, 115, 937–945.

131. Prim, RC. (1957) Shortest connection networks and some generalizations. Bell System Technical Journal, 36:1389-1401.

132. Sarle, WS. (1983) Cubic Clustering Criterion. SAS Technical Report A-108. SAS Institute Inc., Cary.

133. Schaffer, JD. (1984) Some experiments in machine learning using vector evaluated genetic algo- rithms. (Doctoral Dissertation). Nashville, TN: Vanderbilt University.

134. Schwert, GW. (1989) Why does stock market volatility change over time? J. Finance 44 1115–53

135. Sen, P and Yang, JB. (1998). Multiple criteria decision support in engineering design. London: Springer.

136. Sharan, S, Porat, UB and Bleiberg, O. (2006) Analysis of Biology Networks: Network Modules – Clustering and Biclustering. Lecture Notes in Computer Science, November 23 2006. Retrieved from: http://www.cs.tau.ac.il/~roded/courses/bnet-a06/lec05.pdf.

137. Sheng-Tun Li , Shu-Ching Kuo, Knowledge discovery in financial investment for forecasting and trading strategy through wavelet-based SOM networks, Expert Systems with Applications: An International Journal, v.34 n.2, p.935-951, February, 2008

138. Singh, AK, Allen, DE and Powell, RJ. (2013) Intraday Volatility Forecast in Australian Equity Market (August 12, 2013). Available at SSRN: http://ssrn.com/abstract=2308787 or http://dx.doi.org/10.2139/ssrn.2308787

139. Sneath, PH and Sokal, RR. (1973) Numerical Taxonomy. London, UK: Freeman.

140. Spellman, PT, Sherlock, G and et al. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Mol. Biol. Cell, 9:3273– 3297, 1998.

141. Srinivas, N and Deb, K. (1994) Multi-Objective function optimization using non-dominated sorting genetic algorithms, Evolutionary Computation, 2(3), 221–248.

142. Steinhaus, H. (1957) Sur La Division Des Corps Matériels En Parties (in French). Bull. Acad. Polon. Sci., 4, 4.

143. SYMON, MJ. (1977) Clustering criterion and multi-variate normal mixture. Biometrics 77, 35–43.

144. Systematicinvestor (13 Jan. 2012). "Time Series Matching" [Web blog post]. Trading Strategies. Systematic Investor Blog:https://systematicinvestor.wordpress.com/2012/01/13/time-series-matching/

145. Tanay, A, Sharan, R and Shamir, R. (2004) Biclustering Algorithms: A Survey. In Handbook of Computational Molecular Biology, Edited by Srinivas Aluru, Chapman.

146. Tibshirani, R, Walther, G, and Hastie, T. (2001) Estimating the Number of Clusters in a Data Set Via the Gap Statistic. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63, 411-423.

147. Tsay, R. (2002) Analysis of Financial Time Series. John Wiley& Sons. New York.

148. Turner, H., T. Bailey, and W. Krzanowski (2005). Improved biclustering of microarray data demonstrated through systematic performance tests. Com- putational Statistics and Data Analysis 48, 235–254.

149. Verhage, Julie (26 Jan. 2016). “Will history repeat itself” [Web blog post]. Morgan Stanley Analyzed 43 Bear Markets and Here’s What It Found. BloombergBussiness:http://www.bloomberg.com/news/articles/2016-01-26/morgan-stanley-analyzed-43-bear-markets-and-here-s-what-it-found.

150. Wang, H, Wang W, Yang, J and Yu PS. (2002) “Clustering by Pattern Similarity in Large Data Sets,” Proc. 2002 ACM SIGMOD Int’l Conf. Management of Data, pp. 394-405, 2002.

151. WARD, JH. (1963) Data Hierarchical grouping to optimize an objective function. J. Am. Stat. Assoc. 58, 236–244.

152. Ward, JH. (1963) Hierarchical Grouping to Optimize an Objective Function. Journal of the American Statistical Association, 58, 236-244.

153. Wilson DR and Martinez TR. (1997) Improved Heterogeneous Distance Functions. Volume 6, pages 1-34.

154. Wood, RA, Mclnish, TH, Ord, JK. (1985) An investigation of transaction data for NYSE stocks. Journal of Finance 25, 723-739.

155. Xu, R and Wunsch, D. (2005) Survey of clustering algorithms. IEEE Transactions on Neural Networks, 16: 645–678.

156. Yun Xue, Zhiwen Liu, Jie Luo, et al. (2015) Stock Market Trading Rules Discovery Based on Biclustering Method. Mathematical Problems in Engineering, vol. 2015, Article ID 849286, 13 pages, 2015. doi:10.1155/2015/849286

157. Yun Xue, Zhiwen Liu, Jie Luo, et al. (2015) Stock Market Trading Rules Discovery Based on Biclustering Method. Mathematical Problems in Engineering, vol. 2015, Article ID 849286, 13 pages, 2015. doi:10.1155/2015/849286

158. Zakoian, JM. (1994) Threshold heteroskedastic models, Journal of Economic Dynamics and Control, 18, 931-955.

159. Zhang, S. (2011) New Development in Cluster Analysis and Other Related Multivariate Analysis Methods. (Doctoral Dissertation). Stony Brook University, Applied Mathematics and Statistics.

160. Zitzler, E and Thiele, L. (1998) Multiobjective optimization using evolutionary algorithms—A comparative case study. Parallel Problem Solving from Nature, V, 292–301.