Recursive Temporal Meta-cluster of Daily Time Series

by

Farhana Haider (A00381047)

A Thesis Submitted to

Saint Mary's University, Halifax, Nova Scotia

in Partial Fulfillment of the Requirements for

the Degree of Masters of Science in Applied Science

August 2015, Halifax, Nova Scotia

Approved: Dr. Pawan Lingras
Supervisor

Approved: Dr. Sheela Ramanna
External Examiner

Approved: Dr. Paul Muir
Member, Supervisory Committee

Approved: Dr. Dawn Jutla
Member, Supervisory Committee

Date: August 20, 2015

Contents

List of Figures

List of Tables

Abstract

Recursive Temporal Meta-cluster of Daily Time Series

by Farhana Haider (A00381047)

Identifying pattern groups from large temporal data sets, preserving clustering schemes obtained from different heuristic algorithms and presenting temporal pattern profiles for a specific day and previous days are significant concerns in many fields. As clustering schemes created by different heuristic algorithms may not completely agree with each other, researchers have proposed different clustering ensemble techniques to combine such schemes. In the first phase, this research proposes a rough set based ensemble method that preserves the inherent order in clustering. In the second phase, the Recursive Meta-cluster algorithm is used to create meta-profiles having current volatility with historical perspective for the financial daily temporal pattern clusters, which a trader may use while making decisions. Traditionally, any information of the historical or future clustering is not considered for temporal clustering. The proposed algorithm clusters the temporal patterns iteratively using previous clustering results from connected historical patterns.

Keywords: Clustering, Ensemble, Rough Sets, Granular Computing, Meta-cluster, Meta-profile, Financial Time Series, Volatility

August 20, 2015

Chapter 1

Introduction

This chapter describes the overall introductory background and framework of the thesis. Section 1.1 introduces the overview of the underlying research question. Section 1.2 and 1.3 specify the objectives and methodology of the research in brief. Section 1.4 is an introduction to the organization of this report.

1.1 Overview

Mining valuable data from a raw data set is always a significant issue in all aspects of life that incorporate information in a large volume. In a small dimension where the data set is minute, this task can be done with manual human effort. However, when the volume increases, eventually it becomes difficult to mine the useful information. Automated systems can resolve this problem. Data mining systems are such systems that can extract required data from raw data. Clustering is one such data mining mechanism. Numerous studies have focused on this method, prescribing various types and enhancements of the basic idea of

this process. Meta-clustering is one advancement of the basic clustering whereas Recursive Meta-clustering is an extension of Meta-clustering.

With technology advancement and increasing diversity of data, data mining systems became more field as well as data structure specific. An example of a specifc data format is the daily time series. Researchers have been working for years to extract specific information or patterns from such series in an effective manner.

Along with other time series data, financial time series data is playing an important role in our everyday life. As a consequence, techniques to analyze and mine such data are evolving.

1.1.1 Disagreement of Clustering Schemes

Granular computing (GrC) is an emerging computing paradigm of information processing. It concerns the processing of complex information entities called information granules, which arise in the process of data abstraction and derivation of knowledge from information or data. In granular computing an object is represented as an information granule. Zadeh introduced the notion of information granulation in 1979 (Zadeh, 1979) in the context of fuzzy sets. The research community did not immediately grasp the full implications of Zadeh's initial proposal. Pawlak's theory of rough sets using partitions induced by equivalence relations can also be considered as a type of granulation. The subsequent prolific theoretical and practical developments based on rough sets indicated the diverse potential of this new granular computational paradigm. Zadeh further elaborated on information granulation and its central role in human reasoning (Zadeh, 1997), which provided new insights into granular computing. Granular computing encompasses multiple levels or lay-

ers of granularity in thinking, problem solving and information processing (Yao, 2010). A more elaborate formulation of granular computing followed Zadeh's paper in the form a book by early pioneers Bargiela and Pedrycz. They provided an elegant pyramidal information processing paradigm for granular computing (Bargiela and Pedrycz, 2003). Since Bargiela and Pedrycz's book, researchers have proposed many theories, frameworks, models, methodologies and techniques for granular computing. The mainstream granular computing research focused on fuzzy sets, rough sets, interval analysis and cluster analysis (Pedrycz et al., 2008; Yao, 2007, 2008). Researchers are now using granular computation in a broader context.

In the first phase of this research, we view the same object using different information granules depending on the context or point of view. As an example, we have considered a multigranular view for a time series of commodity prices. A trader finds a daily price pattern interesting when it is volatile. The higher the fluctuations in prices, the more volatile the pattern. A Nobel Prize-winning research introduced the concept of the Black Scholes index to quantify volatility of a pattern(Black and Scholes, 1973). The Black Scholes index is a single concise index to identify volatility in a daily pattern. We can segment daily patterns based on values of the Black Scholes index. This segmentation is essentially a clustering of a one dimensional representation (Black Scholes index) of the daily pattern. However, a complete distribution of prices during the day can provide more elaborate information on the volatility. While a distribution consisting of frequency of different prices is not a concise description for a single day, it can be a very useful representation of daily patterns for clustering based on volatility. That means we will have two different ways of grouping daily patterns. The obvious question that arises in this case is which one of the groupings

is better. Instead of using one grouping or the other, clustering ensemble can be used to come up with a consensual grouping. Traditionally, clustering ensemble is applied to crisp clustering schemes and creates another crisp clustering scheme. In crisp clustering schemes, an object is assigned to one and only one cluster. There is no room for ambiguity in such a clustering. However, while combining results from two clustering schemes, there will be situations when the two clustering schemes do not agree with each other. When the two clustering schemes do not agree with each other on the assignment of an object, it should belong to two different clusters in the resulting clustering schemes. That means the clusters will need to overlap.

This thesis proposes the use of Rough Set theory, which includes the concept of a boundary region that will be ideal for representing overlap of clusters from different clustering schemes. The proposed Rough Ensemble clustering is shown to be a natural representation for combining the clustering schemes of multigranular representations of the same object.

1.1.2 Need for Temporal Historical Pattern Profiling

Profiling objects is useful in various real world data sets where extraction of certain characteristics of those objects are required. As an example, transaction characteristics of a customer of a retail store or profiling customer versus product characteristics may allow that store to significantly help in the analysis of the daily transaction properties of the store and therefore update its product stock accordingly. (Lingras et al., 2014) showed how Recursive Meta-clustering can be used to create such profiles. (Lingras and Rathinavel, 2012) (Triff and Lingras, 2013) described two other ways to implement Recursive Meta-clustering for a network and for information granules of businesses and reviewers of a business rating web

site, respectively.

Recursive Meta-clustering can be understood as an advancement of Meta-clustering. A version of Meta-clustering that supports two stage clustering was proposed by (Slonim and Tishby, 2000) where the technique is used for document clustering. (El-Yaniv and Souroujon, 2001) proposed a repetitive version of this approach. (Castellano et al., 2002) introduced double-clustering using the Fuzzy C-means Algbaorithm. (Caruana et al., 2006) proposed a Meta-clustering algorithm to allow users to select reasonable clustering that best fits their need. In this process the base level clusters are grouped by their similarities and presented to users by organizing them in a meta level. (Ramirez-Cano et al., 2010) showed Meta-clustering as the technique to group game-play data based on players' social relations.

The information granule used in the Recursive Meta-clustering algorithm is represented by parts namely static part and dynamic part. The static parts are created from the information directly related to the candidate record and the dynamic part is created considering the information of the record that is related to the record of the static part. On the other hand, the dynamic part is derived recursively from the clustering process until some predefined criteria is satisfied (Lingras and Rathinavel, 2012; Lingras et al., 2014; Triff and Lingras, 2013). Though Lingras and Rathinavel initially used crisp clustering using K-means for this algorithm, the concept was extended later to use fuzzy clustering by means of Fuzzy C-means.

In addition to non-temporal objects, temporal or time-series objects often demand profiling. In different areas, including Economics, Finance, Social Science, Environmental Science where there are time series, automated profile or characteristic representation of current pattern along with historical patterns over certain period are very useful for further

analysis or decision making. For instance, if the current and historical price fluctuation of a product is available to an investor without the burden of analyzing each temporal pattern for the days under consideration, he or she can more easily decide whether to proceed with an investment or not. To profile such temporal patterns this research proposes an updated Recursive Meta-cluster that works on time series data. However, the basic idea used in the proposed algorithm is same as the one proposed by (Lingras and Rathinavel, 2012; Lingras et al., 2014; Triff and Lingras, 2013)

1.2 Objectives

According to the problem domain as discussed in the previous section, the objectives of this research can be defined in two phases:

1. Grouping temporal patterns by preserving different clustering schemes

2. Profiling temporal patterns for a given day and for the last m days

1.2.1 Grouping Temporal Patterns by Preserving Different Clustering Schemes

The first target of this thesis is to propose a clustering scheme that can represent groups of the time series patterns by preserving clustering schemes obtained from different heuristic algorithms.

1.2.2 Profiling Temporal Patterns for a Given Day and for the Last m Days

The second and ultimate objective of this research is to create profiles of current patterns of a temporal data set along with historical patterns for m previous periods. In other words, as an output, the system can represent a human interpretable status for a pattern and patterns from m previous periods.

1.3 Methodologies

Methods required to achieve the prescribed objectives defined in last section are represented in the following sub-sections. The first objective is targeted to be achieved at the end of step 3.4 while the second will be achieved at the end of 3.7. Algorithms of Rough Ensemble Clustering and Recursive Temporal Meta-clustering are given in section 4.3.1 and 5.2 respectively. For clustering and analysis using plotting we will use the statistical language R, whereas to execute the program modules will use shell scripts. In addition, to execute those programs that use a large data set for this research, we are using the linux cluster for Saint Mary's University provided by the Atlantic Computational Excellence Network (ACEnet). The workflow to implement the two algorithms is given in Figure /refworkFlow.

1.3.1 Preparation of Data Set

We will use a data set containing average prices of 223 financial products or instruments at 10 minutes intervals for 121 days or less, comprising a total of 27,012 records. For the

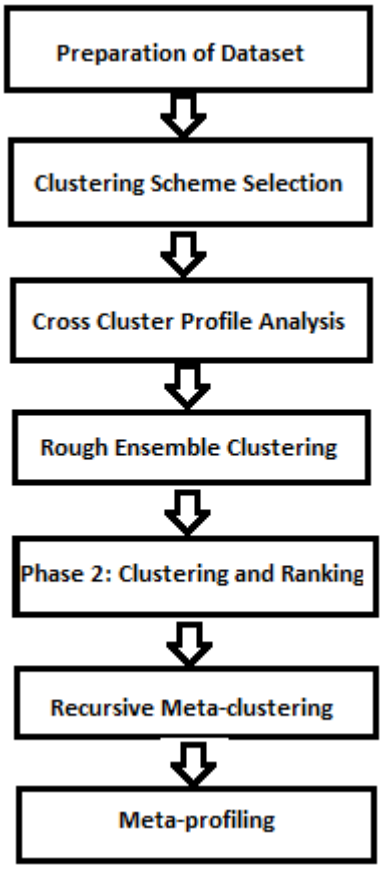


Figure 1.1: Workflow to Implement the Proposed Algorithms

first phase, two data sets will be created from this data set. One of these will be multidimensional and will represent 5 percentile values. The other will be one dimensional and represents the Black Scholes daily volatility. For the second phase, a filter on the data set representing the percentile values as well as the Black Scholes daily volatility will be required to ensure that each commodity has more than m transactions, where m is the number of specified historical periods.

1.3.2 Clustering Scheme Selection

Both the sets of data will be clustered 10 times using the K-means algorithm and the Davies-Bouldin index will be determined. Along with this, we will analyze the cluster results and determine the optimal number of clusters. The final cluster profiles will be preserved.

1.3.3 Cross Cluster Profile Analysis

The profiles created will be compared using overlap checks using an overlapping matrix along with plotting of cross matching patterns. The analysis is targeted to determine agreement and disagreement of the two types of cluster results obtained from the data set.

1.3.4 Rough Ensemble clustering

To ensure quality clustering, significant groupings of both the cluster results should be kept. In order to have such a quality grouping, a Rough Ensemble clustering scheme will be proposed in this stage. At this step, objective 1 will be achieved.

1.3.5 Phase 2 Clustering and Ranking

The filtered data set representing percentile value as well as the Black Scholes volatility index values will be used in this phase. Patterns from the (m+1)th to nth day are treated as the static part in this case. In the first iteration, the static part of the data set will be clustered and ranked according to the volatility index values. In the next consecutive iterations, we will cluster the data set of the concatenating static and dynamic parts and rank them. The

dynamic part which is the connective or relative values of the static part data is created with ranks of the last iteration clusters whose centroids are closest to the day in consideration.

1.3.6 Recursive Meta-clustering

Clustering and ranking as described in the previous step are iteratively observed until the centroids of the dynamic parts have converged or stabilized. The final ranked cluster results found in this way will be used to represent the meta profiles. We will obtain different sets of cluster profiles for each of the data sets used.

1.3.7 Meta-profiling

Once the Recursive Meta-clusters are determined, for any given day existing in data set, the volatility rank of the day and the previous m days can be presented based on the final Meta-cluster results. The volatility rank of a day and the previous m days can be different based on the type of data set (i.e. percentile and Black Scholes daily volatility).

1.4 Organization of the Thesis

The next consecutive chapters of this thesis report are organized as follows:

- **Chapter 2** is the background or study of previous works relevant to this thesis in a categorized manner.

- **Chapter 3** demonstrates the process to prepare the data set for experiment with an example illustration.
- **Chapter 4** describes the theory and overall process of Rough Ensemble clustering with experimental results.
- **Chapter 5** represents the procedure and experimental results for the Recursive Meta-clustering of daily time series patterns.
- **Chapter 6** is a conclusive outline describing the summary of the research outcome and future work.

Chapter 2

Literature Review

This chapter presents a literature review on the basis of searches on literature that are related to this thesis on Recursive Temporal Meta-clustering for daily time series. The overall organization of this chapter is based on 76 relevant articles. This chapter is a background study to review the relevant research contributions more precisely, targeting ultimate advancement of this research.

Considering the major terminologies and branches related to this research, this chapter is organized in five categories; namely Clustering, Meta-clustering, Recursive Meta-clustering, Temporal/Time Series Clustering and Financial Time Series. The last category is chosen considering the type of data that is used in this thesis for empirical study. The sub-sections of the following sections are designed chronologically with respect to the categories specified. The reviews in each section and subsection are organized in order, to represent research advancement with respect to time. Each section starts with a brief introduction on the literature reviews that follow. A summary on the overall study is given at the end of this chapter.

2.1 Categorized Reviews on Literatures

2.1.1 Clustering

This section represents a review of basic work done on clustering. The following three consecutive subsections 2.1.1.1, 2.1.1.2, 2.1.1.3 and 2.1.1.4 correspond to the reviews of crisp clustering, soft clustering, N-clustering and ensemble clustering respectively.

2.1.1.1 Crisp Clustering

A qualitative and quantitative understanding of large amounts of N-dimensional data can be obtained by identifying reasonably good similarity groups called clusters. For a finite set of objects X = $\{\vec{x_1}, ..., \vec{x_n}\}$ represented by m-dimensional vectors, a clustering scheme groups the n objects into k clusters $C = \{c_1, ..., c_k\}$. Clustering is an unsupervised learning process. The correct grouping is unknown. A number of cluster validity measures including the Davies-Bouldin (DB) index (Mitra, 2004) have been proposed to evaluate clustering schemes. For k^n possible clustering schemes, the objective is to find the most optimal grouping. This idea of clustering represents the concept of crisp or boolean clustering. This is the most obvious application of K-means, which we now define.

K-means Algorithm: The name K-means originates from the means of the k clusters that are created from n objects. In this approach assigning a data object to a cluster depends upon the distances of the object from the available distributions. To measure the distances the Euclidean distance is calculated (MacQueen, 1967). The basic definition and approach of K-means was made efficient by Hartigan and Wong (1979). They proposed the standard

K-means algorithm which clusters objects in such a way that sum of squares distance among objects within each cluster is minimized. The objective is to assign n objects from X to k clusters. The process begins by randomly choosing k objects as the centroids of the k clusters. The objects are assigned to one of the k clusters based on the minimum value of the distance $d(x, c_i)$ between an object x and the centroid of cluster c_i. Usually, the centroid of a cluster and the objects are represented by vectors. After the assignment of all the objects to various clusters, the new centroid vectors of the clusters are re-calculated as means of all the objects in the clusters. The process stops when the centroids of the clusters stabilize, i.e. the centroid vectors from the previous iteration are identical to those generated in the current iteration.

2.1.1.2 Soft Clustering

Soft clustering is a technique that overcomes the problem of a tendency toward unnecessary cluster-splitting when clustering is made using traditional crisp clustering approach. There are different types of soft clustering algorithms.

Fuzzy Clustering: Fuzzy clustering can allow a candidate data object be a member of more than one cluster at a time with a different degree of membership between 0 and 1 (DUNN, 1973). The algorithm Fuzzy C-means (FCM) used for this, which was modified later on. This method makes it possible to specify a varying degree of memberships of an object to different clusters where the sum of the coefficients for different clusters of a given

object is 1. This algorithm is based on minimization of the following objective function:

$$\sum_{i=1}^{n}\sum_{j=1}^{k} u_{ij}^{m}\ v(\vec{x_i}, \vec{c_j}) \quad , \quad 1 < m < \infty \tag{2.1}$$

where n is the number of objects and each object is a d dimensional vector. A parameter m is any real number greater than 1, u_{ij} is the degree of membership of the i^{th} object $(\vec{x_i})$ in the cluster j, and $v(\vec{x_i}, \vec{c_j})$ is the Euclidean distance between an object $\vec{x_i}$and a cluster center c_j. The degree of membership, given by the matrix $\vec{u}$, for objects on the edge of a cluster may have a lesser degree than objects in the center of a cluster. However, the sum of these coefficients for any given object x_i must be 1, i.e.,

$$\sum_{j=1}^{k}\ u_{ij}\ =\ 1\quad \forall i. \tag{2.2}$$

The centroid of a fuzzy cluster is the weighted average of all objects, where the weights of each object is its degree of membership to a cluster:

$$\vec{c_j}\ =\ \frac{\sum_{i=1}^{n}\ u_{ij}^{m}\ \vec{x_i}}{\sum_{i=1}^{n}.u_{ij}^{m}} \tag{2.3}$$

FCM is an iterative algorithm that terminates if

$$max\left(\left|u_{ij}^{t+1} - u_{ij}^{t}\right|\right) < \delta, \tag{2.4}$$

where δ is a termination criterion between 0 and 1, and t is the iteration step (Bezdek, 1981).

Rough Clustering: The notion of rough set was proposed by Pawlak (Pawlak, 1982).

For a universe or a finite ordinary set X and an equivalence (indiscernibility) relation $R \subseteq X \times X$, on X, the pair $A = (X, R)$ is called an approximation space. The equivalence relation R partitions the set X into disjoint subsets. Such a partition of the universe is denoted by: $X/R = E_1, E_2, ..., E_n$, where E_i is an equivalence class of R. If two elements $u, v \in X$ belong to the same equivalence class $E \subseteq X/R$, u and v are called indistinguishable. The equivalence classes of R are called the elementary or atomic sets in the approximation space $A = (X, R)$. The union of one or more elementary sets is called a composed set in A. The empty set $\emptyset$ is also considered as a special composed set. $Com(A)$ denotes the family of all composed sets. Since it is not possible to differentiate between the elements within the same equivalence class, one may not be able to obtain a precise representation for an arbitrary subset $Y \subseteq X$ in terms of elementary sets in A. Instead, any Y may be represented by its lower and upper approximations. The lower approximation $\underline{A}(Y)$ is the union of all the elementary sets which are subsets of Y, and the upper approximation $\overline{A}(Y)$ is the union of all the elementary sets which have a non-empty intersection with Y. The pair $\Big(\underline{A}(Y), \overline{A}(Y)\Big)$ is the representation of an ordinary set Y in the approximation space $A = (X, R)$, or simply the rough set of Y. The elements in the lower approximation of Y definitely belong to Y, while elements in the upper approximation of Y may or may not belong to Y.

Rough sets (Pawlak, 1984) enable us to represent such clusters using upper and lower bounds. Lingras (Lingras, 2001) described how a Rough Set theoretic classification scheme can be represented using a rough set genome. In subsequent publications (Lingras and West, 2004), (Lingras et al., 2004), a modification of the K-means approach and Kohonen Neural Network were proposed to create intervals of clusters based on rough set theory. All the

three approaches have been used successfully in clustering of web users. The K-means based and Neural Network approaches have also been used for clustering of supermarket customers. Rough Sets were proposed using equivalence relations by Pawlak (Pawlak, 1982). However, it is possible to define a pair of upper and lower bounds $\left(\underline{A}(X), \overline{A}(X)\right)$ or a Rough Set for every set $X \subseteq U$ as long as the properties specified by Pawlak (Pawlak, 1982, 1992) are satisfied (Yao et al., 1994; Polkowski and Skowron, 1996; Skowron and Stepaniuk, 1999). For a clustering scheme C based on a hypothetical relation R

$$X/R = C = \{c_1, c_2, \ldots, c_k\} \tag{2.5}$$

partitions the set X based on certain criteria. The actual values of c_i are not known. If, due to insufficient knowledge, it is not possible to precisely describe the sets $c_i, 1 \leq i \leq k$, in the partition, it is possible to define each set $c_i \in X/R$ using its lower $\underline{A}(c_i)$ and upper $\overline{A}(c_i)$ bounds based on the available information. We are considering the upper and lower bounds of only a few subsets of X. Therefore, it is not possible to verify all the properties of the Rough Sets (Pawlak, 1982, 1992). However, the family of upper and lower bounds of $c_i \in X/P$ are required to follow some of the basic Rough Set properties such as:

$(\mathrm{PR1})$ An object **x** can be part of at most one lower bound

$(\mathrm{PR2})$ $\mathbf{x} \in \underline{A}(c_i) \Longrightarrow \mathbf{x} \in \overline{A}(c_i)$

$(\mathrm{PR3})$ An object **x** is not part of any lower bound

$$\Updownarrow$$

x belongs to two or more upper bounds.

The rough K-means (Lingras and West, 2004) and its various extensions (Peters, 2006) have been found to be effective in distance-based clustering. A comparative study of crisp, rough and evolutionary clustering depicts how rough clustering outperforms crisp clustering (Joshi and Lingras, 2009). Peters, et al. (Peters et al., 2013) provide a good comparison of rough clustering and other conventional clustering algorithms.

2.1.1.3 N Clustering

The term bi-clustering was first introduced by Mirkin (1996) and then the appearance of tri and n-clustering followed shortly thereafter. The main idea behind this approach was Formal Concept Analysis (FCA) and clustering was done row-wise and columns-wise simultaneously to determine the intersected regions. Afterwards, a relaxation of the formal concept was introduced by the development of concept-based bi-clustering, or alternatively called Object-Attribute clustering, that reduces the time complexity for FCA. It ensures the resultant number of dense bi-clusters is no greater than the number of non-empty cells in the initial relation. This approach was used in a relationship set of firms and terms bought by the firms to recommend terms for a certain firm. Terms bought by other competitive firms were considered for this, where the terms included term(s) of the firm under consideration along with other term(s) (Ignatov et al., 2012). As an advancement of this method, an alternative of Tri-concept called Tri-clustering or Object Attribute Condition (OAC) tri-clustering was introduced. This concept-based clustering reduces computational time with respect to the traditional Formal Concept Analysis by representing a maximum cuboid full of crosses. Using this approach, the dense tri-clusters are formed consisting of triples formed by taking the triboxes of object, attribute and condition. To calculate the

triboxes, prime operator values are considered. The box operator set or value of an object consists of the objects found in the prime operator set of attribute and condition of the triple taken into account. The same process is employed to calculate box operator values of attributes and conditions as well. A tri-cluster is called dense if its density is greater than a minimal threshold (Ignatov et al., 2011). Recently a combination of bi-clustering and tri-clustering approaches to analyze data of a social network was deployed. In this approach, bi-clusters are extracted from two separate object-attribute tables. Bi-clusters with respect to their objects or extents are merged, taking their intersections. The attribute or intent of the first bi-cluster and intent of the second bi-cluster become the intent and modus (or condition) respectively of the newly formed tricluster (Gnatyshak et al., 2012). The conceptual tri-clusters are able to extract densest tri-clusters while spectral clustering can only extract lower dense tri-clusters that are difficult to analyze by human experts. On the other hand, the latter approach is two times faster than the former one. In comparison with TRIAS, which determines absolutely densed tri-clusters, it has been found that TRIAS is the most time consuming algorithm that results in easily interpretable tri-clusters, though the general structure of larger context is not easy to understand (Ignatov et al., 2013). In other research based on a comparative study, it was observed that Dense Prime OAC-tri-clustering and TriBox are good alternatives for the Tri-clustering analysis approach since the total number of tri-clusters for a real data example is considerably less than the number of tri-concepts (Gnatyshak et al., 2013).

2.1.1.4 Ensemble Clustering

Though there are number of clustering algorithms available, no single clustering algorithm is capable of delivering sound solutions for all data sets. Combining cluster results or creating cluster ensembles is an alternate approach to improve the quality of different individual clustering results. For a given set of objects, a method of cluster ensemble works in two major steps; namely, Generation and Consensus. In the Generation stage, a population of diverse multiple clustering partitions are made by generative mechanisms using different feature subsets, clustering algorithms, parameter initialization and projection to subspaces or subsets of objects. In the Consensus stage, partitions are aggregated based on objects co-occurrence using relabeling and voting, co-association matrix, graph and hyper-graph, etc. In addition, Median Partition using Genetic Algorithms, Kernel Methods etc. or Probabilistic Models can also be used for consensus selection (Ghosh and Acharya, 2011). The expected properties of an ensemble clustering result are robustness, consistency, novelty and stability (Vega-Pons and Ruiz-Shulcloper, 2011; Strehl and Ghosh, 2003; Gao et al., 2013).

To define Ensemble clustering more formally, let X be a set of n objects positioned in a m-dimensional space, and P be a set of n partitions of objects in X. Thus $P = \{p_1, p_2, ..., p_n\}$. Each partition in P is a set of disjoint and nonempty clusters $p_i = \{L_i^1, L_i^2, ..., L_i^{K(i)}\}$, $X = \{L_i^1 U L_i^2 U ... U L_i^{K(i)}\}$, and for any $p_i, K(i)$ is the number of clusters in the i-th clustering partition. The problem of clustering ensemble is to find a new partition $E = \{C_1, C_2, ..., C_K\}$ of data X, given the partitions in P, such that the final clustering solution is better than any individual clustering partition (Gao et al., 2013). A clustering C maximizing Summation of Normalized Mutual Information SNMI, maximizes the infor-

mation it shares with all the clusterings in the ensemble. Thus it can be considered to be the best one having general trend in the ensemble (Fern and Lin, 2008).

2.1.2 Meta Clustering

The literature review in this category represents the recent research on meta clustering. Reviews described here provide an idea of the meta clustering itself, research trends in meta clustering considering the application areas covered by those respective research studies, the techniques observed to implement meta clustering and also the superiority of the techniques over other related methods as discussed in the respective work.

A two stage clustering technique via the Information Bottleneck method was introduced by Slonim and Tishby (2000) where word clusters were used for document clustering. In the first stage, word clusters were formed using mutual information of words for the documents. Afterwards, documents were clustered by considering mutual information of documents for the word clusters. An iterative version of this method was proposed by El-Yaniv and Souroujon (2001) that works better than the document clustering especially in noisy settings and it shows competitive performance, even when applied to unsupervised text categorization. Castellano et al. (2002) used a double clustering technique by inducing information granules in the space of numerical data using the FCM algorithm and the prototypes obtained in this way were further clustered for each dimension using hierarchical clustering.

Meta clustering even helps to decide the best clustering when the clustering that is best depends on how the clusters will be used; not on a pre-specified clustering criterion that most clustering techniques prescribe. Since, an appropriate clustering criterion cannot be

defined in advance, alternate reasonable clusters can be made by meta clustering that allows the users to select the one that best fits their needs. It generates the base level clusters, groups the clusters according to their similarities and presents them to users by organizing them in a meta level (Caruana et al., 2006).

The technique of meta clustering can be implemented in areas other than document classification as illustrated by Ramirez-Cano et al. (2010). They used a meta clustering approach to classify players from gameplay data. The process works in three levels. In the first level, partitional clustering using K-means is done on the attribute action/skill. Then in the next level similarity based clusters are made considering preferences of players where Rubner's Earth Mover's Distance is used as a similarity metric of the players' game world exploration and in the third level, clusters are made depending upon the players social relations among themselves using the Multidimensional Scaling Technique as a visualization tool (Ramirez-Cano et al., 2010).

2.1.3 Recursive Meta-clustering

This part of literature review is based on research on Recursive Meta-clustering. In addition to the application area, it specifies the key procedure and basic algorithms used to implement this proposed technique.

The information granule used in Recursive Meta-clustering algorithm is represented by static and dynamic portions where the static ones are created from the information directly related to the candidate record and the dynamic part is created by considering the information of the record that is related to the record of the static one. The dynamic portion is derived recursively from the clustering process until some predefined criteria is satisfied.

The process is effective in creating profiles for user who are connected to other users in a network environment (Lingras and Rathinavel, 2012). Though Lingras and Rathinavel used crisp clustering using K-means for this algorithm previously, the concept was extended later by using fuzzy clustering by means of Fuzzy C-means; as a consequence, a fuzzy meta-clustering algorithm was proposed. In this process, the representation of a granule is updated recursively using the cluster memberships of other connected granules obtained in the previous step of clustering. As a consequence, the problem of assigning member objects forcefully to one cluster, due to incomplete information or information vagueness can be overcome. Crisp clusters do not always exist in real world applications (Rathinavel and Lingras, 2013). The K-means version of Recursive Meta-clustering was used in the information granules of customers and products of a retail store data set and information granules of businesses and reviewers of a business rating web site by Lingras et al. (2014) and Triff and Lingras (2013) respectively.

2.1.4 Time Series / Temporal Data Mining

The papers reviewed in this section are about mining Time Series or Temporal Data. It incorporates literature and surveys on Time Series Data Mining in section 2.4.1 and reviews research on Temporal Mining in diverse fields using a range of algorithms in section 2.4.2.

2.1.4.1 Surveys on Time Series Data Mining

Time series is a collection of observations made sequentially through time. Such series are widely used in many areas including finance, economics, social science and environmental science. Examples of time series in finance and economics are daily stock market,

sales figures, unemployment rates etc. Time series analysis of such data can be useful for prediction and estimates of future values based on the historic data. In addition to these observations, demonstration on various aspects of time series analysis along with definitions of related notions and their applications can be found in the study by Chatfield (2013). An example of time series is shown in Figure 2.1.

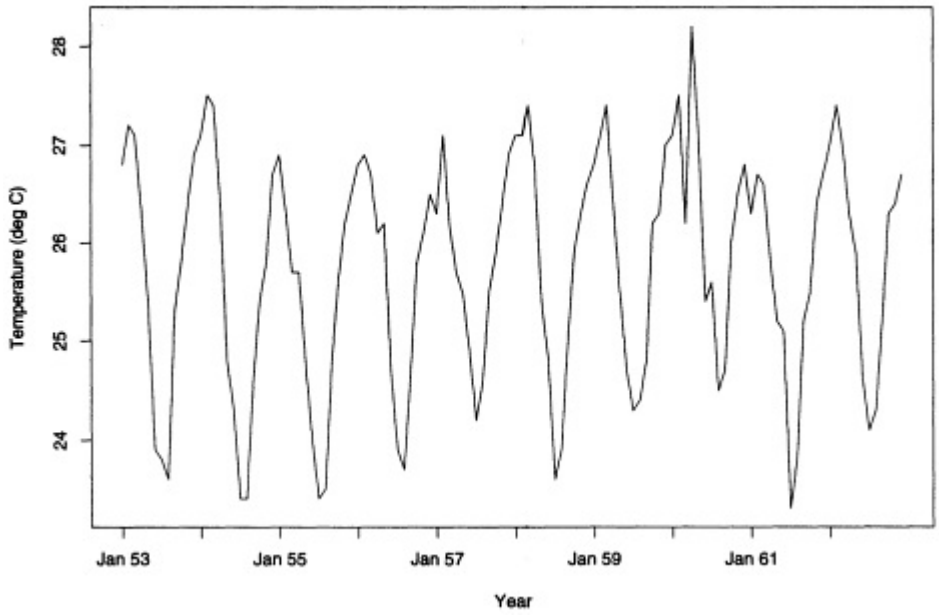


Figure 2.1: Average Air Temperature at Recife, Brazil from 1953 to 1962 (Chatfield, 2013)

Liao (2005) provided an overview of previous work that investigated clustering of time series data from various applications. To present the basics of time series data, he described several commonly used general purpose clustering algorithms, similarity and dissimilarity measures and evaluation criterion. As well, he described clustering approaches considering three categories, namely raw-data-based, feature-based and model-based.

Time Series Clustering can be partitional and hierarchical. Hierarchical Clustering can be done in top-down or bottom up fashion by computing pair wise distance and splitting or

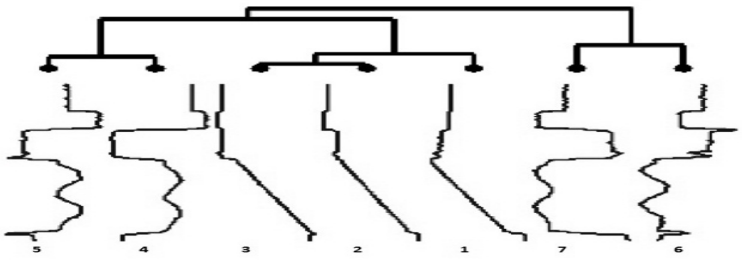


Figure 2.2: A Hierarchical Clustering of Time Series (Ratanamahatana et al., 2010)

merging clusters respectively as shown in Figure 2.2. However, implementation of Hierarchical Clustering is limited to short data sets due to its quadratic computational complexity. On the other hand, Partitional Clustering typically uses the K-means clustering algorithm or some variant to optimize the objective function by minimizing the sum of squared intra-cluster errors. This algorithm perhaps is the most commonly used clustering algorithm, though one of its shortcomings is the number of clusters must be prespecified (Ratanamahatana et al., 2010). While describing different similarity measures, Ratanamahatana et al. mentioned Euclidean distance as the simplest and easiest to compute method; however, the major problem with this method is it does not allow a situation where two sequences are alike but stretched or compressed. Normalization may resolve this problem.

An overall review of the previous and contemporary research on time series data mining can be found in (Fu, 2011). This study can help interested researchers to proceed further in this area. Major recent research directions that are identified here are multi-attribute time series, time series data streams and privacy issues. Furthermore, a more effective and efficient representation scheme for high dimensional time series data including whole sequence and subsequence matching of variant lengths is defined as a fundamental significant issue (Fu, 2011; Ratanamahatana et al., 2010). Esling and Agon (2012) provided a survey of the

techniques applied for time series analysis. This survey also describes the tasks relevant to current trends of time series analysis. The existing work presented there are with respect to three common aspects of time series analysis namely, representation techniques, distance measure and indexing methods. Moreover, various research trends on this area have also been summarized there. This article can be treated as a dictionary as well as a point of reference for researchers who are interested in working with time series.

2.1.4.2 Research on Time Series Data Mining

The following literature reviews are based on research on Time Series Analysis that incorporates mostly Temporal Clustering which refers to factorizing of multiple time series into non-overlapping segments that belong to k temporal clusters (Hoai and la Torre, 2012).

An adaptive method for the discovery of local temporal patterns instead of the use of traditional global models was proposed by Das et al. (1998), where the author considered the problem of finding rules relating patterns in a time series to other patterns of that series. Using vector quantization, sliding windows are formed and then clusters are made on them. Then rules are obtained using simple rule finding methods. Daily stock prices, hourly volumes of telephone calls and daily sea-surface temperatures were used to experiment with the proposed idea. Povinelli and Feng (1999) proposed a method that is capable of handling nonstationary, nonperiodic, irregular time series, including chaotic deterministic time series. This research used the idea of temporal patterns from wavelets. Wijk and Selow (1999) addressed the problem of identifying patterns and trends on multiple time scales (e.g. days, weeks, seasons) simultaneously and propose a solution by clustering the similar daily data patterns. The average patterns are represented as graphs and the corresponding days

are shown on a calendar. Afterwards, Viovy considered the unsuitability of the hierarchical ascending approach for large data sets, and lack of nonsupervisory characteristics as well as slow execution time of K-means algorithm into account. To resolve these drawbacks, he proposed a recursive descending hierarchical method that starts with a meta-cluster of the whole data set and in each iteration, the cluster splits into two distinct meta-clusters if the parent cluster does not hold group of homogeneous points. This process continues until no other split is possible (Viovy, 2000). Kohonen Neural Network or Self Organizing Map (SOM) was used by Fu et al. to discover patterns from a stock time series. They used this popular and superior technique due to its clustering performance. To prepare for the SOM, data sequences were segmented using a continuous sliding window. Afterwards, similar temporal patterns were grouped together. To address the problem of exponentially increasing time needed for the discovery process due to increasing number of data points of patterns, the input patterns were compressed by using Perceptually Important Point (PIP) identification algorithm (Fu et al., 2001).

Incorporating soft clustering algorithms to analyze temporal data is another dimension in temporal data mining as showed by (Liu and George, 2003). An unsupervised fuzzy clustering algorithm based on fuzzy K-means was used in this work for analyzing spatio-temporal data of earth science database. A cluster validity index to determine the optimal number of clusters was also proposed. The specialization of the proposed algorithm is that the clusters found using fuzzy K-Means are merged to satisfy a predefined correlation criteria among the centroids and this process continues till two clusters are left (Liu and George, 2003). Later on, Yan used rough clustering along with fuzzy clustering in web usage and supermarket data sets. In the case of supermarket data sets intervals were clustered using

a 26-week period data. To ensure the clustering is not biased by any specific criteria, a weighting scheme was used. It was shown that rough and fuzzy clusters are more subtle and correct than K-means clustering since these two techniques are more accurate considering the slight differences among the clusters (Yan, 2004). Corduas and Piccolo showed that whatever method among the clustering techniques is chosen, the interpretation result is confined to a descriptive level. On the other hand, the Autoregression (AR) metric instead allows the composition of time series within a testing hypothesis structure and gives a meaningful way to assess their nearness. In addition, it is a tool in itself that can construct time series cluster (Corduas and Piccolo, 2008).

Kremer et al. combined a number of different novel time series clustering and cluster tracing approaches to detect cluster similarities. For this, they considered periodic appearing and disappearing clusters, merged and new patterns, overall changing patterns; even where there is no time series in common. It has been found that while applying sliding window and clustering obtained from shorter time series all together, the temporal position of cluster was not considered. To resolve this, interval information was combined in subsequent clustering. In addition, to detect same pattern stretched by different factors, adaptable distance measure as Time Warping was combined with the clustering algorithm (Kremer et al., 2010). A two stage algorithm called periodica was used by Li et al. (2010) to determine the periodic behavior for moving objects. In the first step they detected periods using Fourier Transform and autocorrelation and in the next step summarized the periodic behaviors using hierarchical clustering.

Yang and Leskovec described an algorithm called K-Spectral Centroid (K-SC) clustering algorithm to uncover the temporal dynamics of online content. The proposed algorithm

outperformed the K-means clustering algorithm in finding distinct shapes of time series. The prescribed method works iteratively like the K-means algorithm except that to calculate the centroid it considers similarity among time series by scaling instead of just calculating the mean of the members of the cluster. In this technique, the time series that share the same shape are treated as the members of the same cluster (Yang and Leskovec, 2011). Detecting Shape similarity is one of the three major objectives of time series clustering namely similarity in time, similarity in shape and similarity in change. For shape similarity-based clustering, Nearest Neighbor Network can be used. For this, the Nearest Neighbor Network that has a node of one time series object and link of neighbor relationship between nodes ts created. The nodes with higher degree of neighbors are chosen to cluster using dynamic time warping distance function and hierarchical clustering algorithm (Zhang et al., 2011).

A bottom up hierarchical clustering algorithm was proposed by Sravya and Sri that uses K-means algorithm to cluster. The process starts with initial clustering and reconcile those partitions to candidate consensus partitions and proceed further until an agreement function in not satisfied. The research aimed to construct statistical models to describe the characteristics of each group of data (Sravya and Sri, 2013).

2.1.5 Financial Time Series

The research reviews in this part of this thesis incorporate ideas on financial temporal data emphasizing ideas on mining financial time series and stock price volatility of historical financial data.

As described by Chatfield, a financial time series is a series routinely recorded in economics and finance (Chatfield, 2013). Figure 2.3 shows an example of a financial time

series.

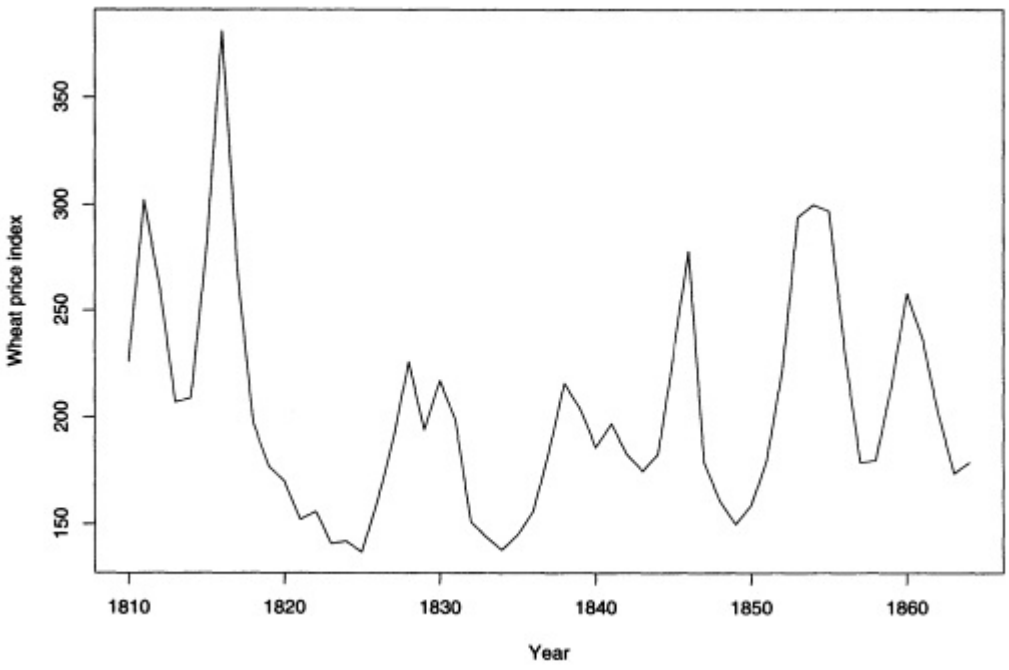


Figure 2.3: The Beveridge Wheat Price Annual Index Series from 1810 to 1864 (Chatfield, 2013)

Though the techniques applicable to any other time series mining can be applicable to financial time series as well, I have tried to study few specific papers on financial temporal data clustering.

Fu et al. used the Kohonen Self Organizing Map for the fragments in the sliding windows to group financial temporal patterns. They used stock time series collected from a Hong Kong stock market to extract stock patterns (Fu et al., 2001). Coronnello et al. adopted hierarchical clustering on financial time series collected from a London stock market to portfolio the stocks traded. They investigated the time series from a daily time interval as well as 5-minute time intervals. It showed differences detected in the structure

of the correlation matrix of high frequency and daily returns. Depending upon the results of a comparative study, they concluded that the application of just a single method cannot extract all economic information that exists in the correlation coefficient matrix of a stock portfolio (Coronnello et al., 2005).

Volatility measure using historical data is a common scenario in identifying stock price pattern. The most familiar, frequently used and easier than any other historical variation estimator is the volatility measure generalized from Black Scholes option valuation equation. The standard historical volatility estimate, according to the Black Scholes formula, represents volatility by calculating the square root of the result obtained by multiplying the variance of the log price relative to the number of observations, where the variance is the summation of the squares of differences between the relative log prices and their mean (Figlewski, 1994; Karoui et al., 1998; Grullon et al., 2012).

2.2 Chapter Summary and Conclusions

This chapter represented a detailed literature study relevant to the thesis ”Recursive Temporal Meta-cluster of Daily Time Series”. According to the research studies conducted for this literature review, two major types of clustering are commonly used. These are crisp and soft clustering. In addition, N clustering and ensemble clustering are other types of clustering. Meta-clustering creates clusters from information about clusters. On the other hand, Recursive Meta-clustering creates clusters from dynamic information obtained by the last created clusters and from static information. A number of articles have been found that encompass these techniques using different types of basic methods and extensions of these methods. Those are experimented with various types of data structures.

A good deal of research is available that discuses time series analysis with a hierarchical or partitioned approach applied to a whole or a subsequence of a time series. Those studies observed advantages of the proposed algorithm over others considered for comparison. In addition, the research papers documenting the overall time series mining trend are truly useful for understanding the definitions, techniques and contributions in this field.

From the articles on Temporal Meta-clustering it is found that K-means, fuzzy K-means and rough clustering are commonly used in this area though other techniques can be used as well, with various advantages and drawbacks.

Finally the reviews covered in the area of Financial Time Series give the ideas on the theories, notions and nature of financial temporal data, specially stock volatility and its calculation using the popular Black Scholes formula. This will help to implement my research thought using Financial Time Series data.

However, the review of this literature also provide an indication of the fact that there is much research on time series meta-clustering whereas the Recursive Meta Clustering of daily series data is indeed a new dimension to the trend. According to the previous research results, Recursive Meta-clustering can represent interesting profiles of the data set objects. Thus, the implementation of this technique on time series data leaves a chance to have competent outcomes accordingly and therefore may open a new door to research.

Chapter 3

Preparation of Data

This chapter provides some idea of the theory and procedures required to prepare the data set for our experiments, with a sample illustration. Section 3.1 specifies the significance and definition of volatility and two different ways to represent volatility. Section 3.2 describes the data set used for experimentation. Conversion of the data set to Percentiles and Black Scholes Volatility are described in Section 3.3 and 3.4 respectively. On the other hand, Section 3.5 describes how the data set is prepared for Recursive Meta-clustering. Section 3.6 is a brief overview of the chapter along with concluding remarks.

3.1 Volatility

Volatility is a measure on variation of price of a financial instrument over time. In financial data series, volatility is an important indicator used by traders to measure the risk of investment. The fluctuation in prices creates trading opportunities. Historical volatility measured by means of Black Scholes model is a one of the various types of volatility

measures. Percentiles of the prices may represent the fluctuation as well.

3.1.1 Black Scholes Volatilty Index

The equation of Black Scholes volatility index is an extension of the Noble Prize-winning Black Scholes model which estimates the price of the option over time. This model is widely used by options market participants. The key idea behind the model is to hedge the option by buying and selling the underlying asset in just the right way and, as a consequence, to eliminate risk. The instantaneous log returns of the stock price considered in this formula is an infinitesimal random walk with drift, or more precisely, it is a geometric Brownian motion. The equation to estimate volatility using this model is:

$$Volatility = \sqrt{LogPriceRelativeVariance \times (Observations - 1)}, \qquad (3.1)$$

where $LogPriceRelativeVariance = \sum (LogPriceRelative - Mean)^2$. The Black Scholes index is a one dimensional concise measure that represents volatility for the day. Observations can be any number of time intervals for which a historical volatility is represented by the formula. Intervals represent equally distant time periods, which can be in any unit including months, days, hours, minutes, etc. In this thesis, we use intervals of 10 minutes for each day.

3.1.2 Percentile

Distribution of prices during the day can provide a more elaborate description of price fluctuations. Percentile calculations may represent such a distribution. For this research we

propose the use of five percentile values, 10%, 25%, 50%, 75% and 90%, to represent the price distribution. 10% of the prices are below the 10th percentile value, 25% of the prices are below the 25th percentile value and so on.

3.2 The Data Set

Our data set contains open prices and average prices at 10 minute interval for 223 instruments transacted on 121 days comprising a total of 27,012 records. Each daily pattern has 39 intervals. A section of the data set is shown in Table 3.1

Instrument	Open Price	avgp0	avgp1	avgp2	avgp3		avgp36	avgp37	avgp38
3_1	329	1	1	0.99696	0.99696		1.00304	1.00304	1.00304
A_103	1788	1	1.002237	1.002796	1.002796		1.027405	1.027405	1.027405
A_10	7925	1	1	1	1.000883		0.997603	0.997603	0.997603
A_113	554	1	0.998195	0.998195	0.998195		0.990975	0.990975	0.990975
A_116	58358	1	1.000171	1.000206	1.000206		0.999949	0.999949	0.999949
A_12	16760	1	1.000239	1.001671	1.001969		1.003222	1.003222	1.003222
A_131	2268	1	1.001764	1.001323	1.001764		0.997795	0.997795	0.997795
A_141	21349	1	0.999204	0.999016	0.999016		1.001593	1.001639	1.001639
A_14	23589	1	1.001823	1.003307	1.00479		1.011912	1.011912	1.011912
A_18	8252	1	1.000364	1.000121	1	...	1.01636	1.01636	1.01636
A_19	2936	1	1.002044	1.003406	1.003747		1.00579	1.00579	1.00579
A_20	1766	1	0.998867	0.998301	0.998301		0.983579	0.983579	0.983579
A_43	3036	1	0.999671	0.999671	0.999671		1.016469	1.016469	1.016469
A_53	402	1	1	1	0.997512		0.997512	0.997512	0.997512
A_64	3238	1	1.000926	1.001235	1.004015		1.004941	1.004941	1.004941
A_6	8887	1	0.998987	0.998875	0.998087		0.999325	0.999325	0.999325
A_71	2080	1	0.997596	0.997115	0.996635		0.987019	0.987019	0.987019
A_7	15940	1	0.999875	0.999686	0.999812		0.991343	0.991343	0.991343
A_84	7253	1	0.99545	0.994899	0.994071		0.986764	0.986764	0.986764
A_89	1396	1	1.000716	1.000716	1.000716		0.998567	0.997851	0.997851
A_95	3574	1	0.999161	0.998601	0.998881		1.002238	1.002238	1.002238

Table 3.1: Sample Section of Data Set

3.3 Conversion to Percentile

The data set described in the previous section is used to create a five dimensional pattern. The pattern represents 10, 25, 50, 75 and 90 percentile values of the prices. The prices are normalized by the opening price so that a commodity selling for $100 has the same pattern as the one that is selling for $10. Afterwards, the natural logarithm of the five percentiles are calculated as shown in Table. 3.2.

Percentile	10%	25%	50%	75%	90%
Percentile of avgp (avgpPerc)	0.9841346	0.9873798	0.9927885	0.9951923	0.9966346
ln(avgpPerc)	-0.0159926	-0.012701	-0.0072376	-0.0048193	-0.0033711

Table 3.2: Calculation of Percentiles for the Sample Record of Figure. 3.4

Table 3.3 shows the converted percentile representation of the sample section of the original data set shown in Table 3.1

Day:Instrument	10p	25p	50p	75p	90p
2012-01-25:Z_2	0	0.002707	0.005821	0.008099	0.033301
2012-01-27:3_1	0	0.003044	0.003044	0.003044	0.006079
2012-01-27:A_103	0	0.003619	0.011093	0.014946	0.021627
2012-01-27:A_10	0	0.001188	0.004089	0.00497	0.005976
2012-01-27:A_113	0	0.003636	0.00545	0.00545	0.00726
2012-01-27:A_116	0	0.000226	0.000329	0.000552	0.000606
2012-01-27:A_12	0	0.001237	0.002544	0.003345	0.004678
2012-01-27:A_131	0	0.000442	0.000442	0.003088	0.004057
2012-01-27:A_141	0	0.00022	0.000759	0.001484	0.001984
2012-01-27:A_14	0	0.005291	0.006968	0.007051	0.007353
2012-01-27:A_18	0	0.000364	0.001455	0.013795	0.017032
2012-01-27:A_19	0	0.001356	0.002034	0.003557	0.004064
2012-01-27:A_20	0	0.005964	0.013369	0.013937	0.014504
2012-01-27:A_43	0	0.004419	0.004909	0.007355	0.010833
2012-01-27:A_53	0	0.002497	0.002497	0.002497	0.002497
2012-01-27:A_64	0	0.000952	0.00172	0.003253	0.003253
2012-01-27:A_6	0	0.000453	0.003841	0.004291	0.00508
2012-01-27:A_71	0	0	0.001946	0.008971	0.009309
2012-01-27:A_7	0	0.000222	0.001396	0.007307	0.011403
2012-01-27:A_84	0	0.002886	0.004068	0.00525	0.006457
2012-01-27:A_89	0	0.001434	0.00215	0.002865	0.005722
2012-01-27:A_95	0	0.000982	0.002243	0.005318	0.005598

Table 3.3: Percentile Representation of a Sample Section of Data Set

3.4 Conversion to Black Scholes Volatility Index

The original data set containing 39 intervals are converted to the second representation of volatility. It represents the one dimensional Black Scholes volatility for the day. Calculation of Black Scholes Volatility is shown in Table 3.4 for a given instrument.

	Sample Record	Intervals	ActualPrice	Xi =ln(Pt/Pt-1)	Mean=∑Xi/T T=Total No of Intervals	MSD =sqr(Xi-Mean)	SMSD = ∑MSD	Daily Volatility =sqrt(SMSD*[T-1])
	832							
avgp0	1	1	832	0	-0.00043513	1.89E-07	1.32E-05	0.0005891
avgp1	0.998798	2	831	-0.00120265		5.89E-07		
avgp2	0.998798	3	831	0		1.89E-07		
avgp3	0.997596	4	830	-0.00120409		5.91E-07		
avgp4	0.996394	5	829	-0.00120555		5.94E-07		
avgp5	0.996394	6	829	0		1.89E-07		
avgp6	0.996394	7	829	0		1.89E-07		
avgp7	0.995192	8	828	-0.001207		5.96E-07		
avgp8	0.995192	9	828	0		1.89E-07		
avgp9	0.995192	10	828	0		1.89E-07		
avgp10	0.995192	11	828	0		1.89E-07		
avgp11	0.993990	12	827	-0.00120846		5.98E-07		
avgp12	0.992788	13	826	-0.00120992		6.00E-07		
avgp13	0.992788	14	826	0		1.89E-07		
avgp14	0.992788	15	826	0		1.89E-07		
avgp15	0.992788	16	826	0		1.89E-07		
avgp16	0.992788	17	826	0		1.89E-07		
avgp17	0.992788	18	826	0		1.89E-07		
avgp18	0.992788	19	826	0		1.89E-07		
avgp19	0.992788	20	826	0		1.89E-07		
avgp20	0.992788	21	826	0		1.89E-07		
avgp21	0.991586	22	825	-0.00121139		6.03E-07		
avgp22	0.991586	23	825	0		1.89E-07		
avgp23	0.990384	24	824	-0.00121286		6.05E-07		
avgp24	0.989183	25	823	-0.00121433		6.07E-07		
avgp25	0.987980	26	822	-0.00121581		6.09E-07		
avgp26	0.987980	27	822	0		1.89E-07		
avgp27	0.987980	28	822	0		1.89E-07		
avgp28	0.987980	29	822	0		1.89E-07		
avgp29	0.986778	30	821	-0.00121729		6.12E-07		
avgp30	0.986778	31	821	0		1.89E-07		
avgp31	0.985576	32	820	-0.00121877		6.14E-07		
avgp32	0.985576	33	820	0		1.89E-07		
avgp33	0.98437	34	819	-0.00122026		6.16E-07		
avgp34	0.98437	35	819	0		1.89E-07		
avgp35	0.983173	36	818	-0.00122175		6.19E-07		
avgp36	0.983173	37	818	0		1.89E-07		
avgp37	0.983173	38	818	0		1.89E-07		
avgp38	0.983173	39	818	0		1.89E-07		

Table 3.4: Calculation of Daily Volatility for a Given Instrument

Table 3.5 shows the converted daily volatility representation of the sample section of the original data set shown in Table 3.1

Day:Instrument	DailyVol
2012-01-25:Z_2	0.002779
2012-01-27:3_1	0.001101
2012-01-27:A_103	0.001041
2012-01-27:A_10	0.000831
2012-01-27:A_113	0.000741
2012-01-27:A_116	6.64E-05
2012-01-27:A_12	0.000497
2012-01-27:A_131	0.000419
2012-01-27:A_141	0.000365
2012-01-27:A_14	0.000555
2012-01-27:A_18	0.001
2012-01-27:A_19	0.000485
2012-01-27:A_20	0.000637
2012-01-27:A_43	0.000931
2012-01-27:A_53	0.000698
2012-01-27:A_64	0.000622
2012-01-27:A_6	0.000714
2012-01-27:A_71	0.00072
2012-01-27:A_7	0.000703
2012-01-27:A_84	0.000747
2012-01-27:A_89	0.000553
2012-01-27:A_95	0.000808

Table 3.5: Daily Volatility Representation of Sample Section of Data Set

3.5 Data Set for Recursive Temporal Meta-cluster

The percentile data set and one dimensional daily volatility data set prepared previously are used here in two different ways. Since the natural logarithm values used in these two sets are very small, they are multiplied by 100 and one more set for each of these two sets

are prepared. This weighting ensures that the small values of static part of the data set are not dominated by the large values of dynamic part of the data set. However, during our experiments all of four sets of data are used and the results obtained are analyzed.

For our experiment, we predefine 10 as the value of m. In other words, we consider 10 as the period for which historical profile has to be determined. Therefore m =10 is the number of days for the dynamic part. Thus 121-10 = 111 days are used for static part, though the data set contains fewer transactions for some instruments.

The actual data set is checked to filter out the instruments that have transactions of less than m days. Instruments *A-19* having 9 days transaction and *H-26* having 5 days transaction fall in this category. Therefore, these transactions have been removed. As a consequence, a total of 27012 - (9 + 5) = 26,998 records are used for the Recursive Meta-cluster.

3.6 Chapter Summary and Conclusions

Our data set for the experiment is a financial temporal set for which some preprocessing has already been already. This data set of average prices of commodities at 10 minutes interval points is further processed as described in this chapter to make it usable in our experiment. This chapter described the overall processing procedure with examples. In addition, we have seen that volatility is an important measure to guide investment strategies. The percentile and Black Scholes volatility index can be used to represent volatility.

The data set prepared at this stage is used as the input to our experiment. We have targeted the Percentile and Black Scholes daily volatility data sets prepared here to use in

two phases to create Rough Ensemble clusters and Temporal Meta-clusters as described in the next two chapters.

Chapter 4

Rough Ensemble clustering

This chapter presents the procedure Rough Ensemble clustering and the associated experimental results. Section 4.1 demonstrates first level clustering process and results, whereas Sections 4.2, 4.3 and 4.4 present a comparative analytical study of the two sets of cluster results, the algorithm for Rough Ensemble clustering and the process of formulating the proposed Rough Ensemble clustering, respectively. In addition Sections 4.5 and 4.6 describe the value of the proposed technique and a comparative study with other relevant methods. Sectiona 4.7 describes computational aspect of the algorithm. The last section is a brief summary and conclusion based on the experiments described in this chapter.

4.1 Initial Ordered Clustering

4.1.1 Optimal Number of Clusters

The data set representing percentiles and the set representing the Black Scholes daily volatility are clustered 10 times using the K-means algorithm for the initial ordered clustering. Cluster profiles for the percentile and daily volatility data set are shown in Figure 4.1 and 4.2 respectively. The optimal number of clusters is found by plotting the Davies-Bouldin (DB) index which is the ratio of within cluster scatter and between cluster distance and also plotting totWithinss which is the distance among the objects in a cluster. The DB index is shown in Figure 4.3 and the scatter in the clusters or withinss is shown in Figure 4.4. The aim is to use the number of clusters corresponding to the lowest DB index and the knee of the curve of the cluster scatter. Based on these two criteria, we chose five as a reasonable number of clusters.

Sizes

	Clusters: 2	Clusters: 3	Clusters: 4	Clusters: 5	Clusters: 6	Clusters: 7	Clusters: 8	Clusters: 9	Clusters:10
cp1	3981	1022	2450	817	4024	850	8245	15	2037
cp2	23031	7118	8367	3349	12405	8293	4536	1196	2090
cp3		18872	15894	14125	21	4836	753	1224	70
cp4			301	8676	8737	17	10320	2543	600
cp5				45	1492	2300	1309	564	13
cp6					333	10526	1675	134	3798
cp7						190	159	5278	9365
cp8							15	8387	7885
cp9								7671	326
cp10									828

Withinss

	Clusters: 2	Clusters: 3	Clusters: 4	Clusters: 5	Clusters: 6	Clusters: 7	Clusters: 8	Clusters: 9	Clusters:10
cp1	1.851976	0.9549265	0.3806406	0.3343017	0.2131535	0.180283	0.112668	0.067348	0.070325
cp2	0.9952046	0.5645833	0.3421071	0.2888087	0.1008832	0.113676	0.105665	0.086084	0.064139
cp3		0.4072694	0.218766	0.1491897	0.09377899	0.159615	0.173573	0.078752	0.067749
cp4			0.5442072	0.2425405	0.174991	0.077502	0.059314	0.06249	0.069489
cp5				0.1884424	0.2159459	0.169048	0.090062	0.15839	0.058783
cp6					0.2219919	0.062784	0.088323	0.119202	0.055564
cp7						0.154461	0.142447	0.101576	0.045748
cp8							0.067348	0.034277	0.083062
cp9								0.067857	0.133972
cp10									0.071607

Betweenss

Clusters: 2	Clusters: 3	Clusters: 4	Clusters: 5	Clusters: 6	Clusters: 7	Clusters: 8	Clusters: 9	Clusters: 10
2.774227	3.694628	4.135687	4.418124	4.600663	4.70404	4.782008	4.845432	4.900969

Figure 4.1: Cluster Profiles of Percentile Data Set

Sizes

	Clusters: 2	Clusters: 3	Clusters: 4	Clusters: 5	Clusters: 6	Clusters:7	Clusters: 8	Clusters: 9	Clusters:10
cdv1	24569	20783	7914	14182	3881	10024	154	7751	367
cdv2	2443	418	17471	684	1077	2089	1385	7211	3471
cdv3		5811	1457	95	211	33	8083	3430	139
cdv4			170	3061	9172	8863	32	5597	18
cdv5				8990	36	716	5590	1746	7678
cdv6					12635	5113	8137	30	7119
cdv7						174	3081	318	58
cdv8							550	794	1783
cdv9								135	819
cdv10									5560

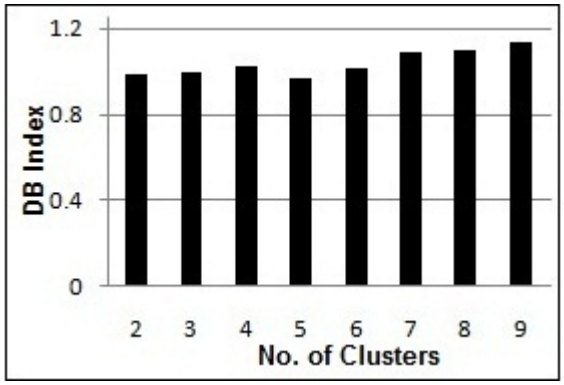
Withinss

	Clusters: 2	Clusters: 3	Clusters: 4	Clusters: 5	Clusters: 6	Clusters:7	Clusters: 8	Clusters: 9	Clusters:10
cdv1	0.002625044	0.001069	0.000512	0.000281	0.000191	0.00011	9.67E-05	4.43E-05	3.40E-05
cdv2	0.003708734	0.00133	0.000546	0.000309	0.000187	0.000106	7.19E-05	5.06E-05	4.08E-05
cdv3		0.001119	0.000444	0.000388	0.000149	0.000123	6.55E-05	4.31E-05	2.93E-05
cdv4			0.000645	0.000261	0.000185	9.88E-05	0.000119	4.37E-05	5.41E-05
cdv5				0.000254	0.000137	0.00012	5.99E-05	4.22E-05	4.26E-05
cdv6					0.000201	0.000105	5.67E-05	0.00011	4.91E-05
cdv7						0.000116	6.22E-05	4.64E-05	3.66E-05
cdv8							9.22E-05	4.40E-05	3.74E-05
cdv9								8.34E-05	3.60E-05
cdv10									4.11E-05

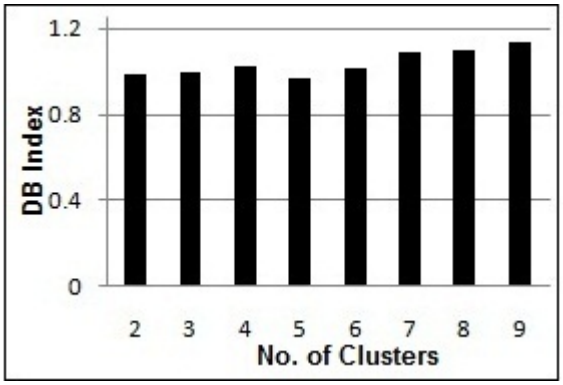
Betweenss

Clusters: 2	Clusters: 3	Clusters: 4	Clusters: 5	Clusters: 6	Clusters: 7	Clusters: 8	Clusters: 9	Clusters: 10
0.006844	0.009659487	0.011031	0.011684	0.012126	0.012399	0.012554	0.01267	0.012776

Figure 4.2: Cluster Profiles of Daily Volatility Data Set

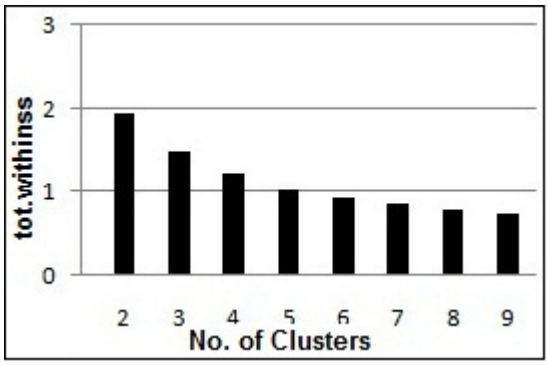


(a) Percentile

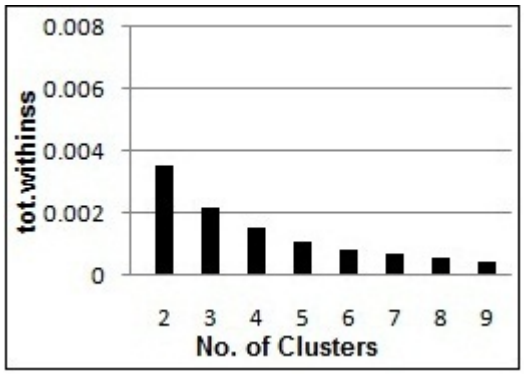


(b) Daily Volatility

Figure 4.3: DB Index



(a) Percentile



(b) Daily Volatility

Figure 4.4: Cluster Scatter

4.1.2 Cluster Ranking

According to the centroids obtained, the clusters are ranked. Figures 4.5 and 4.6 show the plots of centroids for the two clustering schemes. There is a clear ranking of the clusters in terms of the volatility. For example, the top line *cpr5* of percentile values is the most volatile, while the one at the bottom, that is *cpr1* is the least volatile. Similarly, the highest values of Black Scholes volatility indicate high volatility.

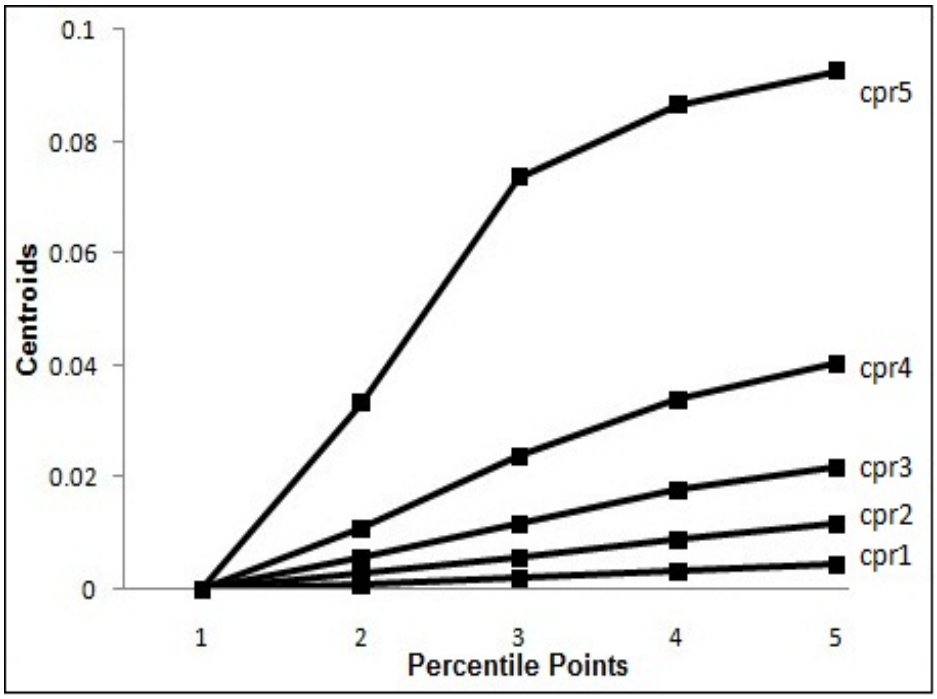


Figure 4.5: Centroids of 5 Percentile Clusters after Ranking

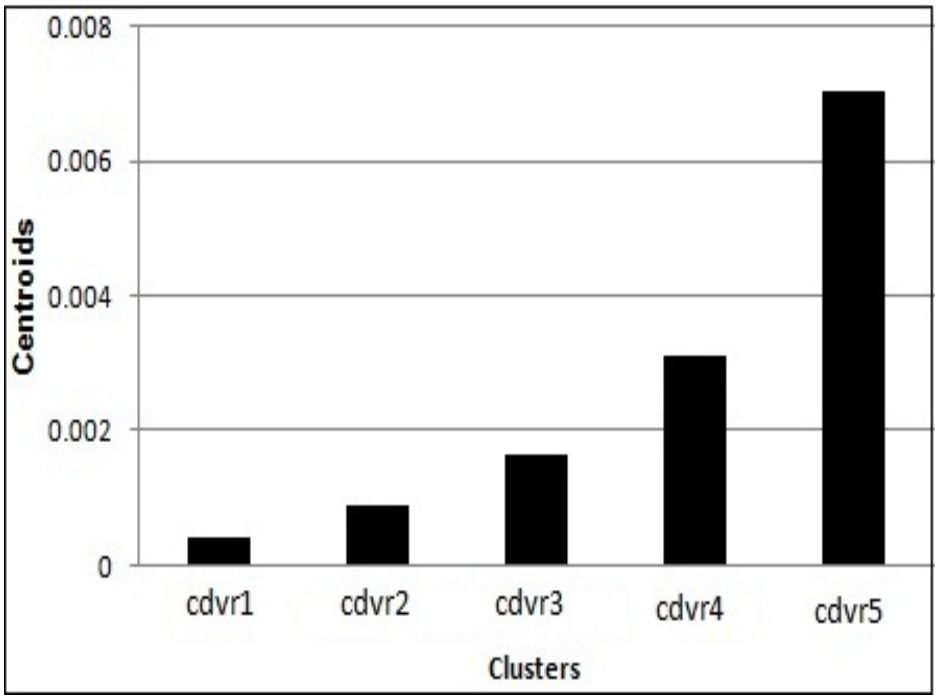


Figure 4.6: Centroids of 5 Daily Volatility Clusters after Ranking

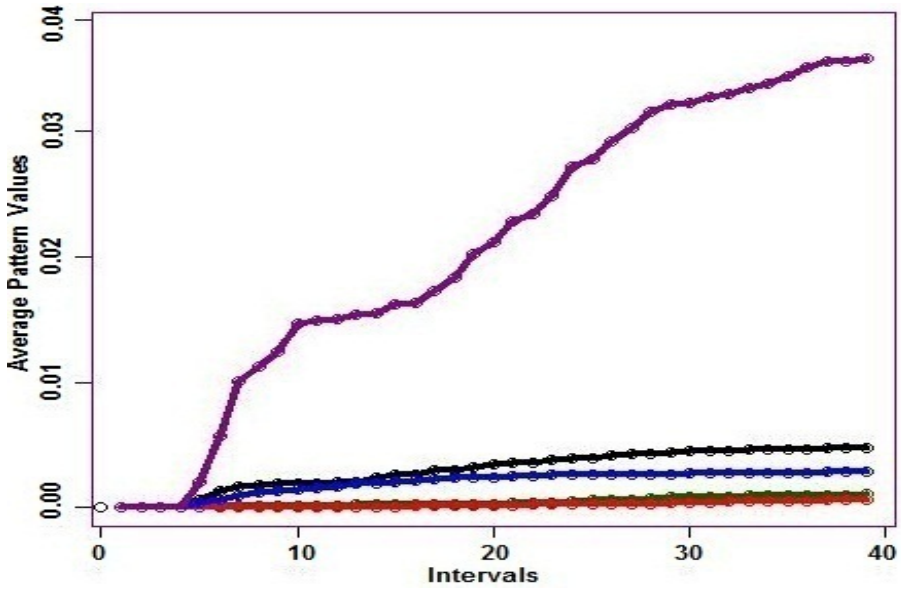
We number the clusters based on the increasing volatility. $cpr = \{cpr_1, cpr_2, cpr_3, cpr_4, cpr_5\}$ is the clustering scheme based on percentile values. $cdvr = \{cdvr_1, cdvr_2, cdvr_3, cdvr_4, cdvr_5\}$ is the clustering scheme based on the Black Scholes volatility. We can define the volatility ranking of an object by the function $cpr : X \rightarrow \{1, 2, 3, 4, 5\}$ for percentile values and $cdvr : X \rightarrow \{1, 2, 3, 4, 5\}$ for Black Scholes volatility. If an object $x \in cpr_i$, $cpr(x) = i$. Similarly, if an object $x \in cdvr_i$, $cdvr(x) = i$. The cluster ranks with respect to cluster centroids are shown in Table 4.1

Percentile Clusters								Daily Volatility Clusters			
Clusters Before Ranking (cp)	Clusters After Ranking (cpr)	Centers at 10 Percentile	Centers at 25 Percentile	Centers at 50 Percentile	Centers at 75 Percentile	Centers at 90 Percentile	Sizes	Clusters Before Ranking (cdv)	Clusters After Ranking (cdvr)	Centers	Sizes
cp3	cpr1	0	0.0008756	0.001916744	0.003155106	0.004404852	14125	cdv1	cdvr1	0.000402921	14182
cp1	cpr2	0	0.00265382	0.005659052	0.008884772	0.011652508	8676	cdv2	cdvr4	0.003108668	8990
cp4	cpr3	0	0.00555565	0.011769217	0.017598031	0.021789257	3349	cdv3	cdvr2	0.007027504	3061
cp2	cpr4	0	0.01104996	0.02360972	0.033990958	0.040298716	817	cdv4	cdvr3	0.001618887	684
cp5	cpr5	0	0.03327886	0.07348316	0.086358925	0.092558784	45	cdv5	cdvr5	0.000886057	95

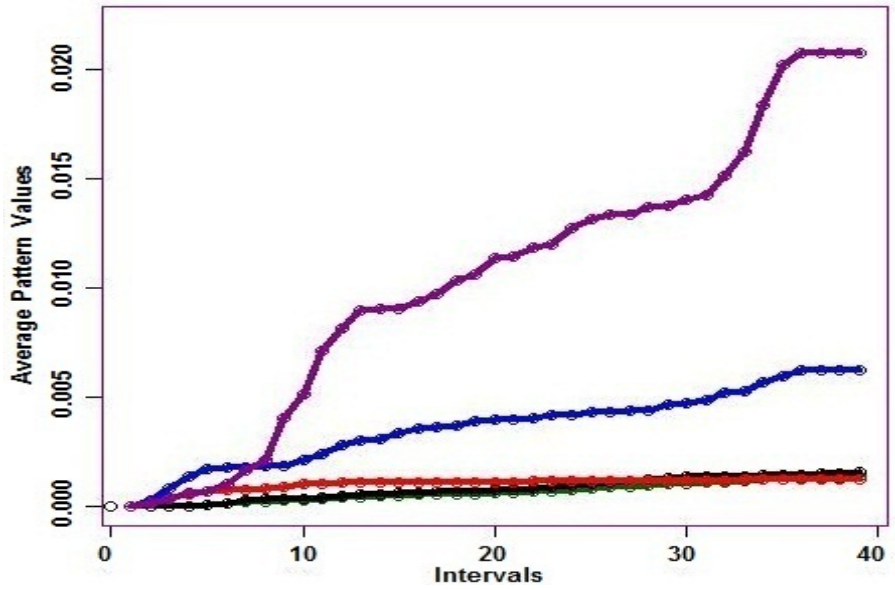
Table 4.1: Ranked Clusters

4.2 Comparative Analysis

Since we are using two different granular representations of daily patterns, we cannot easily tell if they match with each other. Both the granular representations were obtained from the same chronological daily patterns. Therefore, we plot the average chronological patterns in Figure 4.7.



(a) Percentile



(b) Daily Volatility

Figure 4.7: Average Chronological Daily Patterns

We can see that the average patterns do match reasonably well for the highest volatile clusters. For the bottom four clusters, we have a certain amount of disagreement. For example, the second most volatile cluster using Black Scholes index is well separated from the other ones, while the third and fourth most volatile clusters are more or less indistinguishable from the least volatile cluster. On the other hand, the middle three clusters are a little better separated from each other when clustered with percentile values.

In order to further understand the two clustering schemes, we plot every original pattern in all five clusters $cpr_1, \cdots cpr_5$ obtained using percentile values in Figures 4.8-4.12. The clusters $cdvr_1, \cdots cdvr_5$ based on Black Scholes index are shown in Figures 4.13-4.17. We can again visually confirm that there is a reasonable matching for highest volatile clusters.

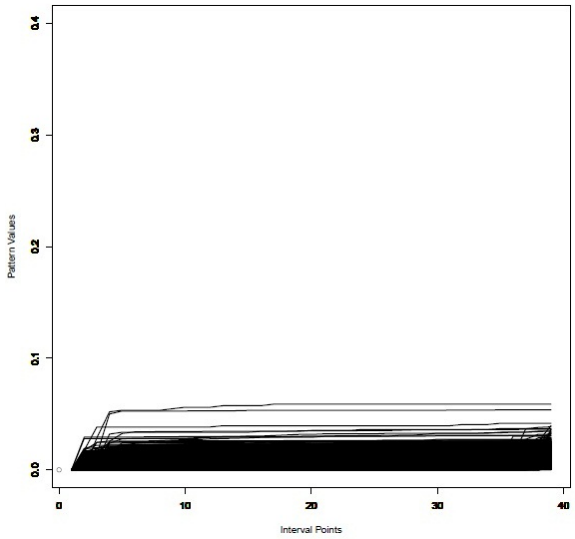


Figure 4.8: Original Patterns in cpr1

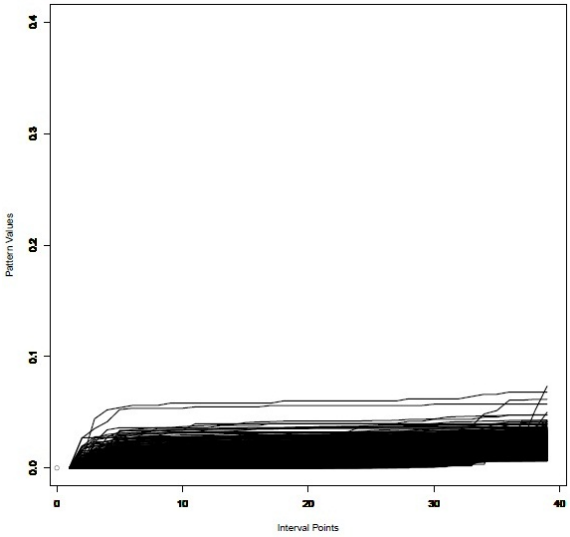


Figure 4.9: Original Patterns in cpr2

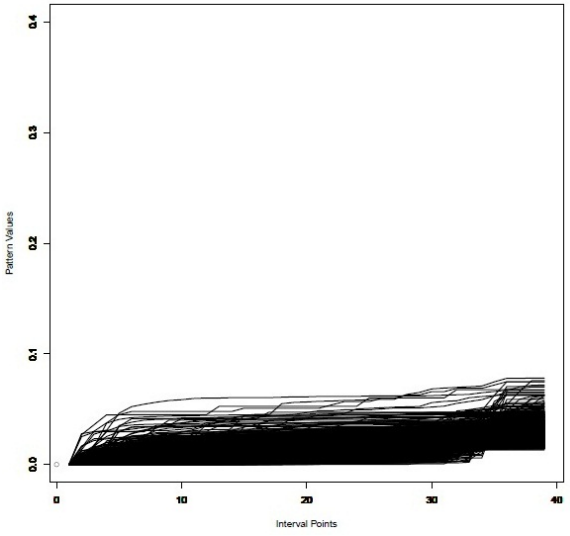


Figure 4.10: Original Patterns in cpr3

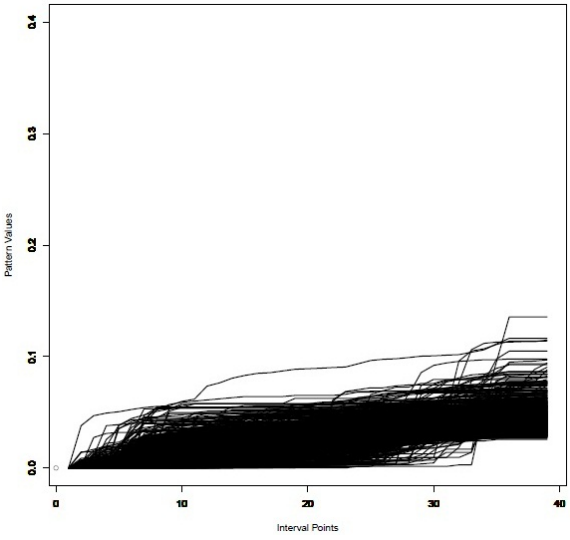


Figure 4.11: Original Patterns in cpr4

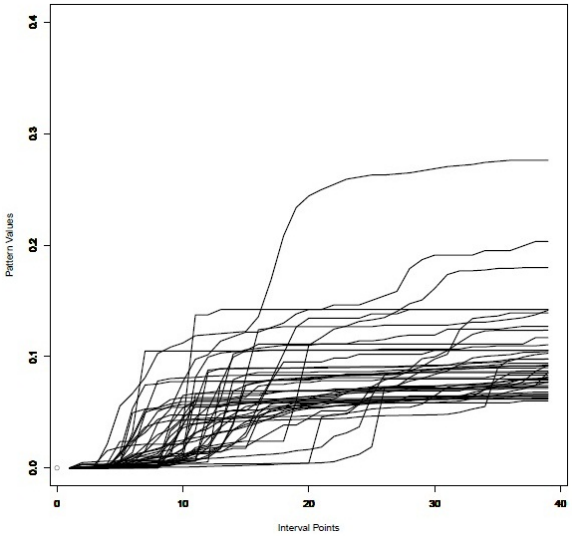


Figure 4.12: Original Patterns in cpr5

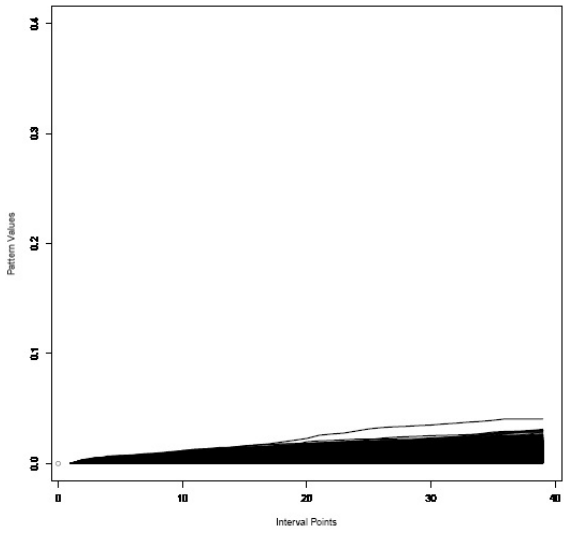


Figure 4.13: Original Patterns in cdvr1

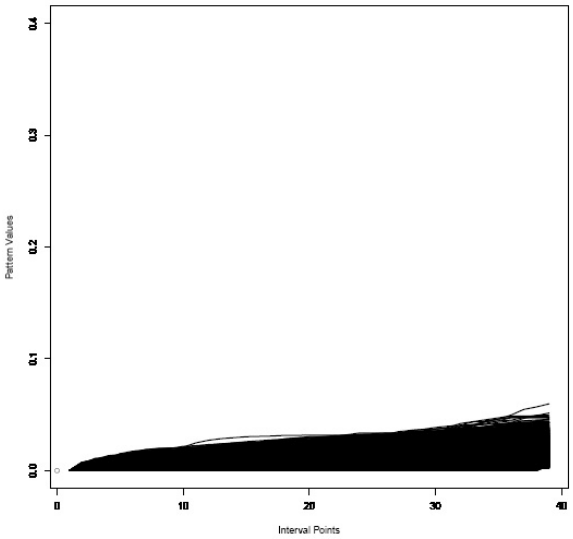


Figure 4.14: Original Patterns in cdvr2

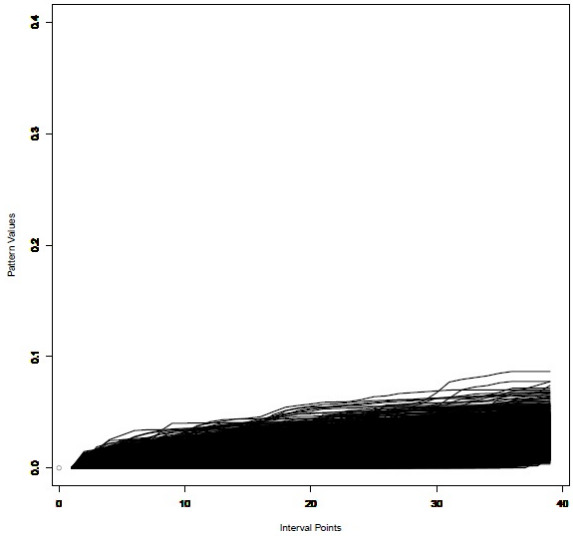


Figure 4.15: Original Patterns in cdvr3

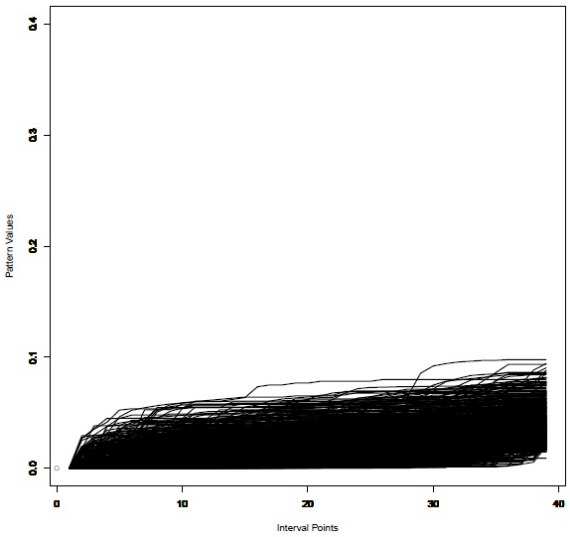


Figure 4.16: Original Patterns in cdvr4

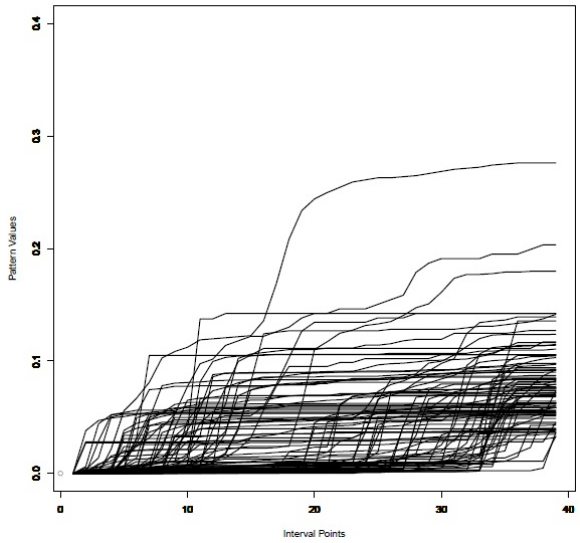


Figure 4.17: Original Patterns in cdvr5

In order to understand the disagreement between the two clustering schemes, we plot the patterns from the overlaps of clusters $cdvr_i \cap cpr_j, 1 \leq i, j \leq 5$ in Figures 4.18-4.40.

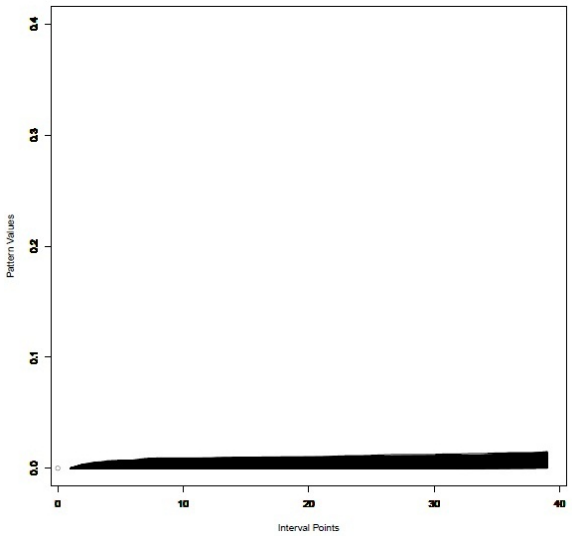


Figure 4.18: Pattern Overlaps: cpr1, cdvr1

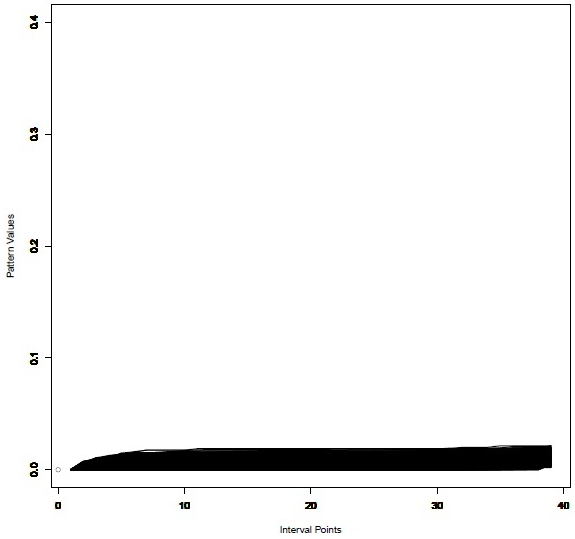


Figure 4.19: Pattern Overlaps: cpr1, cdvr2

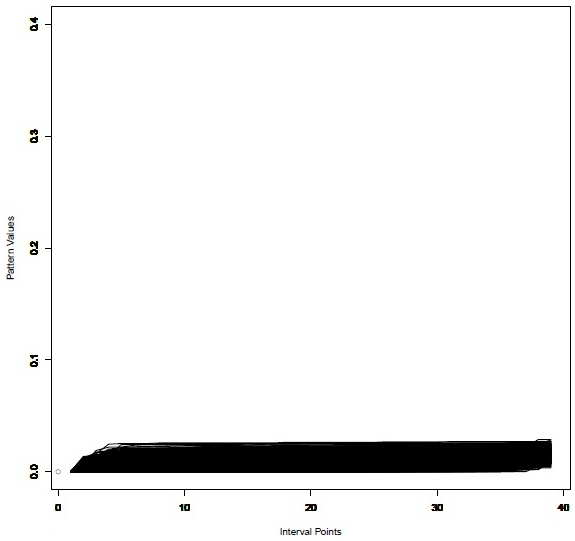


Figure 4.20: Pattern Overlaps: cpr1, cdvr3

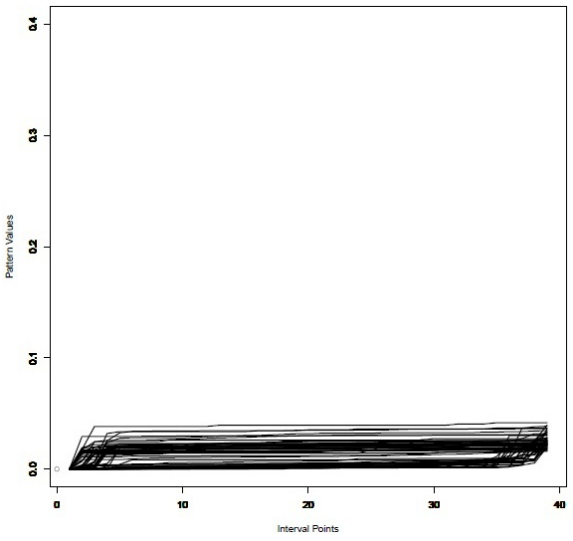


Figure 4.21: Pattern Overlaps: cpr1, cdvr4

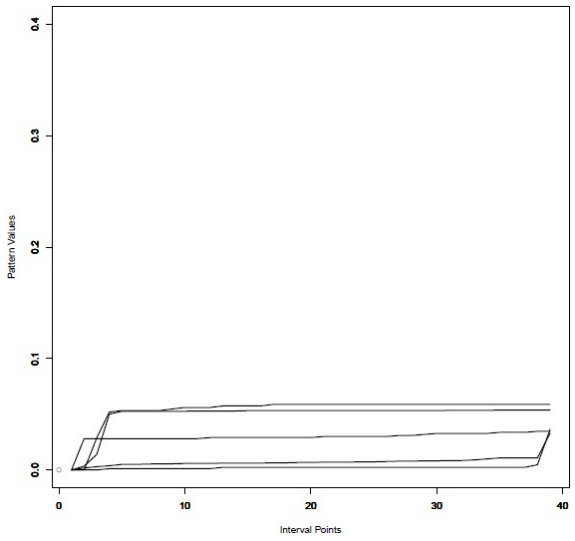


Figure 4.22: Pattern Overlaps: cpr1, cdvr5

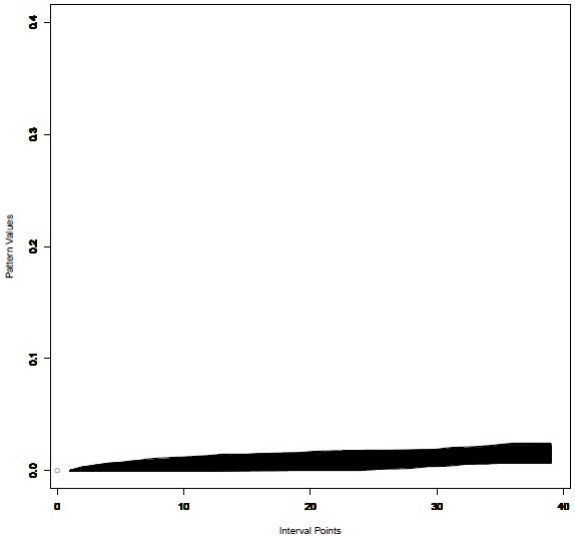


Figure 4.23: Pattern Overlaps: cpr2, cdvr1

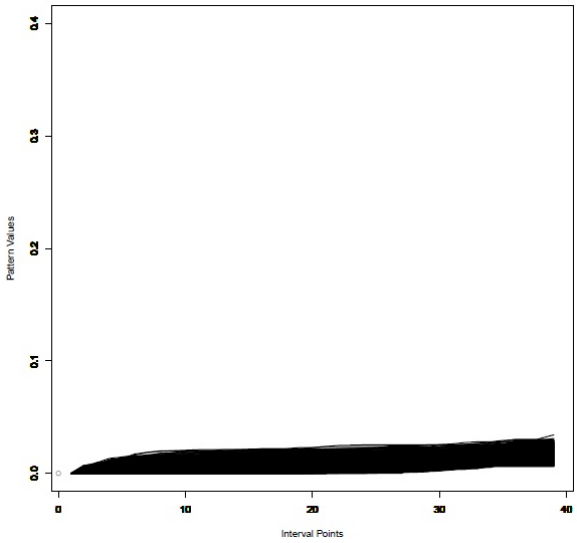


Figure 4.24: Pattern Overlaps: cpr2, cdvr2

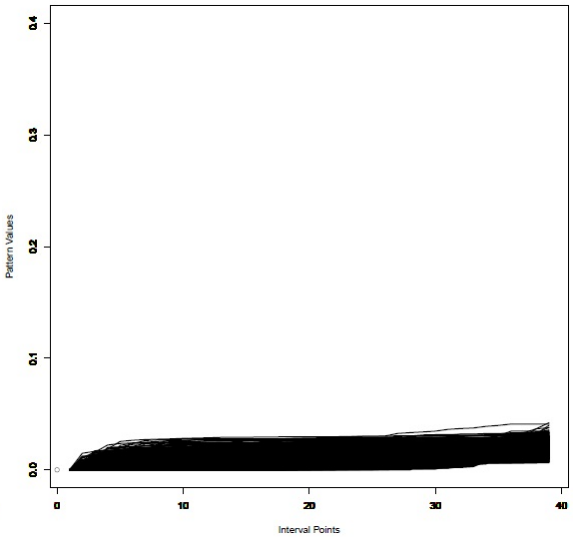


Figure 4.25: Pattern Overlaps: cpr2, cdvr3

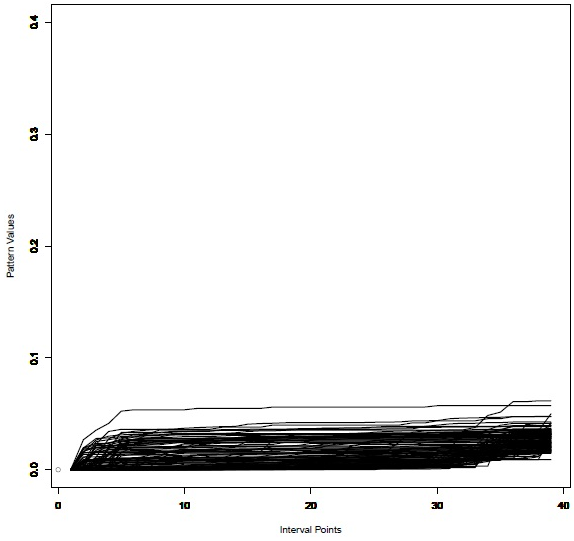


Figure 4.26: Pattern Overlaps: cpr2, cdvr4

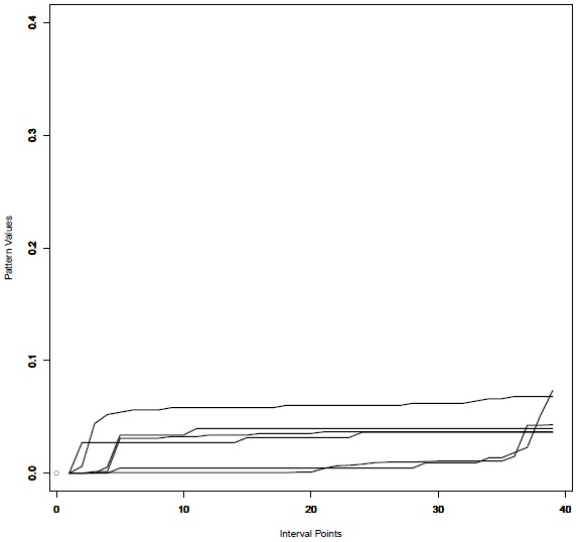


Figure 4.27: Pattern Overlaps: cpr2, cdvr5

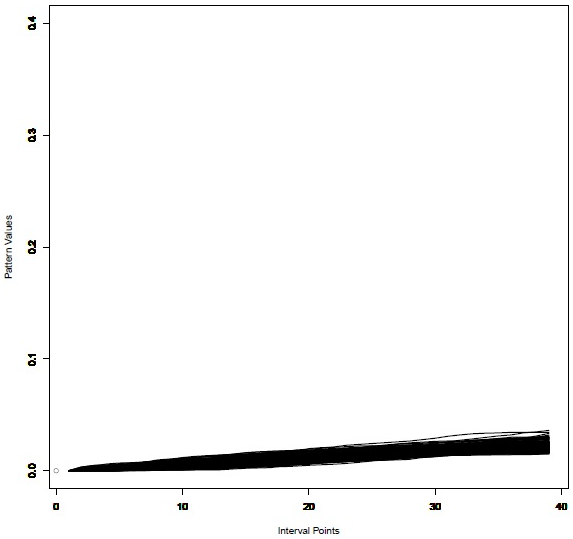


Figure 4.28: Pattern Overlaps: cpr3, cdvr1

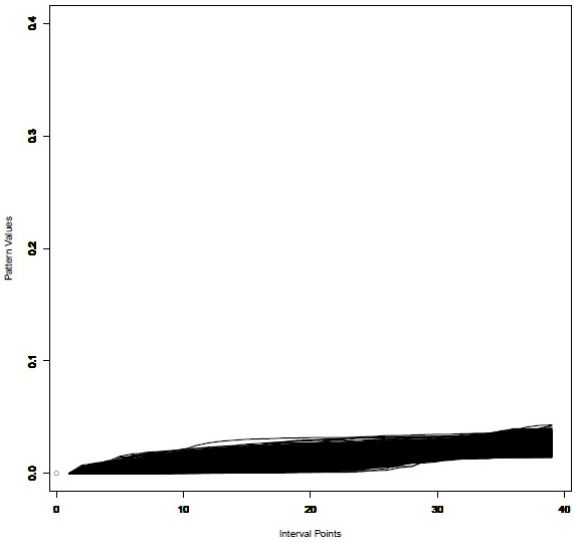


Figure 4.29: Pattern Overlaps: cpr3, cdvr2

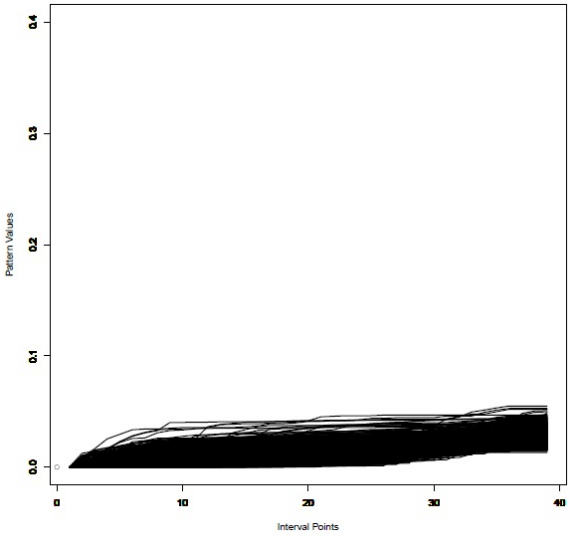


Figure 4.30: Pattern Overlaps: cpr3, cdvr3

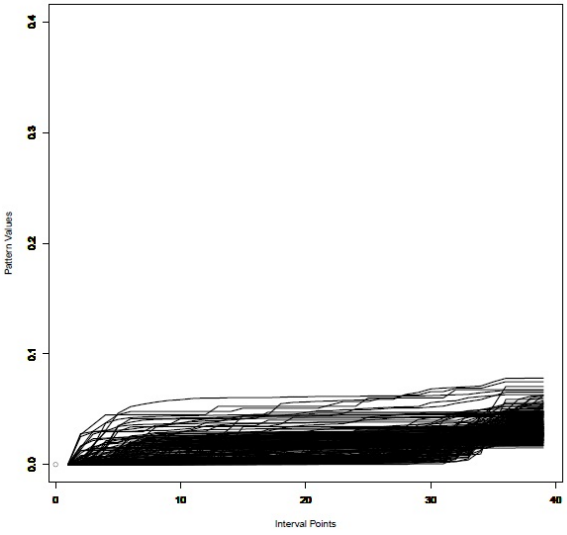


Figure 4.31: Pattern Overlaps: cpr3, cdvr4

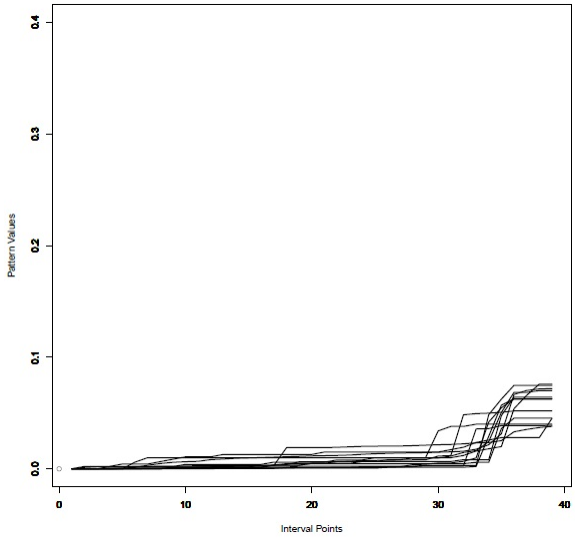


Figure 4.32: Pattern Overlaps: cpr3, cdvr5

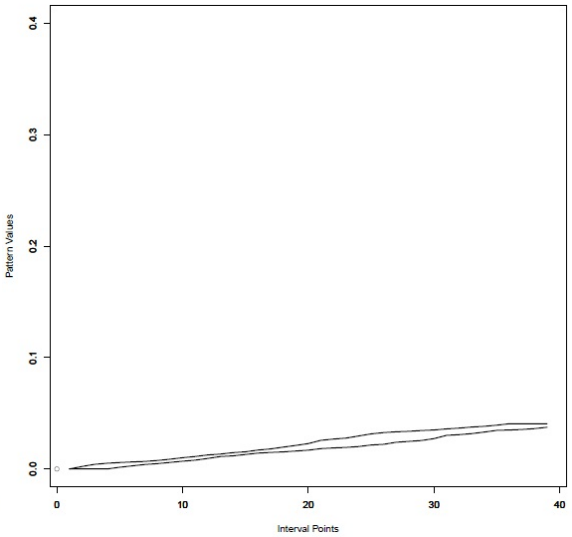


Figure 4.33: Pattern Overlaps: cpr4, cdvr1

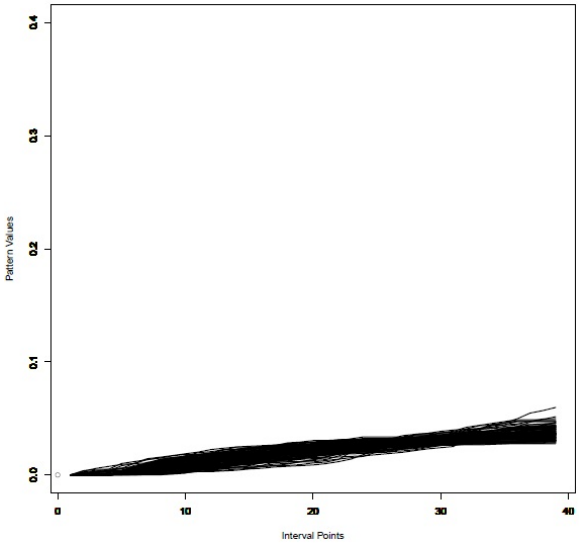


Figure 4.34: Pattern Overlaps: cpr4, cdvr2

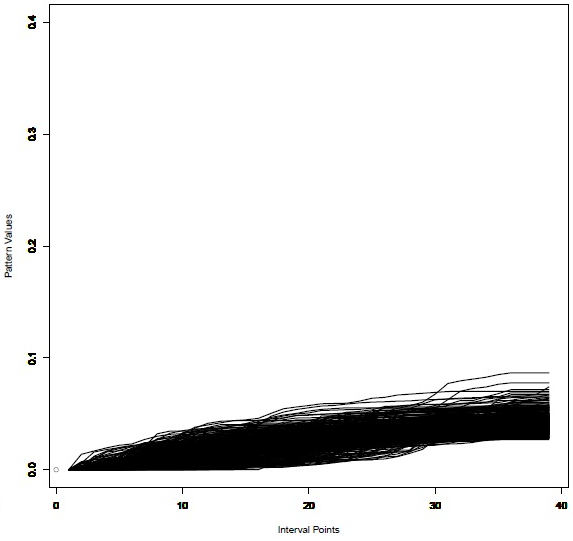


Figure 4.35: Pattern Overlaps: cpr4, cdvr3

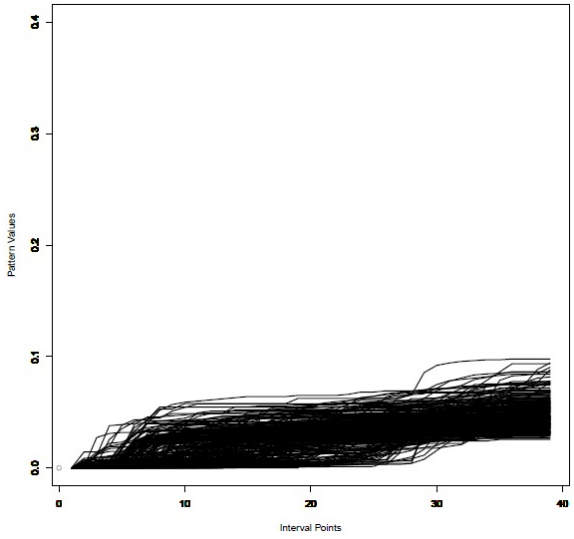


Figure 4.36: Pattern Overlaps: cpr4, cdvr4

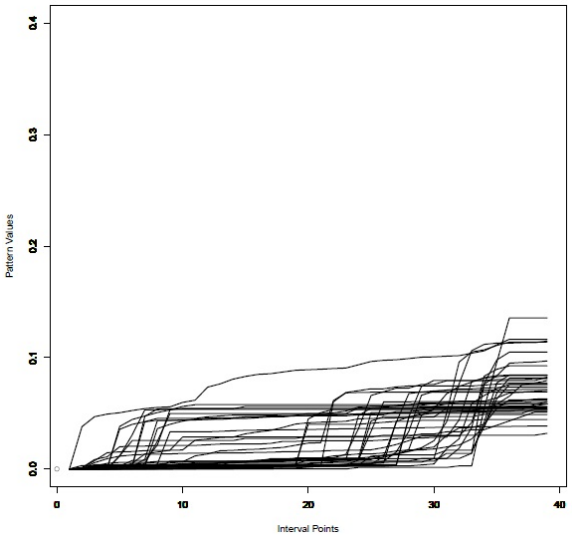


Figure 4.37: Pattern Overlaps: cpr4, cdvr5

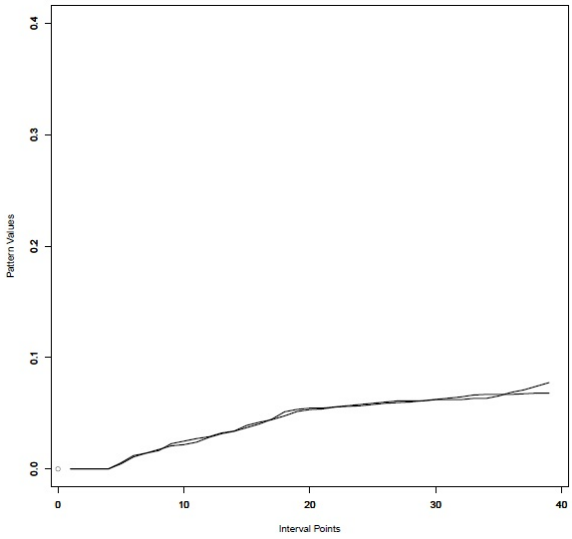


Figure 4.38: Pattern Overlaps: cpr5, cdvr3

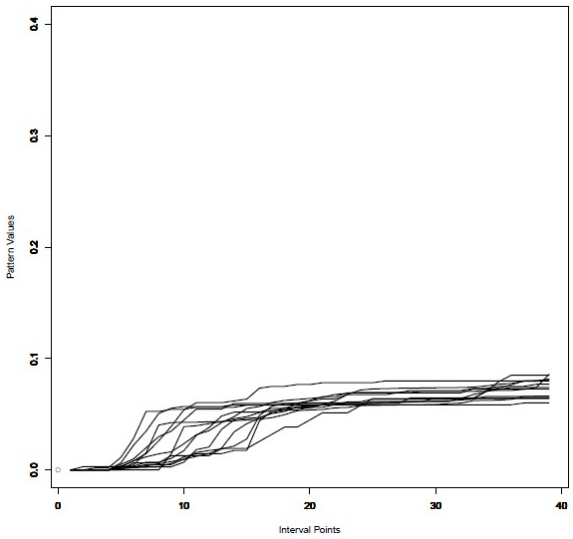


Figure 4.39: Pattern Overlaps: cpr5, cdvr4

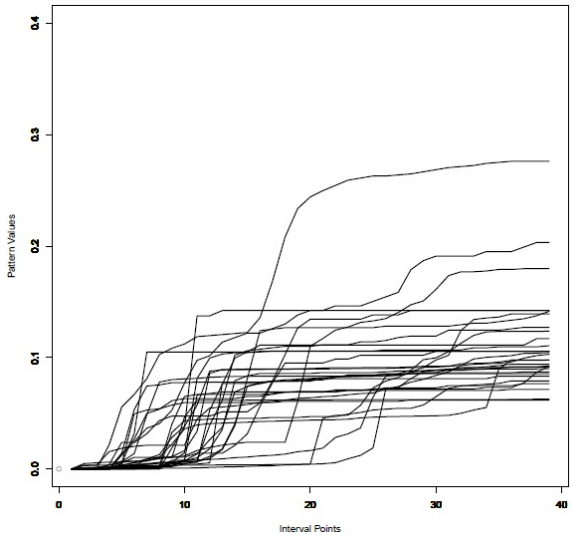


Figure 4.40: Pattern Overlaps: cpr5, cdvr5

Table 4.2 shows the intersections of clustering schemes $cdvr$ and cpr. There is reasonable agreement for clusters 1 and 2, whereas the other clusters have more disagreements with respect to cluster memberships.

	cdvr1	cdvr2	cdvr3	cdvr4	cdvr5
cpr1	**10430**	3104	519	67	5
cpr2	3411	**4047**	1089	123	6
cpr3	339	1727	1047	223	13
cpr4	2	112	404	258	41
cpr5	0	0	2	13	30

Table 4.2: Cluster Intersections

In order to compare the two clustering schemes, let us revisit Figure 4.7 that shows the corresponding average chronological daily patterns. We can see that the percentile values uniformly separates patterns with lower volatility, while the Black Scholes index tends to separate the higher volatility patterns a little better. The lower values of the last points of patterns are indistinguishable in the Daily Volatility Clusters compared to the clusters for the Percentiles, as we can see in Table 4.3.

(a) Daily Volatility

Cluster	Min. Value	Max. Value
cdvr1	0	0.040600367
cdvr2	0.001851852	0.059711432
cdvr3	0.003355705	0.086645469
cdvr4	0.009009009	0.097841727
cdvr5	0.031986532	0.276154571

(b) Percentile

Cluster	Min. Value	Max. Value
cpr1	0	0.058823529
cpr2	0.006289308	0.073394495
cpr3	0.013100437	0.078106272
cpr4	0.025575448	0.135616438
cpr5	0.060126582	0.276154571

Table 4.3: Min-Max Values of Clusters

Further inspection of patterns of the days where the rankings do not match suggest that the Black Scholes volatility focuses on the maximum peak more than the stability of the overall pattern. The Percentile distribution, on the other hand, considers the stability of a pattern.

This analysis suggests that both measures of volatility represent significant information from different perspectives. Our Clustering Ensemble should accommodate both the volatility rankings.

4.3 Rough Cluster Ensemble Formulation for Ordered Clustering

We propose the use of Rough Set theory for creating an ensemble. Since our clusters are ordered from $1, 2, \cdots, k$, we want to create clusters based on consensus in ordering. Let us use cdv and cpr as the two clustering schemes that will be combined to form a Cluster Ensemble. For an object $x \in X$, we have previously defined the ranks as $cdvr(x)$ and $cpr(x)$ from the two clustering schemes. Let $C = cdvr \otimes cpr = \{C_1, C_2, \cdots, C_k\}$ be the Rough Cluster Ensemble, and the rank of an object $x \in X$ will be defined as an interval:

$$cdvr \otimes cpr(x) = [min(cdvr(x), cpr(x)), max(cdvr(x), cpr(x))] \tag{4.1}$$

We can then define each cluster $c_i \in C$ as Rough Sets with upper and lower bounds as follows:

$$\underline{A}(c_i) = cdvr_i \cap cpr_i \tag{4.2}$$

$$\overline{A}(c_i) = cdvr_i \cup cpr_i \tag{4.3}$$

We can easily verify that the lower and upper bounds given by Eqs. 4.2-4.3 satisfy the properties for rough clustering (PR1)-(PR3) specified in section 2.1.1.2 while describing rough clustering.

We can use the intersection table shown in Table 4.2 to describe the lower and upper approximations for our example. Let us look at the table as a two dimensional matrix $table$, and $table[i][j]$ is the cell in row i and column j. All the objects in the diagonal correspond

to the lower bounds, that is the cardinality of the lower approximations,

$$|\underline{A}(c_i)| = table[i][i]. \tag{4.4}$$

That means, $\underline{A}(c_1)$ has 10430 objects, $\underline{A}(c_2)$ has 4047 objects, and so on. For the cardinality of upper approximation,

$$|\overline{A}(c_i)| = table[i][i] + \sum_{i \neq j} table[i][j] + \sum_{i \neq j} table[j][i] \tag{4.5}$$

That means objects in row i and column j belong to $\overline{A}(c_i)$. We use the condition $i \neq j$ in the sums because we do not want to count the diagonal twice. In our table that means

$\overline{A}(c_1)$ has 10430+3104+519+67+5+3411+339+2+0 = 17877 objects.

On the other end of the spectrum,

$\overline{A}(c_5)$ has 30+0+0+2+13+5+6+13+41 = 110 objects.

We can verify that the ranking in the Rough Cluster Ensemble is consistent with the ranking from the initial clustering schemes. For all $x \in \underline{A}(c_i)$, $cdv(x) = cpr(x) = i$. While for all $x \in \overline{A}(c_i)$, $min(cdvr(x), cpr(x)) \leq i \leq max(cdvr(x), cpr(x))$. We have managed to preserve the original ordering of clusters in the Cluster Ensemble with the help of interval values. The following section describes the complete Rough Ensemble clustering algorithm.

4.3.1 Algorithm: Rough Ensemble clustering

Algorithm Name: roughEnsembleClustering

Input: clSet1, clSet2, noCls

Output: rEnClusLo, rEnClusUp

Body:

rEnClusLo ← Empty

rEnClusUp ← Empty

1. for cnt = 1 to noCls do step 2 to 3
2. makeCluster(rEnClusLo[cnt], clSet1[cnt], clSet2[cnt])
3. for i = 1 to noCls do step 4
4. if(i $\neq$ cnt) do step 5 and 6
5. makeCluster(rEnClusUp[cnt], clSet1[cnt], clSet2[i])
6. makeCluster(rEnClusUp[cnt], clSet2[cnt], clSet1[i])

Procedure: makeCluster(rEnClus, cls1, cls2)

1. for each i = 1 to size(cls1) do step 2
2. for each j = 1 to size(cls2) do step 3
3. if (clSet1[i] = clSet2[j]) do step 4
4. rEnClus ← clSet1[i]

4.4 Robustness of the Proposed Rough Set Ensemble

Real-world data is susceptible to noise and likely to contain outliers. The outliers may be genuine outliers or result of noise in the data. Our two stage process, which determines the appropriate number of clusters using the knee of the curve of scatter in the clusters, will usually show a sudden increase in the cluster scatter when an outlier is absorbed into a homogeneous cluster. Therefore, a proper selection of the number of clusters will most

likely keep outliers in clusters by themselves. If a pattern was shown as an outlier in one of the clustering schemes and part of a cluster in another clustering scheme, a conventional cluster ensemble will force the outlier into a cluster. On the other hand, the rough clustering ensemble can reconcile such a disagreement by putting the outlier in the boundary region. While our dataset does not have noticeable outliers, the most volatile fifth clusters $cdvr_5$ with a cardinality of 95 and cpr_5 with a cardinality of 45 represent less than 1% of the population. They are on the extreme end of the spectrum. The disagreement between the clustering scheme leads to only 30 of them ending up in the combined most volatile cluster. The rest of them are scattered in various boundary regions.

4.5 Comparison with Other Clustering Ensemble Methods

The concept of the clustering ensemble is relatively new - first proposed a little more than a decade ago by Fred (2001). The early research focused on the crisp clustering ensemble. The lower bound of the clusters in our rough ensemble will always form the core of the corresponding cluster in any of the crisp ensemble, since there is no disagreement between the individual clustering schemes. The crisp clustering ensemble forces the non-diagonal objects in Table 4.2 into one of the clusters. Our proposal will not disagree with any of the conflict resolution strategies proposed by different clustering ensemble methods, since our upper bounds of the clusters will include the corresponding clusters from all the crisp clustering ensembles. Therefore, our proposal is consistent with any of the conventional crisp clustering ensemble approaches. In addition, one of the unique aspects of our

proposal is an ability to preserve an implicit ranking between the clusters obtained by different clustering schemes. To the best of our knowledge, there is no other clustering ensemble proposal that addresses the issue of ordered clusters. The uniqueness of our approach is also a limitation. The rough ensemble technique described here is based on combination of ordered clustering schemes. However, the lessons learned from the proposed approach will be useful in creating a more general Rough Clustering Ensemble technique.

4.6 Computational Requirements and Scalability for Rough Ensemble Clustering Algorithm

The proposed rough ensemble clustering algorithm has inherent opportunities for parallel processing. Therefore, while it will require significant computational resources, they can be distributed among multiple processors resulting in a reasonable chronological time requirement. In this section, we discuss the computational requirements and describe how the algorithm can be parallelized. The implementation of parallel rough ensemble clustering is a separate research topic in itself, and is being investigated as part of our ongoing research.

The problem of obtaining an optimal clustering scheme is NP-hard. Let us assume that there are n objects that need to be grouped into k clusters. Each object can be assigned to any one of the k clusters, resulting in $k \times k \times \cdots k = k^n$ possible clustering schemes. The clustering scheme that provides minimum scatter within clusters and maximum separation between clusters will then be selected as the optimal one. Therefore, finding the optimal clustering scheme will require $O(k^n)$ calculations of cluster quality.

It is possible that if the cluster quality measure is carefully chosen, it may be possible to optimize it without having to consider all possible clustering schemes. For example, k-means algorithm can converge towards local minimum for cluster scatter. Running k-means multiple times with different starting centroids increase the chances of finding global minimum without having to consider k^n schemes. Each iteration in k-means requires $O(k \times n)$ distance calculations. Therefore, k-means time requirements are $O(k \times n \times iter)$, where $iter$ is the number of iterations. However, the clustering scheme resulting from k-means depends on the initial choice of cluster centers. As mentioned before, one needs to apply k-means multiple times and choose a clustering scheme that provides minimum scatter within clusters and maximum separation between clusters. This can be done in parallel.

The creation of cluster ensembles are independent to each other. That is to create ensemble on one cluster we do not have to wait for the cluster ensembles to be created for other clusters. Therefore, cluster ensemble creation can be done separately.

We have used the environment provided by ACEnet, UNIX operating System, R as the clustering, plotting tool and the language to implement the algorithm, Unix script to execute R program. For our experiment, we have implemented the algorithm in linear fashion whereas such linear implementation may cause notable increase in runtime when the size of data grows significantly . Our implementation took approximately 20 minutes to complete the full execution using R. It is possible to reduce the chronological time by using a distributed environment as follows:

1. Apply k-means algorithm in parallel on multiple nodes and choose the clustering scheme with the best quality.

2. Create cluster ensemble for the different clusters in parallel to facilitate faster com-

putations.

4.7 Chapter Summary and Conclusions

Given the approximate and unsupervised nature of clustering, data mining practitioners feel the need to perform groupings using different clustering algorithms. Researchers are increasingly focusing their attention on combining these different clustering schemes using Clustering Ensemble techniques. In this phase, we use the daily prices of commodities to show an additional need for Clustering Ensemble based on preservation of cluster orderings. In order to group daily price patterns based on volatility, we represent daily patterns using a single dimensional information granule based on the Black Scholes index. Another grouping is based on an information granule that consists of a more elaborate distribution of prices during the day. The clusters within these two groupings can be ordered based on their volatility.

While these groupings tend to have a general consensus on volatility for most of the daily patterns, they disagree on a small number of patterns. A closer inspection of the patterns suggests that both points of view have some merits. That means we have a certain amount of order ambiguity in the resulting Clustering Ensemble. This paper proposes a novel Rough Clustering Ensemble algorithm for representing the ambiguity in combined clustering using intervals.

Chapter 5

Recursive Meta-clustering

This chapter represents the working procedure and experimental results for the Recursive Meta-cluster algorithm implemented on the data sets extracted from percentile and daily volatility data sets as described in Chapter 3. Section 5.1 describes the original Recursive Meta-cluster technique. Section 5.2 states the algorithm of Recursive Temporal Meta-cluster. The next consecutive sections illustrate the steps of the algorithm with sample experimental outputs and also computational aspect of the algorithm. In addition, at the end of this chapter, there is a brief summary and conclusion.

5.1 Basic Recursive Meta-clustering

According to the basic idea of Recursive Meta-cluster as described by (Triff and Lingras, 2013; Lingras et al., 2014; Lingras and Rathinavel, 2012; Rathinavel and Lingras, 2013), the process starts with the creation of a data set called the static part. It contains attributes representing information directly related to the object in consideration. This static part is clustered and then the dynamic part containing the associated or indirectly related in-

formation of the candidate records is created. The information of dynamic part is collected from the last clusters obtained.

Next the static and dynamic parts are concatenated and another clustering is performed. This process of clustering with the two parts and updating the dynamic part continues as long as the two consecutive dynamic portions of the clusters are divergent. The overall process is shown as a flowchart in Figure 5.1.

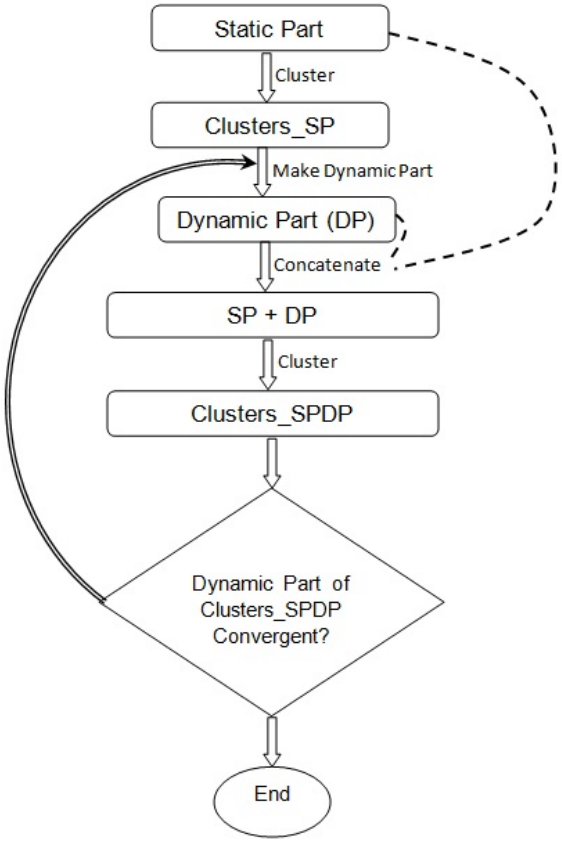


Figure 5.1: Flowchart of Recursive Meta-cluster

5.2 Algorithm: Recursive Temporal Meta-cluster

In this section the algorithm to implement recursive temporal meta-cluster using Temporal data is shown. The major difference with the basic recursive meta-clustering and temporal recursive meta-clustering is, to create the dynamic part m period is specified and m+1 to last day patterns are used in the static part with m days in dynamic part. The complete algorithm for recursive temporal meta-cluster is given next:

Name of Algorithm: recursiveMetaCluster

Body:

dyn ← Empty

prevCenters.DynCls ← Empty

noCls ← No of Clusters

m = Period of Dynamic Part

1. stat ← createStaticPart(m+1)
2. statPlusDyn ← stat + dyn
3. centers ← cluster(statPlusDyn)
4. rankedCentersStat ← rank(centers.StatClsPart)
5. if centers.DynClsPart == prevCenters.DynClsPart
6. Stop
7. prevCenters.DynClsPart ← centers.DynamicClsPart
8. dyn ← createDynamicPart(rankedCentersStat, m, noCls)

Procedure: createStaticPart(mPlus1)

1. for each i = mPlus1 to (NumOfRecords.Item) do step 2

2. add record[i] to statProfile

3. return statProfile

Procedure: rank(centers)

1. sort centers in ascending order

2. return centers

Procedure: createDynamicPart(rankedCenters, m, noCLs)

1. for each i = 1 to (NumOfRecords.Item-m) do step 2 to 4

2. for dy = i to (i+(m-1)) days do step 3

3. rank[dy]=rankDay(dy,rankedCenters,noCls)

4. add rank to dynProfile

5. return dynProfile

Procedure: rankDay(dy,rankedCenters,noCls)

1. Initialize min with -99 for each of (m+1) cells

2. for k = 1 to noCls do step 3 to 6

3. dis=abs(rankedCenters[k]-record[dy])

4. if dis<min do step 5 to 6

[i.e. if most of the values of dist <values of min]

5. min=dis

6. rank =k

7. return rank

5.3 Experimental Results

The two key steps in the algorithm are the creation of the static and dynamic parts. A daily pattern of a stock is naturally connected to the daily patterns of the same stock from previous days. It is fair to assume that a sustained activity in a stock does not last for more than two weeks (ten trading days). Based on this assumption, we can create a graph where each daily pattern is connected to the daily patterns of the same stock from previous ten days. That means the representation of a daily pattern has data from that day (obtained statically from the database). This static part consists of the natural logarithm of five percentile values, 10%, 25%, 50%, 90% as described in the data processing section. The historical volatility of the same stock over the last ten trading days constitutes the dynamic part of the representation of a daily pattern. More specifically, the dynamic part will use the volatility ranking of the last ten trading days for the same stock based on meta-clustering information. In order to have ten days of history available in the representation of a daily pattern, our data set consists of patterns starting from the 11^{th} trading day onwards. Since the natural logarithm values of the static part data set are very small, large values of the dynamic part data set may dominate the full data set. Considering this impact, we created two versions of the static data set: one with the natural logarithm values and the other with the weighted values obtained by multiplying the natural logarithm by 100. These weighting ensures that the small values of the static part data set are not dominated by the large values of the dynamic part data set. We experimented on both of these versions to see the difference while implementing our proposed Recursive Temporal Meta-clustering algorithm. Thus we experimented on four sets of data: Percentile (PD), Daily Volatility (DVD), Weighted Percentile (WPD) and Weighted Daily Volatility (WDVD) Data sets.

5.3.1 Creation of Static Part

We created the static part of the total data set for the Recursive Temporal Meta-cluster by taking records of five percentiles of the 11th to the last day for each object. For the Black Scholes daily volatility data set we do the same. The two versions of the static parts of percentile data and daily volatility data for m=10 periods are represented as Table 5.1, Table 5.2, Table 5.3 and Table 5.4 respectively.

Day:Instrument	p10	p25	p50	p75	p90
2011-08-16:3_1	0	0.0133357	0.026177	0.03095	0.036017
2011-08-17:3_1	0	1.145E-16	0.001744	0.004354	0.005222
.					
.					
2012-01-31:3_1	0	0	0.006711	0.013378	0.020661
.					
.					
2011-08-16:Z_2	0	0.0012129	0.002021	0.002424	0.002424
.					
.					
2012-01-31:Z_2	0	0.0004788	0.000878	0.001475	0.005016

Table 5.1: Static Part of Percentile Data

Day:Instrument	p10	p25	p50	p75	p90
2011-08-16:3_1	0	1.33357	2.617697	3.095025	3.601675
2011-08-17:3_1	0	1.14E-14	0.174368	0.435351	0.522194
.					
.					
2012-01-31:3_1	0	0	0.671143	1.337813	2.066051
.					
.					
2011-08-16:Z_2	0	0.121286	0.202061	0.242424	0.242424
.					
.					
2012-01-31:Z_2	0	0.047877	0.087758	0.147549	0.501573

Table 5.2: Weighted Static Part of Percentile Data

Day:Instrument	dv
2011-08-16:3_1	0.001191
2011-08-17:3_1	0.002273
.	
.	
2012-01-31:3_1	0.001994
.	
.	
2011-08-16:Z_2	0.00034
.	
.	
2012-01-31:Z_2	0.000959

Table 5.3: Static Part of Daily Volatility Data

Day:Instrument	dv
2011-08-16:3_1	0.119149
2011-08-17:3_1	0.227295
.	
.	
2012-01-31:3_1	0.199389
.	
.	
2011-08-16:Z_2	0.034041
.	
.	
2012-01-31:Z_2	0.095949

Table 5.4: Weighted Static Part of Daily Volatility Data

5.3.2 Clustering Static Part

The static part created as shown in the previous subsection is clustered. The five clusters obtained are then ranked. The cluster having the lowest valued centroids is ranked 1, the cluster having next lowest valued centroids is ranked 2 and so on. Ranked clusters for percentile data after clustering the static part are shown in Table 5.5 and 5.6. Table 5.7 and 5.8 show the ranked clusters for daily volatility data after clustering the respective static part.

Rank	Cluster	Center P10	Center P25	Center P50	Center P75	Center P90
1	C3	0	0.000923	0.001999	0.0033	0.004603
2	C5	0	0.002892	0.006084	0.009479	0.012443
3	C4	0	0.006402	0.013285	0.01941	0.023789
4	C2	0	0.013115	0.026881	0.038433	0.045564
5	C1	0	0.036534	0.102173	0.111379	0.116105

Table 5.5: Ranked Clusters after First Iteration (PD)

Rank	Cluster	Center p10	Center p25	Center p50	Center p75	Center p90
1	C2	0	0.092325	0.199883	0.329958	0.46027
2	C1	0	0.289221	0.60838	0.947911	1.24431
3	C5	0	0.640178	1.328481	1.94103	2.37894
4	C4	0	1.311539	2.688112	3.843337	4.55637
5	C3	0	3.653376	10.2173	11.13789	11.6105

Table 5.6: Ranked Clusters after First Iteration (WPD)

Rank	Cluster	CenterDV
1	C2	0.000406
2	C5	0.000895
3	C3	0.00164
4	C1	0.003134
5	C4	0.007072

Table 5.7: Ranked Clusters after First Iteration (DVD)

Rank	Cluster	Center DV
1	C5	0.0401206
2	C2	0.0879398
3	C1	0.160484
4	C3	0.3085872
5	C4	0.7072064

Table 5.8: Ranked Clusters after First Iteration (WDVD)

5.3.3 Creation of Dynamic Part

For the percentile data set the dynamic part is formed using the percentiles of each 10(m) days and ranking with the ranks of the static part clusters that are closest to them. The percentile values of the first 11 days used to calculate rank of the dynamic part for first two records (first 10 days for first record and next 10 days for second record) are shown in Table 5.9, Table 5.10.

Day:Instrument	p10	p25	p50	p75	p90
2011-08-01:3_1	0	0.0032921	0.008755	0.011173	0.012622
2011-08-02:3_1	0	0.0015259	6.60E-03	7.86E-03	0.01164
2011-08-03:3_1	0	0	0	0.002628	0.002628
2011-08-04:3_1	0	0.0021974	0.00542	0.00542	0.006706
2011-08-05:3_1	0	0	2.81E-03	0.011189	0.012579
2011-08-08:3_1	0	0.0015516	0.003101	0.006192	0.007734
2011-08-09:3_1	0	0.0033389	0.023142	0.024774	0.0251
2011-08-10:3_1	0	1.55E-03	0.003091	0.003091	0.003091
2011-08-11:3_1	0	0.0015886	0.001589	0.003175	0.003175
2011-08-12:3_1	0	0.0034327	0.011191	0.017354	0.017354
2011-08-16:3_1	0	0.0133357	0.026177	3.10E-02	0.036017

Table 5.9: Percentiles Used for First Two Records of Dynamic Part (PD)

Day:Instrument	p10	p25	p50	p75	p90
2011-08-01:3_1	0	0.329208	0.875492	1.11733	1.262153
2011-08-02:3_1	0	0.152594	0.659566	0.785908	1.163981
2011-08-03:3_1	0	0	0	0.262812	0.262812
2011-08-04:3_1	0	0.219738	0.542007	0.542007	0.670624
2011-08-05:3_1	0	0	0.280899	1.118893	1.257878
2011-08-08:3_1	0	0.155159	0.310078	0.619197	0.773399
2011-08-09:3_1	0	0.33389	2.314153	2.477418	2.510039
2011-08-10:3_1	0	0.154679	0.309119	0.309119	0.309119
2011-08-11:3_1	0	0.158856	0.158856	0.317461	0.317461
2011-08-12:3_1	0	0.343268	1.119067	1.735401	1.735401
2011-08-16:3_1	0	1.33357	2.617697	3.095025	3.601675

Table 5.10: Percentiles Used for First Two records of Dynamic Part (WPD)

For the Black Scholes volatility index data set, the dynamic part is formed using the daily volatility of each 10(m) days and ranking with the ranks of the corresponding static part clusters that are closest to them. The daily volatility values of the first 11 days used to calculate rank of the dynamic part for first two records (first 10 days for first record and next 10 days for second record) are shown in Table 5.11 and Table 5.12.

Day: Instrument	dv
2011-08-01:3_1	0.000589
2011-08-02:3_1	0.000848
2011-08-03:3_1	0.000421
2011-08-04:3_1	0.000862
2011-08-05:3_1	0.000855
2011-08-08:3_1	0.000705
2011-08-09:3_1	0.001356
2011-08-10:3_1	0.000345
2011-08-11:3_1	0.000444
2011-08-12:3_1	0.000881
2011-08-16:3_1	0.001191

Table 5.11: Daily Volatilities Used for First Two Records of Dynamic Part (DVD)

Day:Instrument	dv
2011-08-01:3_1	0.058908028
2011-08-02:3_1	0.084817842
2011-08-03:3_1	0.042070966
2011-08-04:3_1	0.086209428
2011-08-05:3_1	0.085474913
2011-08-08:3_1	0.070530337
2011-08-09:3_1	0.135598382
2011-08-10:3_1	0.034537266
2011-08-11:3_1	0.044420523
2011-08-12:3_1	0.088116996
2011-08-16:3_1	0.11914886

Table 5.12: Daily Volatilities Used for First Two Records of Dynamic Part (WDVD)

The dynamic parts created with the cluster results obtained using the static part of percentile data set are shown in Table 5.13 and 5.14. On the other hand, the dynamic parts created using the static part cluster results of daily volatility data set are shown in Table 5.15 and Table 5.16.

day_{m+1}:instrument	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
2011-08-16:3_1	2	2	1	1	2	1	3	1	1	2
2011-08-17:3_1	2	1	1	2	1	3	1	1	2	4
.										
.										
2012-01-31:3_1	1	2	3	1	1	3	4	1	1	3
.										
.										
2011-08-16:Z_2	2	1	1	1	1	1	1	1	1	2
.										
.										
2012-01-31:Z_2	1	2	2	1	3	1	2	3	1	2

Table 5.13: Dynamic Part of Percentile Data After First Iteration (PD)

	Dynamic Part									
Day:Instrument	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
2011-08-16:3_1	2	2	1	1	2	1	3	1	1	2
2011-08-17:3_1	2	1	1	2	1	3	1	1	2	4
.										
.										
2012-01-31:3_1	1	2	3	1	1	3	4	1	1	3
.										
.										
2011-08-16:Z_2	2	1	1	1	1	1	1	1	1	2
.										
.										
2012-01-30:Z_2	2	1	2	2	1	3	1	2	3	1
2012-01-31:Z_2	1	2	2	1	3	1	2	3	1	2

Table 5.14: Dynamic Part of Percentile Data After First Iteration (WPD)

day_{m+1}:instrument	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
2011-08-16:3_1	1	2	1	2	2	2	3	1	1	2
2011-08-17:3_1	2	1	2	2	2	3	1	1	2	2
.										
.										
.										
2012-01-31:3_1	2	2	3	1	2	3	4	2	2	3
.										
.										
.										
2011-08-16:Z_2	1	1	1	1	1	1	3	1	1	1
.										
.										
.										
2012-01-31:Z_2	1	2	2	2	3	1	1	4	2	2

Table 5.15: Dynamic Part of Daily Volatility Data After First Iteration (DVD)

Day:Instrument	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
2011-08-16:3_1	1	2	1	2	2	2	3	1	1	2
2011-08-17:3_1	2	1	2	2	2	3	1	1	2	2
.										
.										
2012-01-31:3_1	2	2	3	1	2	3	4	2	2	3
.										
.										
2011-08-16:Z_2	1	1	1	1	1	1	3	1	1	1
.										
.										
2012-01-31:Z_2	1	2	2	2	3	1	1	4	2	2

Table 5.16: Dynamic Part of Daily Volatility Data After First Iteration (WDVD)

5.3.4 Adding Static and Dynamic Parts

During the execution using the percentile data set, the static and dynamic parts of Table 5.1, 5.2 and Table 5.13, 5.14 are concatenated as shown in Table 5.17 and 5.18 to make it available for the next step clustering.

On the other hand, during the execution using the daily volatility data set, the static and dynamic part of Table 5.3, 5.4 and Table 5.15, 5.16 are concatenated as shown in Table 5.19 and 5.20. These added data are made available for clustering in next iteration.

5.3.5 Clustering Added Static and Dynamic Parts

The concatenated profile having 15 attributes (5 percentiles and ranks of last 10 days) as shown in the last subsection are clustered. The resultant cluster profiles are shown in Table 5.21 and 5.22.

For daily volatility set of data the concatenated profile contains 11 attributes (1 Black Scholes volatility index and ranks of last 10 days) as shown in the last subsection. The added profile is clustered, as we did for the percentile set. The resultant cluster profiles are shown in Table 5.23 and 5.24.

	Static Part					Dynamic Part									
Day:Instrument	p10	p25	p50	p75	p90	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
2011-08-16:3_1	0	0.0133357	0.026177	0.0309502	0.0360168	2	2	1	1	2	1	3	1	1	2
2011-08-17:3_1	0	1.145E-16	0.0017437	0.0043535	0.0052219	2	1	1	2	1	3	1	1	2	4
.															
.															
2012-01-31:3_1	0	0	0.0067114	0.0133781	0.0206605	1	2	3	1	1	3	4	1	1	3
.															
.															
2011-08-16:Z_2	0	0.0012129	0.0020206	0.0024242	0.0024242	2	1	1	1	1	1	1	1	1	2
.															
.															
2012-01-31:Z_2	0	0.0004788	0.0008776	0.0014755	0.0050157	1	2	2	1	3	1	2	3	1	2

Table 5.17: Concatenated Static and Dynamic Part After First Iteration (PD)

	Static Part					Dynamic Part									
Day:Instrument	p10	p25	p50	p75	p90	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
2011-08-16:3_1	0	1.33357	2.617697	3.095025	3.601675	2	2	1	1	2	1	3	1	1	2
2011-08-17:3_1	0	1.14E-14	0.174368	0.435351	0.522194	2	1	1	2	1	3	1	1	2	4
.															
.															
2012-01-31:3_1	0	0	0.671143	1.337813	2.066051	1	2	3	1	1	3	4	1	1	3
.															
.															
2011-08-16:Z_2	0	0.121286	0.202061	0.242424	0.242424	2	1	1	1	1	1	1	1	1	2
.															
.															
2012-01-30:Z_2	0	0.02003	0.12012	0.439737	0.885491	2	1	2	2	1	3	1	2	3	1
2012-01-31:Z_2	0	0.047877	0.087758	0.147549	0.501573	1	2	2	1	3	1	2	3	1	2

Table 5.18: Concatenated Static and Dynamic Part After First Iteration (WPD)

	Static Part	Dynamic Part									
Day:Instrument	dv	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
2011-08-16:3_1	0.001191	1	2	1	2	2	2	3	1	1	2
2011-08-17:3_1	0.002273	2	1	2	2	2	3	1	1	2	2
.											
.											
2012-01-31:3_1	0.001994	2	2	3	1	2	3	4	2	2	3
.											
.											
2011-08-16:Z_2	0.00034	1	1	1	1	1	1	3	1	1	1
.											
.											
2012-01-31:Z_2	0.000959	1	2	2	2	3	1	1	4	2	2

Table 5.19: Concatenated Static and Dynamic Part After First Iteration (DVD)

	Static Part	Dynamic Part									
Day:Instrument	dv	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
2011-08-16:3_1	0.119149	1	2	1	2	2	2	3	1	1	2
2011-08-17:3_1	0.227295	2	1	2	2	2	3	1	1	2	2
.											
.											
2012-01-31:3_1	0.199389	2	2	3	1	2	3	4	2	2	3
.											
.											
2011-08-16:Z_2	0.034041	1	1	1	1	1	1	3	1	1	1
.											
.											
2012-01-31:Z_2	0.095949	1	2	2	2	3	1	1	4	2	2

Table 5.20: Concatenated Static and Dynamic Part After First Iteration (WDVD)

Rank	Cluster	p10	p25	p50	p75	p90	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
1	C1	0	0.002005	0.00422	0.00645	0.00833	1.361	1.3203	1.3526	1	1.2765	1.338	1.3529	1.2502	1.3565	1.3504
2	C5	0	0.002266	0.004808	0.007335	0.00952	1.447	1.3865	1.4565	2.3773	1.3606	1.486	1.4237	1.3403	1.4339	1.4027
3	C3	0	0.002673	0.005716	0.008859	0.01143	1.849	1.3417	1.8851	1.6566	2.9031	1.8	1.8845	1.6096	1.849	1.7935
4	C4	0	0.002761	0.005789	0.008867	0.01154	1.688	1.4692	1.664	1.6172	1.4059	1.869	1.743	2.8242	1.801	2.0553
5	C2	0	0.002934	0.006155	0.009336	0.01197	1.987	2.7388	2.0002	1.7455	1.6574	1.866	1.9558	1.5396	1.91	1.8143

Table 5.21: Cluster Centers after Clustering with Concatenated Profile (PD)

Rank	Cluster	p10	p25	p50	p75	p90	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
1	C4	0	0.14382	0.30398	0.48126	0.64097	1.34878	1.32783	1.2302	1.20089	1.32349	1.32168	1.32218	1.31453	1.32833	1.35230
2	C2	0	0.20478	0.43184	0.68104	0.91311	1.67672	1.69947	2.67788	1.53145	1.71014	1.52193	1.80877	1.69088	1.51659	1.60060
3	C5	0	0.20340	0.4361	0.68921	0.91442	1.51463	1.69113	1.37914	2.65108	1.61151	1.78801	1.56211	1.68801	1.68561	1.51438
4	C3	0	0.22243	0.47037	0.74715	0.99583	2.15922	1.98918	1.5421	1.52399	2.08161	2.07479	1.99106	2.02117	2.14393	2.20696
5	C1	0	0.91330	1.91100	2.69839	3.21708	1.68656	1.72732	1.74478	1.72101	1.71276	1.75546	1.787	1.73362	1.79379	1.77583

Table 5.22: Cluster Centers after Clustering with Concatenated Profile (WPD)

Rank	Cluster	dv	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
1	C3	0.000581	1.244876	1.232362	1.247033	1.191909	1.243581	1.243581	1.185976	1.194175	1.239159	1.239051
2	C5	0.000817	1.922866	2.026016	1.897079	2.249886	1.639662	1.65016	1.406435	1.417618	1.505705	1.521451
3	C2	0.000827	1.641562	1.477151	1.632882	1.427623	1.907327	1.630074	2.468726	1.453153	2.012254	1.529232
4	C1	0.000872	1.507823	1.599641	1.516799	1.487561	1.56117	1.832265	1.454476	2.446781	1.694024	2.205694
5	C4	0.001275	2.442462	2.433538	2.505846	2.468	2.498154	2.500615	2.455077	2.450154	2.448308	2.429846

Table 5.23: Cluster Centers after Clustering with Concatenated Profile (DVD)

Rank	Cluster	dv	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
1	C2	0.0578349	1.213534	1.255977	1.245229	1.246984	1.187102	1.254332	1.234372	1.218798	1.262119	1.227901
2	C1	0.0813556	1.421884	1.628032	1.619289	1.647208	2.623801	1.712352	1.829385	1.451777	1.695431	1.438522
3	C4	0.0813672	2.462082	1.83882	2.046574	1.786637	1.442819	1.608876	1.676177	1.501341	1.648378	1.50695
4	C5	0.086204	1.388828	1.645176	1.495032	1.688673	1.403842	1.813866	1.710091	2.218812	1.820269	2.283065
5	C3	0.1268952	2.389019	2.475467	2.49153	2.510222	2.46729	2.498832	2.506133	2.476636	2.476928	2.424357

Table 5.24: Cluster Centers after Clustering with Concatenated Profile (WDVD)

5.3.6 Recursion and Convergence

The cluster centroids of the dynamic parts of data obtained as shown in last subsection are compared with the cluster centroids of the dynamic part obtained in the previous iteration executed for the respective data set. Since we do not have any dynamic part in the clusters of first iteration, the comparison is applicable only from the 3rd iteration. From the 3rd iteration, we can compare the dynamic part of cluster centroids with that of 2nd iteration and continue like this.

As soon as the dynamic parts of the cluster profiles obtained from two consecutive iterations are the same or stable, the process is supposed to be stopped. The rounded centroids of the dynamic part of two consecutive clusters are compared for this. The system executed 65 iterations for both of the types of data sets.

In the case of the unweighted percentile data set a maximum of a 86% match was found for the results of iterations 50 and 51. The dynamic part of the 50th and 51th clusters representing rounded centroids are shown in Tables 5.25 and 5.26.

Cluster Rank	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	3	3	2	2
3	5	2	2	1	2	1	1	1	1	1
4	2	2	2	5	2	2	2	1	1	1
5	1	3	3	1	2	2	1	1	1	1

Table 5.25: Rounded Cluster Centroids of Dynamic Part at 50th Iteration (PD)

Cluster Rank	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	3	3	3	2	2
3	4	1	3	1	2	1	1	1	1	1
4	2	2	2	5	2	2	2	1	1	1
5	2	5	2	1	2	2	1	1	1	1

Table 5.26: Rounded Cluster Centroids of Dynamic Part at 51th Iteration (PD)

On the other hand, for the weighted percentile data set a maximum of a 98% match was found for the results of iterations 9 and 10. The dynamic part of the 9th and 10th clusters representing rounded centroids are shown in Tables 5.27 and 5.28.

Cluster Rank	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	3	3	3	3
3	3	3	3	2	2	2	1	1	1	1
4	3	3	3	3	3	3	3	3	3	3
5	1	1	1	1	1	1	1	1	1	1

Table 5.27: Rounded Cluster Centroids of Dynamic Part at 9th Iteration (WPD)

Cluster Rank	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
1	1	1	1	1	1	1	1	1	1	1
2	1	1	2	2	2	2	3	3	3	3
3	3	3	3	2	2	2	1	1	1	1
4	3	3	3	3	3	3	3	3	3	3
5	1	1	1	1	1	1	1	1	1	1

Table 5.28: Rounded Cluster Centroids of Dynamic Part at 10th Iteration (WPD)

While executing Recursive Meta-clustering for the unweighted data set of Black Scholes volatility index, a maximum of 94% match was found for the results of iteration 40 and 41. The dynamic part of 40th and 41th clusters representing rounded centroids are shown in Tables 5.29 and 5.30.

Cluster Rank	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	2	2	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
4	3	3	2	2	1	1	1	1	1	2
5	3	3	3	3	3	3	2	1	1	1

Table 5.29: Rounded Cluster Centroids of Dynamic Part at 40th Iteration (DVD)

Cluster Rank	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
4	3	3	2	2	1	1	1	1	1	2
5	3	3	3	3	4	3	2	2	1	1

Table 5.30: Rounded Cluster Centroids of Dynamic Part at 41th Iteration (DVD)

For the weighted data set of Black Scholes volatility index, a maximum of 88% match was found for the results of iteration 14 and 15. The dynamic part of the 14th and 15th clusters representing rounded centroids are shown in Tables 5.31 and 5.32.

Cluster Rank	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2
3	2	2	1	2	2	2	3	4	4	3
4	5	3	3	2	2	2	2	2	2	2
5	2	3	3	4	4	3	3	2	2	2

Table 5.31: Rounded Cluster Centroids of Dynamic Part at 14th Iteration (WDVD)

Cluster Rank	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2
3	2	2	2	2	2	3	3	4	4	4
4	5	3	2	2	2	2	2	2	2	2
5	3	3	4	4	4	3	3	2	2	2

Table 5.32: Rounded Cluster Centroids of Dynamic Part at 15th Iteration (WDVD)

Considering the cluster results of the 50th iteration and 9th iteration of unweighted and weighted percentile data as final, the final cluster centroids and their ranks are shown in Table 5.33 and Table 5.34 respectively. For one dimensional unweighted and weighted daily volatility data, we consider the cluster results of the 40th and 14th iteration as final. Centroids of the clusters of the 40th iteration for unweighted daily volatility set and their ranks are shown in Table 5.35. On the other hand, centroids of the clusters of the 14th iteration and their ranks for weighted daily volatility data set are shown in Table 5.36.

Rank	Cluster	p10	p25	p50	p75	p90	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
1	C4	0	0.00236	0.00501	0.00765	0.00992	1.00621	1.00358	1.00267	1.00403	1.01133	1.00933	1.00240	1.00249	1.00353	1.01817
2	C1	0	0.00263	0.00542	0.00799	0.01004	1.14622	1.16352	1.23427	1.11792	2.07704	2.36163	2.72641	2.54088	2.38522	1.71383
3	C5	0	0.00260	0.00546	0.00885	0.01134	4.75887	2.30769	1.95562	1.14940	1.57396	1.38757	1.18491	1.09467	1.10207	1.07100
4	C3	0	0.00308	0.00654	0.00993	0.01241	2.29545	2.25	2.41477	4.81392	2.37215	2.05965	1.83806	1.49573	1.31960	1.25284
5	C2	0	0.00321	0.00633	0.00980	0.01264	1.12597	3.43234	3.25194	1.18507	1.75427	1.58009	1.19595	1.09953	1.07309	1.05598

Table 5.33: Final Ranked Centers (PD)

Rank	Cluster	p10	p25	p50	p75	p90	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
1	C4	0	0.17534	0.37242	0.58997	0.78547	1.04914	1.02839	1.05153	1.04509	1.03633	1.04382	1.03902	1.03768	1.06150	1.10285
2	C3	0	0.20646	0.42539	0.66686	0.87847	1.27127	1.29929	1.46560	1.67021	1.97198	2.35992	2.70141	2.98829	3.01808	2.93971
3	C2	0	0.20454	0.43570	0.68550	0.90632	3.31182	3.17725	2.71814	2.35777	1.95536	1.56402	1.33431	1.21179	1.20202	1.23004
4	C5	0	0.25030	0.52988	0.80458	1.04298	2.92101	3.05917	3.18111	3.31695	3.39053	3.34473	3.19706	2.99331	2.86982	2.71546
5	C1	0	0.91822	1.92571	2.72231	3.24948	1.27055	1.21782	1.25423	1.22724	1.23289	1.19962	1.19271	1.17514	1.16572	1.21343

Table 5.34: Final Ranked Centers (WPD)

Rank	Cluster	dv	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
1	C1	0.000786	1.01442	1.004554	1.028689	1.024438	1.026108	1.028081	1.021554	1.020795	1.045082	1.088798
2	C4	0.000795	1.368455	1.258565	1.404654	1.759535	2.177117	2.548804	2.820297	2.996768	3.03426	2.973497
3	C3	0.0008	2.985588	3.014277	3.023006	3.013127	3.003789	3.004736	3.012856	3.0136	3.015833	2.990662
4	C2	0.000821	3.371703	3.093525	2.177458	1.564748	1.082734	1.131894	1.21223	1.293765	1.464029	1.603118
5	C5	0.000856	2.712371	2.805155	3.140206	3.346392	3.287629	2.639175	1.989691	1.489691	1.203093	1.365979

Table 5.35: Final Ranked Centers (DVD)

Rank	Cluster	dv	$Day_{(m-9)}$	$Day_{(m-8)}$	$Day_{(m-7)}$	$Day_{(m-6)}$	$Day_{(m-5)}$	$Day_{(m-4)}$	$Day_{(m-3)}$	$Day_{(m-2)}$	$Day_{(m-1)}$	$Day_{(m)}$
1	C2	0.07406	1.1005	1.089841	1.068523	1.046335	1.043507	1.046117	1.047205	1.050903	1.085926	1.176202
2	C5	0.079609	1.931926	1.995272	1.99009	1.993717	2.017877	2.021569	1.983742	1.965024	1.994235	2.019302
3	C3	0.079874	1.634738	1.543423	1.49106	1.532567	1.846105	2.482759	3.272031	3.803959	3.573436	3.083014
4	C1	0.087341	4.538998	3.27584	2.665821	2.123018	1.771084	1.717185	1.720989	1.699429	1.783133	1.818643
5	C4	0.09231	1.953866	2.708902	3.496426	4.037037	3.824561	3.151397	2.618583	2.131904	1.895387	1.841455

Table 5.36: Final Ranked Centers (WDVD)

5.3.7 Meta-profile Representation

Now that we have the static and dynamic profile for the days of all the financial instruments (stocks) as shown graphically in Figure 5.2, 5.3, 5.4 and 5.5, we can create the meta-profiles of each cluster as follows:

i. *Meta-profiles of Unweighted Percentile Data*:

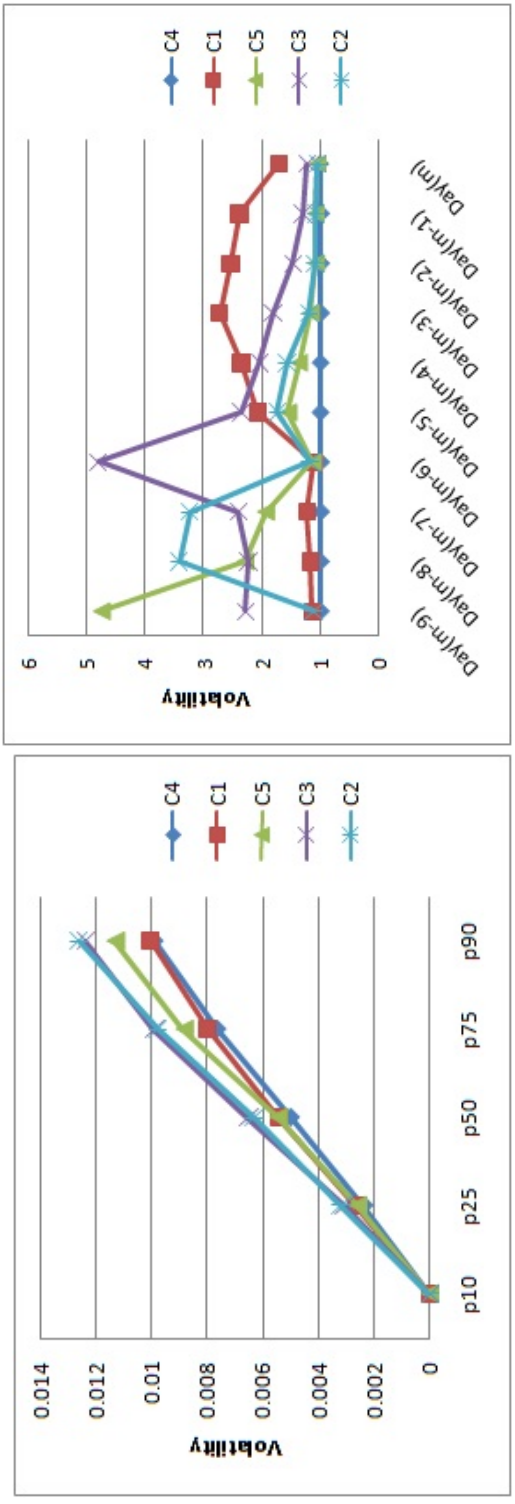
Cluster C4 (Rank 1) - least volatile The stocks in this cluster are not volatile today nor have they shown any volatility for last two weeks (10 trading days).

Cluster C1 (Rank 2) - low volatility today, but volatile over last week The stocks in this cluster are not volatile today. However, they were volatile last week (5 trading days). The volatility in these stocks may be subsiding and it may be relatively safer to sell them.

Cluster C5 (Rank 3) - moderate volatility today and first week and no volatility last week Stocks in this group are moderate volatility today and first week (5 trading days) while almost not volatile last week. There are possibilities of rising in the next little while and therefore selling after a while would be profitable.

Cluster C3 (Rank 4) -high volatile today, downgrading volatility last two weeks with an exception Stocks of this cluster are high volatility today, in last two weeks volatility trend was decreasing except for a sudden pick (4th trading day) at the middle of the first week. This trend indicates now is the best time to sell and shortly there will be opportunities to buy stocks.

Cluster C2 (Rank 5) - high volatility today, volatile in the beginning of first week (3



(a) Current Day

(b) Last 10 Days

Figure 5.2: Final Ranked Centers (PD)

trading days) but not volatile in the rest of the week and next The stocks in this cluster are attracting interest of the traders. They may be in the early phase of activity and potential buying opportunities.

ii. *Meta-profiles of Weighted Percentile Data*:

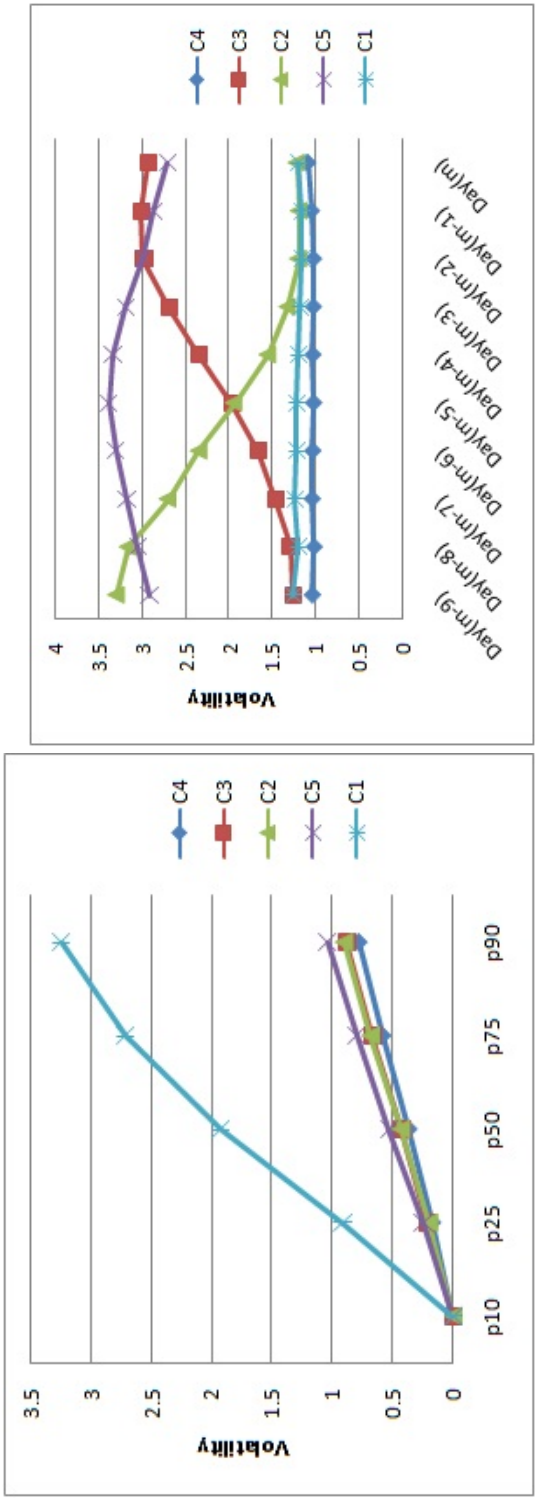
Cluster C4 (Rank 1) - least volatile The stocks in this cluster are not volatile today nor have they shown any volatility for last two weeks (10 trading days).

Cluster C3 (Rank 2) - low volatility today, but volatile over last week The stocks in this cluster are not volatile today. However, they were volatile last week (5 trading days). The volatility in these stocks may be subsiding and it may be relatively safer to sell them.

Cluster C2 (Rank 3) - moderate volatility today and last week, but volatile two weeks ago The stocks in this cluster are somewhat volatile today. They have not shown much volatility last week (5 trading days) either. However, they were quite active two weeks ago. The volatility in these stocks has definitely subsided and it may be better to sell them as they are unlikely to rise in the next little while.

Cluster C5 (Rank 4) - moderate volatility today, but volatile for last two weeks The stocks in this cluster are not volatile today. However, they were volatile over the last two weeks (10 trading days). The volatility in these stocks seems to have come to a screeching halt. It may be good idea to study the news on these stocks and trade accordingly.

Cluster C1 (Rank 5) - high volatility today, but was not volatile for last two weeks The stocks in this cluster are attracting interest of the traders. They may be in the early phase of



(a) Current Day

(b) Last 10 Days

Figure 5.3: Final Ranked Centers (WPD)

activity and potential buying opportunities.

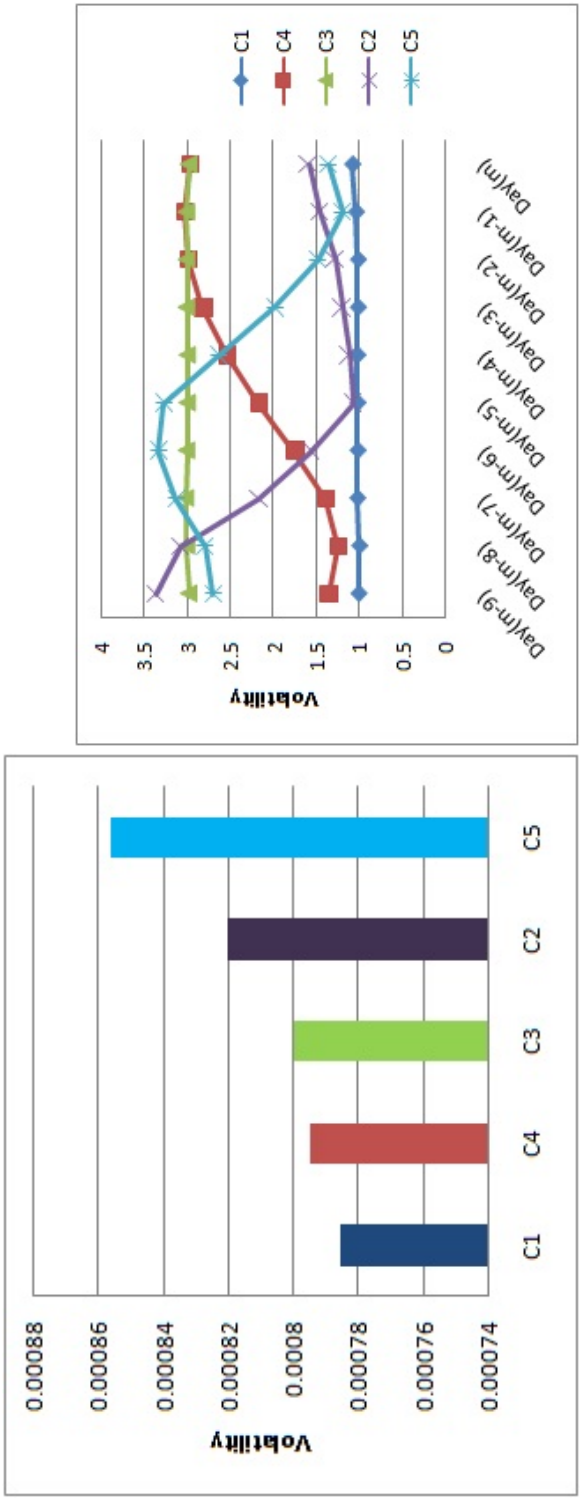
iii. *Meta-profiles of Unweighted Daily Volatility Data*: **Cluster C1 (Rank 1) - least volatile** The stocks in this cluster are not volatile today nor have they shown any volatility for the last two weeks (10 trading days).

Cluster C4 (Rank 2) - low volatility today, but volatile over last week The stocks in this cluster are not volatile today. However, they were volatile last week (5 trading days). The volatility in these stocks may be subsiding and it may be relatively safer to sell them.

Cluster C3 (Rank 3) - moderate volatility today, but volatile in last two weeks The stocks in this cluster are somewhat volatile today. However, they were quite active for the last two weeks. The volatility in these stocks seems to be decreasing or coming to a halt. It may be good idea to study the news on these stocks and trade accordingly.

Cluster C2 (Rank 4) - moderate volatility today, downgraded volatility one week ago The stocks in this cluster are moderately volatile today and low volatile for the last week. However, they were volatile one week ago. The volatility in these stocks seems to be increasing and therefore providing trading opportunities. Traders may like to sell the stocks now or waiting a bit would be profitable.

Cluster C5 (Rank 5) - high volatility today and one week ago The stocks in this cluster are now in the rising region with a possibility of low volatility next. Traders should take their trading decision now or a risk may arise shortly.



(a) Current Day

(b) Last 10 Days

Figure 5.4: Final Ranked Centers (DVD)

iv. *Meta-profiles of Weighted Daily Volatility Data*:

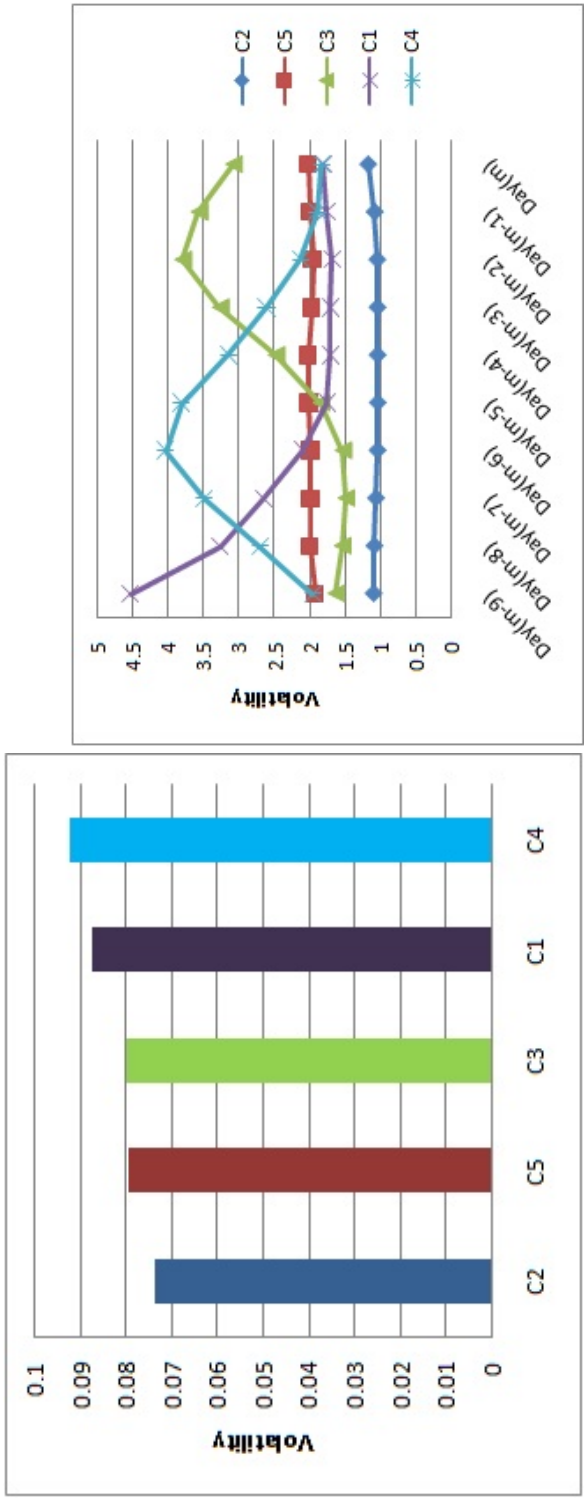
Cluster C2 (Rank 1) - least volatile The stocks in this cluster are not volatile today nor have they shown any volatility for last two weeks (10 trading days).

Cluster C5 (Rank 2) - moderate volatility today and last two weeks The stocks in this cluster are moderately volatile today and the last two weeks (11 trading days). Traders may like to buy or sell as per current scenario. They should keep an eye on the news.

Cluster C3 (Rank 3) - moderate volatility today, low volatility one week ago and volatile last week The stocks in this cluster are somewhat volatile today. They have not shown much volatility one week ago (5 trading days). However, thy were quite active last week. The volatility in these stocks are reaching to a halt. It may be better to sell them as they are unlikely to rise in the next little while.

Cluster C1 (Rank 4) - volatility today and one week ago, but low volatile for last week The stocks in this cluster are volatile today and were one week ago as well. However, they were not volatile almost all over last week (6 trading days). The volatility in these stocks seems to be in the most fluctuating region now. Traders may enjoy trading opportunities right now and there is a risk of halt in next days.

Cluster C4 (Rank 5) - high volatility today and middle of first week The stocks in this cluster are volatile today as it was in the middle of first week. It seems there is a possibility of slow downgrading of volatility in the coming days. Therefore, traders may take trading decisions accordingly.



(a) Current Day

(b) Last 10 Days

Figure 5.5: Final Ranked Centers (WDVD)

The cluster profiles described here put the volatility in historical perspective and may allow traders to look at stocks with a more informed decision.

We can also provide a graphical representation of a stock over the previous two weeks. As for instance, for the day 2011-08-16 of instrument 3_1 we have the ranks for 11 days including the day itself and previous 10 days as shown in Figure 5.6 and 5.7 where the meta-clusters are created from unweighted and weighted percentile data sets respectively.

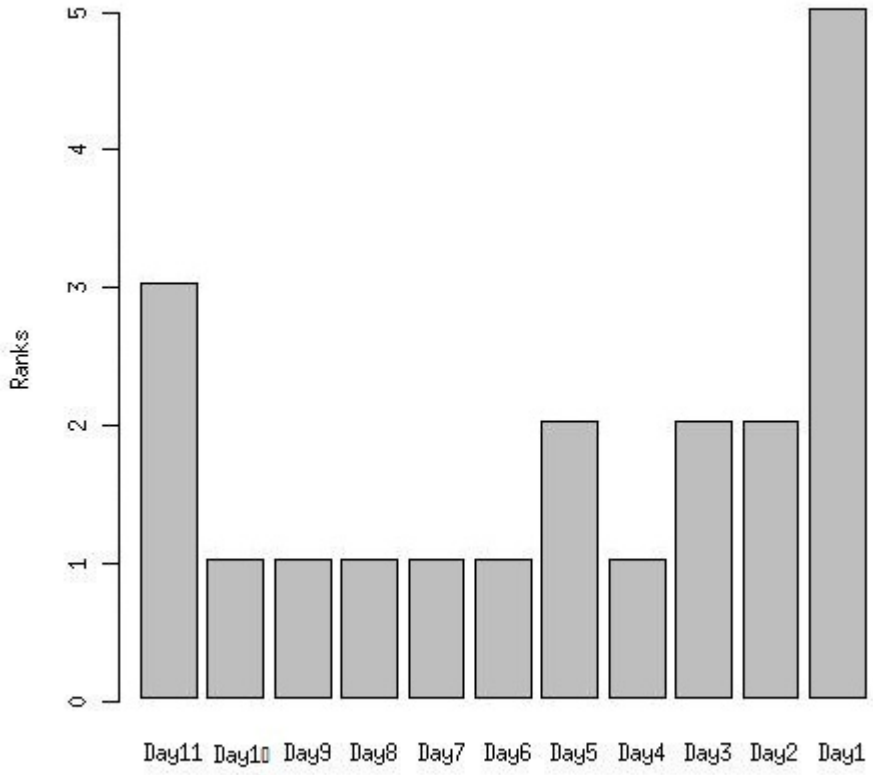


Figure 5.6: Ranks of day 2011-08-16 and last 10 days of Instrument 3_1 (PD)

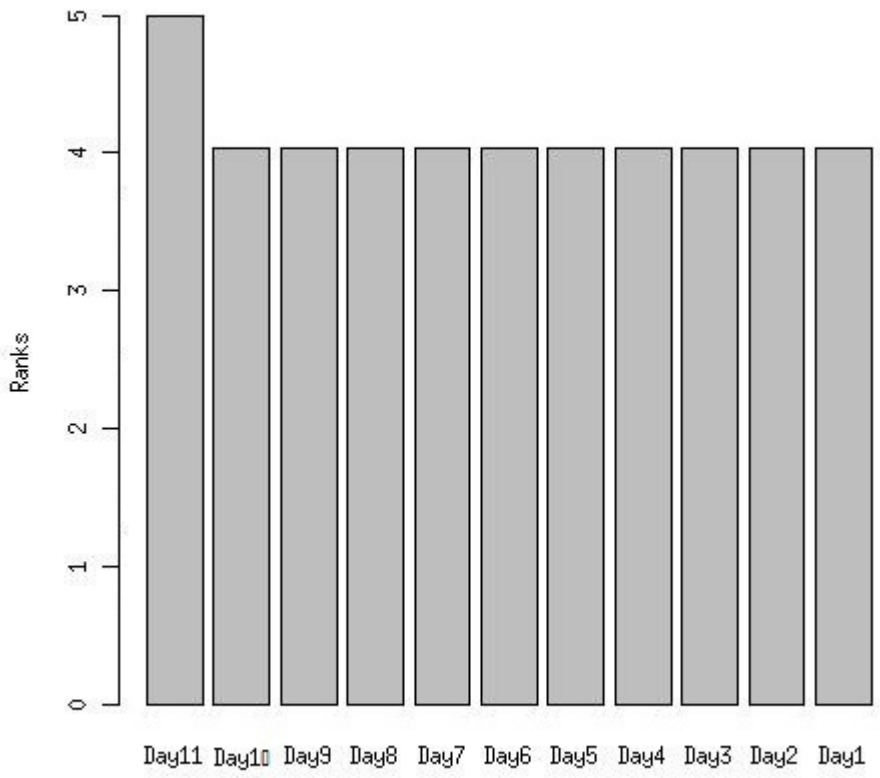


Figure 5.7: Ranks of day 2011-08-16 and last 10 days of Instrument 3_1 (WPD)

For the same instrument and same day, we have 11 days ranks as shown in Figure 5.8 and 5.9 when the meta-clusters are created from the two versions of daily volatility data set.

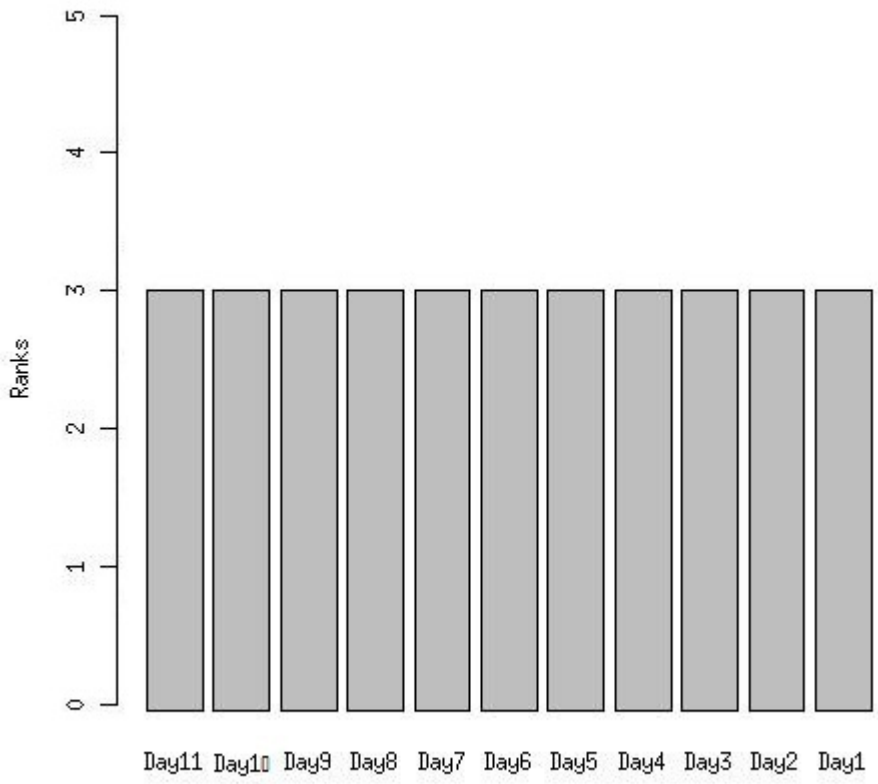


Figure 5.8: Ranks of day 2011-08-16 and last 10 days of Instrument 3_1 (DVD)

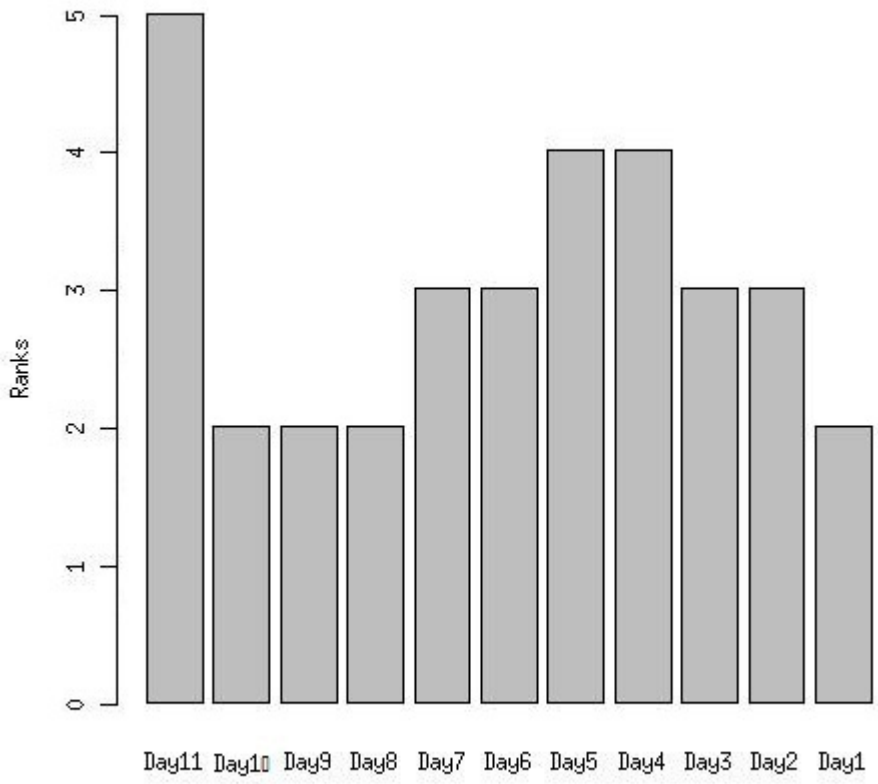


Figure 5.9: Ranks of day 2011-08-16 and last 10 days of Instrument 3_1 (WDVD)

5.4 Computational Requirements and Scalability for the Meta-clustering Algorithm

The primary objective of the proposed meta-clustering algorithm is to generate semantically more meaningful profiles based on connections between granules. It is necessary to strike a balance between reliable and useful profiles versus computational efficiency.

The proposed meta-clustering algorithm has inherent opportunities for parallel processing. Therefore, while it will require significant computational resources, they can be distributed among multiple processors resulting in a reasonable chronological time requirement. In this section, we discuss the computational requirements and describe how the algorithm can be parallelized. The implementation of parallel meta-clustering is a separate research topic in itself, and is being investigated as part of our ongoing research.

The problem of obtaining an optimal clustering scheme is NP-hard. For n objects that need to be grouped into k clusters, each object can be assigned to any one of the k clusters, resulting in $k \times k \times \cdots k = k^n$ possible clustering schemes. The clustering scheme that provides minimum scatter within clusters and maximum separation between clusters will then be selected as the optimal one. Therefore, finding the optimal clustering scheme will require $O(k^n)$ calculations of cluster quality.

It is possible that if the cluster quality measure is carefully chosen, it may be possible to optimize it without having to consider all possible clustering schemes. For example, k-means algorithm can converge towards local minimum for cluster scatter. Running k-means multiple times with different starting centroids increase the chances of finding global minimum without having to consider k^n schemes. Each iteration in k-means requires $O(k \times n)$ distance calculations. Therefore, k-means time requirements are $O(k \times n \times iter)$, where $iter$ is the number of iterations. However, the clustering scheme resulting from k-means depends on the initial choice of cluster centers. As mentioned before, one needs to apply k-means multiple times and choose a clustering scheme that provides minimum scatter within clusters and maximum separation between clusters. However, these multiple runs can be easily run in parallel, keeping the same chronological time.

The proposed meta-clustering algorithm uses multiple applications of a conventional clustering algorithm such as the k-means. In addition, the resulting clustering schemes will be used to create the dynamic representations for each object. The creation of a dynamic representation will require $10 \times n = O(n)$ computations, where n is the number of temporal patterns and we connect them to 10 historical patterns.

As the computational resources, we have used ACEnet system, Unix operating System, R as the tool to cluster, plot and implement the algorithm and Unix script to execute programs written in R. Our experiments used a linear application of the clustering algorithms and took approximately three hours to complete the full execution using R. The linear implementations will require significant chronological time when the values of n are of the order of millions. It is possible to reduce the chronological time in a distributed environment as follows:

1. Apply k-means algorithm in parallel on multiple nodes and choose the clustering scheme with the best quality.

2. The creation of dynamic profiles involves sorting and searching lists. There are many parallel implementations of sorting and searching that can be used to facilitate faster computations.

3. Creation of dynamic profile in parallel by distributing group of records in nodes, as creation of dynamic part for an instance is independent of creation of dynamic part for other instances in same iteration.

5.5 Chapter Summary and Conclusions

In this chapter, we described a meta-clustering process in a temporal environment using financial markets as an example. The daily patterns of the stocks were clustered based on their volatility. We assumed that sustained activity in a stock lasts for a two week pattern (ten trading days). Therefore, each daily pattern was connected to the daily patterns of the same stock from the previous ten days. That means the representation of a daily pattern has data from that day (obtained statically from the database), and extent of volatility of the same stock over the last ten trading days. The historical volatility is recursively derived from the clustering itself. This constitutes the dynamic part of the representation of a daily pattern. However, the daily patterns are clustered using only the static representation initially. The first and subsequent clusterings are used to assign volatility ranks to the historically connected daily patterns.

In conventional clustering, we can only represent the daily volatility of a stock. However, the profiles based on only a single day's volatility cannot provide a much needed historical perspective of the volatility for that stock. A daily pattern of a stock is naturally connected to the daily patterns of the same stock from previous days. Thus, a relative cluster representation might be more convenient and helpful. Therefore, the resulting meta-profiles created as shown here not only describe the current volatility of the stock but whether the stock is at the beginning or and end of a volatility cycle and thus can help traders to make trading decisions.

Chapter 6

Conclusion

This chapter summarizes the research outcome of the thesis and next scope of work. Section 6.1 and 6.2 illustrate the summary and conclusive findings respectively while Section 6.3 represents the future work.

6.1 Summary

With an understanding of the importance to identify temporal pattern groups from a large time series data set that preserves clustering schemes obtained from different heuristic algorithms and to present a temporal historical pattern profile, this thesis is aimed to propose two fold clustering technique that can be convenient solutions for these two issues.

We preprocessed our data set, which consisted of average prices of commodities at 10 minutes interval points, into two sets namely the percentile and Black Scholes daily volatility. Percentile is based on an information granule that consists of a more elaborate distribution of prices during the day whereas Black Scholes daily volatility is a single dimensional

information granule representing volatility of the day. We created two sets of clusters for these two types of volatility representation and used them to create rough cluster ensemble. These ensemble clusters are created on the basis of rough set theory.

We implemented the Recursive Meta-clustering algorithm on the two types of data sets (percentiles, Black Scholes daily volatility) prepared for Rough Ensemble clustering. However, among the 121 days transactions, 10 days transactions have been used to create the static cluster patterns and the remaining transaction days have been used for the dynamic part. We assumed that a sustained activity in a stock lasts for a two week pattern (ten trading days). Therefore, each daily pattern was connected to the daily patterns of the same stock from previous ten days. Therefore, representation of a daily pattern contains data from the day obtained statically from the database, and extent of volatility of the same stock over last ten trading days. The historical volatility is derived repeatedly from the clustering itself. This constitutes the dynamic part of the representation of a daily pattern. Initially, the daily patterns are clustered using only the static representation. The first and subsequent clustering is used to assign volatility ranks to the historically connected daily patterns.

6.2 Conclusions

Clustering ensemble, N-clustering and time series meta-clustering established numerous research. However, Rough Ensemble clustering and Recursive Meta-clustering of daily series data are indeed new dimensions to the trend. According to the previous research results, Recursive Meta-clustering can represent interesting profiles of the data set objects. We found similar results while implementing this technique on time series data. On the other hand, we showed with evidential data that Rough Ensemble clustering can group and

represent different temporal cluster results created on same data with different representations. This ensemble clustering may preserve and show the common (overlapping) as well as distinct temporal groupings derived from the baseline clusterings.

The clusters within the two groupings used in this thesis can be ordered based on their volatility. While these groupings tend to have a common agreement on volatility for most of the daily patterns, they disagree on a small number of patterns. A closer inspection of the patterns suggests that both points of view have their respective merit. This research work proposes a novel Rough Clustering Ensemble algorithm for representing the ambiguity in combined clustering using intervals along with the unambiguous patterns.

In conventional clustering, we can only represent the daily volatility of a stock. The profiles created in this way, based on only a single day's volatility, does not provide a convenient historical perspective of the volatility for that stock. Since a daily pattern of a stock is naturally relative to the daily patterns of the same stock from previous days, a historical cluster representation based on this relation, might be more convenient. Thus, the resulting meta-profiles created here, on the basis of such relations, not only describe the current volatility of the stock but whether the stock is at the beginning or and end of a volatility cycle. This outcome can help traders to make trading decisions.

For both the type of representations, we successfully executed the program and ended up with meta-clusters. Using the meta-clusters obtained in this process, cluster ranks or profiles of any unit time and specified previous periods can be represented. In our case, using the meta-clusters obtained, we represented volatility of any specified day and last 10 days. Naturally, the results for the data set representing percentiles and the data set representing Black Scholes volatility index were not always same, since the two types of

representations consider volatility in different ways. In addition, the normalized data values play significant role while creating meta-clusters. If they are not normalized, the large values of dynamic part may dominate the grouping and therefore effect the result. We observed this evidence while the two versions of each of the data sets representing weighted or normalized and unweighted values resulted in two different meta-profile sets.

6.3 Future Work

The proposed Rough Clustering Ensemble technique will be further studied in situations when there is no obvious ordering of clusters.

For the Temporal Meta-clustering, we had our experiment with financial time series data set, though the proposed algorithm can be used as a generic one that can be useful for any similar time series data set. Experiments on a few other similar time series data sets can be considered as good evidence of the overall idea.

References

Bargiela, Andrzej and Witold Pedrycz (2003), *Granular computing: an introduction*. Springer.

Bezdek, James C. (1981), *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers.

Black, Fischer and Myron Scholes (1973), "The pricing of options and corporate liabilities." *The Journal of Political Economy*, 81, 637–654.

Caruana, Rich, M. Elhaway, Nam Nguyen, and Casey Smith (2006), "Meta clustering." In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, 107–118, IEEE.

Castellano, Giovanna, Anna Maria Fanelli, and Corrado Mencar (2002), "Generation of interpretable fuzzy granules by a double-clustering technique." *Archives of Control Science*, 12, 397–410.

Chatfield, Chris (2013), *The analysis of time series: an introduction*. CRC press.

Corduas, Marcella and Domenico Piccolo (2008), "Time series clustering and classification by the autoregressive metric." *Computational Statistics and Data Analysis*, 52, 1860–1872.

Coronnello, C, M Tumminello, F Lillo, S Micciche, and RN Mantegna (2005), "Sector identification in a set of stock return time series traded at the london stock exchange." *Acta Physica Polonica B*, 36, 2653.

Cotofrei, Paul and Kilian Stoffel (2004), *From temporal rules to temporal meta-rules*, 169–178. Data Warehousing and Knowledge Discovery, Springer.

Cotofrei, Paul and Kilian Stoffel (2009), *Time Granularity in Temporal Data Mining*, 67–96. Foundations of Computational, IntelligenceVolume 6, Springer.

Das, Gautam, King-Ip Lin, Heikki Mannila, Gopal Renganathan, and Padhraic Smyth (1998), "Rule discovery from time series." In *KDD*, volume 98, 16–22.

DUNN, JC (1973), "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters." *J. Cybernetics*, 3, 32–57.

El-Yaniv, Ran and Oren Souroujon (2001), *Iterative double clustering for unsupervised and semi-supervised learning*, 121–132. Machine Learning: ECML 2001, Springer.

Esling, Philippe and Carlos Agon (2012), "Time-series data mining." *ACM Computing Surveys (CSUR)*, 45, 12.

Fern, Xiaoli Z and Wei Lin (2008), "Cluster ensemble selection." *Statistical Analysis and Data Mining*, 1, 128–141.

Figlewski, Stephen (1994), "Forecasting volatility using historical data." *Available at SSRN 5618*.

Fred, Ana (2001), "Finding consistent clusters in data partitions." In *Multiple classifier systems*, 309–318, Springer.

Fu, Tak-chung (2011), "A review on time series data mining." *Engineering Applications of Artificial Intelligence*, 24, 164–181.

Fu, Tak-chung, Fu-lai Chung, Vincent Ng, and Robert Luk (2001), "Pattern discovery from stock time series using self-organizing maps." In *Workshop Notes of KDD2001 Workshop on Temporal Data Mining*, 26–29, Citeseer.

Gao, Can, Witold Pedrycz, and Duoqian Miao (2013), "Rough subspace-based clustering ensemble for categorical data." *Soft Computing*, 9, 1643–1658.

Gao, Jia-Hong and Seong-Hwan Yee (2003), "Iterative temporal clustering analysis for the detection of multiple response peaks in fMRI." *Magnetic resonance imaging*, 21, 51–53.

Ghosh, Joydeep and Ayan Acharya (2011), "Cluster ensembles." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1, 305–315.

Gnatyshak, Dmitry, Dmitry I. Ignatov, Alexander Semenov, and Jonas Poelmans (2012), *Gaining Insight in Social Networks with Biclustering and Triclustering*, 162–171. Perspectives in Business Informatics Research, Springer.

Gnatyshak, Dmitry V., Dmitry I. Ignatov, and Sergei O. Kuznetsov (2013), "From triadic FCA to triclustering: Experimental comparison of some triclustering algorithms." *CLA 2013*, 249.

Grullon, Gustavo, Evgeny Lyandres, and Alexei Zhdanov (2012), "Real options, volatility, and stock returns." *The Journal of Finance*, 67, 1499–1537.

Hartigan, J. A. and M. A. Wong (1979), "Algorithm as 136: A k-means clustering algorithm." *Journal of the Royal Statistical Society.Series C (Applied Statistics)*, 28, 100–108, URL `http://www.jstor.org/stable/2346830`.

Hoai, Minh and Fernando De la Torre (2012), "Maximum margin temporal clustering." In *Proceedings of 15th International Conference on Artificial Intelligence and Statistics*.

Husin, Husna Sarirah, Lishan Cui, Herny Ramadhani Husny Hamid, and Norhaiza Ya Abdullah (2013), "Time series analysis of web server logs for an online newspaper." In *Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication*, 1, ACM.

Ignatov, DI, SO Kuznetsov, LE Zhukov, and J Poelmans (2013), "Can triconcepts become triclusters?" *International Journal of General Systems*, 42, 572–593.

Ignatov, Dmitry I., Sergei O. Kuznetsov, Ruslan A. Magizov, and Leonid E. Zhukov (2011), *From triconcepts to triclusters*, 257–264. Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, Springer.

Ignatov, Dmitry I., Sergei O. Kuznetsov, and Jonas Poelmans (2012), "Concept-based biclustering for internet advertisement." In *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on*, 123–130, IEEE.

Joshi, Anupam and Raghu Krishnapuram (1998), "Robust fuzzy clustering methods to support web mining." In *Proc. Workshop in Data Mining and knowledge Discovery, SIGMOD*, 15–1, Citeseer.

Joshi, Manish and Pawan Lingras (2009), "Evolutionary and iterative crisp and rough clustering ii: Experiments." In *Pattern Recognition and Machine Intelligence*, 621–627, Springer.

Karoui, Nicole El, Monique JeanblancPicqu, and Steven E. Shreve (1998), "Robustness of the black and scholes formula." *Mathematical finance*, 8, 93–126.

Kremer, Hardy, Stephan Gunnemann, and Thomas Seidl (2010), "Detecting climate change in multivariate time series data by novel clustering and cluster tracing techniques." In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, 96–97, IEEE.

Li, Zhenhui, Bolin Ding, Jiawei Han, Roland Kays, and Peter Nye (2010), "Mining periodic behaviors for moving objects." In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1099–1108, ACM.

Liao, T. Warren (2005), "Clustering of time series dataa survey." *Pattern Recognition*, 38, 1857–1874.

Lingras, Pawan (2001), "Unsupervised rough set classification using gas." *Journal of Intelligent Information Systems*, 16, 215–228.

Lingras, Pawan, Ahmed Elagamy, Asma Ammar, and Zied Elouedi (2014), "Iterative meta-clustering through granular hierarchy of supermarket customers and products." *Information Sciences*, 257, 14–31.

Lingras, Pawan and Farhana Haider (2015a), "Combining rough clustering schemes as a

rough ensemble." In *Proceeding of the 2015 International Joint Conference on Rough Sets (IJCRS 2015)*. Accepted.

Lingras, Pawan and Farhana Haider (2015b), "Rough ensemble clustering." *Special Issue of the journal Intelligent Data Analysis*, 19, to appear.

Lingras, Pawan, Mofreh Hogo, and Miroslav Snorek (2004), "Interval set clustering of web users using modified kohonen self-organizing maps based on the properties of rough sets." *Web Intelligence and Agent Systems*, 2, 217–225.

Lingras, Pawan and Kishore Rathinavel (2012), "Recursive meta-clustering in a granular network." In *Intelligent Systems Design and Applications (ISDA), 2012 12th International Conference on*, 770–775, IEEE.

Lingras, Pawan and Chad West (2004), "Interval set clustering of web users with rough k-means." *Journal of Intelligent Information Systems*, 23, 5–16.

Liu, Zhijian and Roy George (2003), *Fuzzy cluster analysis of spatio-temporal data*, 984–991. Computer and Information Sciences-ISCIS 2003, Springer.

MacQueen, James (1967), "Some methods for classification and analysis of multivariate observations." In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, 14, California, USA.

Mirkin, B. (1996), "Mathematical classification and clustering kluwer."

Mitra, Sushmita (2004), "An evolutionary rough partitive clustering." *Pattern Recognition Letters*, 25, 1439–1449.

Nanni, Mirco and Dino Pedreschi (2006), "Time-focused clustering of trajectories of moving objects." *Journal of Intelligent Information Systems*, 27, 267–289.

Pawlak, Zdzisław (1982), "Rough sets." *International Journal of Computer and Information Sciences*, 11, 341–356.

Pawlak, Zdzisław (1984), "Rough classification." *International Journal of Man-Machine Studies*, 20, 469–483.

Pawlak, Zdzisław (1992), "Rough sets: A new approach to vagueness." In *Fuzzy logic for the management of uncertainty*, 105–118, John Wiley & Sons, Inc.

Pedrycz, Witold, Andrzej Skowron, and Vladik Kreinovich (2008), *Handbook of granular computing*. John Wiley & Sons.

Peters, Georg (2006), "Some refinements of rough k-means clustering." *Pattern Recognition*, 39, 1481–1491.

Peters, Georg, Fernando Crespo, Pawan Lingras, and Richard Weber (2013), "Soft clustering–fuzzy and rough approaches and their extensions and derivatives." *International Journal of Approximate Reasoning*, 54, 307–322.

Polkowski, Lech and Andrzej Skowron (1996), "Rough mereology: A new paradigm for approximate reasoning." *International Journal of Approximate Reasoning*, 15, 333–365.

Povinelli, Richard J. and Xin Feng (1999), "Data mining of multiple nonstationary time series." *proceedings of Artificial Neural Networks in Engineering, St.Louis, Missouri*, 511–516.

Ramirez-Cano, Daniel, Simon Colton, and Robin Baumgarten (2010), "Player classification using a meta-clustering approach." In *Proceedings of the 3rd Annual International Conference Computer Games, Multimedia and Allied Technology*, 297–304.

Ratanamahatana, Chotirat Ann, Jessica Lin, Dimitrios Gunopulos, Eamonn Keogh, Michail Vlachos, and Gautam Das (2010), *Mining time series data*, 1049–1077. Data Mining and Knowledge Discovery Handbook, Springer.

Rathinavel, Kishore and Pawan Lingras (2013), "A granular recursive fuzzy meta-clustering algorithm for social networks." In *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint*, 567–572, IEEE.

Siersdorfer, Stefan and Sergej Sizov (2004), "Restrictive clustering and metaclustering for self-organizing document collections." In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 226–233, ACM.

Skowron, Andrzej and Jaroslaw Stepaniuk (1999), "Information granules in distributed environment." In *New Directions in Rough Sets, Data Mining, and Granular-Soft Computing*, 357–365, Springer.

Slonim, N. and N. Tishby (2000), "Document clustering using word clusters via the information bottleneck method." In *23rd annual international ACM SIGIR conference on Research and development in information retrieval*, 208–215.

Sravya, Alla Naga and M Nalini Sri (2013), "A novel approach of temporal data clustering via weighted clustering ensemble with different representations." 4, 623–629.

Strehl, Alexander and Joydeep Ghosh (2003), "Cluster ensembles—a knowledge reuse framework for combining multiple partitions." *The Journal of Machine Learning Research*, 03, 583–617.

Triff, Matt and Pawan Lingras (2013), *Recursive Profiles of Businesses and Reviewers on Yelp. com*, 325–336. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Springer.

Vega-Pons, Sandro and José Ruiz-Shulcloper (2011), "A survey of clustering ensemble algorithms." *International Journal of Pattern Recognition and Artificial Intelligence*, 25, 337–372.

Viovy, N. (2000), "Automatic classification of time series (acts): a new clustering method for remote sensing time series." *International Journal of Remote Sensing*, 21, 1537–1560.

Wijk, Jarke J. Van and Edward R. Van Selow (1999), "Cluster and calendar based visualization of time series data." In *Information Visualization, 1999.(Info Vis' 99) Proceedings. 1999 IEEE Symposium on*, 4–9, 140, IEEE.

Yan, Rui (2004), "Temporal mining of the web and supermarket data using fuzzy and rough set clustering."

Yang, Jaewon and Jure Leskovec (2011), "Patterns of temporal variation in online media." In *Proceedings of the fourth ACM international conference on Web search and data mining*, 177–186, ACM.

Yao, JingTao (2007), "A ten-year review of granular computing." In *Granular Computing, 2007. GRC 2007. IEEE International Conference on*, 734–734, IEEE.

Yao, YY (2008), "Granular computing: past, present, and future." *Lecture Notes in Computer Science*, 5009, 27–28.

Yao, YY (2010), "Human-inspired granular computing." *Novel developments in granular computing: applications for advanced human reasoning and soft computation*, 1–15.

Yao, YY, X Li, TY Lin, and Q Liu (1994), "Representation and classification of rough set models." In *Proceeding of Third International Workshop on Rough Sets and Soft Computing*, 630–637.

Zadeh, Lotfi A. (1979), "Fuuzy sets and information granularity." *Advances in Fuzzy Set Theory and Applications*, 18.

Zadeh, Lotfi A. (1997), "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic." *Fuzzy sets and systems*, 90, 111–127.

Zhang, Xiaohang, Jiaqi Liu, Yu Du, and Tingjie Lv (2011), "A novel clustering method on time series data." *Expert Systems with Applications*, 38, 11891–11900.