Int. J. Advanced Networking and Applications
Volume: 03, Issue: 04, Pages:1281-1291 (2012)

Clustering Approach to Stock Market Prediction

[1]M.Suresh Babu, [2]Dr. N.Geethanjali, [3]Prof B.Satyanarayana
[1]Principal, Intel Institute of Science, Anantapur, Andhra Pradesh , India.
[2]Associate Professor, Department of Computer Science, S.K. University, Anantapur. India
[3]Professor & Chairman, Board of Studies, Department of Computer Science, Sri Krishnadevaraya Univesity, Anantapur.

----------ABSTRACT----------
Clustering is an adaptive procedure in which objects are clustered or grouped together, based on the principle of maximizing the intra-class similarity and minimizing the inter-class similarity. Various clustering algorithms have been developed which results to a good performance on datasets for cluster formation. This paper analyze the major clustering algorithms: K-Means, Hierarchical clustering algorithm and reverse K means and compare the performance of these three major clustering algorithms on the aspect of correctly class wise cluster building ability of algorithm. An effective clustering method, HRK (Hierarchical agglomerative and Recursive K-means clustering) is proposed, to predict the short-term stock price movements after the release of financial reports. The proposed method consists of three phases. First, we convert each financial report into a feature vector and use the hierarchical agglomerative clustering method to divide the converted feature vectors into clusters. Second, for each cluster, we recursively apply the K-means clustering method to partition each cluster into sub-clusters so that most feature vectors in each subcluster belong to the same class. Then, for each sub cluster, we choose its centroid as the representative feature vector. Finally, we employ the representative feature vectors to predict the stock price movements. The experimental results show the proposed method outperforms SVM in terms of accuracy and average profits.

Date of Submission: September 15, 2011 Date of Acceptance: November 20, 2011

1.0 Classification and prediction

We treat many things as a group of things e.g., financial data, stock quotes etc. In order to define a class (a group of entities) a set of models that define and distinguish data classes or concept are delineated together. Using this class we get the ability to predict whether any new model belongs to this class or not i.e. a data model for whose class value is unknown can be predicted based on classification rules. There are various ways to apply rules e.g., classification (if-then) rules, decision trees, mathematical formula or neural networks. Classification can be used to predict the class label of the data object. However, classification is most useful in predicting certain missing values or unavailable data within a class. Normally, when classification is used to predict missing values in numeric data this is referred to as prediction. Data values prediction is more useful over class label assignment to an unknown object.

1.1.1 Cluster Analysis

Clustering, analysis a data set without consulting a known class label. Class labels are not present in the training data, as they are not known to begin with. Clustering is used to divide a data set into classes (by generating labels for them) using the principle of maximizing the intra class similarity and minimizing inter class similarity. Within the data set clusters are formed so that objects which are similar are grouped together and objects that are very different fall into other clusters. Clustering, thus also facilitates taxonomy formation i.e. organization of observed objects into a hierarchy of classes that group similar things together.

1.1.2. The problem of classification

The problem of classification is to learn the mapping from an input vector to an output class in order to generalize to new examples it has not necessarily seen before.

Classification

Classification rather than continuous function approximation will be the focus because it is the most common question to be answered in data mining situations. Binary classification is the frequently encountered situation where there are two categories. A set of cases or instances is partitioned into two subsets based on whether each has or does not have a particular property. Binary classification is also our focus because there are clear criteria for judging binary classification efforts percentage correctly classified and receiver operating characteristic (ROC) curves. Increased knowledge of the accuracy of various classification methods will allow data mining analysts to select from those that are most effective. Knowledge of which classifiers perform best may suggest directions for those seeking to construct new algorithms or to improve upon existing ones.

1.2 Stock Market Prediction using Classification:

Stock market prediction is an appealing topic not only for research but also for commercial applications. In stock market research, the random walk theory (Malkiel 1973) suggested that short term stock price movements were governed by the random walk hypothesis and thus were

unpredictable. On the other hand, the efficient market hypothesis (Fama 1964) stated that the stock price was a reflection of complete market information and the market behaved efficiently so that instantaneous price corrections to equilibrium would make stock prediction useless. However, prior researches (Brown & Jennings 1989; Abarbanell & Bushee 1998) made use of a variety of methods to gain future price information. They proposed two types of stock market analysis. First, the fundamental analysis derives stock price movements from financial ratios, earnings, and management effectiveness. Second, the technical analysis identifies the trends of stock prices and trading volumes based on historical prices and volumes.

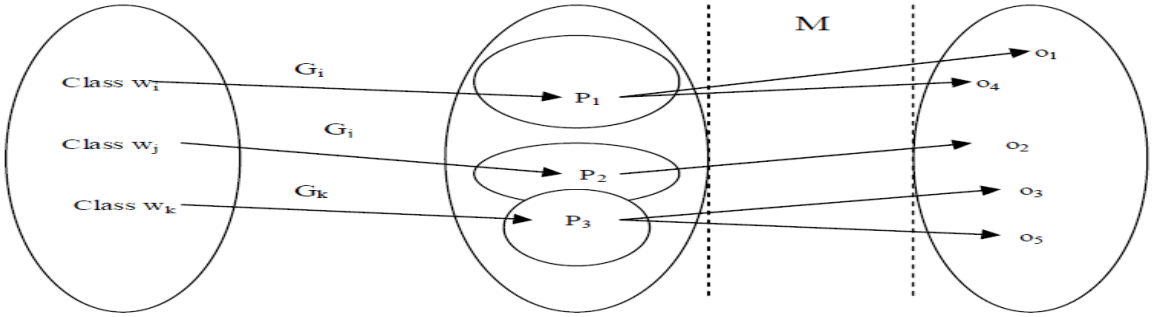
Stock market prediction based on structured data such as price, trading volume and accounting items has been widely employed on numerous researches (Chan et al. 2002; Lin et al. 2009). However, it is much more difficult to predict stock price movements based on unstructured textual data. One kind of unstructured textual data for stock market prediction is collected from financial news published on the newspapers or Internet. The methods used news articles to predict stock prices in a short period after the release of news articles (Schumaker & Chen 2009). Another kind of unstructured textual data is gathered from financial reports, which contain not only textual data but also numerical data. The numerical data provides quantitative information and the textual data contains a large amount of qualitative information related to the company performance and future financial movements. Moreover, incorporating the quantitative and qualitative information into stock market analysis can improve the prediction ability (Chen et al. 2009; Kogan et al. 2009). Thus, we propose a method and use both quantitative and qualitative information in financial reports to predict stock price movements.

1.2.1 Pattern recognition

Pattern recognition can be defined as the science relevant to the description or classification of measurements. It is generally characterized as an information reduction, information mapping or information labeling process. As an important component in a lot of intelligent systems for data preprocessing and decision making, pattern recognition techniques have found application in multiple areas such as image processing, seismic analysis, financial forecasting, medical diagnosis, etc. Information is always a manifest of multiple factors resulting in the form of patterns. Stock price time series can hence be seen as a series of different patterns resulted from trading activities involving multiple market participants including institutional and individual traders.

1.2.2 Classification and clustering

Classification and clustering are both the central concepts of pattern recognition. Both the techniques are used as important knowledge discovery tools in modern machine learning process. Classification assigns input data into one or more pre-specified classes based on extraction of significant features or attributes and the processing or analysis of these attributes. A high level abstraction of the classification problem is depicted in the following figure:



Class membership of measurement space o — pattern space P — observation or Description space C

Figure 1.1: An example of mapping in an abstract representation of pattern generation/classification system

A mapping can be postulated between the pattern space, P and the class membership space, C via a relationship G. In other words, each class, w_n where $n = i, j, k,....$ generates a subsets of patterns in the pattern space where the pattern is denoted by p_n, where $n = i, j, k,.....$ Another relation, M can be used to map patterns from subspace of P into the observations denoted by o_n , where $n = i, j, k,.....$From the above illustration, classification or characterization can be understood simply as the process of identifying and inverting the mappings G and M for all $n = i, j, k,.....$ Classification generally uses supervised training/learning approaches by assuming a labeled training set. However, in many cases, there is no clear structure in the sampled data or training set where classes are not defined. The number of classes is unknown and the relationship of typical attributes to the classes may not seem obvious.

1.2.3 Cluster Analysis

Clustering, analysis a data set without consulting a known class label. Class labels are not present in the training data, as they are not known to begin with. Clustering is used to divide a data set into classes (by generating labels for them) using the principle of maximizing the intra class similarity and minimizing inter class similarity. Within the data set clusters are formed so that objects which are similar are grouped together and objects that are very different fall into other clusters. Once the data derived for

any object inclusion Clustering, thus also facilitates taxonomy formation i.e. organization of observed objects into a hierarchy of classes that group similar things together.

Cluster analysis is the automatic identification of groups of similar objects or patterns. For example, if a set of data denoted by *x*, is very similar to a few other sets of data, we may intuitively tend to group *x* and these sets of data into a natural cluster. By maximizing inter group similarity and minimizing intra group similarity, a number of clusters would form on the measurement/observation space. We can then easily recognize and assign to the clusters suitable label or feature description.

There are generally two types of learning approaches relevant to cluster analysis. The parametric partitional approach attempts to cluster the set directly, in a manner that depends on a set of parameters. These parameters are then adjusted to optimally satisfy a chosen criterion of separation and compactness of clusters. Whereas, the non-parametric approach hierarchical approach proceeds from a provisional initial clustering and iteratively merges/or split clusters until a required degree of similarity holds for the elements of the clusters.

There have been a lot of works in the area of cluster analysis. Generally some typical requirements for a good clustering technique can be defined (Han and Kambert,2000):

- Scalability: The cluster method should be applicable to huge data sets and performance should decrease linearly with data size increase.
- Versatility: Clustering objects could be of different types – numerical data, Boolean data, categorical data, time series, etc. Ideally a clustering method should be suitable for all different types of data objects.
- Ability to discover clusters with different shapes: This is an important requirement for spatial data clustering. Many clustering algorithms can only discover clusters with spherical shapes.
- Minimal input parameter: The method should require a minimum amount of domain knowledge for correct clustering. However, most current clustering algorithms have several key parameters and they are thus not practical for use in real world applications.
- Robust with regard to noise: This is important because noise exists everywhere in practical problems.

1.2.4 Mutual information

Cluster analysis is used as a tool to examine the dependence between time series. The most common measure of dependence between two sets of random variables is probably the coefficient of linear correlation. However, the usage of correlation coefficients is only limited to cases where there exists a pure linear relationship or at least a linear transformed relationship (Granger and Lin, 1994; Bernhard and Darbellay, 1999).

As most of the real-world data sets, such as economic variables, display a non-linear dependencies (if there should exist), a measure of 'global' dependence that captures both the linear and non-linear relationships is required. Mutual information, originated from the theory of communication, can be used to examine the 'global' dependencies in time series. Shannon first introduces the measure of 'mutual information' in 1948. The theory was then generalized and extended to use mutual information as a measure of dependence (Bernhard and Darbellay, 1999; Darbellay 1997).

Mutual information can be derived from the computation of entropies. Entropy H(X) is defined as the uncertainty about *X*. For discrete distributions, H(X) is given by:

$$H(X) = -\sum_{k=-K}^{K} p_k \log p_k$$

Where X is the discrete random variable as X = $\{x_k | k= 0,\pm1 \pm 2.....\pm k\}$

$\mathbf{P_k = P(X = x_k)}$

H(X,Y) is called the joint entropy of X and Y and is given by :

$$H(X,Y) = -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(x,y)$$

where p(x, y) = joint probability density function.

Mutual information between two time series is defined by the following expression:

$$I(X,Y) = H(X) + H(Y) - H(X,Y)$$

Based on the definition of mutual information, Pompe (1998) presents some of the properties of mutual information to the discrete variables:

(i) $I(X,Y) = 0$ if and only if *X* and *Y* are statistically independent

(ii) $I(X,Y) = H(X)$ if and only if *X* is a function of *Y*

(iii) $I(X,Y) = H(Y)$ if and only if *Y* is a function of *X*

1.2.5 Global correlation coefficient

For cluster analysis, some criteria have been defined to qualify a method as a good measure of linear and non-linear dependence (Granger *et al.*, 2002):

a) Must be well defined for both the discrete and continuous cases

b) Must be normalized so that value lies between -1 and +1, and value = 0 if the sets of variables are found to be statistically independent

c) The modulus value should equal 1 if there is an exact non-linear relationship between the sets of variables

d) Must be similar to the linear correlation coefficient in the case of bivariate normal distribution so that comparison can be made to satisfy the above-mentioned properties, a standard measure called global correlation coefficients, derived from mutual information is defined:

$$\lambda = \sqrt{1 - e^{-2I(\vec{X},\vec{Y})}}$$

Where $I(\vec{X},\vec{Y})$ is the mutual information of $\vec{X}, \vec{Y}$

The values of λ ranges from 0 to 1 and is able to capture the 'global' dependence, both linear and non linear, hence is most suitable to be used as the dependence or similarity measure used in most cluster analysis.

1.3 k-means clustering

The *k* means algorithm is an unsupervised partitioning method used for automatic classification. The method begins by choosing *k* classes with an initial guesses of *k* reference points. Based on their respective distances to these reference points, the data is then segregated into *k* distinct regions. Consider a training set $X = (x_1, x_2, \ldots, x_M)$

where $x_i \in R^n$ is an n dimension vector in Euclidean space and $i = 1, 2, \ldots, M$. The segregation of the training set into k clusters using the initial guesses of cluster centroids $Y = (y1, y2, \ldots, yk)$ where $y_j \in R^n$ and $j = 1, 2, \ldots, k$ is performed by minimizing the cost function D:

$$D = \frac{1}{M} \sum_{i=1}^{M} \min_{y_j \in Y} (d(x_i, y_j))$$

Each region is then evaluated with the reference point as the centroid of the region. Data points are 'clustered' with the reference point based on 'nearest neighbour' evaluation. The nearest neighbour description is defined by the membership function:

$$u_j(x_i) = \begin{cases} 1 & if \quad d(x_i, y_j) = d_{\min}(x_i) \\ 0 & otherwise \end{cases}$$

The initial reference points (cluster centers) will then be recomputed and moved to minimize the distance between itself and its members.

$$y_j = \frac{\sum_{i=1}^{M} u_j(x_i) x_i}{\sum_{i=1}^{M} u_j(x_i)}$$

This membership is reassessed in each iteration until the algorithm converges upon a solution (the movement of the reference points or centroids approaches zero).

1.3.1 Data

For the cluster analysis, we have chosen the interday stock price time series instead of the intraday stock price data. The choice is made considering the fact that the intraday data is very much "noisier" and more of random nature. Price changes for different stock counters over short period at intraday level of resolution present few distinct features, making it difficult to differentiate for clustering and classification. For the proposed implementation, we have used a dataset of 35 randomly selected interday stock price time series. The stocks chosen span a variety of industries including service, hotel, telecommunication, consumer, plantation and construction. It is believed that this diversity will make our data more suitable for the testing of our proposed clustering and classification framework

1.3.2. Dimension reduction and clustering

We use the Sammon mapping codes from the SOM toolbox to achieve dimension reduction. In our implementation of Sammon mapping, we use the 'global correlation coefficient' computed as the stress function. Deriving from mutual information, global correlation coefficient measures the linear and non-linear dependence between different stock price time series and is most suitable for our cluster analysis.

We choose Sammon mapping for dimension reduction, for its unsupervised learning capability and its ability to handle the non linearity nature of the stock price time series. By projecting the input data onto two or three dimensional visualization space, Sammon mapping also automatically separates the stocks into different clusters. Through this projection, we can better understand the underlying structure of the data and determine the clusters through visual interpretation. However, Sammon mapping implementation is more important in providing low dimension representations for the subsequent auto classification procedure using K-means algorithm.

1.3.3 Classification

The last step of the proposed stock clustering analysis framework is automated classification using k-means algorithm. We, again, make use of the codes from the SOM software package (Vesanto *et al.,* 2000) to achieve our purpose. The algorithm automatically classifies the stocks to k different groups represented by the centroids. The complication involved in the implementation of this algorithm is perhaps to choose the 'right' k corresponding to the natural number of clusters in the data. For our k-means implementation, we use two validation methods to verify the number of natural cluster and hence the accuracy of cluster analysis and classification: Davies–Bouldin index and the minimum sum square error (SSE) method. Davies-Bouldin index is a function of the ratio of the sum of within cluster scatter to between cluster separations, it uses both the clusters and their sample means. The minimum of the Davies-Bouldin function can be used to locate the optimal number of clusters. The second method, minimum SSE, is based on the monotonically decreasing property of the cost function when k increased. Empirically, the cost function decreases regularly up to certain point and then slows down. This point corresponds to the optimal number of clusters.

1.4 Effective Clustering Method

K-MEANS CLUSTERING: The basic step of k-means clustering is simple. In the beginning, we determine number of cluster K and we assume the centroid or center of these clusters. We can take any random objects as the initial centroids or the first K objects can also serve as the initial centroids. Then the K means algorithm will do the three steps below until convergence.Iterate until *stable* (= no object move group):

1. Determine the centroid coordinate
2. Determine the distance of each object to the centroids
3. Group the object based on minimum distance (find the closest centroid). This is showed in figure 1.2 in steps.

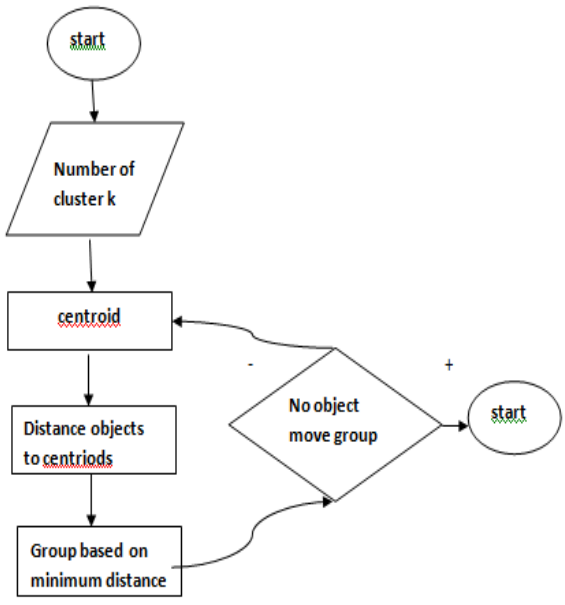


Figure 1.2 K means clustering Process.

Volume: 03, Issue: 04, Pages:1281-1291 (2012)

The K-means clustering method (K-means for short) is a widely-used clustering method. However, its major disadvantages can be described in two aspects. First, the number of clusters is often unknown in different datasets but it is required to be specified in advance. Second, randomly choosing initial centroids of the clusters makes it impossible to obtain reliable results. On the other hand, HAC (Hierarchical Agglomerative Clustering method) produces better resultant clusters and provides a more interpretative hierarchical understanding of the document collection (Steinbach et al. 2000).

However, as the size of a cluster grows, the centroid of a cluster might no longer be adequate to represent any feature vectors in the cluster. This drawback makes further investigation into the characteristics of the clusters difficult. Numerous hybrid methods have been made to mitigate the disadvantages in both approaches. Cheu et al. (2004) combined the K-means, HAC or SOM (Self-Organizing Maps) for the two-level clustering. In the first level of clustering, the prototypes of vectors are generated to reduce the number of samples for the second level of clustering. Chen et al. (2005) and Hu et al. (2007) presented a hybrid clustering method by using HAC to divide the data into clusters and then using K-means to group the clusters generated by HAC. Han et al. (2009) proposed the parameter free hybrid clustering algorithm, which uses HAC to generate initial clustering and then iteratively uses K-means to choose the best number of centroids.

Therefore, in this paper, we propose an effective clustering method, which combines the advantages of K-means and HAC, to perform stock market prediction. Hierarchical clustering algorithms are either top-down or bottom-up. Bottom-up algorithms treat each document as a singleton cluster at the outset and then successively merge (or *agglomerate*) pairs of clusters until all clusters have been merged into a single cluster that contains all documents. Bottom-up hierarchical clustering is therefore called *hierarchical agglomerative clustering* or *HAC*. Top-down clustering requires a method for splitting a cluster. It proceeds by splitting clusters recursively until individual documents are reached. Unlike the previous hybrid clustering methods, we first utilize HAC to do the initial clustering and then *recursively* perform K-means to do the second clustering. The proposed method consists of three phases. First, we convert each financial report into a feature vector and use HAC to divide them into clusters. Second, for each cluster, we recursively apply K-means to partition each cluster into sub-clusters so that most feature vectors in each sub-cluster belong to the same class. Then, for each sub-cluster, we choose its centroid as the representative feature vector. Finally, we employ the representative feature vectors to predict the stock price movements.

First, we use a weight to consolidate both qualitative and quantitative features to analyze financial reports. Second, we combine the advantages of the K-means and HAC methods to develop an effective clustering method to cluster financial reports and select the representative feature vectors. Third, we employ the proposed method to investigate the relationships between financial reports and short-term stock price movements. Finally, the experimental results show the proposed method outperforms SVM (Support Vector Machine) in terms of accuracy and average profits.

1.4.1 Review of Effective clustering method

The methods used unstructured textual data to predict stock prices or market indices have to extract relevant information from a large number of text documents. LeBaron et al. (1999) suggested that the relationships between news articles and stock prices do exist. They developed a stock trading system with simulated traders and discovered a lag between the release of information and the price movements. Lavrenko et al. (2000) employed naïve Bayes and language model to predict forthcoming trends in stock price. Schumaker & Chen (2009) employed SVM to predict stock prices at the time of news release and showed that their model containing both article terms and stock price had the best performance on predicting the stock prices of twenty minutes later.

Public companies are required to file periodic financial reports through the SEBI to section 13 or 15(d) of the Securities Exchange Act. Thus, the financial reports are important data sources for stock market prediction. Many methods used the numerical information of the financial reports to predict stock price movements (Carnes 2006; Chen & Zhang 2007). Besides, Kloptchenko et al. (2004) suggested that the textual information in the financial reports contains not only the description of events, but also explains why they have happened and how long the effect of such events will continue. Chen et al. (2009) built an earning prediction model by incorporating the textual information about the risk sentiment contained in financial reports, which significantly improved the accuracy of earning prediction. Moreover, the textual information holds some forward looking statements about the future performance of the company. Exploiting the related textual information in addition to the numeric information should increase the quality of prediction.

Back et al. (2001) used SOMs to cluster the companies based on the quantitative and qualitative information in the annual reports. They compared the resultant clusters and suggested that the performance of considering both quantitative and qualitative information is better than that of using just quantitative or qualitative information. Kloptchenko et al. (2004) combined SOMs and prototype matching methods to analyze the quantitative and qualitative information of quarterly reports. The experimental results suggested that the quantitative part reflects the past financial performance, but the qualitative part holds some messages about the future performance of the companies. Magnusson et al. (2005) analyzed the effects of seven financial ratios by SOMs and the effects of the qualitative data by collocational networks (Williams 1998). They concluded that: (1) a change in the textual data usually indicates a change in the financial data of the following quarter; and (2) the relationship is a consequence of the fact that the texts reflect the plans and future expectations, whereas the ratios reflect the current financial situation of the company.

Many stock prediction methods based on SVM have been proposed (Qiu et al. 2006; Schumaker & Chen 2009). Qiu et al. (2006) built SVM based predictive models with different feature selection methods from ten years of annual reports. The results showed that document frequency threshold is efficient in reducing feature space while maintaining the same classification accuracy compared with other feature selection methods. Furthermore, the results showed the feasibility of using text classification on current year's annual reports to predict next year's company financial performance, namely the return on equity ratio.

It has been shown that the performance of considering both quantitative and qualitative information is better than that of using just quantitative or qualitative information. However, quantitative and qualitative information of financial reports are considered separately in the previous studies (Back et al. 2001; Kloptchenko et al. 2004; Magnusson et al. 2005). In this paper, we use a weight to combine both qualitative and quantitative information together and propose an effective clustering method to predict the stock price movements.

1.4.2 PROPOSED FRAMEWORK

We first extract a feature vector for each financial report. Each feature vector comprises two parts, namely qualitative and quantitative. The qualitative part is extracted from the textual contents of the financial reports. To obtain the qualitative part, we first transform financial reports into bag of words by the stemming algorithm (Porter 1980) and removing stop words. Then, we compute the TF-IDF weight of each term by multiplying the term frequency and the inverse document frequency. The term frequency tft, d represents the number of occurrences of term t in the financial report d. The inverse document frequency idft is defined as log2(n/dft), where n is the total number of financial reports in the collection, and dft is the number of financial reports containing term t in the collection. We select the terms with top k TF-IDF weights to form the qualitative part of a feature vector. In addition, the quantitative part of a feature vector comprises some ratios about the performance of the company. Based on the prior research (Magnusson et al. 2005), we select five important financial ratios regarding company performance, namely operating margin, return on equity (ROE), return on total assets (ROTA), equity to capital, and receivables turnover. Incorporating the qualitative information with the quantitative information of the financial reports may generate more valuable information to explain the stock price dynamics.

Thus, each feature vector contains k qualitative features and five quantitative features. The similarity between two feature vector, f_1 and f_2, is defined by α times the Euclidean distance of qualitative features plus 1-α times the Euclidean distance of quantitative features of f_1 and f_2, where the combination weight α is used to measure the relative importance of qualitative and quantitative features.

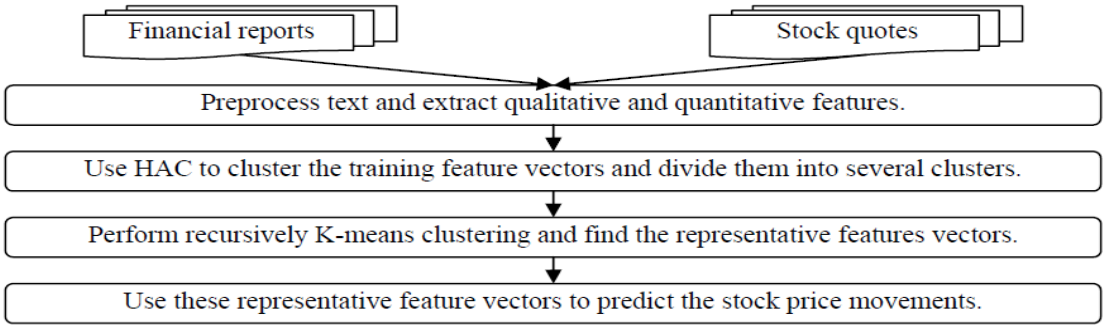


Figure 1.3 : The proposed framework for stock market prediction.

To distinguish the influence of financial reports on the direction of stock price movements, we classify the financial reports into three categories: "rise", "no movement", and "drop", which are represented by 1, 0, and -1, respectively. Specifically, we follow the categorization scheme used in Mittermayer (2004). We define the time window for a financial report from the release day to one trading day after the release. Then, we label a financial report as "rise" if it leads to a peak, with an increase of at least 3% and triggers a shift of average price at least 2% above the open price of the release day during the defined time window. Similarly, we label a financial report as "drop" if it leads to a drop, with a decrease of at least 3% and triggers a shift of average price at least 2% below the open price of the release day during the defined time window.

Next, we propose an effective clustering method, HRK (Hierarchical agglomerative and Recursive K means clustering), for stock market prediction as shown in Figure 1.2. The proposed method consists of three phases.

First, we apply HAC to cluster the training feature vectors and divide them into clusters.

Second, from the clusters generated by HAC, we recursively perform K-means to accomplish further clustering until the purity of the cluster exceeds a predefined purity threshold p, where the purity is defined as the number of feature vectors of the dominant class divided by the total number of feature vectors in the cluster. Then, we compute the centroid for each cluster. The centroids are called the representative feature vectors of the clusters. Finally, we use these representative feature vectors to predict the stock price movements.

1.4.3 Hierarchical Agglomerative Clustering

First, we perform HAC to do initial clustering and construct a dendrogram, where the centrioid clustering is used and the similarity is computed by the Euclidean

distance between feature vectors. An HAC clustering is typically visualized as a *dendrogram* as shown in Figure . Each merge is represented by a horizontal line. The y-coordinate of the horizontal line is the similarity of the two clusters that were merged, where documents are viewed as singleton clusters. We call this similarity the *combination similarity* of the merged cluster. By moving up from the bottom layer to the top node, a dendrogram allows us to reconstruct the history of merges that resulted in the depicted clustering.

A fundamental assumption in HAC is that the merge operation is *monotonic*. Monotonic means that if s_1, s_2,……s_{k-1} are the combination similarities of the successive merges of an HAC, then $s_1 \geq s_2 \geq \ldots . \geq s_{k-1}$ holds. A non-monotonic hierarchical clustering contains at least one *inversion* and contradicts the fundamental assumption that we chose the best merge available at each step.

Hierarchical clustering does not require a pre specified number of clusters. However, in some applications we want a partition of disjoint clusters just as in flat clustering. In those cases, the hierarchy needs to be cut at some point. A number of criteria can be used to determine the cutting point:

- Cut at a pre specified level of similarity. For example, we cut the dendrogram at 0.4 if we want clusters with a minimum combination similarity of 0.4. In Figure, cutting the diagram at yields 24 clusters (grouping only documents with high similarity together) and cutting it at yields 12 clusters (one large financial news cluster and 11 smaller clusters).
- Cut the dendrogram where the gap between two successive combination similarities is largest. Such large gaps arguably indicate ``natural'' clusterings. Adding one more cluster decreases the quality of the clustering significantly, so cutting before this steep decrease occurs is desirable. This strategy is analogous to looking for the knee in the K-means graph.
- Apply Equation :

  $K\square = \arg\min_{K\square}[RSS(k\square) + \lambda k\square]$

  Where $k\square$ refers to the cut of the hierarchy those results in clusters, RSS is the residual sum of squares and λ is a penalty for each additional cluster. Instead of RSS, another measure of distortion can be used.
- As in flat clustering, we can also prespecify the number of clusters K and select the cutting point that produces K clusters.

SIMPLEHAC(d_1,…..d_N)
1. for n ← 1 to N
2. do for i ← 1 to N
3. do C[n][i] ← SIM(d_n,d_i)
4. I[n] ← 1 (keeps track of active cluster)
5. A ← [] (assembles clustering as a sequence of merges)
6. for k ← 1 to N – 1
7. do <i,m> ← arg max $_{\{<i,m> : i \neq m \square I[i] = 1 \square I[m] = 1\}}$ C[i][m]
8. A.APPEND ({<i,m> }) (storage merge)
9. for j ← 1 to N
10. do C[i][j] ← SIM (i,m,j)
11. C[j] [i] ← SIM(i,m,j)
12. I[m] ← 0 (deactivate cluster)
13. return A

The clustering process of HAC is described as follows. Let us consider a document collection consist of nine financial reports {*d*1, *d*2, …, *d*9}, where the incidence matrix is shown in Table 1. The feature vector of the financial report *d*i is illustrated in the i^{th} column. The last five values are the quantitative features. After applying HAC, the resultant dendrogram is shown in Figure 4.2, where each financial report is represented by a node, and two merged clusters is linked by an edge.

Financial report		D1	D2	D3	D4	D5	D6	D7	D8	D9
Qualitative features	Efficient	1	1	0	1	0	0	1	0	0
	Growth	1	1	0	0	0	0	0	0	0
	Advantage	1	1	1	0	0	0	0	0	0
	Improvement	1	0	0	0	0	0	0	0	0
	Deficient	0	0	0	1	1	0	0	0	0
	Reorganise	0	0	0	1	1	0	0	1	1
	Difficulty	0	0	0	1	1	1	1	0	0
	Complaint	0	0	0	1	0	0	0	0	1
Quantitative features	Operating margin	0.4	0.38	0.37	0.1	0.07	0.08	0.05	0.06	0.03
	ROE	0.3	0.28	0.27	0.01	0.04	0.04	0.07	0.02	0.05
	ROTA	0.25	0.23	0.22	0.02	0.05	0.07	0.04	0.01	0.04
	Equity to capital	0.8	0.78	0.77	0.45	0.4	0.5	0.45	0.5	0.55
	Receivables turnover	2.5	2.4	2.45	1.4	1.3	1.5	1.2	1.1	1.5
Class label		1	1	0	-1	-1	-1	0	-1	-1

Table 1.1 : An example data set

Next, we divide the dendrogram constructed in the above step into *s* groups. If we want to split it into *s* groups, we remove the *s*-1 longest links, where the *s*-1 longest links refer to the links that merge two clusters in the last *s*-1 iterations in HAC. The reason why we could remove the longest links is that the longest links must merge clusters which are most dissimilar. Each group forms a cluster, which will be input to the K-means clustering method. In

the example shown in Figure 1.4, if we want to obtain three clusters after the initial clustering, we just need to remove the two longest links. Consequently, we obtain three clusters: {*d*1, *d*2, *d*3}, {*d*4, *d*5, *d*6, *d*7}, and {*d*8, *d*9}.

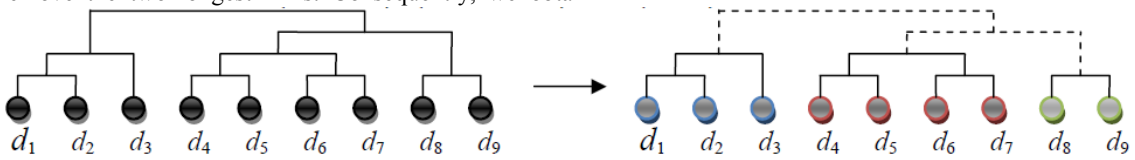


Figure 1.4 : Dendogram constructed by HAC and the clusters formed after removing the links.

1.5 K-means Clustering Method

We perform recursively the K-means clustering method to divide each cluster into sub-clusters until most feature vectors in each sub-cluster belong to the same class. However, to avoid the over-fitting problem, we use a purity threshold *p* in the recursive K-means clustering. When the purity of a cluster exceeds *p*, the recursion is finished. In addition, the class label of the resultant sub-cluster is set to the label of the majority class. In the proposed method, we modify the K-means clustering method in two aspects. First, the number of sub clusters is determined by the number of different classes within a cluster. Second, the centroid of each sub cluster is determined by averaging the features vectors belonging to the same class. We employ these two modifications to overcome the inherent weaknesses of the K-means clustering method.

For each cluster (or sub-cluster), we first examine how many different classes within the cluster (or sub-cluster), where the centroid of each class is determined by averaging the feature vectors which belong to the class. For example, there are two classes within the cluster {*d*1, *d*2, *d*3}, namely class 0 and class 1. Thus, the number of sub-clusters in the K-means clustering method is set to 2. The centroid of class 0 is (0, 0, 1, 0, 0, 0, 0, 0, 0.37, 0.27, 0.22, 0.77, 2.45), which is the average of the feature vectors of *d*1 and *d*2, and the centroid of class 1 is (1, 1, 1, 0.5, 0, 0, 0, 0, 0.39, 0.29, 0.24, 0.79, 2.45). That is, the cluster {*d*1, *d*2, *d*3} is further divided into two sub-clusters: {*d*1, *d*2}, and {*d*3}. The purity of each cluster obtained is 1.0. Thus, the recursion is finished. Next, let us consider the cluster {*d*4, *d*5, *d*6, *d*7}. After the first iteration of the K-means clustering method, the cluster is divided into two sub-clusters: {*d*4, *d*5}, and {*d*6, *d*7}. However, there are two classes within the sub-cluster {*d*6, *d*7}. Thus, the sub-cluster is further divided into two sub-clusters: {*d*6}, and {*d*7}. Since the purity of each cluster obtained is 1.0, the recursion is finished. Moreover, there is only one class in the cluster {*d*8, *d*9}, and thus we don't need to perform the K-means clustering method. Finally, we obtain six clusters: {*d*1, *d*2}, {*d*3}, {*d*4, *d*5}, {*d*6}, {*d*7}, and {*d*8, *d*9}.

For each resultant sub-cluster, its centroid is computed by averaging the feature vectors within the sub-cluster. These centroids are regarded as the representative feature vectors of the resultant subclusters, which is used to predict the stock price movements.

1.5.1 Stock Price Movements Prediction

When a financial report is released, we will transform it into a feature vector *f* according to the steps described. Next, we assign *f* to the nearest representative feature vector. Then, we predict the direction of the stock price movement according to the class label of the nearest representative feature vector. For example, if the transformed feature vector *f* is assigned to the representative feature vector of cluster {*d*1, *d*2}, we predict the direction of the stock price movement to be "rise". Hence, we make a buy stock decision based on the prediction. On the other hand, if the prediction is "drop", we make a short stock decision. We don't make any trading decision if the prediction is "no movement".

1.5.2 Classification Analysis.

Experiments are conducted to compare HRK with SVM. HRK was implemented by Java, C++ and SVM was implemented by LIBSVM (Chang & Lin 2001). We chose the polynomial kernel and set all its other parameters to their default values since the polynomial kernel outperformed the others for the dataset.

1.5.3 Dataset and Evaluation Metrics

We gathered financial reports and financial ratios from the yahoo, BSE, NSE financial databases. We focused on the companies listed in the BSE30 index as of Sep. 30, 2008, and collected all available quarterly and annual reports released from Jan. 1, 1996 to Dec. 31, 2009. Besides, the daily open and close stock quotes were gathered. We also conducted the GICS (Global Industrial Classification System) experiments to investigate the performance of company groups based on their industry sectors, where the GICS was developed by Morgan Stanley in 1999. Therefore, we classified the companies into ten industry sectors according to the definition of their principal business activity. The codes and corresponding industry sectors are described in Table 1.2. In the experiments, we used the financial reports before Jan. 1, 2010 as the training reports. The remaining financial reports were testing reports.

There are 20,884 training reports and 5,371 testing reports. In the GICS experiments, the numbers of training and testing reports are shown in Table 1.2.

Code	Industry sector	Number of training reports	Number of testing reports
10	Energy	1,710	442
15	Materials	1,226	329
20	Industrials	2,688	651
25	Consumer discretionary	3,159	887
30	Consumer staples	1,890	442
35	Health care	2,361	585
40	Financials	2,684	743
45	Information technology	3,118	831
50	Telecommunication services	355	100
55	Utilities	1,333	361

Table 1.2 : The GICS dataset.

We use two matrices to evaluate the performance in the experiments. One is the accuracy of the prediction. The other is the average profit per trade, which simulates the buy and short trading based on the predictions in the short-term stock market. If the prediction is "rise" (or "drop"), we make a buy (or short) decision at the open of the day of the financial report releases and even up at the close of the next trading day. Based on the prior research (Lavrenko et al. 2000; Schumaker & Chen 2009), we assume the transaction cost is zero since the trading costs are absorbed if the trading volume is large. The average profit per trade is calculated by averaging the profit rate of each trade.

1.6 Experimental Results

To decide the value of each parameter, we randomly sampled 10% of the data from each industry sector to conduct a series of experiments and found that HRK have the best performance when the number of qualitative features is 1,000 and the number of clusters generated by HAC is 10. Then, we used the rest data of each industry sector to evaluate the performance of HRK and SVM. Figure 1.3(a) shows the accuracy and average profit versus the combination weight, where the purity is 0.9. The experimental result shows that we have the highest average profits when the weight is set to 0.5. Moreover, Figure 1.3(b) illustrates the accuracy and average profit versus the purity, where the purity is from 0.8 to 1.0. The experimental result shows that HRK is most profitable when the purity is 0.9. Hence, we set the purity to 0.9 in the following experiments. Note that the accuracy decreases slightly and the average profit increases sharply when the purity varies from 0.8 to 0.9. When the purity threshold is low, the feature vectors of class 0 dominate some clusters. Hence, the feature vectors of class -1 and class 1 in these clusters would be merged into class 0. That makes the prediction bias toward class 0. Therefore, the average profit is low since fewer trades are executed. On the other hand, the accuracy decreases slightly and the average profit decreases sharply when the purity varies from 0.9 to 1. When the purity threshold is high, the resultant clustering becomes over-fitted. Therefore, the accuracy and average profit are lower.

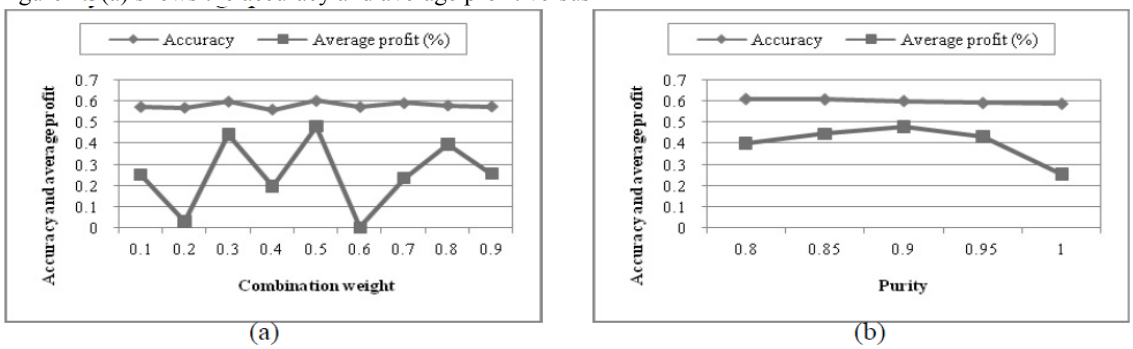


Figure 1.5. The accuracy and average profit: (a) combination weight and (b) purity.

Next, we compare HRK with SVM, HAC, and K-means methods, where the combination weight is set to 0.5 and the purity is set to 0.9 in HRK. The experimental results are shown in Figure 4.4. In this experiment, we adopt two settings of K-means clustering, namely K-means (avg_seed) and K-means (rand_seed). The difference between them is in the process of seed initialization. The seeds of Kmeans (avg_seed) are calculated as the average of the feature vectors of each class within a cluster, while the seeds of K-means (rand_seed) are randomly selected among the feature vectors within a cluster. Note that both of them are recursively performed until the purity of each cluster exceeds the purity threshold. Besides, we adopt three settings of HRK: HRK (with ratio) includes 1,000 qualitative features retrieved from financial reports and five financial ratios, HRK (w/o ratio) excludes the financial ratios, and HRK (ratio) only includes the financial ratios in the feature vectors.

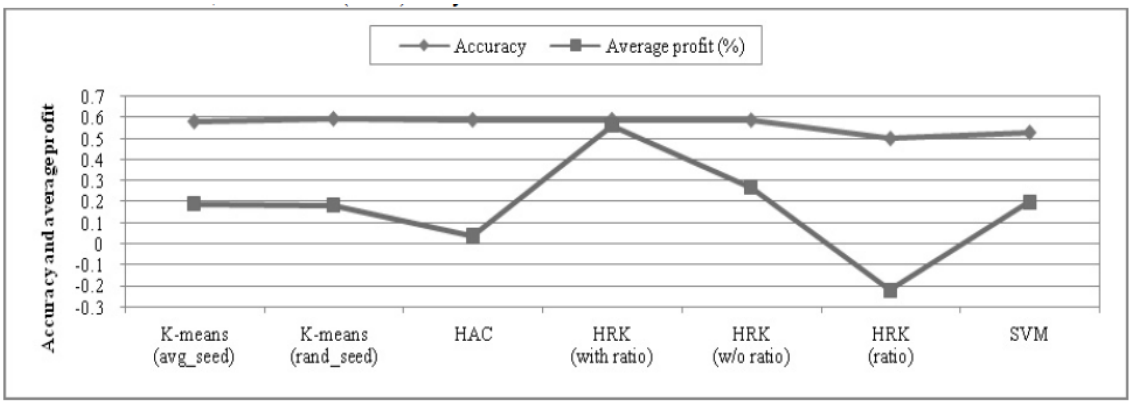
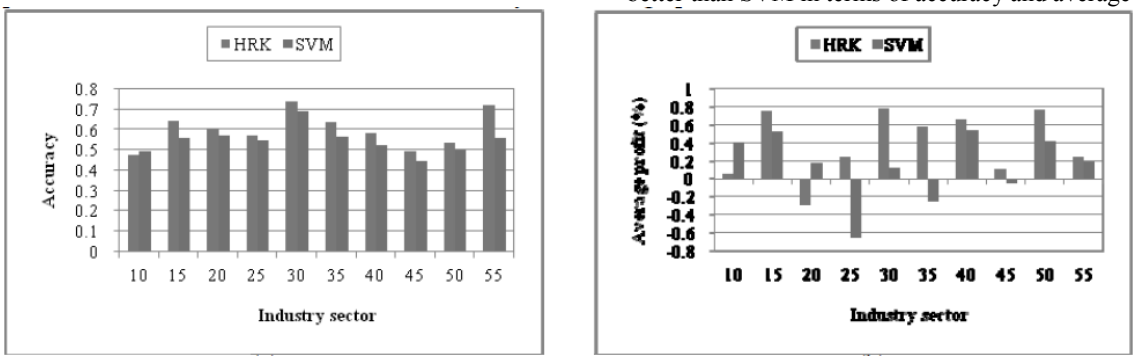


Figure 1.6. Comparing HRK with the K-means, HAC, and SVM methods.

By comparing two settings of the K-means clustering, we find that K-means (avg_seed) has better average profit. That is, initializing the seeds as the average of the feature vectors of each class within a cluster contributes to the better quality of the clustering. By comparing three settings of HRK, we could confirm that the performance of considering both qualitative and quantitative features in financial reports is better than that of only considering the qualitative or quantitative features. Moreover, HRK (with ratio) outperforms K-means (avg_seed). Since HRK uses HAC to divide the feature vectors into several clusters and HAC localizes the resultant clusters, the average profit is better than K-means (avg_seed). Besides, HRK (with ratio) outperforms HAC method as well. The results show that HRK combines the advantages of two clustering methods and the performance is better than that of using K-means clustering or HAC method only. Furthermore, HRK (with ratio) performs better than SVM in terms of accuracy and average profits.



(a) **(b)**

Figure 1.7. The (a) accuracy and (b) average profit of the GICS experiment.

Figure 1.7 shows the accuracy and average profit of the GICS experiment. For the accuracy, HRK outperforms SVM in nine industry sectors. By employing paired t-test over the results at 95% confidence level, the results show HRK performs significantly better than SVM with p-value 0.0027. For the average profit, HRK outperforms SVM in eight industry sectors. Furthermore, the total average profit of 10 industry sectors of HRK is 3.95%, while the total average profit of SVM is 1.46%. The results of the GICS experiment further validate that HRK is better than SVM.

1.6 Summary

Stock market analysis is recognized as a highly complex domain of research, Nevertheless, the potential financial awards that stand to be reaped has attracted the interest of many researchers. In this work, we have proposed a novel and effective framework for classifying stock time series based on similarity in the price trends. The proposed clustering and classification framework for stock time series, benefits from the multi-resolution capability to analyze the nonlinear similarities between stock price time-series at different time horizon. This could help investors or financial analysts in identifying inter relationship between the long term underlying trends of stock prices normally masked by the short term fluctuations. HRK outperforms SVM in terms of accuracy and average profit. HRK can attribute its better performance to three aspects. First, we consider both qualitative and quantitative features in financial reports. Second, we combine the advantages of two clustering methods to propose an effective clustering method. Third, choosing an appropriate number of splits in HAC can localize the clusters generated and thus improve the quality of the clustering generated by the K-means clustering.

References :

[1]. Han J. and Kamber M.: "Data Mining: Concepts and Techniques," *Morgan Kaufmann Publishers*, San Francisco, 2000.

[2]. Xiaozhe Wang, Kate Smith and Rob Hyndman: "Characteristic-Based Clustering for Time Series Data", *Data Mining and Knowledge Discovery, Springer Science + Business Media, LLC Manufactured in the United States,* 335–364, 2006.

[3]. Li Wei, Nitin Kumar, Venkata Lolla and Helga Van Herle:"A practical tool for visualizing and data mining medical time series", *Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05)* 106- 125, 2005.

[4]. Eamonn Keogh, Selina Chu,David Hart and Michael Pazzani:" An online algorithm for segmenting time series",*0-7695-1 119-8/01 IEEE,* 2001.

[5]. Xiao-Tao Zhang, Wei Zhang and Xiong Xiong:" A model based clustering for time-series with irregular interval", *Proceedings of the Third International Conference on Machine Learhg and Cybernetics, Shanghai,* 26-29, August 2004.

[6]. Hui Ding, Goce Trajcevski and Eamonn Keogh: "Querying and mining of time series data: Experimental comparison of representations and distance measures", *PVLDB '08,* August 23-28, 2008, Auckland, New Zealand,2008.

[7]. Ehsan Hajizadeh, Hamed Davari Ardakani and Jamal Shahrabi:"Appilication of data mining techniques in stock market", *Journal of Economics and International Finance Vol. 2(7),* pp. 109-118, July 2010.

[8]. Back, B., Toivonenb, J., Vanharanta, H., Visa, A. (2001). Comparing numerical data and text information from annual reports using self-organizing maps. International Journal of Accounting Information Systems, 2, 249-269.

[9]. Brown, D.P., Jennings, R.H. (1989). On technical analysis. The Review of Financial Studies, 2 (4),527-551.

[10]. Carnes, T.A. (2006). Unexpected changes in quarterly financial-statement line items and their relationship to stock prices. Academy of Accounting and Financial Studies Journal, 10 (3)

[11]. Chan, M.C., Wong, C.C., Tse, W.F., Cheung, B., Tang, G. (2002). Artificial intelligence in portfolio management. Intelligent Data Engineering and Automated Learning, 403-409.

[12]. Chang, C.C., Lin, C.J. (2001). LIBSVM: A library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[13]. Chen, B., Tai, P.C., Harrison, R., Pan, Y. (2005). Novel hybrid hierarchical-K-means clustering method (H-K-means) for microarray analysis. IEEE Computational Systems Bioinformatics Conference, 105-108.

[14]. Chen, K.T., Chen, T.J., Yen, J.C. (2009). Predicting future earnings change using numeric and textual information in financial reports. In Proceedings of Pacific Asia Conference on Knowledge Discovery and Data Mining, 54-63.

[15]. Chen, P., Zhang, G. (2007). How do accounting variables explain stock price movements? Theory and evidence. Journal of Accounting and Economics, 43, 219-244.

[16]. Cheu, E.Y., Kwoh, C.K., Zhou, Z. (2004). On the two-level hybrid clustering algorithm. International Conference on Artificial Intelligence in Science and Technology, 138-142.

Authors Biography

M.Suresh Babu, obtained his MCA from Osmania University and M.Phil Degree in Computer Science from Bharathiar Univeristy. At Present he is doing research in Data Mining. He has organized several workshops and training Programmes in the field of Computer Science and attended a number of Workshops and seminars. He is a life member of ISTE and Science & Society. He was elected as State Treasurer, A.P.Jana Vignana Vedika.

Dr.N.Geethanjali, has obtained PhD in 2004 from S.K.University. She is working as Head, Department of Computer Science & Technology. She has more than 20 years of teaching experience for both UG and PG Courses.Her area of interest are Data mining, Data Communications, Artificial Intelligence, Cryptography, Network Security, Programming Languages.

Prof B.Satyanarayana, has obtained PhD in 2000 from S.K.University, Anantapur. He is working as Professor / Chairman (Board of Studies), Department of Computer Science & Technology. He has more than 25 years of teaching experience. He has contributed more than 60 papers to various National and International Journals. His areas of interest are Cryptography, Design and Analysis of Algorithms, Artificial Intelligence, Software Engineering. Nearly 10 students have completed PhD and 20 students have completed M.Phil under his esteemed guidance.