Abstract

Objectives: The stock market performance has more impact on national economy. The purpose of this work is to generate a portfolio to reduce the uncertainty of stock in short term basis. Methods: Hierarchical clustering is more efficient while non-determinism is of concern when compared with flat clustering. Hierarchical agglomerative Clustering is used, which results in more informative structure than flat clustering on unstructured data. Single-link clustering is taken into account as it does not pays more attention to outliers and amalgamation criterion is local than complete-link clustering and results in intuitive cluster structure. Dendrogram is used to represent the progressive formation of clusters in HAC. Findings: Flat clustering K-means algorithm is used to combine the clusters generated by Hierarchical agglomerative clustering (HAC). As the number of samples has been reduced, iterative use of k-means will choose better centroid. Applications: The final list of the recommended stocks is then showcased to the investor on short term basis. The baseline data is downloaded from National Stock Exchange (NSE).

Keywords: Dendrogram, Hierarchical Agglomerative Clustering (HAC), Single-Link Clustering

1. Introduction

With ever evolving technologies it becomes easy to recognize the stock market prices instantly. Despite the price of the selected stocks, all of the stock related news is also obtainable at the same instant. Yet the naive problem associated with the stock market prediction of vague parameters is that influence of the market dynamics still survives. Regardless of the company’s performance there exists certain factors such as political news, economic situation; financial reports influence the value of the share. Each day there is a battle sandwiched between bulls and bears. Investor constantly monitors the movement of stocks which has influence on future investments. Different scenarios like psychology and emotions drive our decisions other than unpredictable value of the stock. Each stock pulls the market in different directions causing frustration on the other side. The nature of the stock is volatile in nature. Hence it always requires risk taking factor before investment. A stock market has different sectors like automotive, information technology, power, services, banking, health care etc. The existing systems cater the prediction of a single stock exclusively using diversified methods.

Existing systems support prediction of single stock, by association rule mining. A rebalancing formula is used to remove the negatively performing stocks. Financial report is converted to feature vectors. First HAC is applied to convert feature vectors to clusters and then k-means is applied recursively to form sub clusters. Centroid is chosen for each sub cluster as its representative feature vectors to determine the stock price. Apriori is implemented to determine stock category association and K-means is implemented to determine the stock clusters, the parameter used in both the algorithm is stock indices. Neural network models for level estimation and classification and cross-validation technique are employed on the parameters like production price index, industrial production index, consumer price index on long term basis to forecast stock returns. BSE-Sensex is

*Author for correspondence
selected as a parameter for analysis. Genetic algorithm is used to optimize the parameter and Decision tree is used to improve accuracy. The hybrid approach is implemented using Support vector machine. Survey of various techniques like decision tree, neural networks and ARM for various applications is discussed. Original data set is converted into smaller set of prototype vectors by cluster means and three methods like K-means, hierarchical clustering and self-organizing map are used. A survey is done on the data mining techniques and how it is used in stock market, clustering is used to uncover hidden patterns and predict future trends and behaviors in financial markets. As stock market is non-linear, neural networks would more suited, and in this paper various factors like hidden layers, learning rate and momentum rate is considered to predict stock movements. In this paper relationship between daily and intraday stock data is established using neural networks, in this exhaustive experiments are performed to compare ARMAX, LMS, BPN, Radial basis functional neural networks and the Recursive least square prediction, and finally ARMAX is recommended. A linear relationship of non-model based co relation and linear regression based predictability and non linear neural network is found to exist between intraday as well as AHIPMI. NASDAQ datasets like open, close, high, low prices of intraday and AHIPMI are considered as parameters. phase of stock price trends, and accuracy of results increases, so in this paper algorithms like pattern extraction algorithm, data stream correlation extraction algorithms has been employed on time series data. In foreign exchange market movement is displayed by the best offer and best asking price, which in turn can be determined by neural networks and regression trees and it is found that it is more accurate than simple averaging and waning.

2. Clustering Analysis

2.1 Hierarchical Clustering

Connectivity based clustering results in clusters, with high within-cluster similarity and low inter-cluster similarity. There are many variants to define closest pair clusters. Single Link aims at grouping the instances based on most cosine-similar. Complete-link aims at grouping the instances based on least cosine similar. Centroid based aims at clustering data based on most cosine similar centroid. Average-link based on average cosine between pairs of elements. The main target is to single out distinguished features.

2.1.1 Single-Link Agglomerative Clustering

In this work single-link agglomerative clustering is used. The clusters are formed based on maximum similarity and progressive of cluster formation is visualized using dendrogram.

\[
sim(c_i, c_j) = \max_{x \in c_i, y \in c_i} \sim(x, y)
\]

After merging \( c_i \) and \( c_j \), the similarity of the resulting cluster to another cluster, \( c_k \), is

\[
sim(c_i \cup c_j, c_k) = \max(\sim(c_i, c_k), \sim(c_j, c_k))
\]

**Procedure**

- Initially consider each instance as disjoint clusters at level \( K \) (m), \( m=0 \).
- Calculate the dissimilarity matrix \( (N \times N) \) to hold all distances \( d(i, j) \).
- Locate similar clusters based on minimum distance, \( D(pi, pj) = \min_{i \in p, j \in q} d(i, j) \), \( D \) is the Euclidean distance between clusters \( p \) and \( q \).
- Move to next sequence \( m=m+1 \). Merge clusters \( p \) and \( q \) into a single cluster to form the next level cluster \( k \) (m) = \( d([p] \cup [q]) \).
- Modify the similarity matrix \( D \), by removing the row and column values corresponding to clusters \( r \) and \( s \) and adding a row and column corresponding to the newly formed cluster. The proximity between the new cluster \( p \) and old cluster in the previous level \( k \) is denoted as \( D([k] \cup [p]) = \min d([k], [p]), d([k], [q]) \).
- Repeat step 2 until single cluster formation.

2.2 K-Means

K-Means algorithm is a heuristic algorithm that converges to local optimum. K-means is more sensitive to outliers. Flat clustering used to create partitions independent of each other. For the given data set \( D \) the K-Means algorithm partitions the data set into K clusters. Each cluster has a cluster center called centroid. The outliers can affect the mean centroid a lot. K-median is a better alternate for data with outliers. K-means will result in better output for equal or density clusters and poor results for non-convex shape clusters.
**Procedure**

Objective of K-Means algorithm is to minimize the total distortion. K-means is more sensitive to cluster center initialization.

Consider the data set \( D = \{ s_1, s_2, s_3, s_4, \ldots, s_n \} \)

Step 1: Let \( \{ k_1, k_2, k_3, \ldots, k_n \} \) be chosen as ‘n’ initial centroid.

Step 2: Calculate the distance for each instance \( \{ s_1, s_2, s_3, s_4, \ldots, s_n \} \) in data set \( D \) to each centroid.

Step 3: Assign each \( s_i \{ s_1, s_2, s_3, s_4, \ldots, s_n \} \) in \( D \) to the closest centroid cluster.

\[
C_n = \{ i: n = \arg \min_{n} \| s_i - k_n \|^2 \}
\]

Where: \( C_n \) are the data sets closest to cluster \( k_n \)

Step 4: Update the centroid of each cluster based on the new instance in the cluster.

\[
K_n = \frac{1}{|C_n|} \sum_{i \in C_n} S_i
\]

where:

\( |C_n| = \) size of cluster \( C_i \)

\( K_n = \) is the centroid of the cluster ‘n’ after adding the new instance.

Step 5: Continue step 2 until the convergence criteria.

**Convergence Criteria**

- Insignificant re-assignment of instances to different clusters.
- No or insignificant change in centroid value of the clusters.
- Insignificant decrease in the sum of squared error

\[
E = \sum_{i=1}^{N} \sum_{s \in C_i} d(s, m_i)^2
\]

\( N = \) total number of clusters. \( C_i = i^{th} \) cluster

\( M_i = \) Centroid of \( i^{th} \) cluster

\( d(s, m) = \) distance between instance \( s \) and centroid of cluster \( i \)

3. Implementation and Results

In this work stock prediction using cluster analysis precisely employs two methods. Hierarchical agglomerative clustering is used as the hierarchical methods and the K-means algorithm as the partitioning method.

**Hierarchical agglomerative clustering**

There are two approaches in the hierarchical methods, agglomerative approach and divisive approach. In this work the former approach is used. As each share is considered as single object, objects are clustered bottom up. This means it starts with the single object and forms separate groups to form clusters. Dendrogram is represented in Figure 1.

From Table 1, the percentage of increase of different stocks for different days is calculated using Eq(1).

\[
Input_{i,j} = \frac{[\text{closeprice}_d - \text{Openprice}_e]}{\text{Openprice}_d} * 100/n
\]

Where, \( s_i = \) stock, \( d - \) day (n business days in 2 weeks)

Dissimilarity matrix is calculated as shown in Table 2 to identify the similarity between two stocks which controls the formation of clusters.

\[
\text{Dis-sim}(i,j) = d((\text{input} s_i), (\text{input} s_j))
\]

Where, Dis-sim is dissimilarity matrix

- Group the stocks with minimum similarity index to form new clusters.

\[
S_i, S_j = \min (\text{Dis-sim}(S_i), (S_j))
\]

- Update dendrogram with \( S_i, S_j \) as follows.

\[
\text{Replace zero to the values in the } i^{th} \text{ and } j^{th} \text{ column}
\]

\[
\text{Repeat until dissimilarity value is very large}
\]

In the above diagram Figure 3 – x axis depicts the sample stocks that are considered for analysis, y axis depicts the dissimilarity level at which stocks are clustered simultaneously a weightage factor for each stock is calculated which is used further rank the stocks as shown in Table 3.

\[
\text{Weight}_s = \sum_{d=0}^{9} \frac{[\text{closeprice}_d - \text{openprice}_d]}{\text{openprice}_d} * w
\]

**K-Means**

The input to the k-means algorithm would be the resultant stocks from the hierarchical agglomerative clustering and percentage of raise is calculated using Eq(5).
Table 1. Sample Actual Data Sets downloaded from NS

<table>
<thead>
<tr>
<th>YESBANK</th>
<th>ZEEL</th>
<th>ZENSARTECH</th>
<th>ZODIACLOTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Price</td>
<td>Close Price</td>
<td>Open Price</td>
<td>Close Price</td>
</tr>
<tr>
<td>380.00</td>
<td>379.55</td>
<td>290.10</td>
<td>287.80</td>
</tr>
<tr>
<td>365.00</td>
<td>379.20</td>
<td>287.00</td>
<td>290.25</td>
</tr>
<tr>
<td>362.45</td>
<td>362.00</td>
<td>285.50</td>
<td>289.30</td>
</tr>
<tr>
<td>360.60</td>
<td>360.35</td>
<td>280.90</td>
<td>287.95</td>
</tr>
<tr>
<td>365.90</td>
<td>362.05</td>
<td>280.55</td>
<td>278.35</td>
</tr>
<tr>
<td>367.25</td>
<td>365.40</td>
<td>283.50</td>
<td>280.15</td>
</tr>
<tr>
<td>359.50</td>
<td>363.70</td>
<td>274.30</td>
<td>282.35</td>
</tr>
<tr>
<td>355.80</td>
<td>355.85</td>
<td>288.50</td>
<td>285.90</td>
</tr>
<tr>
<td>368.10</td>
<td>361.35</td>
<td>279.20</td>
<td>280.55</td>
</tr>
</tbody>
</table>

Table 2. Sample dissimilarity matrix of sample 9 stocks

<table>
<thead>
<tr>
<th>3MINDIA</th>
<th>ABAN</th>
<th>ABB</th>
<th>ABGSHIP</th>
<th>ABIRLANUVO</th>
<th>ACC</th>
<th>ADANIENT</th>
<th>ADANIPORTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Price</td>
<td>Close Price</td>
<td>Open Price</td>
<td>Close Price</td>
<td>Open Price</td>
<td>Close Price</td>
<td>Open Price</td>
<td>Close Price</td>
</tr>
<tr>
<td>3MINDIA</td>
<td>0</td>
<td>0.6114</td>
<td>1.187887</td>
<td>0.972478</td>
<td>0.384962</td>
<td>0.798992</td>
<td>0.89337</td>
</tr>
<tr>
<td>AARTIIND</td>
<td>0.20381311</td>
<td>0.815214</td>
<td>1.3917</td>
<td>1.176291</td>
<td>0.58775</td>
<td>1.002805</td>
<td>1.097183</td>
</tr>
<tr>
<td>ABAN</td>
<td>0.61140045</td>
<td>0</td>
<td>0.576486</td>
<td>0.361077</td>
<td>0.226439</td>
<td>0.187592</td>
<td>0.094378</td>
</tr>
<tr>
<td>ABB</td>
<td>1.18788694</td>
<td>0.576486</td>
<td>0</td>
<td>0.215409</td>
<td>0.802925</td>
<td>0.388985</td>
<td>0.294517</td>
</tr>
<tr>
<td>ABGSHIP</td>
<td>0.97247769</td>
<td>0.361077</td>
<td>0.215409</td>
<td>0</td>
<td>0.587516</td>
<td>0.173486</td>
<td>0.079108</td>
</tr>
<tr>
<td>ABIRLANUVO</td>
<td>0.38496165</td>
<td>0.226439</td>
<td>0.802925</td>
<td>0.587516</td>
<td>0</td>
<td>0.41403</td>
<td>0.508408</td>
</tr>
<tr>
<td>ACC</td>
<td>0.79899212</td>
<td>0.187592</td>
<td>0.388985</td>
<td>0.173486</td>
<td>0.41403</td>
<td>0</td>
<td>0.094378</td>
</tr>
<tr>
<td>ADANIENT</td>
<td>0.89336976</td>
<td>0.281969</td>
<td>0.294517</td>
<td>0.079108</td>
<td>0.508408</td>
<td>0.094378</td>
<td>0</td>
</tr>
<tr>
<td>ADANIPORTS</td>
<td>0.40510012</td>
<td>0.2063</td>
<td>0.782787</td>
<td>0.567378</td>
<td>0.020138</td>
<td>0.393892</td>
<td>0.48827</td>
</tr>
</tbody>
</table>

\[ input_s = \left( \frac{\text{closeprice}_m - \text{openprice}_m}{\text{openprice}_m} \right) \times 100 \]  

\[ m_{i=0}^k = \text{rand(Input}_s) \]  

Assign each stock to the cluster to which the stock is most similar based on the mean value of the stocks in the cluster.

\[ \text{cluster}(m, y) = \min (\text{diff}(m, \text{input}_s), \text{diff}(m, \text{input}_s)) \]  

The Figure 2 shows the formation of clusters representation using the index representation of each stock. The
The Figure 3 depicts the results obtained by this work with the actual scenario at the end of the day. The advantage of k-means algorithm is it is more scalable and efficient. The algorithm is also sensitive to noise and outlier data. The final result obtained by portfolio generator is compared with the day's actual result for accuracy as shown in Figure 3. Comparison between results of HAC and k-means is shown in Figure 4. Figure 5 represents the movement of stock of AMTEKAUTO Company in the course of a day.

### Table 3. Sample intermediate results of the Hierarchical Agglomerative Clustering (HAC)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Weightage</th>
<th>Stocks</th>
<th>Stocks no.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16</td>
<td>AMTEKAUTO</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>POLARIS</td>
<td>363</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>MERCATOR</td>
<td>308</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>CAPF</td>
<td>77</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>VIPIND</td>
<td>484</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>DEN</td>
<td>109</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>BEMl</td>
<td>54</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>PRISMCEM</td>
<td>368</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>AMTEKAUTO</td>
<td>21</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>PRAKASH</td>
<td>366</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>MONNETISPA</td>
<td>314</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>AUTOAXLES</td>
<td>36</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>BRFL</td>
<td>70</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>DYNAMATECH</td>
<td>119</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>JKTYRE</td>
<td>256</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>KSBPUMPS</td>
<td>280</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>APOLLOTYRE</td>
<td>26</td>
</tr>
</tbody>
</table>

The sample stock marked in Figure 2 is represented with the names of the stocks grouped together.

- Now calculate the new centroid value for the newly formed cluster using Eq(8).

$$m_j = \frac{1}{n} \sum_{i=0}^{n} [\text{cluster}(i,f)]$$ (8)

- Repeat from step 2 until no significant change in centroids for repeated iterations.

---

**Figure 2.** Sample clusters formed in subsequent iterations.

**Figure 3.** Comparison of the portfolio generator with actual result.

**Figure 4.** Comparison results of HAC and K-means.

**Figure 5.** Final result with sample graph of AMTEKAUTO Company.
4. Conclusion

Since the stock market has huge amount of data sets, it is viewed as a complex domain of research analysis. This work depicts how clustering approaches in data mining such as hierarchical agglomerative clustering and the K-means algorithm is employed to predict the value of stocks to decide over future investments. The application is tested with the sample stocks collected from national stock exchange it could be extended to other stock exchanges also. The graph provided at the end proves us that the portfolio recommends the consistently performing stock through the specified period. Though we have restricted the data sets for a period of 22 business days, it could be extended in future to increase the reliability. As well extend the application for intraday trading exclusively.

5. References