Big Data Analysis Using Fuzzy Logic and K-mean Clustering Algorithm

Aparna Sharma
Computer Science and Engineering
SRM Institute of Science and Technology
Kattankulathur, Chennai-603203
aparna9517@gmail.com

Shubham Chatter
Computer Science and Engineering
SRM Institute of Science and Technology
Kattankulathur, Chennai-603203
Shubhamjain15@gmail.com

ABSTRACT:

Stock market prediction is one of the most important exertions in the finance world. It is a process which is full of uncertainty and affected by many factors. Both technical and fundamental analysis is applied to overcome this problem. In this paper we focus on finding a better technical analysis method which would give quantitative results instead of the binary one. The technical analysis is done by applying fuzzy time series prediction algorithm on historical financial data. We compared the accuracy of the predictions made using different clustering algorithms. Finally, we explore additional techniques like SCM to further increase the accuracy of the results.

Keywords— Fuzzy logic, K- MEAN, SCM, MapReduce, Big Data

Introduction (**Heading 1***)*

Predicting the stock market is one of the most important area for research today. With advancement in big data and aide and advent of social media has created the waves among researchers.

Two traditional principles of any financial derivatives are[1][2]:

- Some investment is necessary for any profit
- No opportunity to generate profit is free of risk

The concepts to be considered the stock prediction model are:

- Random walk theory: the financial data are independent of each other as they have the same distribution, so the past movement of financial data can't be used to predict it's future movement[1].

It's is given by the formula:

$$v(t) = v(t-1) + c(t)$$

$$\Delta v(t) = \frac{v(t) + d(t) - v(t-1)}{v(t)}$$

$v(t)$: *price of stock at time t*

$v(t-1)$: *price of stock at time t − 1*

$\Delta v(t)$: *change in price of stock at time t*

$d(t)$: *dividend at time t*

$c(t)$: *adjustment term at time t*

The prediction of $\Delta v(t)$ is difficult since $c(t)$ is the impact of all available information on the stock

- Efficient market hypothesis it's states that the assimilation of all information available mirror the state of the market. The system enters an unbalanced state when new information enters the market and the predicted correct change is eliminate by the new price. Hence the given information is not possible to predict the future price of stock[3]. EMH has three possible forms:
  - Weak form: only passed information is considered
  - Semi strong form: all publicly available information is used
  - Strong form: all information publicly and privately available are used

There are two approaches that could be followed to models stock market prices[4].

Technical: statistical analysis of stock prices. **Fundamental**: semantic analysis of all available data.

In this paper we focused on fundamental analysis using fuzzy time series forecasting algorithm on financial data. Form of EMH for prediction is semi strong since only publicly available information is used for prediction.

The forecasting method used is the one proposed by [5]. The accuracy of the prediction mainly depends on the clustering algorithm used to create the intervals in the universe of discourse.

I. LITERATURE REVIEW

Many research groups are exploring ways for stock market predictions.
Using logistic regression to get binary result for market trends[6].

Mapreduce: Simplified data processing on large clusters
MapReduce is a programming model and an associated implementation for processing and generating large datasets that is amenable to a broad variety of real-world tasks. Users specify the computation in terms of a map and a reduce function, and the underlying runtime system automatically parallelizes the computation across large-scale clusters of machines, handles machine failures, and schedules inter-machine communication to make efficient use of the network and disks. Programmers find the system easy to use: more than ten thousand distinct MapReduce programs have been implemented internally at Google over the past four years, and an average of one hundred thousand MapReduce jobs are executed on Google's clusters every day, processing a total of more than twenty petabytes of data per day.
Creating model based on analytical and fundamental analysis to predict stock market[7].
The new approach based on logistic regression model predicted the stock price of next month by using the stock prices of the current month[6].
The semantic analysis and logistic regression are used to create the model to predict the stock market[7].
The multivariate fuzzy time series forecasting algorithm is used to perform predictions using historical data for different discipline[5].
Thea method is proposed to improve the quality of clusters formed by the K -means clustering algorithm using axiomatic fuzzy set and subtractive clustering method.

II. Background

In this paper we propose to improve upon the methodologies for technical analysis suggested in [7] to be able to better predict the stock market trend. [7] uses logistic regression to get a binary answer of whether stock would go up or down. But this approach has a few problems

- The output is binary: a direct yes or no answer
- It does not reflect the magnitude of the change
- How to find the accurate values based on the raw data because many parameters affect the accuracy.
- How to evaluate the semantic strength (SS) of algorithms?
- How to design a new clustering algorithm by integrating
- semantic fuzzy concept with SCM, which performs better than FCM?
- How can we modify new algorithm to deal with the big data sets efficiently?

To deal with this problem we need to we first need to know about some theories

A. Fuzzy K-Mean Clustering

In fuzzy clustering, each point had a probability of belonging to each cluster, rather than completely belonging to just one cluster as it is the case in the traditional k-means. Fuzzy k-means specifically tries to deal with the problem where points were somewhat in between centers or otherwise ambiguous by replacing distance with probability, which of course would be some function of distance, such as having probability relative to the inverse of the distance. Fuzzy k-means used a weighted centroid and based on those probabilitie. Processes of initialization, termination and iteration were the same as the ones used in k-mean. The resulting clusters were best analyzed as probabilistic distributions rather than a hard assignment of labels. One should know that k-means was a special case of fuzzy k-means when the probability function use was simply one if the data points is closest to a centroid and zero otherwise.

The fuzzy k-means algorithm steps are following:

1. **Assume** the clusters with a fixed number
2. **Initialization**: Initialize the k-mean randomly associated with the cluster and the probability should be computed that each data point was a member of a given cluster *k*, P(point x_i label $k|x_i, k$), P(point x_i has label $k|x_i, k$).
3. **Iteration**: the centroid was recalculated of the cluster. The weighted centroid gives the probabilities of membership of all data points

$$\mu_k(n+1) = \frac{\sum_{x_i \in k} x_i \times P(\mu_k|x_i)^b}{\sum_{x_i \subset k} P(\mu_k|x_i)^b}$$

4. **Termination**: Iteration should be done until convergence or until number of iterations of user-specified number had been reached.

B. Subtractive Clustering Method (SCM)

SCM was used for computing the clusters centroid. It identify the numbers of cluster formed and based on the raw data it found the cluster centroid. Each data point was used to find out the potential cluster. Formula used to calculate the potential is:

$$M_1(x_i) = \sum_{j=1}^{n} \exp\left(-\frac{\| x_i - x_j \|^2}{({}^{\tau_1}/_2)^2}\right)$$

τ_1: Neighbour radius

Neighbor radius influences the scope of a cluster centroid. Larger the value of is, the greater its impact will be. The first centroid is the, data point with maximum mountain function. Then update the mountain function of each data according to the following equation:

$$M_l(x_i) = M_l(x_i) - M^*_{l-1} \ exp\left(-\frac{\| x_i - x_i^* \|^2}{(^{\tau_2}/_2)^2}\right)$$

τ_2 : influencing weight of the last cluster centroid

We need to avoid getting cluster centroids closer to each other because the data points near the first cluster centroid will have greatly reduced potential and thus unlikely to be the next cluster centroid. In general, $1.5\tau_1 = \tau_2$ s.

III. Proposed Work

The steps to forecast the data are as follows

- Apply subtractive clustering method to get the centroid of the data sets
- We get the cluster centroid which is the input of fuzzy K-mean
- Apply Fuzzy K-Mean
- Construct the Fuzzy logical relationship and groups
- calculate the forecasted output

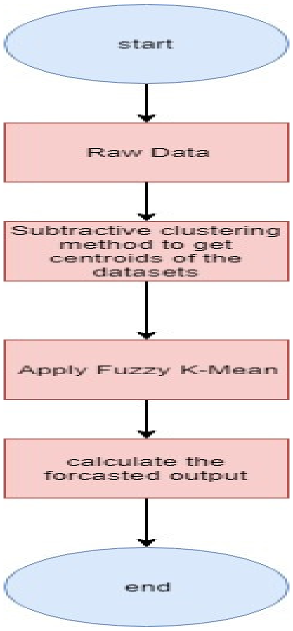
start
Raw Data
Subtractive clustering method to get centroids of the datasets
Apply Fuzzy K-Mean
calculate the forcasted output
end

Figure 1: Proposed Algorithm

In [9] the subtractive clustering algorithm is described as follows:

Considered the group of n data points $\{x_1,x_2,...,x_n\}$, where x_i was a vector in the feature space. There is no loss of generality, we assumed that the feature space was normalized so that all data were bounded by the unit hypercube. We considered each data point as the potential cluster centre and defined a measure of the point to serve as the cluster centre. The potential of x_i, denoted as P_i, was computed by Eq. 1.

$$P_i = \sum_{j=1}^{n} \exp\left(-\frac{\|x_i - x_j\|^2}{(r_a/2)^2}\right) \qquad 1)$$

where, r_a is positive constant defining a neighbourhood radius, denoted Euclidean distance. A data point with many neighbouring data points would have a high potential value and the point outside r_a had little influence on its potential.

First cluster centre c_1 was chosen as the point having the highest potential. The potential of c_1 was denoted as PotVal(c_1). Next, the potential of each data point x_i was revised as follows:

$$P_i = P_i - \mathrm{PotVal}(c_1)\exp\left(-\frac{\|x_i - c_1\|^2}{(r_b/2)^2}\right) \qquad 2)$$

where, $r_b = 1.5r_a$ is usually set to avoid obtaining closely centres of spaced cluster. The data points near the first cluster centre would have greatly reduced their potential and would unlikely be selected as the next cluster centre. After the potential of all data points had been reduced according to Eq. 2, the one with the highest potential was selected as the second cluster centre. Then, potential of the remaining points was again reduced. [9]Generically, cluster centre c_k is determined after the k^{th} cluster centre c_k . The potential is revised as follows:

$$P_i = P_i - \mathrm{PotVal}(c_k)\exp\left(-\frac{\|x_i - c_k\|^2}{(r_b/2)^2}\right) \qquad 3)$$

where, c_k is the location of the kth cluster centre and PotVal(c_k) was its potential value.[9] The process continues until the stopping criterion defined in (Li *et al.*, 1999) was reached.

From the clustering process, two conclusions are:

The point with high potential had more chance to be selected as cluster centre than the point with less potential. Each cluster centre was a point with relatively very high potential.

Cluster centres were selected only from data points whether not the actual cluster centres were in the dataset. However, the actual cluster centres were not necessarily located at one of the data points[9].

Step 1: The potential of each data point was computed using Eq. 1; set the number of cluster centres as k = 1[9].

Step 2: The point was selected with the highest potential denoted as c_k, the data point surrounding c_k with radius smaller than r_a were denoted as $(x_1^{(k)},x_2^{(k)},...x_{m(k)}^{(k)})$[9]. Then, the weighted mean cluster centre $\overline{c_k}$ is computed using Eq. 4:

$$\overline{c_k} = \frac{\sum_{j=1}^{m(k)} \mathrm{PotVal}(x_j^{(k)}) * x_j^{(k)}}{\sum_{j=1}^{m(k)} \mathrm{PotVal}(x_j^{(k)})} \qquad 4)$$

where, m(k) is the number of data points surrounding c_k with radius smaller than r_a.

Step 3: The potential of each data point was revised as follows:

$$P_i^{(k+1)} = P_i^{(k)} - \mathrm{PotVal}(\overline{c_k})\exp(\frac{\left\|x_i - \overline{c_k}\right\|^2}{(r_b/2)^2}) \qquad 5)$$

Where:

$$\mathrm{PotVal}(\overline{c_k}) = \sum_{i=1}^{n}\exp(-\frac{\left\|\overline{c_k} - x_i\right\|^2}{(r_a/2)^2}) - \sum_{j=1}^{k-1}\exp(-\frac{\left\|\overline{c_k} - \overline{c_j}\right\|^2}{(r_b/2)^2})$$

Step 4: If the stop criterion was met, then stop the process; otherwise, set k = k + 1, return to Step 2[9].

The point with high potential had a comparatively big impact on the cluster centre. In weighted mean subtractive clustering, the location of the cluster centre is decided by not only one data point but all data points in a neighbouring area. A measure of the influence is based on its potential, the more potential, the more influence.

A. Implementation of Fuzzy K-Mean using Map Reduce:

We needed some vectors. These vectors represent our data, and then k-centre's needed. These were vectors too, sometimes they were just a subset of the input vectors, but sometimes they were random points or points-of-interest to which we were going to cluster them.

Since this was a MapReduce version the keys and values were used. This was really simple, because it was just using a vector, a vector can be a cluster centre as well. So treat the cluster centre-vectors always like keys, and input values.

In the map step

- Read cluster centers into the memory from sequence file
- Cluster centre should be iterated for each input key/value pair.
- The distances were measured and saved the nearest centre which had the lowest distance to the vector
- Write the cluster centre with its vector to the filesystem.

In the reduce step (we get associated vectors for each centre)

- Value of each vector was iterated and calculates the average vector.
- This was the new centre, saved it into the SequenceFile.
- The convergence between the cluster centres should be checked that was stored in a key object.
- If they were not equal, increment the update counter Run this whole thing until nothing was updated.

IV. Comparison between K-Means and C-Means

K-mean clustering:

Hard C-Means clustering or K-means clustering is used as a portioning method applicable to analyze data and use observation according to the objects of data based on the distance and locations between different data points of input. Centroid is the centre point of each cluster. The distance use in clustering is most of the times do not actually represents the spatial distance0. The only solution to the problem of finding global minimum was exhaustive choice of starting points but use of different replicates with random starting point's leads to the solution that is a global solution [2, 6, 14]. In datasets, a desired number of clusters K and a set of k initial starting points, the K-Means clustering algorithm found the desire number of distinct clusters and their centroids. Centroids are the points whose coordinates are obtained by means of computing the average of each of the co-ordinates of points of samples assigned to the clusters.

Fuzzy C-mean clustering:

Fuzzy C- mean clustering method comes from the Hard C-mean clustering algorithm. It is an unsupervised clustering method that is applied to various problems like feature analysis, classifier design, astronomy, geology, chemistry and medical diagnosis. T he algorithm was used for analysis based on the distance between different input data. The cluster was formed depending on the distance between data points and cluster centers are formed for each cluster.

FCM is the data clustering algorithm in which a data set was grouped into n number of clusters with every data point in the dataset related to every cluster and it would have the highest degree of connection to that cluster and another data point that lies far away from the centre of the cluster, where centre of the cluster will have a low degree of connection to that cluster.

Comparison of Time Complexity of FCM and K-Means:

Time complexity of FCM [4] is O(ndc^2 i) and The time complexity of K-means is O(ncdi). If we keep the number of data points constant we assume that
n = 100
d = 3
i = 20 and varying number of clusters
n = number of data points
c = number of cluster
d = number of dimension
i = number of iterations
The following tables represent the comparison in details

TABLE I. COMPARATIVE ANALYSIS OF K-MEANS AND FCM

Algorithm	**Time complexity**	**Elapsed time (seconds)**
K-Means	O(ncdi)	0.443755
FCM	O(ndc^{2}i)	0.781679

TABLE II. TIME COMPLEXITY OF K-MEANS AND FCM WHEN NUMBER OF CLUSTERS VARYING

S.No.	**Number of clusters**	**K-Means Time Complexity**	**FCM Time Complexity**
1	1	6000	6000
2	2	12000	24000
3	3	18000	54000
4	4	24000	96000

TABLE III. TIME COMPLEXITY OF K-MEANS AND FCM WHEN NUMBER OF ITERATIONS VARYING

S.No.	**Number of iterations**	**K-Means time complexity**	**FCM Time complexity**
1	5	3000	6000
2	10	6000	12000
3	15	9000	18000
4	20	12000	24000

The time complexity of the K-Means algorithm is O(ncdi) and the time complexity of FCM algorithm is O(nc^{2}di). From the obtained results we may conclude that K-Means algorithm is better than FCM algorithm. FCM produces close results to K-Means clustering but it still requires more computation time than K-Means because of the fuzzy measures calculations involvement in the algorithm.

Acknowledgment

In this paper, aiming to provide with effective methods to improve the accuracy of the predicted results. First propose a clustering methods SCM and fuzzy K-Means. s. Moreover, it decreases the risk of imprecise operations management using FCM. Experiment results indicate that SCM has stronger clustering SS than FCM. Then we deploy the fuzzy K-Mean using the Map Reduce. After that we compare between different clustering algorithms which help create the intervals and find that the c-mean clustering algorithm doesn't gives better results compared to other clustering methods.

In the future, we will improve the algorithm to validate the effectiveness in terms of other risk analysis and using more abundant implementation platform.

References

[1] Hellstrom T. A Random Walk through the Stock Market, Licentiate Thesis,Department of Computing Science, Umea University, Sweden,1998

[2] Lawrence Shepp, "A Model for Stock Price Fluctuations Based on Information", IEEE Transactions On Information Theory, Vol. 48, No.6, June 2002

[3] Burton G. Malkiel, 'The Efficient Market Hypothesis and Its Critics", The Journal of Economic Perspectives,Vol. 17, No. I (Winter, 2003), pp. 59-82

[4] Laszlo Gerencser, Balazs Torma and Zsanett Orlovits, "Fundamental Modelling of Financial Markets," ERCIM News ,vol 78, pp 52,Jul. 2009

[5] Chen SM, Tanuwijaya K (2011) Multivariate fuzzy forecasting based on fuzzy time series and automatic clustering techniques. Expert Syst Appl 38:10594–10605

[6] R. Chodhury, K. Garg, "A hybrid machine learning system for stockmarket forecasting", Proceeding of World Academy of Science, Engineering and Technology, vol. 29 (2008) ISSN 1307-6884

[7] Girija V Attigeri ,Manohara Pai M M ,Radhika M Pai and Aparna Nayak," Stock market prediction: A big data approach", Manipal Institute of Technology

[8] Wenjie Bi, Meili Cai, Mengqi Liu and Guo Li,"A big data clustering algorithm for mitigating the risk of customer churn", IEEE TRANSACTION ON INDUSTRIAL INFORMATICS, VOL. 12, NO. 3, JUNE 2016

[9] JunYing Chen, Zheng Qin and Ji Jia, 2008. A Weighted Mean Subtractive Clustering Algorithm. Information Technology Journal, 7: 356-360