Stock Trends Prediction by Hypergraph Modeling

Yang Shen
State key lab of software engineering
School of Computer Science, Wuhan University
Wuhan, China

Jicheng Hu, Yanan Lu, Xiaofeng Wang
State key lab of software engineering
School of Computer Science, Wuhan University
Wuhan, China

***Abstract*—This paper presents a new stock price trends prediction algorithm using hypergraph model. Hypergraph modeling offers a significant advantage over traditional graph modeling in terms of triadic or higher relationship description within different stock portfolios over a certain period of time. Under the hypergraph model, each stock will be abstracted as a vertex of hypergraph; the hyperedges can be built by seeking the synchronous relationship of the stocks trends. In order to acquire more refined hyperedges and to avoid the tremendous growing quantity of hyperedges, we employ the frequent item sets to construct hyperedges. Therefore the prediction problem for stock trends is converted to hypergraph partitioning problem. Multilevel paradigm is then applied to do hypergraph partitioning instead of the traditional recursive bisection paradigm. Thus we get a series of stocks section, and the stock price trends can be concluded by analysis the whole section. Experiment result shows that our proposed scheme achieves fine stock trend prediction and the computation is significantly fast as well.**

Keywords–stock trends prediction, frequent item sets, multilevel, hypergraph partitioning

I. Introduction

Professional stock brokers are always trying to determine and compare the future value of different company stocks by carefully analyzing the history trading charts to dig out the relationship from different stock portfolios. A lot of stock prediction software systems are brought to light to help them to generate accurate stock marketing forecasting.

Supply and demand of stock exchange are driven by economic factors, political factors, social psychology of investing, etc., none of which can be easily well simulated by certain efficient modeling method. Since so many complicated factors may affect the price of stocks, we can hardly find out accurate ways to predict stock price trends in an instantly varying free market.

Traditional approaches to predict have always seen huge failure or produce a lot of mistakes. While numerous attempts have been made over the recent years, the difficulty has always centered. But we can still see many valuable trials using a variety of methods include decision trees [1], financial news [2], rule induction [3], Bayesian belief networks [4], neural networks [5], evolutionary algorithms [6], classifier systems [7], fuzzy sets [8] [9], association rules [10], and graph-based evolutionary algorithm [11].

However, many of the former trials is analyzing on the basis of a single stock trend, it is inaccurate and susceptible to many unexpected factors. Sometimes the problem of discrete stock price prediction can be analyzed using a synthesis of linguistic, financial and statistical techniques [2]. It gives us a wonderful idea. [2] tried to approach the prediction using textual representation and statistical machine learning methods on financial news articles, which is partitioned with industry and sector. But we find the complexity of stock price trend is not just inflected by the financial news, sometimes it is not very rational from that single view.

In recent years the concept of neural networks has been used in the trading community for prediction tasks [5]. Prediction is not simple and not just close to the random-walk stock time series behavior. In [5] the neural networks show good effect. But the method is still mainly based on the past trend of a single stock, as the stock price may abruptchange due to a special event.

The graph-based method is used in [11] [12] to analyze the stock market. [11] proposes a graph-based extended evolutionary method GNPRL, which combined with evolution and reinforcement learning in order to create effective graph structures and acquire better results in dynamic environments. [12] use a genetic programming in the stock trading markets to producing trading rules. [13] puts forward the genetic algorithms (GAs) method to determine the characteristics of discrete and the connection weights for artificial neural networks (ANNs) to predict the stock price index. It more directly points out a kind of stock forecast method. But the graph-based method is not sufficient.

Clustering in data mining is a process of discovery. In the course of clustering, the data sets that the intracluster similarity is maximized and the intercluster similarity is minimized can be grouped [14]. Clustering of data in a large dimension space is of a great interest in many data mining applications. [15] propose a method for clustering of data in a high dimensional space based on a hypergraph model. The method is amazing, and it let us partition stocks into different sections. Stocks in the same section usually have some correlations and will bring a strong synchronism in price trend. Received the inspiration in [15], we utilize a hypergraph model to describe the relations between the stocks. Hypergraph is similar to graph, but the former can build edges contact three or more vertices (hyperedges), which let it have more advantages in the dealing the high dimensional data. The method of frequent item sets in [15] also contributes a lot in the building of the more accurate hypergraph model.

In this paper, we present a three-step algorithm for analyzing and predicting the stock trend. In the first step we use a novel scheme to portray a relationschema of different

978-1-4673-2008-5/12/$31.00 ©2012 IEEE

Authorized licensed use limited to: New York University. Downloaded on April 26,2024 at 19:12:29 UTC from IEEE Xplore. Restrictions apply.

stocks using a hypergraph with the method of the frequent item sets. In the second step a special multilevel k-way hypergraph partitioning algorithm is used to partition the vertices of the hypergraph. In the third step, we will analyze the stock price trend according to the result of partitioning. The experiment can get a fine effect of our algorithm. The problem is that, the most suitable parameter spaces sometimes vary, but there may not be any easy way of doing this. The algorithm still gives a good example in predicting the stock price trend.

The rest of the paper is organized as follows. In Section 2 we introduce the proposed framework; we address the algorithm of the stock price prediction by hypergraph partition, which includes hyperedges segmentation and stock price analysis two parts, in Section 3; experiment are reported in Section 4, and followed by the conclusions finally.

II. HYPERGRAPH MODELING OF STOCKS

The stock price prediction scheme can be reduced to a problem of hypergraph partitioning. But first of all, it is essential to introduce the proposed framework, including the concept of hypergraph and frequent item sets, the create procedure of hypergraph based framework.

A. Hypergraph

In classical graph theory people represent the relationship among objects by pairwise vertices jointed by edges. Such kind of modeling is very useful for one-to-one relation problems. Under this kind of model, the vertices represent the objects, and any two vertices that related are joined by an edge. However, in the real world, the pairwise relationships among objects are not suitable to describe the problems. Hypergraph is similar to graph, but its edges (hyperedges) contact three or more vertices, which lets it have more advantages in the dealing the high dimensional data.

Formally, a hypergraph is an ordered pair G = (V, E) comprising a set of vertices V and a set of hyperedges E, where each hyperedge is a subset of the vertex set V. A hypergraph with every hyperedge containing just two vertices is a simple graph. Furthermore, a weighted hypergraph is a hypergraph that has a positive number ω (e) associated with each hyperedge e, called the weight of hyperedge e. We can represent a weighted hypergraph by$G = (V, E, \omega)$. In general, the weight of a hyperedge e is defined by the number of vertices in the hyperedge e, but in this paper we use a special method to weigh the hyperedge, and details in the 2.3. Sometimes we define a weight of the vertex v, which called degree $d\ (v)$. It is valuable in some cases, yet we cannot find a fittest method to combine it with our hypergraph model, we firmly believe that it will contribute a lot in the future scheme.

Figure 1 show an example to explain how to construct a hypergraph we described. It is obvious that the latter express the relationships among vertices using less edges (or hyperedges), and the graph cannot represent the relationship such as $e1$, which contains four vertices that having close connection with each other ($v1$, $v2$, $v3$, $v4$). At the same time, (b) also show a general way of partitioning the hypergraph, it gives a fine segmentation keeping the primary structure as possible. It means the segmentation can put the objects which have close relationships together in one section.

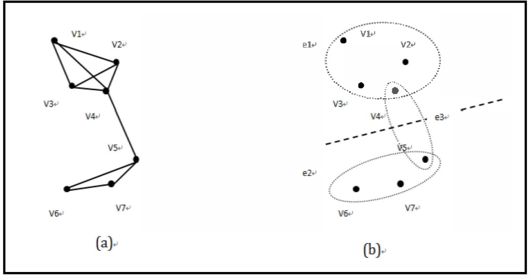

Figure 1. (a): A simple graph, it has 7 vertices ($v1$, $v2$, …, $v7$) and 10 edges (as (a) showed and not numbered). (b) The hypergraph which shows the similar structure of the graph in (a), it has 7 vertices (the same as (a)) and 3 hyperedges ($e1$, $e2$, $e3$). The dotted line means a general partition which is made on $e3$.

See more details about hypergraph in [16].

B. Frequent item sets

The method of frequent item sets is mainly used in the building of hyperedges and weighing their hyperedge weighs in a hypergraph.

The frequent item sets are sets of terms co-occurring in more than a threshold percentage of all documents of a database [15]. Such frequent sets computed by an association rule algorithm such as Apriori [17] are excellent candidates to find such related items. In general, we only find frequent item sets that have support greater than the threshold, which may be determined in a specific domain. This approach allows us to reduce drastically the size of model, which is efficient even for very large databases, and provides an understandable description of the discovered clusters by their frequent term sets. A frequent item-based approach of clustering is efficient, especially in hypergraph model. It allows us to reduce the scale of hyperedges as well as the large dimensionality of the vector space.

C. Hypergraph Based Framework of Stock Price Trend

In this paper, we develop a clustering method based on the hypergraph. Before the two main procedures, segmentation and prediction, we give an outline of the hypergraph based framework of stock price trend.

In the hypergraph model of stock transaction, we assume each stock as a vertex in the hypergraph. Note that the stock price varies with time. As an analysis model, we just take one day as minimum sampling unit. It means that we only pay attention to the opening price and closing price of a stock in one day. The change of the stock price as our benchmark can contribute in the building of hyperedges. Because stocks synchronously rise or fall in one day may be in a huge quantity, we set a limit value p to build a frequent item sets. Only the stocks synchronize in more than p days continuously can be included in the same hyperedge, and the

Authorized licensed use limited to: New York University. Downloaded on April 26,2024 at 19:12:29 UTC from IEEE Xplore. Restrictions apply.

p is related with the sample size *M* and have to be gained via experiments. At the same time, the weight of hyperedges should be given. In this algorithm, we find that the frequency of the frequent item sets as the weight of hyperedges is sufficient already, and the Apriorior the average confidence of the association rules cannot give an improvement dramatically enough. For example, we have several stocks that rise or fall synchronously in *n* consecutive days ($n \geq p$), a hyperedge can be built and given a weight by *n*.

III. Stock Price Prediction by Hypergraph Partition

In this section, we represent a special segmenting algorithm of hypergraph firstly, and stock price analysis based on the hypergraph segmented followed.

A. Hyperedge Segmentation

In this paper, we mainly do the partitioning with hmetis algorithm which describe in [18]. We assume that a hypergraph model have been built with fitting edges and weights. And a function is defined over the hyperedges is optimized. In this paper, the objective function we used, is also one of the most commonly used objective function, is to minimize the total number of hyperedges that span multiple partitions, and we called it the hyperedge-cut of the partitioning.

Here we use a method that computes a k-way partitioning directly based on the multilevel paradigm, which shows a better and faster partitioning than the recursive bisection paradigm in [18]. The partitioning framework is divided into three phases, which illustrated in Figure 2.

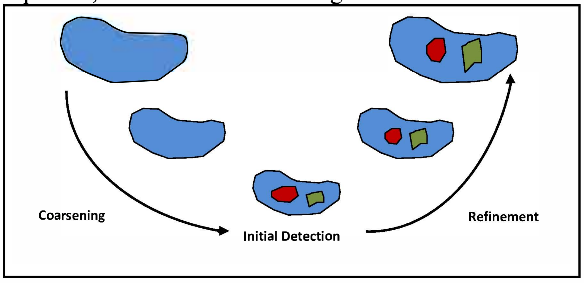

Figure 2. The three phases of the multilevel k-way partitioning algorithm. In the coarsening phase, the size of the hypergraph is successively decreased; in the initial detection phase, we give a k-way partitioning to the small hypergraph (two of these divided parts are showed in red and green for example in the figure); and in the refinement phase, the partitioning is successively modified and refined.

First of the phases is coarsening phase. In this phase, a series of successively smaller hypergraphs is set up. And the coarsening phase allows local refinement techniques such as FM (in the third phase) to become effective and helps reducing the sizes of the hyperedges without leading a pronouncedly worse result. We can use a scheme called hybrid first-choice (HFC) scheme [19], which is a combination of first-choice (FC) and greedy first-choice (GFC) scheme. And see more details in [18] [19].

The second phase of a multilevel k-way partitioning algorithm, called initial detection phase, is to compute a k-way partitioning of the coarsest hypergraph such that the balancing constraint is satisfied and the partitioning objective is optimized. And we use 10 initial partitioning and then filter out the bad ones, which described in [20], and it can be random or produced from some seeds.

In the third phase, refinement phase, we use a Fiduccia-Mattheyses (FM) scheme. In the FM, we find the fittest section for each vertex with counting the hyperedge-cut and moving, the parts can be changed gradually, which described in detail in [20].

After the three phases, our hypergraph can be partition into k parts, where the k may be determined according the stocks sample.

B. Stock Price Analysis and Prediction

We know that predicting a stock's price trend according its historical data may be quite difficult due to stock's inherent indeterminacy. However, we can find that the indeterminate stocks may have a synchronous trend in a long dimension of time. So we can analyze a stock by observing the closely related stocks in the same section which determined with the previous synchronicity. The overall trend of synchronous stocks in the same section can give us a constructive suggestion in predicting the target stock.

Now the stocks in the hypergraph model have been segmented into *k* parts, and we can analyze the stock's price trend with the clustering result. First, we need an overall parameter of rise or fall in the section, which is defined in (1).

$$f(S) = \sum_{v \in S} \mathrm{T}\,(v) \qquad (1)$$

In (1), *S* means sets of vertices in the section. And *v* means a vertex in *S*. *T* (*v*) means the average of the gains of a stock in *m* days, positive if rise, or negative. Then a $T'(u)$ is showed in (2).

$$T'(u) = \sum_{i=1}^{m} \frac{i \cdot T_i\,(u)}{\sum_{j=1}^{m} j} \qquad (2)$$

In (2), $T_i\,(u)$ means the stock *u* gains *i* days ago, positive if rise, or negative. Then the stock price trend can be predicted according the $R\,(u)$in (3). The $R\,(u)$reflects if the stock *u* will rise or fall next day by judging if it is positive or negative, but cannot reflect the range of variation.

$$R(u) = 0.7 * T'(\mathrm{u}) + 0.3 * f(S) \qquad (3)$$

IV. Experiments

In order to validate the proposed stock price prediction algorithm, we have conducted experiments using some real stock transaction data recorded 4 years ago. All these real data comes from Chinese Shanghai stock market, which is publicly available at http://www.10jqka.com.cn/.

For simplicity, we randomly select 3 stock portfolios to verify the performance of our algorithm. In each stock portfolio we check 20 stocks' price (M = 20 in 2.3) trend over an intermediate period span 3 months (from 2008.1.1 to 2008.4.1). And we take a *p* which mentioned in 2.3 as 3 (suitable in this case, and we suggest it can be 3 ~ 7, for reference only), we take a *k* which mentioned in 3.1 as 5

Authorized licensed use limited to: New York University. Downloaded on April 26,2024 at 19:12:29 UTC from IEEE Xplore. Restrictions apply.

(suitable in this case, and by 2012.3.16, the sum of stocks in China A-share market is 2331 and the suggested k can be 327, for reference only), we take a *m* which mentioned in 3.2 as 3(suitable in this case, and we suggest it can be 3 ~ 7 or the same as *p*, for reference only). Then the results will be compared with the real trend of 2012.4.2.

We show the detail experiment result of case 1 in Table I. And Figure 3 shows the report of result in three groups.

TABLE I. PREDICTION RESULT IN CASE 1

STOCKS	SECTION	PREDICTION	ACTUALITY	RESULT
600005	0	↓	↑	X
600019		↓	↓	√
600022		↓	↓	√
600015	1	↑	↑	√
600016		↑	↑	√
600018		↑	↑	√
600026		↑	↑	√
600004	2	↓	↓	√
600007		↓	↓	√
600017		↓	↑	X
600027		↓	↓	√
600009	3	↑	↑	√
600010		↓	↓	√
600012		↓	↓	√
600020		↓	↓	√
600000	4	↑	↑	√
600006		↓	↓	√
600008		↑	↓	X
600011		↓	↑	X
600021		↓	↓	√

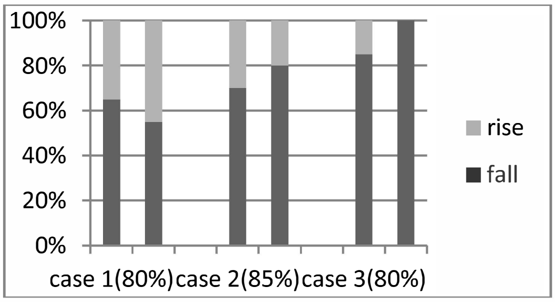

Figure 3. The predicting result of three cases, the left bar is the prediction and the right one is the actuality .

Finally, we can find that our stock price prediction scheme can give a prediction with a precision rate from 80% to 85%. It is a quite fine and inspiring result of prediction.

V. CONCLUSION

The stock price prediction scheme using a hypergraph based clustering in this paper already presented in a very fine level, and the result can give a quite useful suggestion to the stock investors. It is totally able to perform an excellent job in the real stock predicting work. Furthermore, the algorithm in some details of the modeling and segmentation can surely be improved for a faster and more accurate result. In the future work, we will add more excellent idea into our framework. Any questions or advices are welcome.

REFERENCES

[1] Y. M. Chae, S. H. Ho, K. W. Cho, D. H. Lee, and S. H. Ji, "Data mining approach to policy analysis in a health insurance domain," International Journal of Medical Informatics, vol. 62, no. 2, pp. 103−111, 2001.

[2] Robert P. Schumaker and Hsinchun Chen, "A Quantitative Stock Prediction System based on Financial News," Information Processing & Management, vol. 45, no. 5, pp. 571−583, 2009.

[3] J. A. Gentry, M. J. Shaw, A. C. Tessmer, and D. T. Whitford,"Using inductive learning to predict bankruptcy," Journal of Organizational Computing and Electronic Commerce, vol. 12, no. 1, pp. 39−57, 2002.

[4] R. K. Wolfe, "Turning point identification and Bayesian forecasting of a volatile time series," Computers and Industrial Engineering, vol. 15, pp. 378−386, 1988.

[5] Karl Nygren. "Stock Prediction – A Neural Network Approach," Master Thesis, Royal Institute of Technology, KTH, Apr. 2004.

[6] M. A. Kanoudan,"Genetic programming prediction of stock prices," Computational Economics, vol. 16, pp. 207−236, 2000.

[7] S. Schulenburg and P. Ross,"Explorations in LCS models of stock trading," Advances in Learning Classifier Systems, 2001, pages 151−180.

[8] O. Castillo and P. Melin, "Simulation and forecasting complex financial time series using neural networks and fuzzy logic," In Proceedings of IEEE Conference on Systems, Man, and Cybernetics, pages 2664−2669, 2001.

[9] Y. F. Wang,"Predicting stock price using fuzzy gray prediction system," Expert Systems with Applications, vol. 22, no. 1, pp. 33−38, 2002.

[10] R. Veliev, A. Rubinov, and A. Stranieri,"The use of an association rules matrix for economic modelling," In International conference on neural information processing, 1999, pages 836−841.

[11] S. Mabu, K. Hirasawa, and J. Hu, "A Graph-Based Evolutionary Algorithm: Genetic Network Programming (GNP) and Its Extension Using Reinforcement Learning," Evolutionary Computation, vol. 15, no. 3, pp. 369-398, 2007.

[12] J. Y. Potvin, P. Soriano, andM. Vallee,"Generating trading rules on the stock markets with genetic programming,"Computers & Operations Research,vol. 31, pp. 1033–1047, 2004.

[13] K. Kim, I.Han, "Genetic algorithms approach to feature discretization in arti-cial neural networks for the prediction of stock price index," Expert Syst. Appl,vol. 19, no. 2, pp. 125–132,2000.

[14] P Cheeseman and J Stutz,"Baysian classification (autoclass): Theory and results," In UM Fayyad, G. Piatetsky-Shapiro, P. Smith, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, 1996, pp. 153-180.

[15] E.H. Han et al,"Hypergraph-Based Clustering in Frequent item sets: A Summary of Results," Bull. Tech. Committee on Data Eng, vol. 21, no. 1, pp. 15-22, 1998.

[16] D. Zhou, J. Huang, and B. Schlkopf, "Learning with hypergraphs: Clustering, classification, and embedding," NIPS, pp. 1601–1608, 2007.

[17] R. Agrawal,R. Srikant,"Fast Algorithms for Mining Association Rules in Large Databases," Proc. VLDB 94, Santiago de Chile, Chile, 1994, pp. 487-499.

[18] George Karypis and Vipin Kumar,"Multilevel k-way Hypergraph Partitioning," VLSI Design, vol. 11, no. 3, pp. 285 - 300, 2000.

[19] George Karypis and Vipin Kumar,"hmetis,"A hypergraph partitioning package version 1.5.3. Technical report, 1998

[20] George Karypis, Rajat Aggarwal, Vipin Kumar, and Shashi Shekhar,"Multilevel hypergraph partitioning: Application in VLSI domain," Proceedings of the Design and Automation Conference, 1997.

Authorized licensed use limited to: New York University. Downloaded on April 26,2024 at 19:12:29 UTC from IEEE Xplore. Restrictions apply.