Research Article

Stock Market Trading Rules Discovery Based on Biclustering Method

Yun Xue, Zhiwen Liu, Jie Luo, Zhihao Ma, Meizhen Zhang, Xiaohui Hu, and Qiuhua Kuang

Laboratory of Quantum Engineering and Quantum Materials, School of Physics and Telecommunication Engineering, South China Normal University, Guangzhou 510006, China

Correspondence should be addressed to Yun Xue; 995438712@qq.com

Received 17 October 2014; Revised 30 January 2015; Accepted 31 January 2015

Academic Editor: Reza Jazar

Copyright © 2015 Yun Xue et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The prediction of stock market's trend has become a challenging task for a long time, which is affected by a variety of deterministic and stochastic factors. In this paper, a biclustering algorithm is introduced to find the local patterns in the quantized historical data. The local patterns obtained are regarded as the trading rules. Then the trading rules are applied in the short term prediction of the stock price, combined with the minimum-error-rate classification of the Bayes decision theory under the assumption of multivariate normal probability model. In addition, this paper also makes use of the idea of the stream mining to weaken the impact of historical data on the model and update the trading rules dynamically. The experiment is implemented on real datasets and the results prove the effectiveness of the proposed algorithm.

1. Introduction

The trend forecasting of the stock market has been a hot research field for a long time. However, it is influenced by many factors such as political events, general economic conditions, and traders' expectations, which make the stock market trend prediction become a challenging task.

The fundamental analysis is one of the main methods in the stock market analysis, which is based on the macro economy, the basic information of the companies, including profitability, industry prospects, and liabilities. Investors need to consider all the factors when they buy or sell a stock.

Technical analysis is another kind of method in the stock market analysis. It summarizes the typical rules in the market and forecasts the future trend by analyzing the historical price and the trading volume of the stocks. According to the efficient market hypothesis in 1960s and 1970s [1, 2], investors can quickly and effectively utilize the potential information in buying and selling stocks, which means all the factors affecting the stock price have been reflected in the price of the stock. Therefore, buy-and-hold (i.e., random selection) is the optimal strategy, and technical analysis of stock is invalid. Whereas the subsequent research result has given a different conclusion, a number of technical analysis methods have

emerged, ranging from traditional time series approaches to artificial intelligence techniques.

Because the stock price is a special kind of time series data, the traditional technical methods to predict the stock price are mainly time series analysis based on statistical models, such as autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA) [3, 4]. However, due to the high noise and the nonlinearity of the stock market, those methods are not satisfying. Therefore, a variety of advanced time series methods have been proposed, which are used to predict the stock time series data.

Allen and Karjalainen [5] generate the trading rules by the genetic algorithm (GA), which use the arithmetic and logical functions to combine the basic technical indicators.

Potvin et al. [6] generate short-term trading rules of 14 Canada companies' stocks by genetic programming which is an extension of GA on the basis of the historical pricing and transaction volume. When the market falls or when it is stable, the trading rules are generally useful, while when the market is rising, the trading rules do not provide any improvement over the buy-and-hold (BAH) approach.

An improved bacterial chemotaxis optimization (IBCO) is established by Zhang and Wu [7] as an effective prediction

model for the prediction of different kinds of stock price indices, which is integrated into the back propagation (BP) artificial neural network.

Chang and Liu [8] developed Takagi-Sugeno-Kang (TSK) type fuzzy rule based system for stock prediction. Their TSK fuzzy model introduced the technical indices as the input variables and the output is a linear combination of the input variables.

A trading system based on multiobjective particle swarm optimization was proposed by Briza and Naval [9]. By using the trading signals from a set of technical indicators, the system develops a trading rule which is optimized for two objective functions: Sharpe ratio and percent profit.

An intelligent hybrid stock trading system was designed by Chavarnakul and Enke [10], which integrated neural network, fuzzy logic, and genetic algorithm. The rules are generated based on single technical indicator-volume adjusted moving average (VAMA). The system shows the advantages of different methods, allowing the investors to make better stock trading decision.

Leigh et al. [11] implement a recognizer for two variations of the “bull flag” technical charting heuristic. They use this identifier to find trading rules and prove its validity.

Although the reliable basis for the technical analysis has not been set up, many investors and the market analysis staff use the technical indicators to analyze the stock price. Most of the technical analysis methods generate trading rules based on one or a few predefined technical indicators. However, it is not always effective by using a technical indicator or the combination of some technical indicators. As the stock prices are affected by a variety of deterministic and stochastic factors, the stock price model should not be static all the time. No technical indicator can be used to construct all the models. Therefore, when the model of the stock price changes, we should choose different technical indicators to build different models.

In order to solve the above problem, we use a data mining method-biclustering technique to find local patterns in the historical data where different patterns contain a subset of technical indicators with different periodic parameters [12]. The model found is regarded as the trading rules, which are applied in the short term prediction of the stock price, combined with the minimum-error-rate classification of the Bayes decision theory under the assumption of multivariate normal probability model. In addition, this paper also makes use of the idea of the stream to update the trading rules dynamically and weaken the impact of historical data on the model.

The main works in this paper are as follows: (1) propose a new biclustering algorithm to find the trading rules in the quantized data; (2) provide guidance of the stock trading by the combination of the trading rules generated and the minimum-error-rate Bayesian decision method under the assumption of multivariate normal probability model; (3) update the trading rules dynamically and weaken the impact of historical data on the model based on the idea of stream; (4) validate the performance of the algorithm using three kinds of historical time series data from stocks and compare it with

several classical technical analysis methods, including buy-and-hold, genetic programming [6], and intelligent hybrid system [10].

The rest of the paper is structured as follows: in Section 2, the introduction of bicluster model is presented; in Section 3, we explain our algorithm in detail; Section 4 proposes the experiment method; Section 5 gives experimental results; Section 6 concludes the paper and gives the main direction of future work.

2. Bicluster Model

Clustering is one of the most important methods to solve the problem of machine learning, data mining, and pattern recognition problems. It has been widely used in various fields, such as biological gene, recommendation system, and financial analysis. The traditional clustering methods, such as hierarchical clustering [13], self-organizing maps [14], and K -means clustering [15], cluster rows or columns, respectively, in a matrix. It could not find local coherent patterns which include subsets of rows and columns. Generally, the local patterns would be more beneficial for mining implicit information. For instance, a trader may care about a small number of technical indicators which could provide the most useful predictions to the market to make trading decisions. Thus, finding a subset of useful indicators, which have similar behaviors in turning points, is important to the analysis of the stock market. To solve this problem, we need biclustering methods, which are actually a special branch of clustering algorithms because it clusters the data both the row and the column simultaneously in a matrix.

Cheng and Church [16] firstly proposed biclustering terms in gene expression data analysis. After that, a series of biclustering algorithms were raised such as FLOC [17], OPSM [18], and Plaid model [19].

Given a $N_r \times N_c$ data matrix, a bicluster could be defined as a submatrix whose row set and column set could be expressed as (R, C) , where $R \subseteq \{1, \dots, N_r\}$ is a subset of rows and $C \subseteq \{1, \dots, N_c\}$ is a subset of columns in the original matrix. The R (a subset of rows) share a certain kind of pattern in C (a subset of columns).

Madeira and Oliveira [20] divide the biclusters into four types: (1) biclusters with constant values; (2) constant values in rows or columns; (3) coherent values; and (4) coherent evolutions. Figure 1 explains ten typical examples for these four types of biclusters. Figures 1(a)–1(e) represent the first three which have the numeric values in the data matrix. The biclusters with coherent evolution are represented by Figures 1(f)–1(j). All kinds of biclusters have their own characteristic and are suitable for different data and fields. In this paper, the biclusters with constant values in columns are chosen.

3. Method

Technical analysis aims at predicting price trend by some rules based on the study of historical data in the market. These methods are based on figures or models which could be described by mathematical formulas that set historical data

FIGURE 1: Examples of different types of biclusters. (a) Constant values in (b) rows and (c) columns. (d) Additive and (e) multiplicative coherent values. (f) Overall coherent evolution in (g) rows and (h) columns. (i) Coherent evolution values in columns. (j) Coherent sign changes in rows and columns.

as input and the trading indicator as output. The rules found could help investors to make better decisions in the markets. Technical analysis is based on the assumption that there exist consistent behavior patterns that are time invariant associated with the stock price and would recur in the future; thus these patterns can be used for predictive purposes. As mentioned above, a bicluster is a submatrix which could be regarded as a local coherent pattern. Biclustering methods could find the coherent patterns in the stock data, which we used to generate the trading rules.

In this paper, we use the historical data of the stock or the financial comprehensive index to build the data matrix, where the rows are the trading days and the columns are the technical indicators and future return. The different combinations of the technical indicators mean different trading rules. Some of them (combinations of the technical indicators) appear in the training sets in many trading days. As Twain's often quoted saying "history does not repeat itself, but it does often rhyme". The patterns corresponding to those combinations will occur in the future and can be regarded as trading signals which imply the stocks rise or fall, so we can refer to the signals to make the decision to buy or to sell. In other words, when some technical indicators fall into a certain range, it can be regarded as the stock change signal. It is meaningful and significant that a technical indicator shows approximate or similar values in different days. Therefore, our method mine biclusters with constant value on columns in the data matrix. Considering that the local information and

manifold information are both important for data clustering, Biclustering-Based Intelligent System (BIS) is proposed to mine the biclusters with constant values in columns inspired by the method of [21].

The procedure is stated as follows: (1) cluster each column in a data matrix by K -means algorithm; (2) mine the biclusters with constant value in columns of the matrix. As a result, only those biclusters that contain both technical indicators and future return are taken into consideration. The detected biclusters are the stock trading rules. When the trading data of the future could match with the trading rules, to buy or sell signal will be determined to make the corresponding stock trading decisions. In addition, in order to find the trend of the current stock price and to weaken appropriately the influence of historical data for prediction, an innovative method based on support is introduced to update the trading rules in the testing process. The biclustering method can find local consistency model in time series data which is in accordance with the characteristics of stock data matrix. Although there is no complex step, it has obtained the good effect; the steps are stated as follows.

3.1. Data Preprocessing. The trading historical data is organized into a matrix, as shown in Figure 2, in which the rows represent the trading days and the columns correspond to technical indicators where the first column is the return and the rest columns correspond to technical indicators with different time spans. In this paper, five popular technical

FIGURE 2: The historical data matrix of the stock market.

FIGURE 3: Different time spans (number of weeks) for the technical indicators MA, RSI, %R, and ROC.

indicators are selected, which are moving average (MA), the relative strength indicator (RSI), the Williams percentage range (%R), the rate of change (ROC), and the trading volume. According to [21], we chose 7 different time spans for MA and 5 different time spans for RSI, %R, and ROC. The distribution of time spans for different technical indicators is shown in Figure 3.

The digits in the cells of Figure 3 represent different time parameters when the corresponding technical indicators are selected.

The indicator of the first column indicates the return value of the i th trading day in a week and its values are calculated according the following:

$$\text{return}(i) = \frac{\text{cpw}(i) - \text{cp}(i)}{\text{cp}(i)}, \quad (1)$$

where $\text{cp}(i)$ is the closing price of the i th trading day and $\text{cpw}(i)$ is the closing price after one week.

The 2nd–8th columns indicate moving average (MA) with 7 different time parameters. MA is a calculation to

analyze data points by creating a series of averages of different subsets of the full dataset, which is calculated as follows:

$$\text{MA}(n) = \frac{1}{n} \sum_{i=0}^n \text{cp}(n-i). \quad (2)$$

In the above formula, $\text{cp}(n-i)$ is the closing price of the $(n-i)$ th trading day.

The 9th–13rd columns are relative strength index (RSI) with different time parameters. The RSI intended to chart the current and historical strength or weakness of a stock or market based on the closing prices of a recent trading period. It is classified as a momentum oscillator, measuring the velocity and magnitude of directional price movements. It is calculated as follows:

$$\text{RSI}(n) = 100$$

$$- 100 \left(1 + \frac{\sum_{i=1}^n \max((\text{cp}(i) - \text{op}(i)), 0)}{\sum_{i=1}^n \min((\text{cp}(i) - \text{op}(i)), 0)} \right)^{-1}, \quad (3)$$

where $\text{op}(i)$ represents the open price of the i th trading day.

The 14th–18th columns are Williams percentage range (%R) with different time parameters. %R shows the current

closing price in relation to the highest and lowest of the past n days. It is calculated as follows:

$$\%R(n) = 100 - \frac{cp(i) - L_n}{H_n - L_n} \times 100, \quad (4)$$

where H_n is the highest price over n previous periods and L_n is the lowest price over n previous periods.

The 19th–23rd columns are the rate of change (ROC) with different time parameters. ROC shows the difference in the closing price between today and n days ago, which is calculated as follows:

$$ROC(n) = \frac{cp(t) - cp(t-n)}{cp(t-n)} \times 100. \quad (5)$$

3.1.1. Data Normalization. Due to the different ranges of the technical indicators value, we use the min-max standard method (as equation (6)) to normalize the data in the matrix (from the second to the twenty-fourth column) into $[0, 1]$:

$$D(i, j) = \frac{D_0(i, j) - \min(D_0(:, j))}{\max(D_0(:, j)) - \min(D_0(:, j))}, \quad (6)$$

where D_0 represents the data matrix before normalization while D is after normalization and $\min(D_0(:, j))$ and $\max(D_0(:, j))$ represent the minimum and maximum value of the j th column, respectively.

3.1.2. Data Quantization. It is believed that each indicator's value can be divided into several classes, and the data falling in the same class will have similar impact on the stock price though they are not strictly equivalent. Therefore, K -means algorithm is used to cluster the normalized data into K classes and the data dropped in the same cluster will be quantized as a constant which reflects the center of the class.

In order to better identify the stock market's seesaw movements, we set threshold value α . If the first column's value (future return) is greater than α , it is replaced by 1. If it is less than $-\alpha$, it is replaced by -1 , otherwise it is set as 0. Among them, 1 represents the rising of the stock, -1 represents the stock devaluation, and 0 represents that the stock price does not change significantly. The changes before and after quantization of the matrix are shown in Table 1, where α is 0.01.

3.2. Training. The training set is a part of history data of one stock. The BIS algorithm is implemented to find the effective stock price prediction model, namely, the stock trading rules, and further to verify the validity of the stock trading rules in the testing phase. The Hang Seng index (HSI) here is introduced as an example, which spans nearly a year from May 3, 2005, to April 28, 2006. Because the return values of the 52th week are acquired in the next week after April 28, 2006, there are only 51 weeks' complete data.

(1) Proposed Biclustering Method. As an efficient data mining tool, biclustering technique is an extension to the traditional

TABLE 1: (a) The matrix after normalization; (b) the matrix after quantization.

(a)					
Return	MA(4)	MA(8)	...	ROC(24)	Volume
−0.03	0.45	0.80	...	0.52	0.09
−0.02	0.60	0.10	...	0.87	0.58
0	0.20	0.56	...	0.45	0.54
0.01	0.80	0.45	...	0.25	0.97
⋮	⋮	⋮	⋮	⋮	⋮
0	0.90	0.40	...	0.43	0.06
0.02	0.25	0.56	...	0.42	0.07
(b)					
Return	MA(4)	MA(8)	...	ROC(24)	Volume
−1	0.57	0.73	...	0.53	0.07
−1	0.57	0.12	...	0.87	0.54
0	0.23	0.48	...	0.43	0.54
1	0.85	0.48	...	0.25	0.97
⋮	⋮	⋮	⋮	⋮	⋮
0	0.85	0.48	...	0.43	0.07
1	0.23	0.48	...	0.43	0.07

clustering methods, which allows simultaneously cluster rows and columns to find the submatrices where the rows show the same pattern in the corresponding column sets. A submatrix represents a local meaningful pattern hidden in the mass data.

The historical data of stock transaction usually forms a matrix where the rows correspond to the dates of the transactions and the columns correspond to the technical indicators. When some technical indicators of stocks fall in a specific range, the trader should make the decision to buy or sell, which are the trading rules. Therefore the trading rules can be represented as some turning points whose specific technical indicators fall in the same range, and the trading dates and technical indicators compose a matrix which accorded with the characteristics of biclusters with constant values in the columns. So our algorithm searches all those biclusters in the training matrix to get the stock trading rules and guide the stock trading.

In order to forecast the stock prices' change according to the historical data of stock trading, an effective trading rule must contain a return and several technical indicators. That is to say, a bicluster should contain two parts: a return (the 1st column) and the technical indicators. Based on the constraints, we raise a new algorithm to find the biclusters with constant values in the columns whose support is beyond the support threshold T_r , namely, BIS algorithm mentioned before. It starts with each column. Gradually the biclusters are merged into biclusters with constant values in two or more columns. The following is an example of detailed procedures.

After data preprocessing, the original data matrix D_0 is transformed into matrix B , as shown in Table 2; the threshold T_r is 2.

TABLE 2: Matrix B .

	Column 1	Column 2	Column 3	Column 4	Column 5
Row 1	1	0.73	0.48	0.34	0.69
Row 2	1	0.73	0.48	0.34	0.69
Row 3	0	0.73	0.74	0.62	0.18
Row 4	-1	0.73	0.74	0.62	0.40
Row 5	-1	0.23	0.18	0.89	0.40

TABLE 3: Records of the support rows of each class in each column.

Column number	Pattern 1	Pattern 2	Pattern 3
1	{1, 2}	{4, 5}	{3}
2	{5}	{1, 2, 3, 4}	
3	{5}	{1, 2}	{3, 4}
4	{1, 2}	{3, 4}	{5}
5	{3}	{4, 5}	{1, 2}

The process of the algorithm is described as follows:

Step 1. After K -means clustering in the data preprocessing, elements in each column have been clustered into several clusters and biclusters of one column $BIC_j(i)$ ($j = 1, 2, \dots, n$, $i = 1, 2, \dots, k$) are obtained, which means the i th bicluster of the j th column. Then add the biclusters of one column whose row numbers are beyond the row threshold T_r into the bicluster set BIC_Set at the same time, and these biclusters are regarded as the set of bicluster seeds (BS) for the next step. The results are shown in Table 3.

Step 2. In BS, $BIC_1(i_1)$ ($i_1 = 1, 2, \dots, k$) in Step 1, that is, all the biclusters of the first column in B are combined with other biclusters $BIC_j(i_2)$ ($j = 2, 3, \dots, n$, $i_2 = 1, 2, \dots, k$) to get the biclusters of two columns. The row sets of two merged biclusters are intersected and the column sets of two merged biclusters are joined. The results which could not satisfy the row threshold T_r are deleted. Finally all the satisfying biclusters of two columns are added to the bicluster set BIC_Set and regarded as the new BS set.

As described in Step 2, each pattern of the first column in Table 3 is merged with patterns of the second column or patterns of other columns. The row sets of two merged patterns are intersected and the column sets are joined to get all the length-2 patterns including the first column. Since there are only two patterns in each column that meet the support in Table 3, at most four new models will be obtained after the merging operation. The result is shown in Table 4.

Step 3. All two biclusters in BS will be merged to get the biclusters of three columns. The satisfying biclusters beyond the support are added to BIC_Set . Because all the biclusters of two columns found in Step 2 include the first column, the merged biclusters contain the first column either. Thus each bicluster in BIC_Set has its own return value.

TABLE 4: Records of the support row sets of each class composed of 2 columns.

Column number	Pattern 1	Pattern 2	Pattern 3	Pattern 4
1, 2	{1, 2}	{4}		
1, 3	{1, 2}	{0}	{0}	{4}
1, 4	{1, 2}	{0}	{0}	{4}
1, 5	{0}	{1, 2}	{4, 5}	{0}

TABLE 5: Records of the support row sets of each class composed of 3 columns.

Column number	Pattern 1
1, 2, 3	{1, 2}
1, 2, 4	{1, 2}
1, 2, 5	{1, 2}
1, 3, 4	{1, 2}
1, 3, 5	{1, 2}
1, 4, 5	{1, 2}

TABLE 6: Records of the support row sets of each class composed of 4 columns.

Column number	Pattern 1
1, 2, 3, 4	{1, 2}
1, 2, 3, 5	{1, 2}
1, 2, 4, 5	{1, 2}
1, 3, 4, 5	{1, 2}

TABLE 7: Records of the support row sets of each class composed of 5 columns.

Column number	Pattern 1
1, 2, 3, 4, 5	{1, 2}

For example, in Table 4 the patterns in $\langle 1, 2 \rangle$ are merged with each pattern in $\langle 1, 3 \rangle$, $\langle 1, 4 \rangle$, and $\langle 1, 5 \rangle$, respectively. The corresponding row sets are intersected to get the corresponding support row set of all the length-3 patterns containing the column set $\langle 1, 2 \rangle$. Then the patterns in column set $\langle 1, 3 \rangle$ are merged with other patterns in $\langle 1, 4 \rangle$ and $\langle 1, 5 \rangle$ to get all the length-3 patterns containing the column set $\langle 1, 3 \rangle$ and their corresponding row sets. The same operation is done with patterns in $\langle 1, 4 \rangle$ and patterns in $\langle 1, 5 \rangle$ to get all the length-3 patterns containing the column set $\langle 1, 4 \rangle$ and their corresponding row sets. The results obtained are shown in Table 5.

Step 4. Repeat Step 3 until BS is empty or the number of columns of the biclusters reaches n .

According to Step 4, the follow-up results are shown in Tables 6 and 7.

Step 5. Filter BIC_Set ; that is, discard the repeated biclusters and the biclusters whose column numbers are less than the column threshold T_c .

FIGURE 4: The flow chart of the BIS algorithm.

The flow chart of the BIS algorithm is shown in Figure 4.

In BIS algorithm, each column (each index) is clustered by K -means clustering; thus we do not need to set a specific threshold to measure the similarity among the indicators of different trading days. After merging, the intersection of the row sets ensures that the obtained biclusters are biclusters with constant value in columns; that is, those trading days share the same patterns in the corresponding indicator sets. It is detailed by the pseudo code in Algorithm 1.

(2) *Generate the Trading Rules.* After all the biclusters with the constant values in columns have been obtained, which satisfy the row support threshold T_r and contain the first column, the biclusters meeting the column threshold T_c are selected as the effective predicting models, as shown in Table 8.

The summary information of the satisfying biclusters is stored including its column set and the corresponding values

as well as the row support. A bicluster in Table 8 is saved in Table 9.

Since each bicluster corresponds to a candidate prediction model, its summary information could be regarded as an effective trading rule.

If the future return of a transaction rule is 1, which shows a rising trend, there is a cue to buy; otherwise, if it is -1 , which shows a falling trend, there is a selling signal. If the return is 0, no action should be taken.

After all the trading rules have been generated from the satisfying biclusters, all the data in the matrix is discarded except the data of the last week in this year, which is left with the data of the second year for updating the transaction rules dynamically.

3.3. Testing. Another part of the historical data of the stock is set as the test data. Take Hang Seng index (HSI) as an example; the data from May 2nd, 2006, to April 30th, 2007, is set as the test data. The trend of the stock price for each trading day in a week is forecasted by the pregenerated trading rules. In addition, the new data is used to update the trading rules and weaken the influence of the historical data, like in stream mining. Next we will take the data on May 2nd, 2006, as an example to predict its stock price's trend in a week and update the trading rules by the new data.

(1) *Predict.* In order to forecast the trend of the stock price after the trading day, the daily data of each column is normalized firstly; then the matching degree with each transaction rule i is computed. The index's value of the trading rule i and the corresponding indices' value of this rule on this day are compared according to formula (7). The smaller the $\text{distance}(i)$ is, the higher matching degree between the trading day and the trading rule i is:

$$\text{distance}(i) = \sqrt{\sum_{j \in \text{ST}} \frac{(v_{\text{td}}(j) - v_{\text{rl}}(j))^2}{n_j}}, \quad (7)$$

where ST is a collection of technical indices of the trading rule i ; n_j is the number of the technical indices of the trading rules i ; $V_{\text{rl}}(j)$ is the j th index's value of the trading rules i , and $V_{\text{td}}(j)$ is the technical indices' values corresponding to the trading day.

The trading rules whose $\text{distance}(i)$ is less than the threshold T_{dis} are set as the reference trading rule set of the trading day, and the process of the trading decision is stated as follows.

If there is only one trading rule, or the multiple trading rules have the same return, then to buy or sell the stock is decided by the return of the trading rule. When the return is 1, it suggests a purchase signal on the next trading day; if the return is -1 , it suggests a signal to sell. If the return is 0, no action should be taken.

If there are more than one trading rules satisfying threshold T_{dis} , and they do not have the same return value, then it is regarded as a classification problem to find the best trading rule and each trading rule is considered as a class. Each technical index of a transaction rule is assumed

```

* * * * * Some mathematical symbols involved in this paper * * * * *
% B: stock trading data matrix after preprocessing.
% n: the number of columns in B.
%  $T_r$ : the row threshold of bicluster.
%  $T_c$ : the column threshold of bicluster.
% BS: the set of bicluster seeds.
% BIC_Set: the set of biclusters.
Input      B
Output    BIC_Set
Let       BS = {} and BIC_Set = {}
Description:
(1) * * * * * Obtain all biclusters with one column * * * * *
(2) for  $j = 1 : n$ 
(3)   Apply the  $K$ -means algorithm for the  $j$ th column.
(4)   Then we obtain a set of clusters in the  $j$ th column, denoted as  $BIC_j(i)$  where  $i = 1, 2, \dots, k$ .
(5)   % Update BIC_Set.
(6)   if the number of rows in  $BIC_j(i) \geq T_r$ 
(7)     Add it into BIC_Set.
(8)   end if
(9) end for
(10) BS = BIC_Set.
(11) * * * * * Expand BS * * * * *
(12) % The following steps are implemented in BS.
(13) % Merge all the biclusters of one column to get the biclusters of two columns.
(14) for  $j = 2 : n$ 
(15)   for  $i_1 = 1 : k$ 
(16)     for  $i_2 = 1 : k$ 
(17)        $BIC_{i_1}(i_1)$  is jointed with  $BIC_{i_2}(i_2)$ . % that is, intersect the row sets and join the column sets.
(18)       Update BIC_Set similar to the process mentioned above.
(19)     end for
(20)   end for
(21) end for
(22) The newly obtained biclusters of two columns which satisfy the row threshold  $T_r$  are put into BS.
(23) % Continue merging to get biclusters of more columns.
(24) while BS  $\neq \{\}$  and the number of columns in biclusters is less than  $n$ 
(25)   Merge every two length- $j$  biclusters (biclusters whose number of columns are  $j$ ) in BS,
(26)   to get length- $j + 1$  biclusters whose number of columns are  $j + 1$ .
(27)   Add the biclusters that meet the threshold  $T_r$  into BIC_Set.
(28)   Put the satisfying length- $j + 1$  biclusters into BS
(29) end while
(30) * * * * * Output BIC_Set * * * * *
(31) Filter out the duplicate biclusters and delete the biclusters whose column number is less than  $T_c$ .
(32) Output BIC_Set

```

ALGORITHM 1: The pseudo code of BIS biclustering algorithm.

TABLE 8: An effective bicluster.

	Return(1)	$R(10)$	$R(12)$	$R(16)$	$R(24)$	% $R(4)$	Volume
Day 13	-1	0.0889	0.0924	0.0996	0.0828	0.5430	0.1333
Day 50	-1	0.0889	0.0924	0.0996	0.0828	0.5430	0.1333
Day 171	-1	0.0889	0.0924	0.0996	0.0828	0.5430	0.1333

TABLE 9: The summary information of a bicluster.

Return(1)	$R(10)$	$R(12)$	$R(16)$	$R(24)$	% $R(4)$	Volume	Support
-1	0.0889	0.0924	0.0996	0.0828	0.5430	0.1333	3

TABLE 10: Bicluster 1.

Return	MA(4)	MA(8)	MA(12)	MA(16)	MA(24)	MA(36)	RSI(8)
1	0.1338	0.1276	0.1084	0.0926	0.1105	0.0890	0.3760
1	0.1338	0.1276	0.1084	0.0926	0.1105	0.0890	0.3760
1	0.1338	0.1276	0.1084	0.0926	0.1105	0.0890	0.3760

TABLE 11: Bicluster 2.

Return	%R(8)	ROC(4)	ROC(12)	ROC(16)	ROC(24)	Volume
-1	0.5429	0.0889	0.0922	0.0997	0.0828	0.1335
-1	0.5429	0.0889	0.0922	0.0997	0.0828	0.1335
-1	0.5429	0.0889	0.0922	0.0997	0.0828	0.1335

as a random variable which obeys the normal distribution. The best trading rule is selected by the minimum-error-rate classification of the Bayes decision theory under the assumption of the multivariate normal distribution. The discriminant function for each trading rule i is shown as follows:

$$g_i(X) = -\frac{1}{2}(X - \mu_i)^T \Sigma_i^{-1}(X - \mu_i) - \frac{d}{2} \ln(2\pi) - \frac{1}{2} \ln |\Sigma_i| + \ln p(\omega_i), \quad (8)$$

where d is the number of technical indices of the trading rule, X is part of the indices' value (d dimensional column vector) corresponding to the specific transaction rules, Σ_i is the $d \times d$ covariance matrix, here it is a unit matrix, and $p(\omega_i)$ is the prior probability calculated in the following:

$$p(\omega_i) = \frac{\text{support}(i)}{n}, \quad (9)$$

where the $\text{support}(i)$ is the support of the trading rule i and n is the total number of rows in the training data.

We select the best trading rules (whose $g_i(X)$ is the maximum) to buy or sell a stock according to its return value. An example is depicted to explain how to make the trading decision.

- (1) Two biclusters are detected in the matrix, as shown in Tables 10 and 11.
- (2) Generate the trading rules, as shown in Tables 12 and 13.
- (3) The values of the set of indicators corresponding to the two trading rules for a specific trading day in testing period are shown in Tables 14, 15, and 16.

According to (7), it is obvious that the set of values matches both of the two trading rules well.

According to (8), $g_1(X) > g_2(X)$, rule 1 is the best trading rule, so we will sell the stock in the next trading day.

(2) *Update the Transaction Rules Dynamically.* It is believed that the most recent data affects the future stock price more than the historical data, which gradually reduces its influence as the time passes by. Therefore, the data stream is used

to update the trading rules dynamically and weaken the influence of historical data.

When the data on May 2nd, 2006, is acquired, the return in a week of the earliest day (April 24th, 2006) retained could be obtained. It is combined with the retained data into a complete record, as shown in Table 17. The complete record is used to update the transaction rules.

The data is normalized and quantized; then the matching trading rules are found and the row support of the trading rules which match successfully are updated. For each of the trading rules, calculate the matching degree according to formula (7). If the value is 0, it shows that the day's data exactly matches the trading rule, so the support of the trading rule adds 1. If there are several matching trading rules, then all the corresponding supports add 1. Finally, the total number of rows in the training data also increases 1.

After the trading rules have been updated and the transaction decision has been made for May 2nd, 2006, delete the data of April 24th, 2006, and save the trading data of May 2nd, 2006.

In order to weaken the influence of the historical data on the trading rules, the supports of all the trading rules are multiplied by an attenuation coefficient γ at the end of each month after the renewal of the trading rules by the month's new data. Then delete the trading rules whose row support is lower than the row threshold with consideration that these rules would not recur and therefore would impact little on the future trading decision. The process is repeated until the end of the test.

4. Experimental Methods

In order to evaluate the performance of the BIS algorithm, we compare it with 3 popular existing methods: (1) buy-and-hold (BAH); (2) genetic programming (GP) [6]; (3) intelligent hybrid system (IHS) [10]. The BAH is a classic and simple stock trading strategy whose trading strategy is to buy the shares on the first day and sell it at the end without intermediate operation. The GP extends classical genetic algorithms by allowing the processing of nonlinear structures. It provides a flexible framework for adjusting the trading rules to the current environment. The IHS integrates fuzzy logic, GA, and NN techniques to increase the efficiency of stock market when using VAMA.

TABLE 12: Rule 1.

Return	%R(8)	ROC(4)	ROC(12)	ROC(16)	ROC(24)	Volume	Support
-1	0.5429	0.0889	0.0922	0.0997	0.0828	0.1335	3

TABLE 13: Rule 2.

Return	MA(4)	MA(8)	MA(12)	MA(16)	MA(24)	MA(36)	RSI(8)	Support
1	0.1338	0.1276	0.1084	0.0926	0.1105	0.0890	0.3760	3

TABLE 14: Values of a trading day after normalization.

MA(4)	0.1340
MA(8)	0.1277
MA(12)	0.1086
MA(16)	0.0925
MA(24)	0.1103
MA(36)	0.0893
MA(48)	0.0900
RSI(8)	0.3762
RSI(10)	0.3400
RSI(12)	0.3560
RSI(16)	0.3675
RSI(24)	0.3772
%R(8)	0.5430
%R(10)	0.5436
%R(12)	0.5324
%R(16)	0.5378
%R(24)	0.5460
ROC(4)	0.0900
ROC(8)	0.0932
ROC(12)	0.0924
ROC(16)	0.0996
ROC(24)	0.0826
Volume	0.1333

TABLE 15: Value of the set of indicators corresponding to rule 1.

ROC(4)	ROC(12)	ROC(16)	ROC(24)	%R(8)	Volume
0.0900	0.0924	0.0996	0.0826	0.5430	0.1333

TABLE 16: Value of the set of indicators corresponding to rule 2.

MA(4)	MA(8)	MA(12)	MA(16)	MA(24)	MA(36)	RSI(8)
0.1340	0.1277	0.1086	0.0925	0.1103	0.0893	0.3762

TABLE 17: A latest complete record.

Return(1)	MA(4)	...	RSI(4)	...	%R(4)	...	ROC(4)	...
Day 227	1	2	...	3	...	1	...	3

In the comparison, 3 real stock datasets are used: (1) Hang Seng index (HSI) (Table 18) from 05/03/2005 to 04/30/2007; (2) four Canadian companies stocks [6] (Table 19) from

TABLE 18: Dataset of Hang Seng index.

Index	Training period	Testing period
HSI	05/03/2005–04/28/2006	05/02/2006–04/30/2007

06/30/1992 to 06/30/2000; (3) Standard & Poor (S&P) 500 index (Table 20) from 01/01/1995 to 12/30/2003. All datasets are downloaded from the Yahoo Finance [22]. The corresponding training time and testing time are shown in the tables.

In the following experiments, the trading rules found by the BIS algorithm in the training period are employed to guide the business conduct in the testing. The trading strategy is that if a trading day forecast returns a signal to buy, then open a position in the next trading day until a signal to sell occurs which means to close the position on the next trading day. During the closing time even there are some signals to buy emerge, no operations will be executed. Repeat this trading mode during the test period. If there is not any sell signal after opening a position, close the position in the final day of the testing period.

To measure the profitability, the results will be presented as the stock return accumulated over the entire trading period [10], computed as follows:

$$\text{profit (\%)} = \sum_N \frac{\text{op}(t) - \text{op}(i)}{\text{op}(i)}, \quad (10)$$

where $\text{op}(i)$ is the opening price of the trading day to open a position, $\text{op}(t)$ is the opening price of the trading day to close a position, and N is the number to open a position during the test.

5. Experimental Result

According to parameter setting of [23], we compared the performance of the algorithm on different datasets with other algorithms. The experimental results are shown in Tables 21–23. As it can be seen from the tables, the trading rules found by the BIS algorithm could gain more profits than the trading rules generated by buy-and-hold (BAH), genetic programming (GP), and intelligent hybrid system (IHS) model.

The program is implemented with MATLAB (MatlabR2011b) and run on Windows7 Platform with a configuration of Pentium(R) CPU G2030 and 4 GB memory.

Tables 21–23 give the comparison of the BIS algorithm and three technical analysis methods in the corresponding

TABLE 19: Dataset of four Canadian companies stocks.

Activity sector	Company	Symbol	Period
Precious metals	Barrick Gold Corporation	ABX	Training
Pipelines	Trans Canada Pipelines Ltd.	TRP	06/30/1992–06/25/1999
Oil and gas	Cdn. Occidental Petroleum Ltd.	CXY	Testing
Diversified (conglomerate)	Canadian Pacific Ltd.	CP	06/28/1999–06/30/2000

TABLE 20: Dataset of S&P 500 index.

S&P 500 index	Training period	Testing period
Trending-up market	01/01/1998–12/30/2002	01/01/2003–12/30/2003
Flat market	01/01/1995–12/30/1999	01/01/2000–12/30/2000
Trending-down market	01/01/1997–12/30/2001	01/01/2002–12/30/2002

TABLE 21: Comparison with the BAH on Hang Seng index. “Profits (%)” is excess return. “Trades” is the number of trades.

Index	BAH Profits (%)	BIS	
		Profits (%)	Trades
HSI	20.20	27.19	101

TABLE 22: Comparison with the GP on four Canadian stocks. “Profits (%)” is excess return. “Trades” is the number of trades.

Index	GP profits (%)	BIS	
		Profits (%)	Trades
ABX	31.76	36.54	15
CP	17.36	21.42	1
CXY	29.23	31.45	15
TRP	14.62	13.80	1
Average	23.24	25.80	8

training dataset. The output of the BIS algorithm includes the profit rate and the number of transactions in testing. On the whole, the BIS algorithm outperforms the other three transaction methods in the simulation experiment.

From Table 21 we can see that the BIS algorithm could boost the profit rate by 6.99% than BAH strategy in the guidance of HSI stock trading (27.19% by BIS while 20.20% by BAH), which shows its significant advantage in profitability. From Table 22, the BIS method could obtain more profit than GP in the four stocks. Considering the average profit, the BIS algorithm (25.80%) could gain more than GP (23.24%) algorithm. The data in Table 23 shows that the BIS method can get better profits on the index in S&P 500; particularly in Trending-down market, the profit with the method of BIS (13.63%) is a significant improvement compared to the profit of the IHS method (−5.59%).

Through the above analysis of the simulation experiments, the proposed BIS algorithm has obvious advantages compared with other classical strategies in the stock trading. Although there are negative returns in some stocks (such as TRP), generally speaking, the BIS method can obtain better effect in mining stock exchange rules and assisting trading decisions.

FIGURE 5: Trading points for ABX from 03/14/2000 to 06/30/2000.

Figure 5 gives an example of guiding ABX stock trading by the BIS method from 03/14/2000 to 06/30/2000, which intuitively displays the process in the guidance of stock trading by the trading rules of the BIS method. The fold line describes the stock price fluctuations over time while the red dots and the blue dots on it represent stock exchange buying and selling points, respectively. From Figure 5, it can be seen that one could find the right time to buy at the bottoms and the right time to sell at the tops by the trading rules, indicating that BIS algorithm could predict the right time points of stock trading relatively accurately.

6. Conclusions

In this paper, we propose a new advanced time series analysis method to find multiple patterns in the fluctuation of the stock market from the historical data. To overcome the disadvantage of most existing algorithms that rely on the predetermined technical indicators, Biclustering-Based Intelligent System (BIS) could find different patterns which contain a subset of technical indicators with different periodic parameters. The patterns found are regarded as the trading

TABLE 23: Comparison with the IHS on the S&P 500 Index (without transaction costs). “Profits (%)” is excess return. “Trades” is the number of trades.

S&P 500	IHS		BIS	
	Profits (%)	Trades	Profits (%)	Trades
Trending-up market	26.88	6	27.46	78
Flat market	8.34	31.20	11.75	24
Trending-down market	-5.59	4.40	13.63	39

rules, combined with the minimum-error-rate Bayesian decision method under the assumption of multivariate normal probability for short term prediction. In addition, we also make use of the data stream to weaken the influence of the historical data in the model and update the transaction rules dynamically. To validate the performance of the proposed algorithm, we compared it with BAH, GP, and IHS models. Hang Seng index (HSI) data, four Canadian companies stock, and Standard & Poor (S&P) 500 index are employed to carry on the test to find the trading rules. The experimental results show that the profit earned by the proposed method is more than other three models, which proves its effectiveness.

In the future work, more technical indicators will be taken into consideration to make full use of the advantage of the biclustering algorithm. Besides, the transaction costs may greatly influence the profits, so the number of transactions will also be taken into account.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work is supported by Guangdong Economy & Trade Committee under Grant nos. GDEID2010IS034 and the PCSIRT (Grant no. IRT1243).

References

- [1] E. F. Fama and M. E. Blume, “Filter rules and stock-market trading,” *The Journal of Business*, vol. 39, no. 1, pp. 226–241, 1966.
- [2] M. C. Jensen and G. A. Benington, “Random walks and technical theories: some additional evidence,” *The Journal of Finance*, vol. 25, no. 2, pp. 469–482, 1970.
- [3] A. K. Bera and M. L. Higgins, “ARCH models: properties, estimation and testing,” *Journal of Economic Surveys*, vol. 7, no. 4, pp. 305–366, 1993.
- [4] M. Karanasos, “Prediction in ARMA models with GARCH in mean effects,” *Journal of Time Series Analysis*, vol. 22, no. 5, pp. 555–576, 2001.
- [5] F. Allen and R. Karjalainen, “Using genetic algorithms to find technical trading rules,” *Journal of Financial Economics*, vol. 51, no. 2, pp. 245–271, 1999.
- [6] J.-Y. Potvin, P. Soriano, and M. Vallée, “Generating trading rules on the stock markets with genetic programming,” *Computers & Operations Research*, vol. 31, no. 7, pp. 1033–1047, 2004.
- [7] Y. Zhang and L. Wu, “Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network,” *Expert Systems with Applications*, vol. 36, no. 5, pp. 8849–8854, 2009.
- [8] P.-C. Chang and C.-H. Liu, “A TSK type fuzzy rule based system for stock price prediction,” *Expert Systems with Applications*, vol. 34, no. 1, pp. 135–144, 2008.
- [9] A. C. Briza and P. C. Naval Jr., “Stock trading system based on the multi-objective particle swarm optimization of technical indicators on end-of-day market data,” *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 1191–1201, 2011.
- [10] T. Chavarnakul and D. Enke, “A hybrid stock trading system for intelligent technical analysis-based equivolume charting,” *Neurocomputing*, vol. 72, no. 16–18, pp. 3517–3528, 2009.
- [11] W. Leigh, N. Modani, R. Purvis, and T. Roberts, “Stock market trading rule discovery using technical charting heuristics,” *Expert Systems with Applications*, vol. 23, no. 2, pp. 155–159, 2002.
- [12] P.-C. Chang and C.-Y. Fan, “A hybrid system integrating a wavelet and TSK fuzzy rules for stock price forecasting,” *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 38, no. 6, pp. 802–815, 2008.
- [13] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley-Interscience, New York, NY, USA, 2005.
- [14] P. Törönen, M. Kolehmainen, G. Wong, and E. Castrén, “Analysis of gene expression data using self-organizing maps,” *FEBS Letters*, vol. 451, no. 2, pp. 142–146, 1999.
- [15] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.
- [16] Y. Cheng and G. M. Church, “Biclustering of expression data,” *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB ’00)*, vol. 8, pp. 93–103, 2000.
- [17] J. Yang, W. Wang, H. Wang, and P. S. Yu, “ δ -clusters: capturing subspace correlation in a large data set,” in *Proceedings of the 18th International Conference on Data Engineering*, pp. 517–528, San Jose, Calif, USA, February–March 2002.
- [18] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini, “Discovering local structure in gene expression data: the order-preserving submatrix problem,” *Journal of Computational Biology*, vol. 10, no. 3–4, pp. 373–384, 2003.
- [19] L. Lazzeroni and O. Art, “Plaid models for gene expression data,” *Statistica Sinica*, vol. 12, no. 1, pp. 61–86, 2002.
- [20] S. C. Madeira and A. L. Oliveira, “Biclustering algorithms for biological data analysis: a survey,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 1, no. 1, pp. 24–45, 2004.
- [21] Q. Huang, “A biclustering technique for mining trading rules in stock markets,” in *Applied Informatics and Communication, Communications in Computer and Information Science*, pp. 16–24, Springer, Berlin, Germany, 2011.

- [22] Yahoo, "Yahoo Finance," 2014, <http://finance.yahoo.com/>.
- [23] Q. Huang, T. Wang, D. Tao, and X. Li, "Biclustering learning of trading rules," *IEEE Transactions on Cybernetics*, 2014.