Biclustering Learning of Trading Rules

Qinghua Huang, Ting Wang, Dacheng Tao, *Senior Member, IEEE,*
and Xuelong Li, *Fellow, IEEE*

***Abstract*—Technical analysis with numerous indicators and patterns has been regarded as important evidence for making trading decisions in financial markets. However, it is extremely difficult for investors to find useful trading rules based on numerous technical indicators. This paper innovatively proposes the use of biclustering mining to discover effective technical trading patterns that contain a combination of indicators from historical financial data series. This is the first attempt to use biclustering algorithm on trading data. The mined patterns are regarded as trading rules and can be classified as three trading actions (i.e., the buy, the sell, and no-action signals) with respect to the maximum support. A modified *K* nearest neighborhood (*K*-NN) method is applied to classification of trading days in the testing period. The proposed method [called biclustering algorithm and the *K* nearest neighbor (BIC-*K*-NN)] was implemented on four historical datasets and the average performance was compared with the conventional buy-and-hold strategy and three previously reported intelligent trading systems. Experimental results demonstrate that the proposed trading system outperforms its counterparts and will be useful for investment in various financial markets.**

***Index Terms*—Biclustering, machine learning, technical analysis, trading rules.**

I. Introduction

TRADING on the stock market has been a popular investment channel due to its potential for excess profits. It is a nonlinear dynamic system that is influenced by a lot of factors such as national policies, the economic environment, supply-demand relationships, etc. However, it is not easy for investors to make a correct trading decision at the right time, because the stock market is a highly complicated and dynamic system [1], [2]. Therefore, it is of great benefit to analyze the price movement and provide trading rules to guide investors in making trading decisions.

Technical analysis aims at devising appropriate trading rules in the stock market. It studies the historical data, primarily price and volume, to forecast the direction of prices and make trading decisions based on the predictions [3]. Voluminous literature exists on technical analysis in various financial domains. Results obtained in the 1960s and 1970s supported the efficient market hypothesis [4], [5], one of the most widely accepted theories in economics, which states that the financial markets are efficient and the current market price fully reflects all the available information. There is no opportunity to consistently gain profit over the average in the market, and it is pointless to predict future market price behavior, whether by fundamental analysis or technical analysis. However, a number of studies since the 1980s have implied that historical data can help to predict future prices [6]–[10].

In recent years, much research into the stock market using data mining and computational intelligence techniques have shown positive results. Frequently used techniques include genetic programming (GP), artificial neural networks (NN) and template matching. Liu *et al.* [11] presents a decision-making model described by a recurrent NN for dynamic portfolio optimization. The results substantiated the effectiveness and illustrated the characteristics of the proposed NN. Potvin *et al*. [12] used GP to generate short-term trading rules on stock markets. Instead of using a composite stock index, trading rules were adjusted to individual stocks from the Toronto Stock Exchange Market. Lin *et al*. [13] enhanced conventional technical analysis with genetic algorithms (GA) by learning trading rules from history for individual stock, and then combined different rules together with Echo state network to provide trading suggestions. Chen *et al*. [14] used genetic network programming with Sarsa learning to create a stock trading model. Kwon and Moon [15] proposed a hybrid system combining the recurrent NN and a GA for stock trading. Chavarnakul and Enke [16] explored the profitability of stock trading by using a NN model developed to assist the trading decisions of the volume adjusted moving average (VAMA) and the ease of movement. Chien and Chen [17] proposed a GA-based algorithm used to build an associative classifier that discovers trading rules from numerical technical indicators. Wang and Chan [18] examined the potential profit of bull flag technical trading rules using a template matching technique based on pattern recognition for the Nasdaq composite index (NASDAQ) and Taiwan weighted index (TWI).

Manuscript received August 31, 2013; revised June 29, 2014 and October 13, 2014; accepted October 29, 2014. Date of publication December 4, 2014; date of current version September 14, 2015. This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2012CB316400, in part by the National Natural Science Foundation of China under Grant 61125106 and Grant 61372007, in part by the Australian Research Council Linkage Project under Grant 140100569, in part by the Projects of Innovative Science and Technology, Department of Education, Guangdong Province under Grant 2013KJCX0012, and in part by the Key Research Program of the Chinese Academy of Sciences under Grant KGZD-EW-T03. This paper was recommended by Associate Editor P. Bhattacharya.

Q. Huang and T. Wang are with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: qhhuang@scut.edu.cn; helloating@gmail.com).

D. Tao is with the Centre for Quantum Computation and Information Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney, Ultimo, NSW 2007, Australia (e-mail: dacheng.tao@uts.edu.au).

X. Li is with the Center for Optical Imagery Analysis and Learning, State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China (e-mail: xuelong_li@opt.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2014.2370063

The results indicated that the trading rules correctly predicted the direction of change in the NASDAQ and TWI. Leigh *et al*. [8] implemented a recognizer for two variations of the bull flag technical charting heuristic (TCH) and used the recognizer to discover trading rules on the New York Stock Exchange (NYSE) composite index. Out-of-sample results indicated that the rules were effective.

Basically, all these methods belong to technical analysis and all the research showed that trading success can be achieved with these methods. However, drawbacks exist for all these methods. The GP is designed for global optima, which cannot guarantee to find the best solution, and it is difficult to avoid the problem by making appropriate design choices. The NN are nonlinear and nonparametric models. Although the NN is dominant in stock prediction, the models have limitations due to noise and complex dimensionality of data, resulting in unconvincing results. Therefore, to deal with the high dimension of the input data matrix, feature selection and feature learning are mainly two ways for dimensionality reduction [19], and matrix factorization approaches such as singular value decomposition and nonnegative matrix factorization has recently been used for representation of the original data matrix [20]. In addition, the problem of overfitting in the NN cannot always be easily avoided. Template matching is used for the fitting, but the fitting process may be unstable, which may cause the rising flag template fit to be lower than that of the declining flag and lead to missing the trading opportunities.

It has been empirically and theoretically demonstrated that learning technical indicators jointly always gains better performance than learning each indicator independently [16], [21]. To this end, we propose a novel method incorporating a biclustering algorithm and the K nearest neighbor (BIC-K-NN) to discover significant technical trading patterns and to determine the trading actions. This is the first attempt to use biclustering algorithm on financial data series. The biclustering algorithm is a type of data mining method that can extract coherent patterns, each of which contains a subset of indicators with different time lengths. Thus, a combination of different technical indicators in a coherent pattern can be found using the biclustering method if they behave in a similar manner for the same sort of turning points. The detected patterns are regarded as trading signals, including the buy, sell, and no-action signals, and are eventually used in deciding trading actions in financial markets. Furthermore, the proposed method is a framework that can be extended to add more technical indicators, different time spans and trading terms, which makes it a general model that very flexible for investors in various financial markets.

The remainder of this paper is organized as follows. Section II provides a brief introduction of biclustering algorithms. Section III describes the proposed algorithm. Section IV shows the experimental results. Section V provides discussions on the proposed method and concludes this paper.

II. Biclustering Algorithms

Clustering, as a fundamental task of many machine learning, data mining and pattern recognition problems [22], identifies objects with the same attributes or the same functions. Given a data matrix, a traditional clustering algorithm classifies the data along either the row or the column separately. However, it is difficult to extract local coherent patterns that include subsets of rows and subsets of columns. Often, clustering objects under a subset of attributes would be more beneficial for mining important information. For instance, a trader may heavily rely on a small number of technical indicators that provide the most useful predictions to the market to make trading decisions. As a result, how to find these useful indicators (i.e., a subset from all indicators considered) that have similar behaviors in predicting turning points (i.e., a subset of trading days) is an interesting problem in practice. To solve this problem, we need to partition the data in both the row and column directions. This approach has been termed as biclustering. Biclustering is actually a special branch of clustering algorithms because it clusters the data along the row and the column simultaneously in a 2-D data matrix [23].

The term biclustering was first used by Cheng and Church [24] in gene expression data analysis. Names such as coclustering, bidimensional clustering, and subspace clustering are often used in the literature to refer to the same problem formulation [25]. In [24], an approach known as mean squared residue score (MSRS) was introduced to assess the coherence of the elements of a bicluster. A bicluster is a sub-matrix with a coherent pattern in a data matrix. Given a $N_r \times N_c$ data matrix, a bicluster can be defined as a subset of rows and a subset of columns of X. It can be expressed as a pair (R, C), where $R \subseteq \{1, \ldots, N_r\}$ is a subset of rows and $C \subseteq \{1, \ldots, N_c\}$ is a subset of columns in the dataset. The goal of biclustering is to identify a subset of rows with certain coherence properties under a subset of the columns. The MSRS is formulated as follows:

$$\begin{aligned} H(R, C) &= \frac{1}{|R|\,|C|} \sum_{i \in R, j \in C} (a_{ij} - a_{iC} - a_{Rj} + a_{RC})^2 \\ \mathrm{a}_{iC} &= \frac{1}{|C|} \sum_{j \in C} a_{ij} \\ \mathrm{a}_{Rj} &= \frac{1}{|R|} \sum_{i \in R} a_{ij} \\ \mathrm{a}_{RC} &= \frac{1}{|R|\,|C|} \sum_{i \in R, j \in C} a_{ij} \end{aligned} \tag{1}$$

where a_{ij} denotes the element value at the ith row and jth column in the bicluster, $H(R, C)$ the value of MSRS for the bicluster, and δ is a homogeneity threshold defining the maximum allowable dissimilarity within the elements of the bicluster. A submatrix is called a δ bicluster if $H(R, C) \leq \delta$ for some $\delta \geq 0$. The homogeneity threshold is set by users.

According to a survey provided by Madeira and Oliveira [25], biclusters can be grouped into four types: 1) biclusters with constant values; 2) constant values in rows or columns; 3) coherent values; and 4) coherent evolutions. Fig. 1 illustrates ten typical examples for these four types of biclusters. The first three are represented by the numeric values in the data matrix, as shown in Fig. 1(a)–(e), which represent biclusters with constant values,

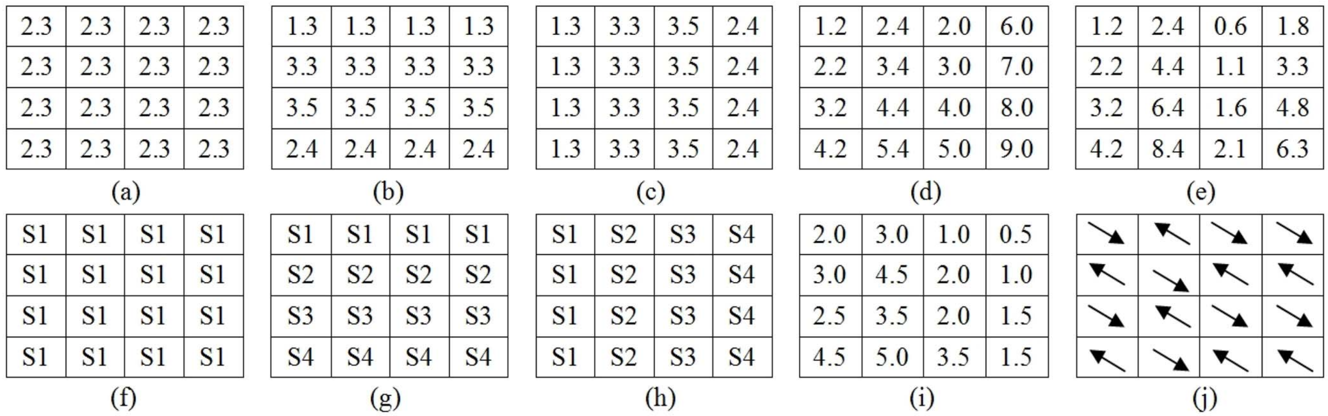


Fig. 1. Examples of different types of biclusters. (a) Constant values in (b) rows and (c) columns. (d) Additive and (e) multiplicative coherent values. (f) Overall coherent evolution in (g) rows and (h) columns. (i) Coherent evolution values in columns. (j) Coherent sign changes in rows and columns.

constant values in rows, constant values in columns, and additive coherent values and multiplicative coherent values, respectively. Fig. 1(f)–(j) represents biclusters with overall coherent evolution, coherent evolution in rows, coherent evolution in columns, coherent evolution values in columns, and coherent sign changes in rows and columns, respectively. A large number of successful applications exist for biclustering in many domains, such as biological data analysis, target marketing, feature selection, and other data mining applications [26]–[29].

III. Methods

Technical analysis tends to forecast future price movement using certain rules based on the study of statistical data that has been generated through market transactions. These trading rules are based on charts or technical indicators that are mathematical formulas that use historical data as input and then return an output signal, which can help investors to make decisions in the markets. As mentioned in Section II, a bicluster defined by a subset of rows (objects) and a subset of columns (attributes) indicates a submatrix, which can be treated as a local coherent pattern (i.e., a subset from all indicators considered that have similar behaviors under a subset of trading days). The detected coherent patterns, including a number of typical turning modes, are identified as trading rules. Therefore, we make use of the biclusters found in a data matrix to generate trading rules in this section.

For a stock or a financial composite index, historical data series are used to form a data matrix, in which the rows correspond to the trading days and the columns correspond to the technical indicators and future return. Observing that local information and manifold information are both crucial for data clustering, and we are inspired from the method proposed by Huang [28] and Yang *et al*. [30], [31]. In this paper, for mining trading rules, we propose an effective biclustering algorithm that converts the problem of searching for biclusters in the data matrix into two procedures: 1) an agglomerative hierarchical clustering (HC) algorithm of rows for each column in the data matrix and 2) a search for the biclusters associated with the clustered rows found in the first procedure. A large number of biclusters may exist after the two procedures, and from these, two factors (i.e., the future returns and the evaluation function) are thereafter taken into consideration. The detected biclusters are then transformed into trading signals with the K nearest neighborhood (K-NN) method. Though the idea of using biclustering algorithm to cluster data objects on a selected subset of features seems to be simple, it detects local coherent patterns from the data matrix well. There are numerous models such as intraday, daily, weekly, and monthly trading for automatic stock trading, depending on behavioral scope. This paper considers only the daily trading model.

TABLE I
CALCULATION PERIODS OF THE TECHNICAL INDICATORS

Indicators	Period1	Period2	Period3	Period4	Period5	Period6
ROC	12	24	36	\	\	\
EMA	12	24	36	\	\	\
ADX	14	28	42	\	\	\
ATR	14	28	42	\	\	\
SMA	10	20	30	40	\	\
%R	9	18	27	36	\	\
RSI	6	12	18	24	30	36

A. Data Preparation

For a stock or a financial composite index, a data matrix is constructed using historical data where the rows correspond to trading days and the columns correspond to 26 technical indicators with different periodic parameters and future returns. Seven popular technical indicators are chosen for this paper to form the data matrix. They are the rate of change (ROC), the exponential moving average (EMA), the average directional index (ADX), the average true range (ATR), simple moving average (SMA), the Williams percentage range (%R), and relative strength index (RSI). With different time lengths associated with each indicator, there are a total of 26 features adopted as the first section of the columns. Table I shows the technical indicators with the different periodic parameters adopted in this paper. Furthermore, future returns are placed in the data matrix as the second section of the columns. They can be obtained by averaging the returns of the following trading days over a period of time. The number of following trading days is determined by users according to their respective applications.

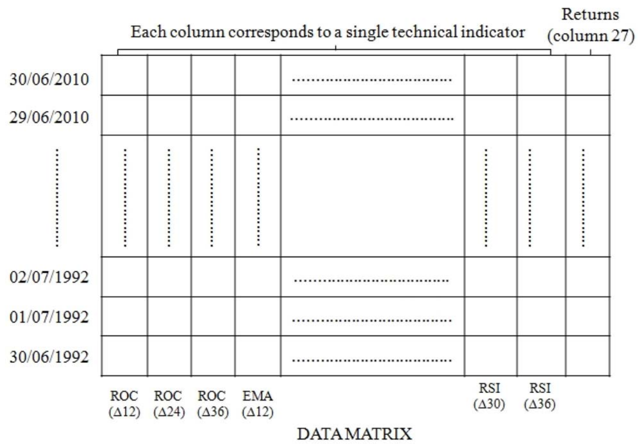


Fig. 2. Data arrangement for the matrix. In the data matrix, each row corresponds to a trading day and each column corresponds to a technical indicator with a specified time length. Column 27 is the sequence of future returns, each of which corresponds to a trading day.

Fig. 2 illustrates the structure of the data matrix. The elements in the data matrix are assigned the values of the indicators (columns 1–26) and the future returns (column 27) corresponding to the trading days. All the technical indicators of a stock or a composite index are computed using the historical data, including the closing prices and transaction volumes. Therefore, a data matrix with a width (column) of 27 and a height (row) of the same as the number of trading days in the training period can be constructed. Please note that the formation of the data matrix in this paper can be regarded as an example and readers can use other technical indicators to form different data matrices for their applications.

It is commonly considered that the price movement can be described by the percentage changes of the price in the sequence of trading days. We replace the original value of elements with the percentage of difference between the original values and the closing prices of corresponding trading days on the columns of the SMA and EMA. The procedure is described by

$$PD\_MA(i,j) = \frac{MA(i,j) - \mathrm{CP}(i)}{\mathrm{CP}(i)} \times 100\% \tag{2}$$

where the $\mathrm{CP}(i)$ denotes the closing price at the ith trading day, $MA(i,j)$ denotes the SMA or the EMA with a periodic parameter j at the ith trading day, and $PD\_MA(i,j)$ denotes the percentage of difference between $MA(i,j)$ and $\mathrm{CP}(i)$.

Since the financial market is time variable and its price variation is a very dynamic system, this environment motivates a need for reassessment of the price movements. Therefore, in column 27, before computing the future return, the closing price corresponding to each trading day is firstly replaced by the mean closing price of the following n trading days (denoted by CP_ave_i) to reflect the price movement trends more convincingly as below:

$$\mathrm{CP}\_ave_i = \frac{1}{n}\sum_{m=i}^{i+n-1} \mathrm{CP}_m \tag{3}$$

where CP_i is the closing price at the ith trading day. In this paper, we consider short-term investment and set the parameter n to be 10. Then the future return in column 27 corresponding to each trading day is computed as

$$FRV_i = \frac{\mathrm{CP}\_ave_i - \mathrm{CP}_i}{\mathrm{CP}_i} \times 100\% \tag{4}$$

where FRV_i denotes the future return in percentage corresponding to the ith trading day, which represents the moving direction of price in a short period.

In order to identify the trends more explicitly in column 27, we use 1 to denote an upward trend, -1 for a downward trend, and 0 movement for a sideways trend in the n following days. If the value of $FRV_{(i,n)}$ is larger than a threshold T_t, it is set to 1; if smaller than $-T_t$, it is set to -1. Otherwise, it is set to 0. In this paper, T_t is 0.5%.

Because the values of different technical indicators vary greatly, we apply a normalization procedure to each column of the technical indicators to map the data into a range of [0, 1]. This procedure is described by

$$V_n(i,j) = \frac{V_o(i,j) - V_{\min}(j)}{V_{\max}(j) - V_{\min}(j)} \tag{5}$$

where the $V_o(i,j)$ denotes the original value of the jth technical indicator at the ith trading day in the data matrix, the $V_{\min}(j)$ denotes lower bound of the jth technical indicator, the $V_{\max}(j)$ denotes the upper bound of the jth technical indicator, and the $V_n(i,j)$ denotes the normalized value of the jth technical indicator at the ith trading day.

B. Discovery of Biclusters

Having formed the data matrix, we search for the biclusters that are a result of the trading rules. Each bicluster has constant columns under a subset of indicators, as shown in Fig. 1(c). On this occasion, different trading rules can be represented by a different subset of technical indicators. This means that the combination of these indicators occur multiple times during the trading days in the training set. The pattern indicated by these indicators has a high possibility to occur again, and may be a useful trading signal. Consequently, using an algorithm to find biclusters from historical financial data is necessary. Meanwhile, it is obviously noted that each indicator should take the same, or similar values for different rows in a discovered trading pattern. Therefore, we propose an algorithm to find the biclusters with constant columns [Fig. 1(c)] from the data matrix.

The first step is to find similar elements in a single column. An agglomerative HC algorithm is initially performed on each column of the data matrix. The average linkage is used and a distance threshold T_{hc} is predetermined to measure the similarity of elements to each of the columns of the data matrix. The detected clusters from all columns are treated as bicluster seeds (BS) and placed in an *HC_Set*. Any of the BS can be expressed as a pair (R, C_j), where $R \subseteq \{1, \ldots, N_r\}$ is a subset of rows and C_j is one of the columns.

The next step is to expand each BS along the column direction. For a BS from *HC_Set*, we process each column in N_c, other than C_j, one by one. A new column can be added to

```
The Biclustering Algorithm
Input    a data matrix, X
Output a set of bicluster (Bic_Set)
Let      HC_Set={} and Bic_Set ={}.
For      each column (l) in X.
         Perform a hierarchical clustering algorithm with a distance threshold Thc on l.
         We obtain a set of clusters in the single column l, denoted as l_Set.
         Add l_Set into HC_Set.
EndFor
For      each cluster (C) in HC_Set
         C' = C.
         For each column l
                Add l to C', we obtain C'', Calculate the MSRS for C'', denoted as lsd.
                If   lsd < Tsd, Then C' = C'', EndIf
         EndFor
         Add C' into Bic_Set.
EndFor
Filter out the duplicate biclusters.
Output Bic_Set
```

Fig. 3. Biclustering algorithm.

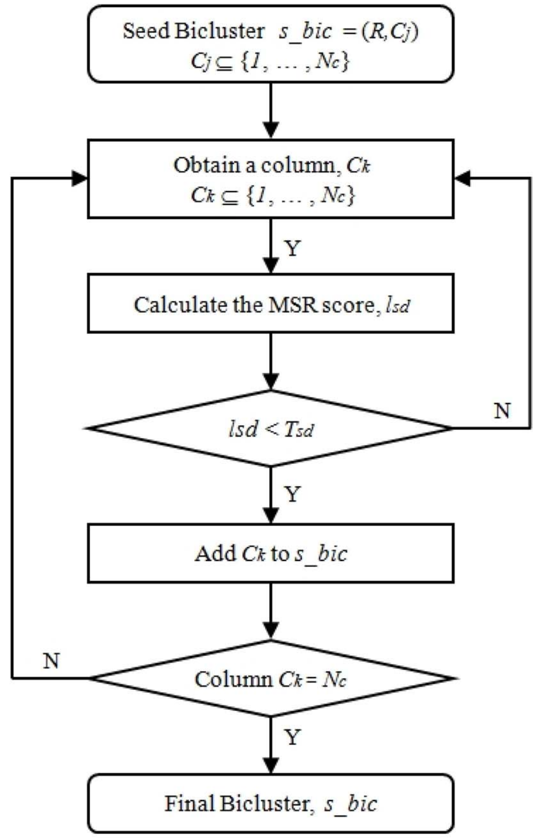


Fig. 4. Flow chart of expanding a BS.

BS if the newly formed sub-matrix satisfies the criterion of MSRS (i.e., the MSRS score is less than a preset threshold T_{sd}). This procedure is repeated until no new column can be added to BS, and then the new BS with expansion in columns can be regarded as a bicluster that is placed in *Bic_Set*. It is worth noting that the bicluster may contain only one column after the expansion, and this is also deemed as a real bicluster. After this procedure is performed for all BS, the *Bic_Set* contains the final output of biclusters, any of which is expressed as (R, C), $R \subseteq \{1, \ldots, N_r\}$ and $C \subseteq \{1, \ldots, N_c\}$. The flow chart of the expanding procedure is shown in Fig. 4.

Note that the search of the biclusters is applied to the sub-matrix, including columns 1–26, without considering the column of future return. The type of a bicluster is then determined by the future returns in column 27. A valid bicluster should contain at least five rows in order to be used as a robust trading rule. Furthermore, when two seed biclusters exist, constructed by the same subset of rows but different columns, (R, C_x) and (R, C_y), the expanding procedure may produce the same bicluster. In this case, duplicate biclusters should be filtered out, keeping only one. The biclustering algorithm is described in Fig. 3.

C. Classification of the Biclusters

A bicluster is translated into a trading rule by averaging each column. Because each column of a bicluster corresponds to a specific technical indicator, the output trading rule is a vector where the value of an element is the mean of the corresponding indicators values over the trading days included by the bicluster. An effective trading rule can help the trader to make a trading decision by predicting a monotone moving direction of the financial price.

Three types of trading actions occur in the stock market, i.e., buy, sell, and no action, so all of the trading rules are then classified into three trading sets: 1) buy signal set (S_b); 2) sell signal set (S_s); and 3) no-action set (S_n). This paper defines a support-based metric as a measure to help make the classification, which is expressed as

$$\begin{aligned} \text{supp}(S_b) &= \frac{\text{No. of rows with } FRV \geq T_t}{\text{All rows}} \\ \text{supp}(S_s) &= \frac{\text{No. of rows with } FRV \leq T_t}{\text{All rows}} \\ \text{supp}(S_n) &= \frac{\text{No. of remaining rows}}{\text{All rows}} \end{aligned} \tag{6}$$

where FRV denotes the future return according to (4), $\text{supp}(S_b)$, $\text{supp}(S_s)$, $\text{supp}(S_n)$ denote the supports of the buy signal, the sell signal and the no-action, respectively.

For a bicluster, three support values $[\text{supp}(S_b), \text{supp}(S_s), \text{supp}(S_n)]$ can be calculated according to (6). To make a trading decision, we have to know the maximum of the supports, as follows:

$$T(\bullet) = \max(\text{supp}(S_b), \text{supp}(S_s), \text{supp}(S_n)) \tag{7}$$

where T denotes the maximum value among $\text{supp}(S_b)$, $\text{supp}(S_s)$ and $\text{supp}(S_n)$, indicating that the type of the bicluster is determined by the signal with the largest support.

To obtain more reliable trading signals, we select the biclusters whose T is larger than a predefined threshold T_{pre} as a valid trading rule. Although larger T_{pre} results in more reliable trading signals, T_{pre} is empirically set to 0.7 in this paper to avoid inadequate valid trading rules found. For instance, given

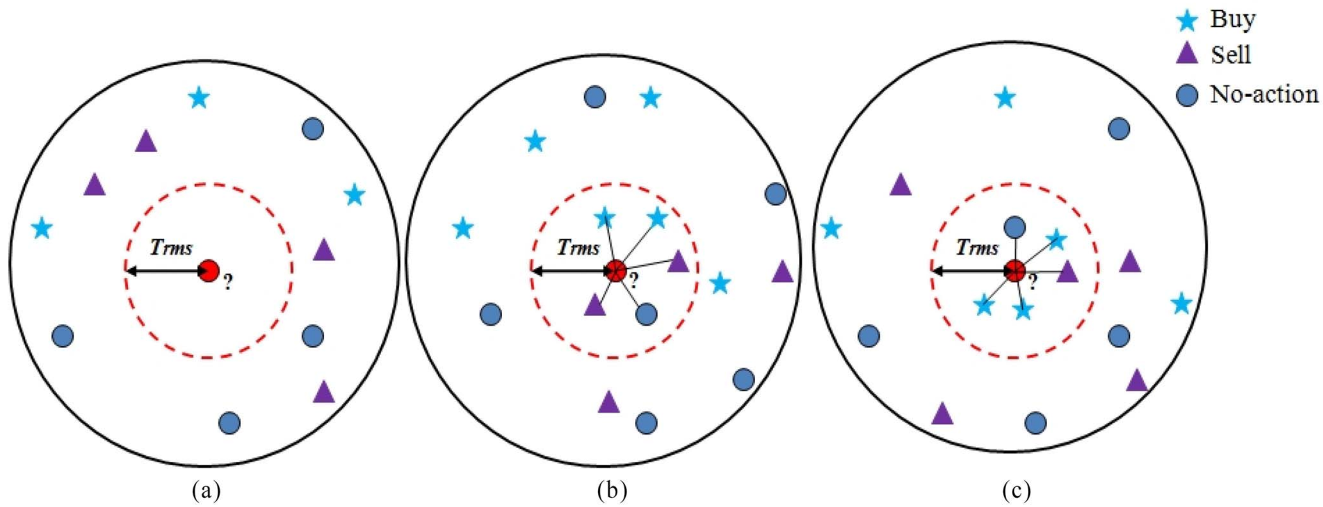


Fig. 5. Illustration on how to use the five-NN classifier to determine the trading actions in the testing period. (a) There is no trading rule falling into the neighborhood. (b) In the neighborhood, the number of buy signals is the same as that of sell signals, hence the trading action is no-action. (c) Number of buy signals is the largest, hence the trading action is buy.

TABLE II
TWO TRADING STRATEGIES ADOPTED IN THIS PAPER

Strategy	Descriptions
1	If there is no position, open a long position when the indicators of current trading day conforms to a buy signal. Thereafter, close it when the indicators of a subsequent trading day conforms to a sell signal. Repeat this trading way during the testing period.
2	If there is no position, open a position when the indicators of the current trading day conforms to either a buy or a sell signal. If a position exists, close it when meeting a reversal trading signal and open a reversal position at the same trading day. Repeat this trading way during the testing period.

that $\mathrm{supp}(S_b) = 0.82$, $\mathrm{supp}(S_s) = 0.13$, and $\mathrm{supp}(S_n) = 0.05$, respectively, the bicluster is classified as a buy signal and put into the buy signal set S_b, because $T = \max(\mathrm{supp}(S_b), \mathrm{supp}(S_s), \mathrm{supp}(S_n)) = \mathrm{supp}(S_b) = 0.82$ is greater than T_{pre}. Therefore, this bicluster is transformed into a trading rule regarded as a buy signal.

D. Determination of Trading Actions Using K-NN Method

Two trading strategies [32] (called strategies 1 and 2) are employed in this paper—described below in Table II. Transaction fee is not considered in this paper.

In the testing period, the trading actions are determined based on the matching of trading rules and trading days. For each trading day, the values of all the technical indicators are calculated. Since a trading rule is a vector of technical indicators with specific values, a trading day is considered to match a trading rule when the values of the corresponding technical indicators of the trading day are very similar to the values of the corresponding technical indicators of a trading rule. In this paper, the root-mean-square (RMS) of the corresponding technical indicators is used to measure the similarity between trading day and trading rule

$$\mathrm{RMS} = \frac{\sqrt{\sum_{\mathrm{k}\in\mathrm{ST}} (V_{td}(k) - V_{tr}(k))^2}}{n_k} \tag{8}$$

where ST denotes the set of technical indicators of a special trading rule, n_k denotes the number of indicators in ST, $V_{td}(k)$ denotes the value of the indicator (corresponding to the kth indicator in ST) for the current trading day, and $V_{tr}(k)$ denotes the value of kth indicator in the trading rule.

Given a predefined threshold T_{rms}, a neighboring region is defined for each trading day in the testing period. Several trading rules may fall into the region (i.e., $\mathrm{RMS} \le T_{\mathrm{rms}}$). In this paper, T_{rms} is set to the mean of all calculated RMS between every trading day and every trading rule in the training period, thus there would be different values of T_{rms} for different training sets. We then use the K-NN method as a classifier [33] to determine the trading signal for the current trading day. We set K to 5 in this paper. In the first case, there is no trading rule falling into the neighborhood, so the trading day provides no trading signal, suggesting no-action on the next trading day. In the second case, there are trading rules falling into the neighborhood of a trading day. We take into account the five nearest rules. Based on a voting strategy, the trading day is classified as a specific pattern (i.e., buy, sell, or no-action) with the largest votes. If no pattern takes only the largest votes, the trading signal is no-action for the next trading day. Fig. 5 provides a simple example illustrating the determination of trading actions in the testing period with a five-NN classifier. Once the trading signal is determined for a trading day in the testing period, corresponding action is made on the next trading day with the opening price. Thus, the proposed method is called BIC-K-NN in this paper. Fig. 6 gives a running example demonstrating how a bicluster can be used as a trading rule in the training period and how to determine the trading action for a trading day based on the matching value (RMS) and the 5-NN classifier in the testing period.

E. Experimental Methods

To assess the performance of the BIC-K-NN, we conduct experiments to make comparisons with four methods: 1) GP [12]; 2) TCH [8]; 3) buy-and-hold (BAH); and

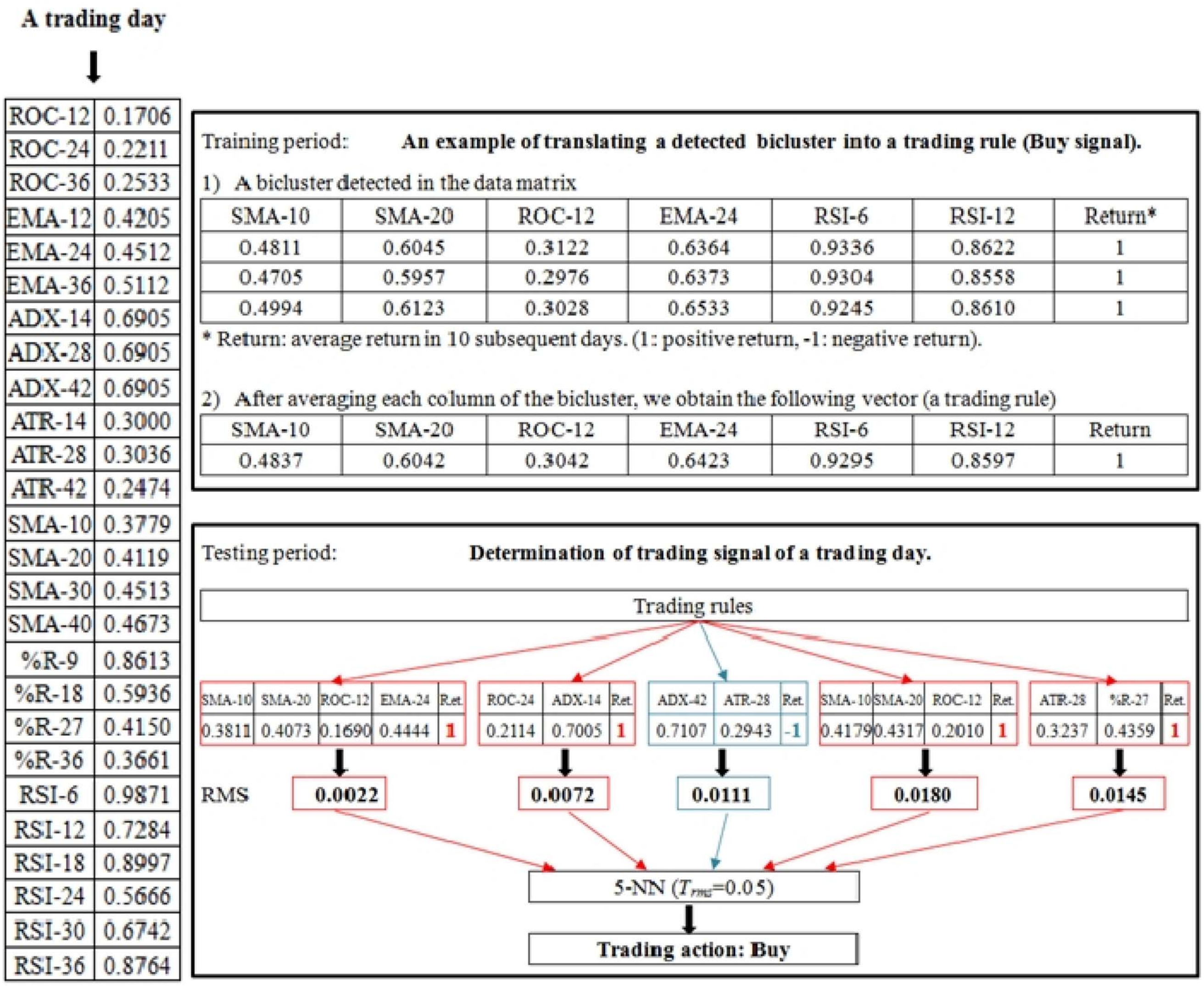


Fig. 6. Example for demonstrating how a detected bicluster is translated into a trading rule in the training period, and how to determine the trading action for a trading day based on the RMS and the five-NN classifier in the testing period.

4) intelligent hybrid system (IHS) [34]. The GP extends classical GA by allowing the processing of nonlinear structures. The TCH compares historical stock price and volume to archetypal chart patterns and predicts future price behavior based on the degree of match. The BAH implies to buy at the start of the period and sell at its end. The IHS integrates NN, fuzzy logic, and GA techniques into the VAMA technical indicator for stock trading.

Four different real-world datasets are used for evaluation of the methods. In particular, three datasets were used by the control methods [8], [12], [34], respectively. Dataset 1 including four Canadian companies stocks was used in [12] (see Table III). Dataset 2 including six major world indices (see Table IV) from 07/02/1997 to 11/13/2012 (downloaded from Yahoo Finance [35]) are used for comparisons between the BAH and BIC-K-NN. Dataset 3 is the NYSE composite index covering the period between 08/06/1980 and 06/08/1999 which was used in [8]. The division of the data into the training and testing periods is the same as [8] in order to make an exact comparison. Dataset 4 is the past Standard & Poor's (S&P) 500 index that was used in [34]. The training and testing periods in this paper are also the same as [34] and the exact time periods used for the experiments are listed in Table V. All historical data consists of the opening price, the highest price, the lowest price, the closing price, and the transaction volume for each trading day. In the BIC-K-NN, the training period is used to construct the data matrices, discover biclusters, and generate trading rules. With the discovered trading rules, the trading actions can be performed on the testing period to evaluate the profiting performance.

As mentioned above, the evaluation of the detected biclusters are determined by two parameters (i.e., T_{hc} and T_{sd}) that should be carefully assigned as they have important influence on the final profits. To find appropriate values for the parameters, preliminary experiments are performed. Considering the time complexity, we have implemented ten independent runs for each dataset and tabulated the profits as well as the standard deviations (SD). For each run, T_{hc} and T_{sd} are tuned by a random combination of a range from 0.005 to 0.25 and we therefore can find the best results for each dataset.

TABLE III
DATASET 1: 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 IV
DATASET 2: SIX MAJOR WORLD INDICES IN ASIA

Market	Indices	Symbol	Period
Bombay	BSE SENSEX	BSESN	Training
Kuala Lumpur	FTSE Bursa Malaysia KLCI	KLSE	07/02/1997 - 03/03/2008
Korea	KOSPI Composite Index	KS11	Testing
Osaka	NIKKEI 225	N225	03/04/2008 - 11/13/2012
Singapore	STRAITS TIMES INDEX	STI	
Taiwan	TSEC weighted index	TWII	

TABLE V
TRAINING AND TESTING PERIODS OF DATASET 4: 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 - 1230/1999	01/01/2000 - 12/30/2000
Trending-down market	01/01/1997 - 12/30/2001	01/01/2002 - 12/30/2002

TABLE VI
COMPARISON BETWEEN THE GP AND BIC-K-NN ON FOUR CANADIAN STOCKS. "PROFITS (%)" IS THE AVERAGE EXCESS RETURN OF TEN RUNS. "TRADES" IS THE AVERAGE NUMBER OF TRADES

Stock	GP	BIC-KNN					
		Strategy 1			Strategy 2		
	Profits (%)	Profits (%)	SD	Trades	Profits (%)	SD	Trades
ABX	31.76	70.59	34.05	45	68.64	15.12	23
CP	17.36	46.32	37.56	11	34.50	10.29	23
CXY	29.23	55.49	26.81	38	32.37	8.97	20
TRP	14.62	−14.05	28.48	47	33.87	9.12	26
Average	**23.24**	**39.50**	31.73	35.25	**42.35**	10.88	23

TABLE VII
COMPARISON BETWEEN THE BAH AND BIC-K-NN ON SIX MAJOR WORLD INDICES IN ASIA. PROFITS (%) IS THE AVERAGE EXCESS RETURN OF TEN RUNS. TRADES IS THE AVERAGE NUMBER OF TRADES

Index	Buy-and-hold	BIC-KNN					
		Strategy 1			Strategy 2		
	Profits (%)	Profits (%)	SD	Trades	Profits (%)	SD	Trades
BSESN	33.70	150.06	62.02	117	28.50	3.34	24
KLSE	46.06	65.78	36.12	167	16.24	2.77	22
KS 11	20.07	126.99	28.49	132	38.06	4.75	22
N225	−32.09	89.45	41.89	179	30.92	11.40	19
STI	5.14	93.11	53.48	40	19.09	6.11	19
TWII	1.59	124.46	45.94	161	20.48	6.20	25
Average	**12.41**	**108.31**	44.66	132.67	**25.55**	5.76	21.83

IV. EXPERIMENTAL RESULTS

The algorithm is programmed using MATLAB (Mathworks, Company, USA). Experimental tests were conducted on a 3.40 GHz PC with an Intel Core 12 Duo CPU. For the training of trading rules, the execution time varies with the sizes of the dataset. The average execution time for dataset 1 (about 2000 days) was approximately 20 s, and that for dataset 2 (3830 days) required approximately 40 s. For dataset 3 (4748 days), the average execution time is summarized in Table VIII. For dataset 4 (about 1500 days), the computation time in each run was approximately 12 s on average.

Tables VI–IX present the average performance for the testing dataset of the four datasets using the two trading strategies in the experiments. The results include the profitability, the SD of profits (for ten runs) and the number of trades discovered in the testing periods. The BIC-K-NN outperforms the other four trading methods in the testing periods, demonstrating significantly promising usefulness.

Specifically, as shown in Table VI the proposed BIC-K-NN under the two trading strategies achieves better average profits (39.50% for strategy 1 and 42.35% for strategy 2) than the GP (23.24%). Table VII demonstrates that the average

TABLE VIII
COMPARISON BETWEEN THE TCH AND BIC-K-NN ON THE NYSE INDEX. "ALL" STANDS FOR ALL TRADING DAYS OF BAH STRATEGY. "RULE1" AND "RULE2" REPRESENT TWO VARIATIONS OF THE BULL FLAG TCH. PROFITS (%) IS THE AVERAGE EXCESS RETURN OF TEN RUNS. "N" IS THE AVERAGE NUMBER OF TRADES

	TCH						BIC-KNN						Average execution time(s)
	All		Rule 1		Rule 2		Strategy 1			Strategy 2			
	Profits (%)	*N*	Profits (%)	*N*	Profits (%)	*N*	Profits (%)	SD	*N*	Profits (%)	SD	*N*	
1982	19.2	108	19.8	5	13.9	36	15.98	11.99	10	9.39	5.70	2	3.24
1983	2.1	251	0	0	13.59	2	30.81	26.10	21	26.71	14.75	15	4.78
1984	5.1	252	0	0	9	2	12.61	13.11	19	3.60	2.59	3	6.98
1985	9.9	250	20.3	10	6.4	42	15.74	9.24	22	14.61	2.17	13	10.53
1986	9	252	0	0	0	0	32.31	14.85	24	31.38	7.52	13	14.50
1987	−2.6	253	0	0	8.9	43	23.55	12.82	9	37.72	6.45	8	18.01
1988	5.4	253	0	0	0	0	12.03	25.86	4	11.37	11.57	2	21.80
1989	4.9	252	0	0	2.4	12	18.20	13.14	8	21.90	6.09	8	25.58
1990	3.1	253	15.9	25	0	0	6.61	11.30	30	8.38	5.25	11	31.31
1991	5.7	252	21.1	9	1.7	13	27.30	13.64	4	20.70	9.55	4	30.88
1992	3	254	2.5	2	−0.1	12	11.88	7.84	13	7.31	2.92	5	33.30
1993	1.9	253	0	0	0	0	6.38	8.68	11	7.36	5.38	8	36.29
1994	1.8	251	0.2	31	0	0	2.30	6.40	15	3.02	2.58	6	39.80
1995	10.6	251	13.5	22	10.2	33	20.20	8.71	11	17.79	7.34	11	39.71
1996	7.6	253	0	0	2.3	17	20.50	6.64	24	17.80	2.96	12	44.56
1997	12.1	252	0	0	16.3	7	34.89	12.36	17	28.90	4.79	10	46.47
1998	4.8	252	11.6	20	−15	1	11.51	12.87	7	18.91	10.08	8	52.67
1999	1.8	106	0	0	0	0	6.69	6.23	2	6.64	3.36	8	42.92
Average	**5.5**	236	11.5	124	7.8	220	**17.19**	12.32	13.94	**16.30**	6.17	8.17	27.96

TABLE IX
COMPARISON BETWEEN THE IHS AND BIC_KNN ON THE S&P 500 INDEX (WITHOUT TRANSACTION COSTS). "G/L ($)" IS THE AVERAGE GAIN OR LOSS OF THE TRADING. PROFITS (%) IS THE AVERAGE EXCESS RETURN OF TEN RUNS. TRADES IS THE AVERAGE NUMBER OF TRADES

S&P 500	IHS			BIC-KNN							
				Strategy 1				Strategy 2			
	G-L($)	Profits (%)	Trades	G-L($)	Profits (%)	SD	Trades	G-L($)	Profits (%)	SD	Trades
Trending-up	238.67	**26.88**	6	226.67	**25.13**	12.22	43	135.97	**14.13**	6.09	8
Flat market	107.90	**8.34**	31.20	133.92	**9.93**	13.89	3	109.87	**8.43**	6.55	3
Trending-down	−55.28	**-5.59**	4.40	154.13	**16.45**	12.09	38	186.73	**18.48**	8.7	3

profits made by the BIC-K-NN are better than those made by the BAH strategy on all indices. In particular, strategy 1 significantly outperforms the BAH, substantially improving the average profitability by 95.90% over the BAH. In Table VIII, the annual profits from 1982 to 1999 made by the BAH, TCH and the BIC-K-NN, respectively, are listed. Our method has an improved overall average profit, and outperforms the BAH and TCH in most years. In the testing periods, the TCH method gains 5.5% on average. In contrast, our method averagely gains 17.19% with strategy 1 and 16.30% with strategy 2, respectively. In another word, the BIC-K-NN further improves the average profitability by 11.69% with strategy 1 and 10.80% with strategy 2 over the TCH methods, respectively. The average number of trades for the BIC-K-NN (i.e., 13.94 trades for strategy 1 and 8.17 trades for strategy 2) is significant decreased comparing to the TCH (236 trades on average). Obviously, lower number of trades help reduce transaction costs in practices.

In Table IX, although it gives slightly poor results in trending-up market, the BIC-K-NN significantly outperforms the IHS in trending-down market and obtains similar performance in the flat market. It can be concluded that most of the predictions on the future trends are correct in the BIC-K-NN method, hence double positive profits can be achieved. The SDs of trading rules with strategy 1 are higher than those with strategy 2. Thus, the average fluctuation range of profits for strategy 1 is higher than that of strategy 2. Although some trades yield negative returns using our method (e.g., dataset 1: TRP), the trading rules discovered by the proposed BIC-K-NN are capable of making more profits than the other four methods. The results are an encouraging evidence indicating that the BIC-K-NN is capable of finding more profitable trading rules in financial markets.

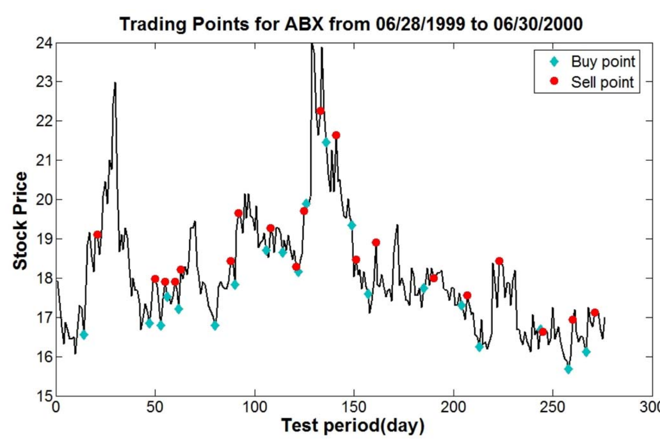


Fig. 7. Trading points for ABX from 06/28/1999 to 06/30/2000.

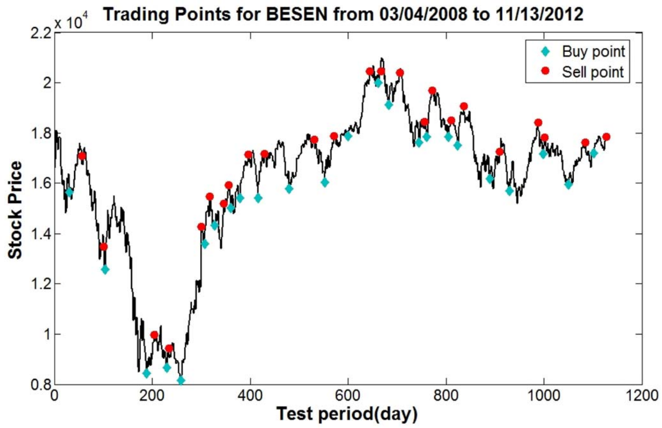


Fig. 8. Trading points for BSESN from 03/04/2008 to 11/13/2012.

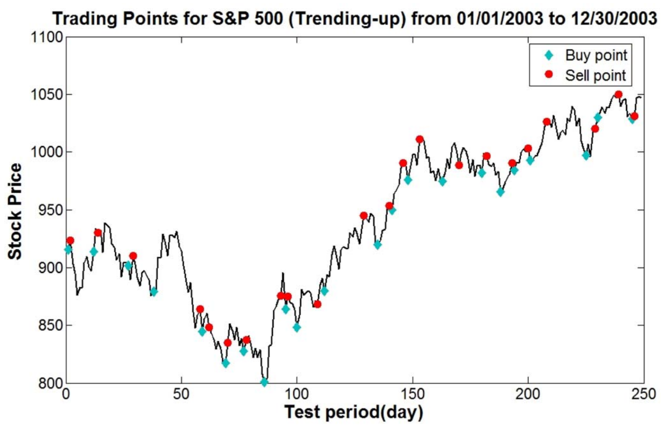


Fig. 9. Trading points for S&P 500 index (trending-up market) from 01/01/2003 to 12/30/2003.

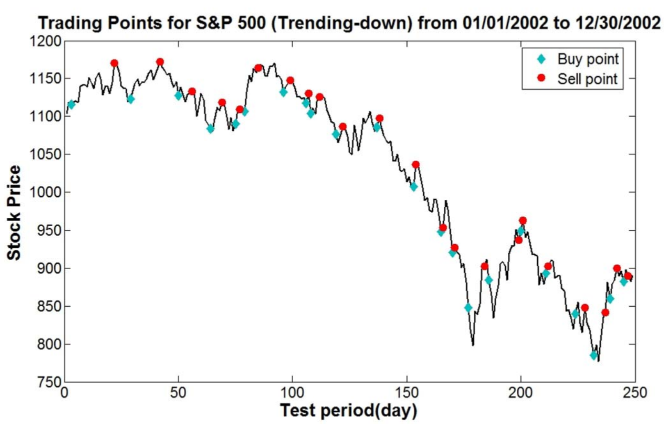


Fig. 10. Trading points for S&P 500 index (trending-down market) from 01/01/2002 to 12/30/2002.

Figs. 7–10 show more understandable examples by marking the buy and sell trading points in the testing periods of four stocks (i.e., ABX, BSESN, and S&P 500 index trending-up and trending-down markets). These trading points are derived from the trading rules discovered in the training periods by the BIC-K-NN. It is observed that the trading rules can successfully detect correct buy points at the bottoms and correct sell points at the tops.

V. Conclusion

In this paper, we propose to use a novel intelligent trading method incorporating BIC-K-NN method to discover trading rules based on a combination of technical indicators in financial markets. To the best of our knowledge, this is the first attempt to apply biclustering algorithm to financial data series for finding trading rules. Seven active technical indicators (ROC, EMA, ADX, ATR, SMA, %R, and RSI) with different time span parameters and future returns are incorporated to form a data matrix. The biclustering algorithm is designed to discover biclusters with constant columns in the data matrix. The detected biclusters are regarded as trading rules and grouped into buy, sell and no-action signals. A five-NN classification method is used to determine the trading action of a trading day in the testing period. The proposed algorithm (BIC-K-NN) is implemented on four datasets and has been compared the average performance with four existing methods. The results demonstrate that our method significantly outperforms its counterparts. Furthermore, the proposed method is a framework that can be extended to add more technical indicators, different time spans and trading terms, which makes it a general and flexible model for investors in various financial markets.

Nevertheless, there are some limitations when using the proposed method. Firstly, we ignored the missing data in the original data matrix, due to the integrity of the data may affect the efficiency of the data analysis and the results of the experiments, it is necessary to estimate missing values to maintain data integrity. Secondly, we selected only a small number of technical indicators to generate the trading rules. As is well known to investors, there are a large variety of technical indicators for investment. Therefore, it is expected that more robust and effective trading rules could be mined using more indicators to generate even better trading systems. Moreover, more selections of the time span for each indicator may be helpful to discover the optimal indicator combinations, which could provide more robust trading rules. Thirdly, several parameters (i.e., T_{hc}, T_{sd}, T_{pre}, and T_{rms}) need to be carefully set when mining the trading rules. T_{rms} is used for determining the distance for the K-NN algorithm and is adaptively set by the mean value. Obviously, smaller T_{rms} will guarantee the similarity between the trading days and the discovered trading rules. T_{hc} and T_{sd} are used for mining biclusters in the training period. They are tuned by a random combination of a range from 0.005 to 0.25 and then the best result for each dataset is reported. T_{pre} is the predefined threshold used for obtaining more reliable trading signals in the testing period. Larger T_{pre} leads to more reliable trading signals. Nevertheless, three parameters have to be empirically set with respect to preliminary trials in this paper. However, in financial practices, this would be inconvenient for investors. A possible solution is to use learning algorithms to find the optimal settings for these parameters. Fourth, we have only considered the periodic cycle length (i.e., ten days), which predicts the future trend in short-term. However, investors may plan to trade in the middle-term or long-term. With different trading cycles, the mined trading rules may be quite different. Finally, the current method for selecting biclusters as trading patterns is relatively simple and may not be adequately reliable.

According to these limitations, future work will extend the research in this paper to a more general model, to develop various trading systems with improvements. First, methods of completion of arbitrary matrices from limited information will be introduced this paper, such as the method proposed by Hu *et al*. [36]. Second, more technical indicators with a larger variety of time spans will be considered for formation of data matrices. Third, intelligent learning algorithms (e.g., multiobjective GP [37] or particle swarm optimization algorithm) will be applied to find the optimal parameters with respect to the metrics of profit ratings. Fourth, different trading cycles will be taken into account to generate more complicated and more profitable trading strategies. Finally, the validations of the detected biclusters will be performed with respect to some measures [38]–[40], and the classification of trading signals can be improved by applying some most advanced learning models, such as the domain adaption learning method [41].

In conclusion, this paper has innovatively proposed to use a BIC-*K*-NN classification method to discover trading rules, and apply them to real financial markets. It is the first attempt to use biclustering algorithm on financial data series. Experimental results demonstrate that the proposed method outperforms four existing trading strategies, indicating its sound performance for trading in financial markets. We expect that the novel idea of converting the biclusters into trading rules will lead to a new research topic in finance theory and applications.

REFERENCES

[1] E. E. Peters *et al.*, *Fractal Market Analysis: Applying Chaos Theory to Investment and Economics*. New York, NY, USA: Wiley, 1994.

[2] V. Milea, F. Frasincar, and U. Kaymak, "TOWL: A temporal web ontology language," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 1, pp. 268–281, Feb. 2012.

[3] J. J. Murphy, *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. London, U.K.: Prentice Hall, 1999.

[4] E. F. Fama and M. E. Blume, "Filter rules and stock-market trading," *J. Bus.*, vol. 39, no. 1, pp. 226–241, 1966.

[5] M. C. Jensen and G. A. Benington, "Random walks and technical theories: Some additional evidence," *J. Financ.*, vol. 25, no. 2, pp. 469–482, 1970.

[6] S. W. Pruitt and R. E. White, "The CRISMA trading system: Who says technical analysis can't beat the market?" *J. Portfolio Manage.*, vol. 14, no. 3, pp. 55–58, 1988.

[7] S. W. Pruitt, K. M. Tse, and R. E. White, "The CRISMA trading system: The next five years," *J. Portfolio Manage.*, vol. 18, no. 3, pp. 22–25, 1992.

[8] W. Leigh, N. Modani, R. Purvis, and T. Roberts, "Stock market trading rule discovery using technical charting heuristics," *Expert Syst. Appl.*, vol. 23, no. 2, pp. 155–159, 2002.

[9] C.-H. Park and S. H. Irwin, "What do we know about the profitability of technical analysis?" *J. Econ. Surveys*, vol. 21, no. 4, pp. 786–826, 2007.

[10] G. Ramazan and S. Thanasis, "Moving average rules, volume and the predictability of security returns with feedforward networks," *J. Forecasting*, vol. 17, nos. 5–6, pp. 401–414, 1998.

[11] Q. Liu, C. Dang, and T. Huang, "A one-layer recurrent neural network for real-time portfolio optimization with probability criterion," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 14–23, Feb. 2013.

[12] J. Y. Potvin, P. Soriano, and M. Vallée, "Generating trading rules on the stock markets with genetic programming," *Comput. Oper. Res.*, vol. 31, no. 7, pp. 1033–1047, 2004.

[13] X. Lin, Z. Yang, and Y. Song, "Intelligent stock trading system based on improved technical analysis and Echo state network," *Expert Syst. Appl.*, vol. 38, no. 9, pp. 11347–11354, 2011.

[14] Y. Chen, S. Mabu, K. Hirasawa, and J. Hu, "Genetic network programming with sarsa learning and its application to creating stock trading rules," in *Proc. 2007 IEEE Congr. Evol. Comput. (CEC)*, Singapore, pp. 220–227.

[15] Y. Kwon and B. Moon, "A hybrid neurogenetic approach for stock forecasting," *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 851–864, May 2007.

[16] T. Chavarnakul and D. Enke, "Intelligent technical analysis based equivolume charting for stock trading using neural networks," *Expert Syst. Appl.*, vol. 34, no. 2, pp. 1004–1017, 2008.

[17] Y. C. Chien and Y. Chen, "Mining associative classification rules with stock trading data—A GA-based method," *Know.-Based Syst.*, vol. 23, no. 6, pp. 605–614, 2010.

[18] J. Wang and S. Chan, "Stock market trading rule discovery using pattern recognition and technical analysis," *Expert Syst. Appl.*, vol. 33, no. 2, pp. 304–315, 2007.

[19] C. Hou, F. Nie, D. Yi, and Y. Wu, "Joint embedding learning and sparse regression: A framework for unsupervised feature selection," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 793–804, Jun. 2014.

[20] Y. Xiao, Z. Zhu, Y. Wei, A. Wei, and X. Li, "Topographic NMF for data representation," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1762–1771, Oct. 2014.

[21] P. Chang, C. Fan, and C. Liu, "Integrating a piecewise linear representation method and a neural network model for stock trading points prediction," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 39, no. 1, pp. 80–92, Jan. 2009.

[22] F. Nie, D. Xu, and X. Li, "Initialization independent clustering with actively self-training method," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 1, pp. 17–27, Feb. 2012.

[23] J. A. Hartigan, "Direct clustering of a data matrix," *J. Amer. Stat. Assoc.*, vol. 67, no. 337, pp. 123–129, 1972.

[24] Y. Cheng and G. M. Church, "Biclustering of expression data," in *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, vol. 8. San Diego, CA, USA, 2000, pp. 93–103.

[25] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: A survey," *IEEE Trans. Comput. Biol. Bioinformat.*, vol. 1, no. 1, pp. 24–45, Jan./Mar. 2004.

[26] Q. Huang, "Discovery of time-inconsecutive co-movement patterns of foreign currencies using an evolutionary biclustering method," *Appl. Math. Comput.*, vol. 218, no. 8, pp. 4353–4364, 2011.

[27] Q. Huang, D. Tao, X. Li, L. Jin, and G. Wei, "Exploiting local coherent patterns for unsupervised feature ranking," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 6, pp. 1471–1482, Dec. 2011.

[28] Q. Huang, "A biclustering technique for mining trading rules in stock markets," in *Codes, Systems and Graphical Models*. New York, NY, USA: Springer, 2011, pp. 16–24.

[29] Q. Huang, D. Tao, X. Li, and W. C. Liewalan, "Parallelized evolutionary learning for detection of biclusters in gene expression data," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 9, no. 2, pp. 560–570, Mar./Apr. 2012.

[30] Y. Yang, D. Xu, F. Nie, S. Yan, and Y. Zhuang, "Image clustering using local discriminant models and global integration," *IEEE Trans. Image Process.*, vol. 19, no. 10, pp. 2761–2773, Oct. 2010.

[31] Y. Yang, Z. G. Ma, A. G. Hauptmann, and N. Sebe, "Feature selection for multimedia analysis by sharing information among multiple tasks," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 661–669, Apr. 2013.

[32] M. Ratner and R. P. Leal, "Tests of technical trading strategies in the emerging equity markets of Latin America and Asia," *J. Bank. Financ.*, vol. 23, no. 12, pp. 1887–1905, 1999.

[33] M. J. Kim, S.-H. Min, and I. Han, "An evolutionary approach to the combination of multiple classifiers to predict a stock price index," *Expert Syst. Appl.*, vol. 31, no. 2, pp. 241–247, 2006.

[34] C. Thira and E. David, "A hybrid stock trading system for intelligent technical analysis-based equivolume charting," *Neurocomputing*, vol. 72, no. 16, pp. 3517–3528, 2009.

[35] (2011, Sep. 22). *Yahoo Finance*. [Online]. Available: http://finance.yahoo.com/

[36] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, "Fast and accurate matrix completion via truncated nuclear norm regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2117–2130, Sep. 2013.

[37] L. Shao, L. Liu, and X. Li, "Feature learning for image classification via multiobjective genetic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 7, pp. 1359–1371, Jul. 2014.

[38] Y. Liu *et al.*, “Understanding and enhancement of internal clustering validation measures,” *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 982–994, Jun. 2013.

[39] J. Wu, H. Xiong, and J. Chen, “Adapting the right measures for K-means clustering,” in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min. (KDD)*, Paris, France, 2009, pp. 877–886.

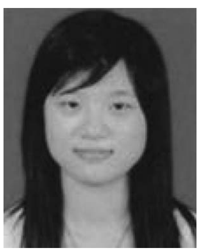
[40] H. Xiong, J. J. Wu, and J. Chen, “K-means clustering versus validation measures: A data distribution perspective,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 318–331, Apr. 2009.

[41] J. Tao, F. Chung, and S. Wang, “A kernel learning framework for domain adaptation learning,” *Sci. China Inf. Sci.*, vol. 55, no. 9, pp. 1983–2007, 2012.

Qinghua Huang received the Ph.D. degree in biomedical engineering from Hong Kong Polytechnic University, Hong Kong, in 2007.

He is currently a Full Professor with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China. His current research interests include pattern recognition, medical imaging, bioinformatics, and intelligent computation and its applications. He has published over 100 technical papers, and holds nine Chinese patents.

Mr. Huang serves as a Guest Editor, an Editorial Board Member, or a Reviewer for almost 30 international journals.



Ting Wang received the B.E. degree in microelectronics from the South China University of Technology, Guangzhou, China, in 2012, where she is currently pursuing the master’s degree with the School of Electronic and Communication Engineering.

Her current research interests include data mining and pattern recognition.

Dacheng Tao (M’07–SM’12) is a Professor of Computer Science with the Centre for Quantum Computation and Intelligent Systems and the Faculty of Engineering and Information Technology, University of Technology, Sydney, Ultimo, NSW, Australia. His current research interests include statistics and mathematics for data analytics, computer vision, data mining, geoinformatics, image processing, machine learning, multimedia, and video surveillance. He has authored and co-authored one monograph and over 100 publications at prestigious journals and prominent conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON CYBERNETICS, JMLR, IJCV, NIPS, ICML, CVPR, ICCV, ECCV, AISTATS, ICDM, ACM SIGKDD, and *Multimedia*.

Prof. Tao was the recipient of the Best Theory/Algorithm Paper Runner Up Award in IEEE ICDM’07, and the Best Student Paper Award in IEEE ICDM’13.

Xuelong Li (M’02–SM’07–F’12) is a Researcher (Full Professor) with the Center for Optical Imagery Analysis and Learning, State Key Laboratory of Transient Optics and Photonics, Xi’an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi’an, China.