A Two-Phase Stock Trading System Using Distributional Differences

Sung-Dong Kim[1], Jae Won Lee[2], Jongwoo Lee[3], and Jinseok Chae[4]

[1] Dept. of Computer System Engineering, Hansung University,
Seoul 136-792, Korea
sdkim@hansung.ac.kr
[2] School of Computer Science and Engineering, Sungshin Women's University,
Seoul 136-742, Korea
jwlee@cs.sungshin.ac.kr
[3] Division of Information and Communication Engineering, Hallym University,
Chunchon-si, Kangwon-do 200-702, Korea
jwlee44@hallym.ac.kr
[4] Dept. of Computer Science and Engineering, University of Incheon,
Incheon 402-749, Korea
jschae@incheon.ac.kr

Abstract. In the context of a dynamic trading environment, the ultimate goal of the financial forecasting system is to optimize a specific trading objective. This paper presents a two-phase (extraction and filtering) stock trading system that aims at maximizing the rates of returns. Extraction of stocks is performed by searching specific time-series patterns described by a combination of values of technical indicators. In the filtering phase, several rules are applied to the extracted sets of stocks to select stocks to be actually traded. The filtering rules are induced from past data using distributional differences. From a large database of daily stock prices, the values of technical indicators are calculated. They are used to make the extraction patterns, and the distributions of the discretization intervals of the values are calculated for both positive and negative data sets. The values in the intervals of distinctive distribution may contribute to the prediction of future trend of stocks, so the rules for filtering stocks are induced using those intervals. We show the rates of returns by the proposed trading system, with the usefulness of the rule induction method using distributional differences.

1 Introduction

The investor's ultimate goal is to optimize some relevant measures of trading objective, such as profit, economic utility or risk-adjusted return. Considerable research efforts have been devoted to the development of financial forecasting models that try to optimize the goal. This paper addresses the problem of financial forecasting to optimize the rates of returns in stock investment.

Most researches in the financial forecasting community have focused on building predictive models. The prediction models incorporate various types of explanatory variables: so-called technical variables (depending on the past price

R. Cicchetti et al. (Eds.): DEXA 2002, LNCS 2453, pp. 143–152, 2002.

sequence), micro-economic stock-specific variables (such as measures of company profitability), and macro-economic variables (which give information about the business cycle) [1]. There are two types of analysis for predicting the stock market: technical and fundamental analyses. Technical analysis is based on the daily price data and it is difficult to create a valid model for a longer period of time. Fundamental analysis considers information concerning the activities and financial situation of each company. Neural networks are considered to provide state-of-the-art solutions to noisy time series prediction problems such as financial prediction [2]. Rule inference for financial prediction through noisy time series analysis is also performed using neural networks [3]. Most studies focus on the prediction of a stock index, not on the prediction of multiple stocks. The use of the prediction system in trading would typically involve the utilization of other financial indicators and domain knowledge. A general approach to decision making in dynamic stock trading environment is to build a trading system that consists of a predictive model and the decision rules converting the prediction into an action [4].

In this paper we propose a two-phase stock trading system that considers multiple stocks in KOSPI and KOSDAQ, two Korean stock markets. The system is composed of an extraction and a filtering phases and its ultimate goal is to maximize the rates of returns. It is also parameterized by a trading policy, which specifies a target profit ratio, a stop loss ratio, and a maximum holding period. Only technical data is used for technical analysis, so the system is adopted to short-term stock trading. Extraction of stocks is performed through pattern matching and filtering is done based on the decision rules. The pattern is a combination of the values of technical indicators. The decision rules, filtering rules, are induced from past data using distributional differences. They are the rules that classify the states of the extracted stocks by which the investment action takes place. In the stock trading system, buy is based on the extraction and the filtering, and sell is directed by a trading policy.

From a database of daily stock prices, the values of technical indicators are calculated. We make the extraction patterns using the values and collect the pattern-matched stocks that serve as the training data. In order to induce the decision rules, the training data is classified into positive and negative and the values are discretized into intervals. The distributions of the value intervals are calculated for positive and negative data sets respectively. Some intervals show a distinctive distribution and the values in those intervals may contribute to the classification of the stocks. The decision rules are induced using the distinguished intervals.

Section 2 explains the environment for building a stock trading system. Overall structure of the proposed stock trading system is given in Section 3. Extraction patterns, the induction process of decision rules, and the trading policy are also described. The results of hypothetical investments are shown in Section 4. Section 5 draws a conclusion and presents further works.

2 Setup Description

2.1 Stock Database

We construct a database of daily stock prices. It contains data for all stocks in KOSPI and KOSDAQ, now about 1,700 stocks. The data is daily and spans 12 years, from January 1990 to December 2001. Raw data (RD) consists of seven fields and can be represented as follows:

$$RD = (name, date, p_o, p_c, p_h, p_l, v) \ ,$$

where p_o is a daily open price, p_c is a daily close price, p_h is the daily highest price, p_l is the daily lowest price, and v is a daily trading volume.

From the raw data, technical indicators are calculated such as price moving average and RSI (Relative Strength Index) [5]. Using the values of the indicators, we construct time-series patterns.

2.2 Time-Series Patterns

In the literature about financial prediction, self-organizing maps [6] and neural networks [7] are used to discover time-series patterns. Recently there have been efforts to identify "change point" with data mining techniques [8].

In this paper we construct patterns using stock expert's knowledge, so the patterns are intuitive and empirical ones rather than automatically acquired patterns from time-series data. Three kinds of pattern sets are established: R, $S1$, $S2$. They are based on the support and resistance levels [5]. First set (R) is for patterns that break the resistance level upward and the others ($S1$, $S2$) are for patterns that relate to the support level. The representative patterns of each pattern set are exemplified in Figure 1. Figure 1-(a) is an example of R, 1-(b) is for $S1$, and 1-(c) is a pattern of $S2$.

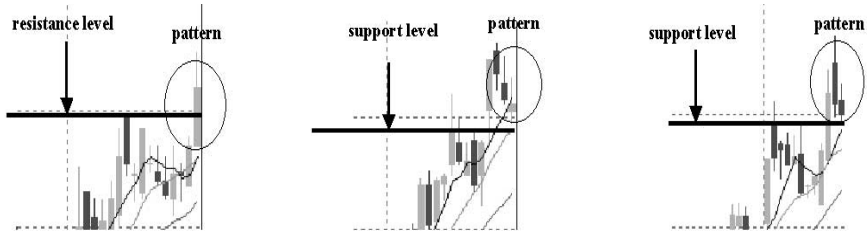

Fig. 1. The examples of patterns in each pattern set

A pattern ($pattern_i$) is represented by a set of questions for the values of the technical indicators:

$$pattern_i = (ques_1(ti_1), ques_2(ti_2), \ldots, ques_n(ti_n)) \ .$$

The following is a concrete example of a pattern[1]:

$$pattern_1 = (Grad5_0 \geq 1.23, RC_0 \geq 0.04, VL_1 < VL_0, Disp5{-}10_1 < Disp5{-}10_0) \ .$$

These patterns will be used in extraction phase of the proposed trading system.

2.3 Training Data

The training data is used to build decision rules for filtering phase in our trading system. From the stock database described in Section 2.1, we extract data of the stocks that are matched with patterns in Section 2.2. From January 1998 to June 2000 (30 months), we extract stocks by pattern matching and collect data for the stocks. The collected data ($D(s)$) consists of the values of technical indicators and several trigram values reflecting 3 days trend of technical indicators:

$$D(s) = (ti_1, \ldots, ti_n, tri_1, \ldots, tri_m) \ , \tag{1}$$

where ti stands for a value of a technical indicator, tri represents a trigram value, and s is an extracted stock. All elements of $D(s)$ are considered in filtering phase, and some of them participate in describing patterns, The trigram consists of 3 ternary symbols ($b_1, b_2, b_3 \in \{-1, 0, 1\}$) obtained by discretizing the original, real value (v) using the "signum" function:

$$b_i = sign(v) = \begin{cases} -1, & \text{if } v < 0 \\ 0, & \text{if } v = 0 \\ 1, & \text{if } v > 0 \end{cases} \ .$$

Because different technical indicators participate in describing each pattern, the elements of each $D(s)$ are different with each other depending on the matched pattern.

Each of the collected data is tagged as *positive* or *negative*, based on the one-day return $R_1(s)$:

$$R_1(s) = \frac{\text{close price - open price}}{\text{open price}} \ , \tag{2}$$

where the prices are the next day prices of the extraction day. The class of each training data ($C(s)$) is determined as follows:

$$C(s) = \begin{cases} positive, & \text{if } R_1(s) \geq 0 \\ negative, & \text{if } R_1(s) < 0 \end{cases} \ .$$

Table 1 shows the number of extracted stocks in each pattern set and statistics of the training data.

[1] In the example, subscript 0 means the today and 1 is yesterday. $Grad5$ is the gradient value of 5-day price moving average, RC is the value of rate of change, VL is a trading volume, and $Disp5-10$ is a disparity value between 5-day and 10-day price moving averages.

Table 1. The statistics of the training data

Pattern Set	Total	Positive	Negative
R	12658	5324	7334
$S1$	5572	2628	2944
$S2$	6624	3000	3624

2.4 Discretization

It has been shown in the past that discretizing real-valued financial time-series into symbolic streams and subsequent use of predictive models on such sequences can be of great benefit in many financial tasks [9][10]. However, the question of the number and position of discretization intervals has been largely dealt with in an *ad hoc* manner. For example, [10] quantized daily returns of exchange rates of five currencies into nine intervals and up to seven quantization intervals for the returns were considered in [9]. A data-driven parametric scheme for quantizing real-valued time-series is introduced in [11].

We determine the number of intervals such that standard deviation of the distribution of intervals would be maximized. The distribution of intervals of a technical indicator ($p_{ti}(int_i)$) is calculated by the formula:

$$p_{ti}(int_i) = \frac{\#\text{ of values in } int_i}{\text{total }\#\text{ of values}} , \tag{3}$$

where int_i is the ith interval. We try the number of intervals from 3 to 10. Therefore, the determination of the number of intervals (n_{ti}) is represented as:

$$n_{ti} = \arg\max_k \sigma(p_{ti}^k) ,$$

where $\sigma(p_{ti}^k)$ means the standard deviation of the distribution when the number of intervals is k. The position of the intervals is determined to keep the size of the interval same. For that purpose, the values (v) are sorted in decreasing order and the ith position (pos_i) is determined as follows:

$$pos_i = \max(v) - (\max(v) - \min(v)) \times i .$$

3 Trading System

We present a stock trading system that consists of an extraction and a filtering phases with a parameterized trading policy. Figure 2 shows the structure of the trading system.

3.1 Extraction Phase

In this phase, a stock is described by a set of values of technical indicators calculated using data from a stock database in Section 2.1. The questions of the

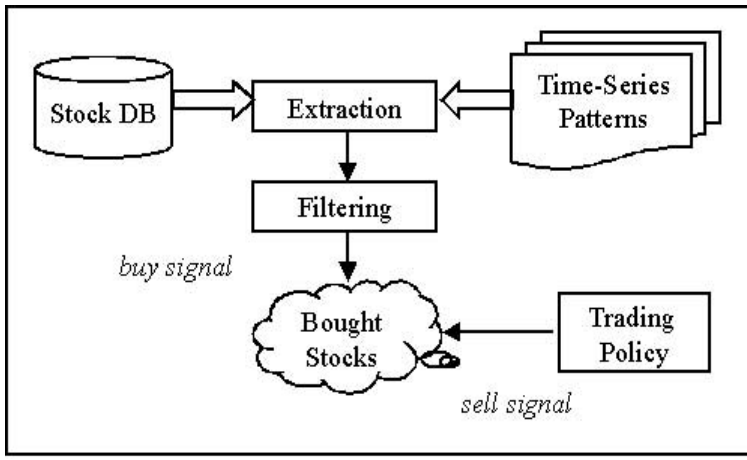

Fig. 2. The structure of the stock trading system

constructed patterns in Section 2.2 are checked with the stock description shown in equation (1). We perform pattern-based extraction rather than the prediction of a single stock because there are multiple stocks in real stock market. The patterns are applied to multiple time-series data and the stocks are matched of which values of indicators satisfy all the questions of a pattern. The set of buying candidates based on the pattern sets ($\mathcal{C}_{\mathcal{P}}$) is expressed as:

$$\mathcal{C}_{\mathcal{P}} = \{cs \mid \forall\, ti_i \text{ of } cs,\ cs(v_{ti_i}) \vdash ques_i(ti_i) \in pattern_j(\in \mathcal{P})\}\ , \tag{4}$$

where the subscript $\mathcal{P}$ means a pattern set, cs is a candidate stock, $cs(v_{ti_i})$ is the value of technical indicator ti_i of cs, and '$\vdash$' means the satisfaction of a question.

3.2 Filtering Phase

In order to take action using the candidates extracted in Section 3.1, we have to select stocks that are most likely to result in high profit. Even if the stocks are extracted using the same pattern, the values of their technical indicators may be different. Therefore, we need to construct classification rules to help the decision making to take a buy action.

Using the training data in Section 2.3, we induce rules that classify the extracted stocks into two categories: *profit-expected, loss-expected.* Only *profit-expected* stocks would be filtered. The overall process of rule induction is illustrated in Figure 3.

The distributions of discretization intervals of technical indicators are calculated respectively for both positive and negative data sets using the equation (3). That is, discretization intervals are considered as random variables and their probability distributions are constructed. For all technical indicators participating in describing a stock, we compare the distributions of intervals. Using

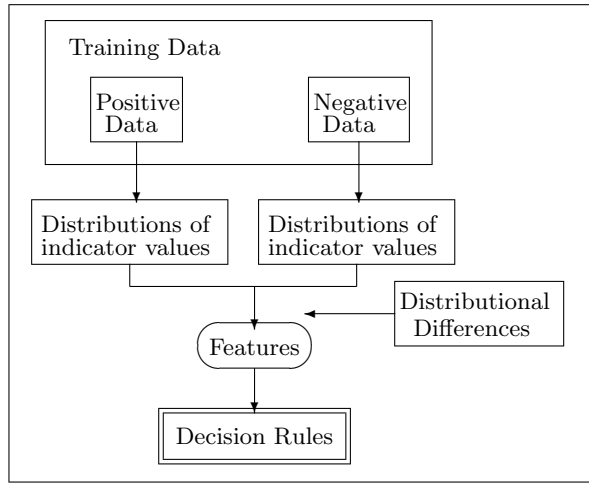

Fig. 3. The overall process of rule induction

distributional differences, we select some intervals as features for building classification rules. A set of selected features (FS_P) is represented as:

$$FS_P = \{f_{ti_i}\ , f_{tri_j} \mid p_{pos}(f_{ti_i}) > p_{neg}(f_{ti_i}) + \epsilon \text{ or } p_{pos}(f_{tri_j}) > p_{neg}(f_{tri_j}) + \epsilon\}\ ,$$

where f_{ti_i} is a discretization interval of ti_i, f_{tri_j} is a trigram value, $p_{pos}(f_{ti_i})$ is a probability of f_{ti_i} in positive data set, and $p_{neg}(f_{ti_i})$ is one in negative data set. The constant ϵ is introduced to determine conspicuous features.

For each element in the set, the following rule (r_i) is generated:

$$\text{if } v_{ti_i} \in f_{ti_i}, \quad \text{then } RS(r_i) \ = p_{pos}(f_{ti_i}) - p_{neg}(f_{ti_i}) \text{ else } RS(r_i) \ = 0\ .$$

A rule score ($RS(r_i)$) is associated with each rule, which can be regarded as a certainty factor.

A buy signal (bs) is generated using the score of the stock and the buy signal function (f_{bs}) is expressed as:

$$f_{bs}(cs) = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} RS(r_i) > \lambda \\ 0, & \text{otherwise} \end{cases}\ ,$$

where cs is an element of $\mathcal{C}_\mathcal{P}$ in the equation (4) and λ is a threshold value. The filtered set of stocks ($\mathcal{BS}_\mathcal{P}$) is defined as follows and a buy action is to be performed for stocks in the set:

$$\mathcal{BS}_\mathcal{P} = \{s \mid f_{bs}(s) = 1 \text{ and } s \in \mathcal{C}_\mathcal{P}.\}\ .$$

3.3 Stock Trading Policy

Stock trading policy (STP) specifies a target profit ratio (tpr) and a stop loss ratio (slr) and a maximum holding period (mhp), which directs the sell action:

$$STP = (tpr, slr, mhp)\ .$$

A buy signal from the filtering phase can be generated after the stock market is closed, so buy is actually performed on the next day. We assume the buy must be done with an open price. Given a stock trading policy, $STP = (\alpha, \beta, \gamma)$, the target profit price (TPP) and the stop loss price (SLP) can be obtained as:

$$TPP = bp \times (1 + \alpha) , \qquad SLP = bp \times (1 - \beta) ,$$

where bp is the buy price. The sell is performed when the target profit price or the stop loss price is reached within the specified maximum holding period. If neither price appears within γ days, the stock would be sold at the close price on the last day of the maximum holding period.

4 Experiments

We simulate stock trading using the proposed trading system on data from July to December 2001 (6 months). Table 2 shows the number of extracted and filtered stocks using three pattern sets in Section 2.2.

Table 2. The statistics of trading stocks

Pattern Set ($\mathcal{P}$)	# of $\mathcal{C}_\mathcal{P}$	# of $\mathcal{BS}_\mathcal{P}$
R	1829	668
$S1$	1116	604
$S2$	1288	895

Using two trading policies, we calculate the rates of returns for each pattern set respectively: $STP_1 = (0.05, 0.05, 1)$, $STP_2 = (0.1, 0.05, 5)$. That is, we evaluate the predictability of daily and weekly returns. Buy is performed on the stocks in $\mathcal{BS}_\mathcal{P}$ and sell is done as a way described in Section 3.3. The rate of return, $R(s)$, is defined somewhat differently from the equation (2):

$$R(s) = \frac{\text{sell price} \ - \ \text{buy price}}{\text{buy price}} \times 100 .$$

And, we introduce *profit per trade* (PPT) as an evaluation measure for trading system:

$$PPT = \frac{\sum_{i=0}^{n} R(s)}{n} \ (\%) , \quad s \in \mathcal{BS}_\mathcal{P} \text{ and } n = |\mathcal{BS}_\mathcal{P}| .$$

Table 3 shows the results of the evaluation. The average holding period in STP_2 is 2.78, 2.88, 2.78 days for pattern sets R, $S1$, and $S2$ respectively. PPT is calculated without the consideration of transaction costs. For performance comparison, the results using decision tree and neural network in filtering phase are also given in Table 3. C4.5 [12] is used to construct decision tree. And, the network is a two-layered feedforward neural network and is trained using the same training data in Section 2.3. As we can see in this table, our method of

Table 3. The results of the evaluation (PPT)

Pattern Set		Extraction	Filtering	Decision Tree	Neural Network
R	STP_1	-0.322	0.107	-0.17	-0.01
	STP_2	-0.58	-0.275	-0.35	0.03
S1	STP_1	0.365	0.782	0.757	0.505
	STP_2	0.937	1.683	1.65	1.165
S2	STP_1	-0.03	0.318	0.04	0.145
	STP_2	0.085	0.265	0.344	0.273

rule induction using distributional differences shows better performance. Moreover, we can see that the transaction costs (0.5% per trade) can be overcome in some cases. During the simulation period, KOSPI increased about 16.8% and KOSDAQ rose about -6.4%. The simple cumulative profits of trading for $S1$ pattern sets are 170.328%[2] and 714.532%[3]. Though the results may be different from those of real trading, we can expect more profit than market average with the consideration of the transaction costs.

5 Arguments and Conclusion

We propose a two-phase stock trading system parameterized by the trading policy, which consists of an extraction and a filtering phases. We aim to construct a prediction model for trading multiple stocks. In our trading system, multiple stocks are extracted at the same time and traded. Our trading system is based on the technical analysis, and technical indicators are used in building patterns and inducing classification rules for filtering stocks.

In acquiring decision rules, we adopt a relatively simple method using distributional differences for selecting features on which rules are constructed. Our method uses features that are both independently and jointly predictive, and it would probably miss feature interactions. But we try to capture somewhat manifest features excluding noisy features, which shows a competitive performance. The feature selection method using distributional differences may help fast induction of rules and the reduction of parameters for neural networks training.

The patterns are modeled by expert's knowledge. This trading system plays a role to verify and enhance the knowledge using past data. The proposed framework can be useful in building trading systems based on expert's knowledge.

The results also suggest that the trading policy can affect the expected rate of return. Therefore, the study of finding optimal parameters of a stock trading policy can be considered as future work. Also, more technical indicators can be considered to enhance the performance of the trading system.

[2] (0.782 - 0.5) × 604 = 170.328.
[3] (1.683 - 0.5) × 604 = 714.532.

Acknowledgment

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Northeast Asian e-Logistics Research Center at University of Incheon.

References

1. Joumana Ghosn, Yoshua Bengio: Multi-Task Learning for Stock Selection, Advances in Neural Information Processing Systems, volume 9, (1997), 946-952, Michael C. Mozer and Micheal I. Jordan and Thomas Petsche editor, The MIT Press.
2. A. Refenes: Neural Networks in the Capital Markets, (1995), John Wiley and Sons.
3. C. Lee Giles, Steve Lawrence, Ah Chung Tsoi: Rule Inference for Financial Prediction using Recurrent Neural Networks, in Proceedings of IEEE/IAFE Conference on Computational Intelligence for Financial Engineering, (1997), 253-259.
4. N. Towers, A. N. Burgess: Optimisation of Trading Strategies using Parameterised Decision Rules, in Proceedings of IDEAL 98, Perspectives on Financial Engineering and Data Mining, (1998), L. Xu et al editor, Springer-Verlag.
5. Optima Investment Research: Interpreting Technical Indicators, Fourth Edition, (1998), http://www.oir.com.
6. Tak-chung Fu, Fu-lai Chung, Vincent Ng, Robert Luk: Pattern Discovery from Stock Time Series Using Self-Organizing Maps, Workshop Notes of KDD 2001 Workshop on Temporal Data Mining, 26-29 Aug., San Francisco, (2001), 27-37.
7. Jinwoo Baek and Sungzoon Cho: Left Shoulder Detection in Korea Composite Stock Price Index Using an Auto-Associative Neural Network, in Intelligent Data Engineering and Automated Learning - IDEAL 2000, Data Mining, Financial Engineering, and Intelligent Agents, Second International Conference, Springer, (2000), Shatin, N.T. Hong Kong, China.
8. V. Guralnik and J. Srivastava: Event Detection from Time Series Data, in Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (1999), 33-42.
9. C. Lee Giles, Steve Lawrence, Ah Chung Tsoi: Noisy time series prediction using a recurrent neural network and grammatical inference, Machine Learning, volume 44, (2000), 161-183.
10. C. P. Papageorgiou: High frequency time series analysis and prediction using Markov models, in Proceedings of IEEE/IAFE Conference on Computational Intelligence Financial Engineering, (1997), 182-185.
11. P. Tino, C. Schittenkopf, G. Dorffner: Volatility trading via temporal pattern recognition in quantized financial time series, Pattern Analysis and Applications, (2001).
12. R. Quinlan: C4.5: Programs for Machine Learning, Morgan Kaufmann, (1992).