An Adaptive Recommendation System for Personalized Stock Trading Advice Using Artificial Neural Networks

Chayaporn Kaensar* and Thanarat Chalidabhongse**

Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
(Tel: +66-2-679-8556; E-mail: chayaporn_kae@freewillsolutions.com)*
(Tel: +66-2-737-2551-4; E-mail: thanarat@it.kmitl.ac.th)**

Abstract: This paper describes an adaptive recommendation system that provides real-time personalized trading advice to the investors based on their profiles and trading information environment. A proposed system integrates Stochastic technical analysis and artificial neural network that incorporates an adaptive user modeling. The user model is constructed and updated based on initial user profile and recorded user interactions with the system. The information presented to each individual user is also tailor-made to fit the user's behavior and preference. A system prototype was implemented in JAVA. Experiments used to evaluate the system's performance were done on both human subjects and synthetic users. The results show our proposed system is able to rapidly learn to provide appropriate advice to different types of users.

Keywords: Artificial Neural Networks, Intelligent System, Adaptive Control, Modeling and Simulation.

1. INTRODUCTION

In the past decade, the growth of the Internet and the WWW astoundingly increases. There are now more than 60 million web sites on the Internet as reported by Netcraft [1]. It now becomes the main route for information exchange in many areas including investment market. As the growth of financial related webs continues to accelerate, the investors become facing with a serious case of information overload. They are flooded with large masses of information, and find it hard to extract the information that is really relevant or useful to them in making a decision.

In this research, we have developed an online stock trading system that provides personalized recommendation to the investors. Typical systems usually recommend one of the five different actions: buy, buy warning, hold, sell warning, and sell on a company's stock based on fundamental and technical analysis without taking the investor's profile and preference into account. However, there are many types of investors out there that are different in investing personalities, styles, interests, risk aversion, etc. Some experienced investors like to see more information and analyze the stock on their own, while some novice or no-time investors might just need a simple recommendation. We propose a system that is intended to make investment information access easier and customized to the investors. The system provides real-time information and guidance based on trading environment and individual investor's profile.

The remaining sections are structured as follows: Section 2 explores the related work in the field of IUI (Intelligent User Interface). Section 3 presents the systems architecture. Section 4 describes our recommendation approach. Section 5 discusses the experiments. Finally, Section 6 concludes the presentation.

2. RELATED WORKS

There have been many works on the development of IUI in several areas. For examples, Webber et al. [2] and Tasso et al. [3] employ user-modeling techniques based on student skill and background knowledge to develop Information Tutoring System. Brusilovsky [4] presents browsing-based access to information source area. The author also mentioned about link manipulation, e.g., hide, sort, annotate, and adaptive presentation technique according to user modeling, interest, and knowledge respectively. Information Retrieval and Filtering is another area that is related. Shepherd et al. [5] presents a framework that helps users filtering news, and Kay [6] proposed the Movies Advisor Project that could suggest movies to a specific user based on their interest.

For stock trading systems, most of the existing systems use machine learning and soft computing techniques to analyze a company's stock and suggest an action to the investors. Examples include an online financial informational web service, Tradetrek [7]. Achelis [8] proposed trading algorithm that works by using trading indicators as parameters in mathematics equations. Kuo [9] used Neurofuzzy approach for predicting financial time series. Tseng and Gmytrasiewicz [10] developed the real-time DSS system, based on object oriented technology and using Bayesians Network, to produce investment recommendation. To the best of our knowledge, most of the works mentioned above do not consider the individual investor's profile into the recommendation process. Recently, Yoo et al. [11] proposed a leaning algorithm to personalize advice to different users. However, they did not emphasize on real time recommendation.

The underlying motivation of our work is to build a real-time personalized stock recommendation and portfolio management system that is capable of customizing information based on user's profile. The system must also have ability to adapt user's model as well as information presentation based on the interactions of the user with the system.

3. SYSTEM ARCHITECTURE

The prototype of the system is developed based on client-server architecture. The server contains 5 main components (Figure 1):

- Trading Information Manager (TIM) gathers and manages various data necessary for trading decision such as current stock information, historical trading information, etc.
- User Modeler (UM) component exploits an Artificial

Neural Network to construct an initial user model, and to update and revise the model during user's interaction with the system.

- Knowledge Manager (KM) manages strategies on how to make trading decision.
- Personalized Recommendation Agent (PRA) makes appropriate suggestion for a company's stock based on individual investor profile and trading environment.
- Presentation Module (PM) adaptively and dynamically generates the information to be presented to the user

To create personalized recommendation, the system first constructs an initial user model from user profile using ANNs. Then converting trading constraints into decision rules for trading decision making. Next, the system selects and ranks appropriate information for the presentation based on individual user profile. The user's interactions and responses to previous recommendation are tracked and make these data become activate rule for dynamic version of the user model. This may in turn change the recommendation in the future.

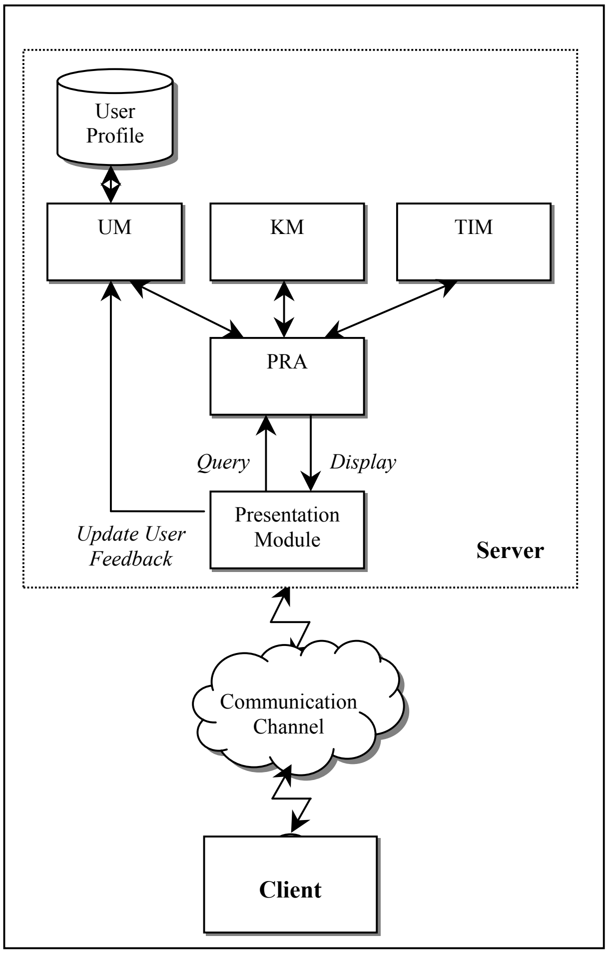

Fig. 1 The System Framework.

4. THE RECOMMENDATION APPROACH

4.1 Integrating User Modeling in Recommending Trading Decision

The UM component generates the user models from a database containing user's profile, which includes education, age, interest, investing experience, level of risk aversion, portfolio status, etc., and the recorded interactions of the users with the system. In this work, a Backpropagation Neural Networks (BPN) is used to learn the user models as well as providing personalized advice based on user model and trading environment. BPN uses the backpropagation algorithm as a learning process that is capable of storing and recognizing the input variables and thereby generating appropriate output [12]. Our network consists of three layers: the input layer, one hidden layer and an output layer. The input layer consists of nodes representing dimensions of user model as described above. The hidden layer consists of nodes which each node has link back to the input layer and output nodes in the output layer. Note that the decision on a number of nodes in hidden layer is still critical issue. Although, there are many guidelines to determine the solution as described in [13], we still need many trail-and-errors to ensure the system yields nearly best performance. The output layer has five nodes with five possible stables such as buy, buy warning, sell, sell warning and hold. The benefit from this approach is adaptivity because the neural net has a build-in capability to adapt their synaptic weights to changes in the surrounding environment. Moreover, it could be easily retrained to adapt to changing environment [14].

In generating the trading recommendation, our system also exploits Stochastic, a simple method for Technical Analysis, as a rule to make decision. This technique examines moving average of both short term and middle term based on stock information (i.e. high/low price, open/close price, index). Typically, they consist of two variables are %K-Line and %D-Line. Moreover, this technique could indicate moving of stock price and opportunity for making transaction when this crossing occurs. Thus, the time to buy stock is when both variables are decreasing and the crossing point has dipped lower than 20%. Conversely, the time to sell is when those are increasing and the crossing point has exceeded over 80%. To compute moving average effectively, we collect information from the latest 3 months. This make the system could clarify the suggestion correctly. Then, the Stochastic results are converted into decision rules for recommending the five different actions: buy, buy warning, sell, sell warning and hold. The buy and sell rules are obtained as described above. For the other actions, a buy warning indicates time to buy in the near future, that is, those variables are likely decreasing and the crossing point has moved between 20% to 40%. Similarly, a sell warning indicates time to sell in the near future, that is, those variables are growing up and their boundary is between 60% to 80%. The other range indicates hold action.

In addition, a set of constraints is converted into decision rules to determine a crossing point between lines. For example, the trend of moving average is converted to the constraints as *m (Slope)*. If *m* is positive value, this means the direction is growing up in the near future. Otherwise, the negative side is applied to the rule. A set of crossing point constraints is represented as

$$S_{History} = \{(P_1, Action_1), \ldots, (P_n, Action_n)\} \qquad (1)$$

where P_i represents crossing point i (x_i, y_i), and $Action_i$ represents the corresponding action.

4.2 Personalizing Recommendation

We make use of the user's profile, usage data, and portfolio status to indicate what stocks that user should be interested with some actions. Briefly, when a user login to the system, a user model will be loaded. While user is interacting with the system, the system learns user behavior and self-adapts the model so that the provided information fits the user's need. We use supervised induction algorithm of ANNs for improving leaning process. We divide the user's interaction tracking tasks into two major tasks as follows:

First, a feedback session is provided to detect the user's acceptance and rejection. If user responses by accepting the advice, a positive example is created. Conversely, if user response is rejection, a negative example is created. These examples are fed back to the system for continuing learning process.

Second, the relevant/interest data to the user is monitored. As mentioned before, user interaction is recorded. To help user coping with the problem of information overload, the system try to understand user's behavior and preferences. Using history of interaction and records of transactions made on a particular stock item to update user model, the system can tailor information that is interest or useful to the user.

5. EXPERIMENTAL EVALUATION

A prototype of real-time personalized stock trading and recommendation system is developed to test whether the proposed adaptive recommendation could help providing the most appropriate trading advice and reduce risk to the investor. To support our hypothesis, we conducted a set of experiments with human and synthetic subjects as follows:

5.1 Experiment with Human Subjects

To validate the system, we evaluated user's satisfaction by measuring successes in providing right recommend stock items to different users. Specifically, for each user, we measured the percentage of acceptance. The higher acceptance rate, the better the system serves the users with the right recommendation. Therefore, we expected the acceptance rate would gradually increase after system learns from user's interactions.

We conducted an experiment with 10 human users who had various backgrounds and interests. We first asked them to provide their personal data such as education, age, investing experience, investing interest, risk aversion level, portfolio status, credit available, etc. To initialize their profile, we then request them to complete one practice before logging in to the system. For this experiment, we used some historic trading data from SET (Stock Exchange of Thailand). After system generated various recommendations, based on the BPN classification, users were requested to provide a feedback whether they accepted or rejected the recommendations. Then, we calculated the acceptance rate which is defined as

$$\text{Acceptance rate} = (S_{rec} \wedge U_{like}) / S_{rec} \quad (2)$$

where S_{rec} is a number of recommendation items that the system recommends to the user, and U_{like} is a number of recommendation items that user determine actually were of interest

Figure 2 shows the acceptance rate as a function of the number of user interactions. The acceptance rate is gradually increased while the system learns and gradually adapts the users' models. The result shows the adaptivity of the system is useful in providing better appropriate advice to users.

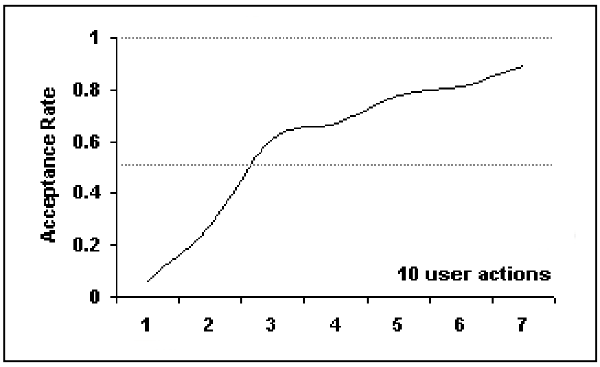

Fig. 2 User's acceptance rate with human subjects.

5.2 Experiment with Synthetic Subjects

Another experiment was done using synthesis subjects. The goal is to compare recommendation effectiveness of the proposed system versus a stock recommendation provided by a well-known brokerage company, which provides stock advice using experienced personnel. In this experiment, we randomly picked 10 stock tickers each day over a week period, and provided users with action recommendations on these 10 stocks. We then measured the effectiveness of the advice by computing the profit return rate. Since the recommendation provides only action, sell or buy or hold, but does not tell the appropriate price to sell or buy. We thus use the market price as an offer price for both sell and buy transactions. Then, we computed the return rate for each day. We then plot the return rates over the week period as shown in Figure 3.

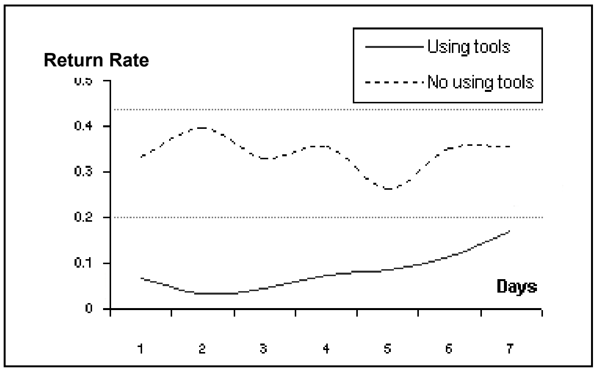

Fig. 3 Comparing return rates between taking advice from our system versus taking advice from a well-known brokerage firm.

The return rates of taking advice from a well-known brokerage firm (dashed plot) show fluctuating return rates at about 30%. The return rates of taking advice from our system (solid plot) show initially small decrease following by gradual and continuous increase. This could be interpreted as follows: at the initial period, the initial model could not adjust itself to fit the market condition immediately. This might also be depending on a number of training, number of sample data, as well as the trading environments. However, after taking some time, the system adapts itself and can provide better advice yielding better return rates. Over a week period, the overall return rate continuously climbs up to almost reaching 20%.

6. CONCLUSIONS

We have described a real-time personalized stock trading and recommendation system that can tailor an advice for each user who has a wide variety of interests and backgrounds. To obtain a good recommendation output, all the information, rules, constraints, and techniques must be applied and integrated. We built a prototype and conducted experiments based on both real human subjects and synthetic subjects. The results support our hypothesis that recommendation system could successfully provide better suggestions to users. However, our objective of building such tool is to help an individual user be able to make his/her own decision effectively rather than making decision for them

REFERENCES

[1] Netcraft, "Netcraft's April 2005 Web Server Survey", Available online: http://news.netcraft.com/archives/web_ server _survey.html.

[2] G. Webber, "A Learning System in the WWW," *17th International Conference on User Modeling*, Banff, Canada, pp. 371-377, 1999.

[3] C. Tasso and P. Giangrandi, "Managing Temporal Knowledge in Student Modeling," *6th International Conference on User Modeling*, pp. 415-426, 1997.

[4] P. Brusilovsky, "Adaptive Hypermedia," *Modeling and User-Adapted Interaction*, Vol. 11, nr. 1-2, pp. 87-110, 2001.

[5] Shepherd et al., "Adaptive User Modeling for Filtering Electronic News," *Proceeding of the 35th Hawaii International Conference on System Sciences IEEE*, pp. 123-145, 2002.

[6] J. Kay, "The User Model Toolkit for Cooperative User Modeling," *User Modeling and User-Adapted Interaction,* pp. 149-196, 1995.

[7] Tradetrek Enterprise and Tradetrek Online, Available online: http://www.tradetrek.com

[8] T. Achelis, "Parameter Tuning in Trading Algorithms Using ASTA," Department of Computer Science, Umea, Sweden, 1995.

[9] R.J. Kuo, "A Decision Support System for the Stock Market Through Integration of Fuzzy Neural Network and Fuzzy Delphi," *Applied Artificial Intelligence,* Vol. 12, pp. 501-520, 1998.

[10] C.C. Tseng and P.J. Gmytrasiewicz, "Real Time Decision Support System for Portfolio Management," *Operations Research*, pp. 123-145, 1999.

[11] J. Yoo et al., "An Adaptive Stock Tracker for Personalized Trading Advice," *International Conference on Intelligent User Interfaces*, pp. 123-145, 2003.

[12] G. Domik, "User Modeling in human-computer interaction," *Seminar in Personalization,* University of Paderborn, 2004.

[13] Heaton, Programming Neural Networks in Java, 1st Edition Revision 2, 2004.

[14] R. Macdonald and D. Silver, "Web-Based User Modeling Using Artificial Neural Networks," Honors Thesis, Acadia University, 2001.

[15] G. Grunst, et al., "Adaptive and Adaptable Systems," In Hoschka, Computer Assistants – A New generation of Support Systems, Hillsdale: Lawrence Erlbaum Associates, 1996.