□특집□

# Intelligent Services on the Internet through Data Mining Agents

Steven H. Kim†

◈ 목 차 ◈



## 1. Introduction

The accelerating pace of transactions in cyberspace highlights the need for intelligent agents to assist in decision making. Ideally, a software agent should possess greater functionality than merely fetch data or broadcast simple information. More specifically, an intelligent agent should be able to perform routine tasks autonomously, glean new knowledge from disparate databases, and improve its own performance through experience. This paper examines a number of critical issues behind the development of such an agent, formulates a general architecture, and presents the key modules for implementation. To clarify the concepts, the general architecture is discussed in the context of a learning agent for predicting financial markets.

The objective of knowledge discovery and data mining is to support decision making through the effective use of information. The practical aspect of knowledge mining lies in the development of learning software to discover patterns, trends, or relations in databases[1].

The automated approach to knowledge discovery is especially useful when dealing with large data sets or complex relationships. For many applications, automated software may find subtle patterns which escape the notice of manual analysis, or whose complexity exceeds the cognitive capabilities of humans. Secondly, software for data mining may ensure the practicality of analyzing large data sets which would require an unjustifiable amount of manual effort. Lastly, automated learning tools may monitor complex environments such as financial markets on a continuous basis, updating their knowledge base immediately with the outbreak of new trends.

## 2. Background

Agents on networks. One way to classify agents is in terms of mobility. A sedentary agent resides permanently on a particular machine and handles local tasks such as serving as a user-friendly interface

† 정회원 : Graduate School of Management Korea Advanced Institute of Science and Technology

to a complex application package [2]. On the other hand, a mobile agent travels across a network to pursue high-bandwidth tasks such as protracted negotiations [3].

An intelligent agent is a flexible program characterized by adaptability and autonomy [4]. The more a user comes to depend on a particular agent, the greater is the need for adaptation to his habits and needs [5].

In a network system, security is a paramount concern for both agents and servers [6]. Agents must be protected against accidental damage or deliberate attempts to extract private information. In a similar way, servers must be made secure against deliberate intrusions or accidental calamities caused by welcoming unknown visitors through the network.

To work in a distributed environment, an agent must be able to accommodate a diverse set of knowledge representation formats and database structures. Such an agent has to recognize common standards such as KQML [7], [8].

The preceding issues represent a number of key considerations in the design and implementation of agents on electronic networks. The general architecture presented in the next section reflects the preceding concerns as well as other important issues in intelligent system design [9].

Architecture. An intelligent agent must be able to work in a distributed environment characterized by diverse hardware platforms, database structures, and application packages. To this end, an adaptive agent must function purposively in a heterogeneous network and interact effectively with other agents.

To clarify the concepts behind generic issues, the ideas in this paper will be discussed against the backdrop of a learning system for predicting financial markets. For a financial information system, the user-level tasks must be supported by intermediate functions such as data procurement and processing engines such as inference or learning. The multilevel partitioning of functions is illustrated in Figure 1.

Figure 2 shows the activation cycle for a predictive agent. The agent first procures data, processes it, and implements the outcome of the decision, as exemplified by dispatching a purchase order.

The agents are collectively identified as the Finbot family and the discussion is tailored to the financial domain for illustrative purposes. However, the same architecture and agent personalities may be cloned for other domains such as marketing information systems, manufacturing intranets, and so on. Figures 3 through 7 depict respectively the primary functions of the Zeus, Athena, Odysseus, Hermes, and Helen agents in the Finbot clan.

To fully realize the promise of software agents, the entities must be able to procure new knowledge and improve their performance over time. One effective approach is a multistrategy technique involving case based reasoning and neural networks. The adaptive functionality is described further below.

Context. The past few decades have witnessed increasing interest in the development of software for knowledge mining. The tools have been applied widely to practical domains especially since the late 1980s.

To date, however, practical applications have tended to utilize single techniques in isolation. However, each tool has its advantages and drawbacks. For this reason, a multistrategy approach to knowledge mining offers the potential to take advantage of the strengths of various techniques while bypassing the limitations of each methodology.

Learning Methods. A neural network (NN) offers many advantages such as robustness and graceful degradation [10]. Perhaps the perception structure employing the backpropagation (BPN) algorithm is more popular than all other neural techniques combined. However, most neural nets suffer form a number of limitations such as the need for long learning times. Another drawback lies in the implicit nature of the acquired knowledge, which cannot be explicitly communicated to a human decision maker.

Another useful technique is found in case based reasoning (CBR). A key advantage of case based reasoning lies in the ability to work with data in their original format. Often the methodology is effective even when applied to an incomplete or partially faulty database. One drawback, however, lies in the tendency of conventional CBR tools to identify similarities based on superficial rather than substantive features of two cases. Another limitation is found in the ability to perform well - such as in predicting stock markets - without yielding an explicit explanation of the underlying causative factors. However, this is a limitation applicable to the entire gamut of learning tools.

Neural networks may also be combined with CBR techniques, as illustrated in Figure 8. In the figure, input data enters both the CBR and NN modules. In addition, the output of the CBR module enters the NN model; consequently the former subsystem may be regarded as a partial preprocessing filter for inputs into the neural component.

## 3. CASE STUDY IN INTEREST RATE PREDICTION

The case study involves a comparative evaluation of learning techniques for predicting a complex process. More specifically, neural networks and case based reasoning are employed - both individually and jointly - in the complex task of forecasting US Treasury bills of 1 years maturity.

The data sets consisted of monthly figures. The learning phase involved observations from January 1981 to May 1989, while the testing phase ran from June 1989 to December 1992. The input variables were as follows: Treasury bill with 1-year maturity (TBILL), money supply (M2), consumer price index (CPI), industrial production index (IPI), housing starts (HS), and the Standard and Poors 500 (SPX);

To obtain stationarity and thereby facilitate prediction, the data were transformed by a logarithm and a difference operation. Moreover, to eliminate the effects of units, the resulting variables were standardized.

The study employed two neural network models. One model, labeled NN_NOW, involved input variables at time t to generate a forecast for t+1. The explanatory variables were TBILL[0], M2[0], CPI[0], IPI[0], HS[0], and SPX[0]; for each variable, the argument within brackets indicates the lag (namely, zero lag here).

The second type of neural net, labeled NN_STEP, used variables selected through stepwise regression. The procedure considered 6 variables, each with several time lags. The candidate variables for predicting the interest rate were lags ranging from 0 to 5 for each of the input variables.

The CBR model for the US interest rate employed the same variables used by NN_NOW. This CBR model was also used as a component of the integrated model, labeled CBR+NN. For validation, each learning model was also compared against a random walk model.

The performance of each model was evaluated according to the mean absolute percent error (MAPE). The best model in each class (CBR or NN) was employed in generating the results in Table 1.

The outcomes indicate that the integrated model is superior according to MAPE or RMSE, while the solitary CBR model is best according to the hit rate. The hit rate involves a binary classification of up or down, with no room for a hold in which $X_{t+1} = X_t$. In that case, a reasonable approach is to assign precise matches randomly to either of the up or down category.

The predictive models were compared by pairs. The predictions differed significantly by model. The results in Table 2 indicate that each model except NN_STEP was superior to random walk according to APE. Moreover, the integrated model outperformed NN_STEP at a weakly significant level ($p < 0.10$). On the other hand, the only significant difference according to AD were the superiority of NN_NOW and the integrated model over random walk.

A set of pairwise tests for the hit rate is given in Table 3. All the models except NN_STEP were significantly superior to random walk. Moreover, CBR was significantly superior to NN_STEP.

## 4. FUTURE DIRECTIONS

The Internet provides an integrated environment for accessing the wealth of information being digitized all over the globe. Not only is access possible, but multimedia capabilities offer a multisensory vehicle for presenting information in a compelling way. For instance, the active role of the user in navigating a virtual space and the instant response of the system provide an immersive experience.

A promising direction for the future lies in advanced services on the Intenet for untethered devices. The rapid growth of wireless communication and the explosion of knowledge highlights the need for mobile instruction and intelligent tutoring based on intelligent agents. A general architecture to support these functions has been developed.

To illustrate this type of application, the overall architecture for a wireless Web-based education system is depicted in Figure 9. The architecture embodies various aspects of the Web-based system for education and instruction. The approach incorporates software technologies such as the Wireless Application Protocol (WAP) for messaging with mobile devices such as cellular phones; Extended Markup Language (XML) for intelligent document processing; and Jini for automatic recognition and communication among hardware devices. In particular, WMLScript provides a convenient language for incorporating intelligent behavior in a tutoring agent.

The user should be able to interact with the system in natural language, including speech. Examples of pertinent queries from the user are as follows.

- What is a quadratic function?
- Show me a parabolic dish in 3-D format.
- Why is translation of an object a commutative operation, but rotation is not?
- Explain Keplers Law on sweeping equal areas of an ellipse in equal time.

The responses to such questions must be handled intelligently, depending on the expertise of the user as well as the needs of the moment. The period ahead will witness a variety of interesting developments in Web-based education.

A key direction for the future lies in autonomous learning capabilities. The maturation of machine learning capabilities offers a means of developing intelligent tutoring systems which can tailor a presentation not only to the basic level of expertise for a particular student, but also to his or her changing level of knowledge. The learning capabilities may be implemented using techniques such as case based reasoning, neural networks, and induction.

Such intelligence may be implemented as a kernel of a smart system for multimedia presentations. A framework for such intelligent systems has already been developed [11]. The framework for intelligent presentations has been adapted to software agents and their embodiment as icons [12]. The framework may also be tailored readily to the educational environment.

## 5. CONCLUSION

The rapid growth of Internet hosts and the accelerating pace of business highlights the need for online services based on intelligent agents. A general architecture to support these functions has been presented, and the associated society of agents defined. Moreover, certain advanced functionssuch as multi-strategy learninghave been implemented and validated.

## REFERENCES

[1] Fayyad, U. M., et al. *Advances in Knowledge Discovery and Data Mining*, AAAI Press / The MIT Press, 1996.

[2] Lieberman, H. Attaching Interface Agent Software to Applications, MIT, http://lieber.www.media.mit.edu/people/lieber/Lieberary/Attaching/Attaching.ps, 1996.

[3] Coen, H. M. SodaBot: A Software Agent Construction System, MIT AI Lab, http://www.cs.umbc.edu/agents/papers/sodabit.ps, 1996.

[4] Franklin, S., et al. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, Univ. of Memphis, ftp://ftp.msci.memphis.edu/comp/caat/agentprog.ps.z, 1996.

[5] Sheth, D. B. A Learning Approach to Personalized Information Filtering, MIT, ftp://media.mit.edu/pub/agents/interface-gents/news-filter.ps, 1994.

[6] Chess, D., et al. Itinerant Agent for Mobile Computing, IBM, http://www.cs.umbc.edu/kqml/papers/itinerent.ps, 1995.

[7] Finin T., et al. A Language and Protocol to Support Intelligent Agent Interoperability, http://www.cs.umbc.edu/kqml/papers/cecals.ps, 1992.

[8] McKay, P. D., et al. An Architecture for Information Agents, UMBC, http://www.cs.umbc. edu/kqml/papers/arpi.ps, 1996.

[9] Kim, S. H. *Designing Intelligence*, New York: Oxford Univ. Press, 1990.

[10] Hopfield, J. J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proc. Nat. Acad. Sciences USA*, v. 79(8), 1982: 2554-2558.

[11] Bordegoni, M., Faconti, G., Maybury, M.T., Rist, T., Ruggieri, S., Trahanias, P., Wilson M. A Standard Reference Model for Intelligent Multimedia Presentation Systems, http://www.dfki.

de/imedia/lidos/papers/cis97, 1997.

[12] Andre, E. WIP and PPP: A Comparison of two Multimedia Presentatin Systems in Terms of the Standard Reference Model. Computer Standards and Interfaces. v. 18 (6-7), pp. 555-564, 1997.

(Table 1) List of models, their associated variables, and performance results in the case of US interest rate prediction. The following abbreviations are used: NN for the neural network model, CBR for the case based reasoning model, and CBR+NN for the integrated model .

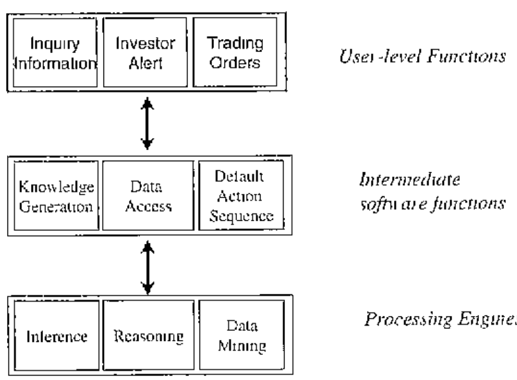
| Model | MAPE (%) | RMSE | Hit Rate (%) |
|---|---|---|---|
| NN_NOW | 3.366 | 0.216 | 74 |
| NN_STEP | 3.745 | 0.247 | 63 |
| CBR | 3.455 | 0.231 | 81 |
| CBR + NN | 3.166 | 0.208 | 74 |
| Random | 4.022 | 0.250 | 50 |

(Table 2) Pairwise t-tests for the differences in residuals for US interest rate prediction. The upper right triangle tabulates the t-tests based on the absolute percentage errors (APE) of residuals, with the significance level in parentheses. The lower left triangle lists the t-tests based on the absolute values of the deviations (AD).

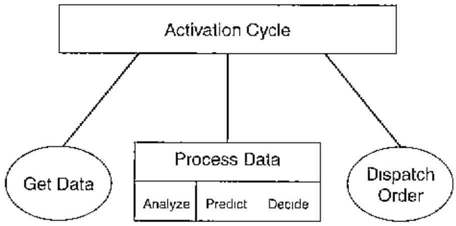
| | NN_NOW | NN_STEP | CBR | CBR+NN | RANDOM |
|---|---|---|---|---|---|
| NN_NOW | | -1.33 (0.190) | -0.32 (0.747) | 0.91 (0.367) | -3.17 (0.003) |
| NN_STEP | 1.17 (0.250) | | 0.91 (0.371) | 1.73 (0.091) | -0.82 (0.414) |
| CBR | 0.35 (0.725) | -0.75 (0.459) | | 0.88 (0.386) | -1.83 (0.074) |
| CBR+NN | -0.80 (0.430) | -1.61 (0.115) | -0.89 (0.380) | | -2.41 (0.020) |
| RANDOM | 3.25 (0.002) | 0.70 (0.485) | 1.61 (0.115) | 2.37 (0.023) | |

(Table 3) Pairwise test for proportions using the metric of hit rate in predicting US interest rates. The first entry in each cell presents the z-value while the significance level is shown in parentheses.

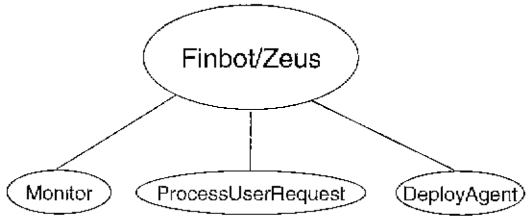
| | NN_STEP | CBR | CBR+NN | RANDOM |
|---|---|---|---|---|
| NN_NOW | 1.16 (0.246) | -0.78 (0.435) | 0.00 (1) | 2.34 (0.019) |
| NN_STEP | | -1.92 (0.055) | -1.16 (0.246) | 1.20 (0.230) |
| CBR | | | 0.78 (0.435) | 3.07 (0.002) |
| CBR+NN | | | | 2.34 (0.019) |



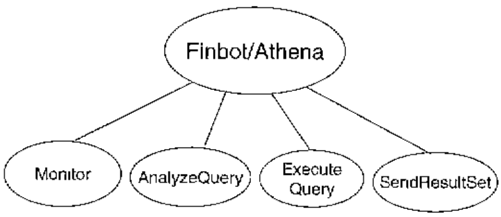

(Figure 1) A framework for system functionality and primary processing methods for a financial information system.



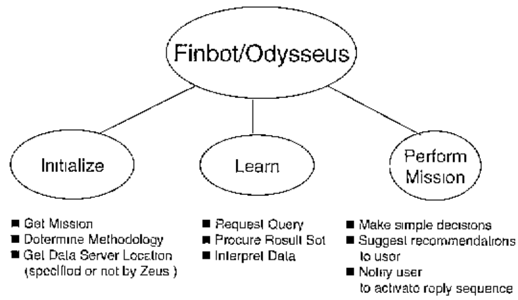

(Figure 2) Schematic of a processing cycle for a predictive agent.



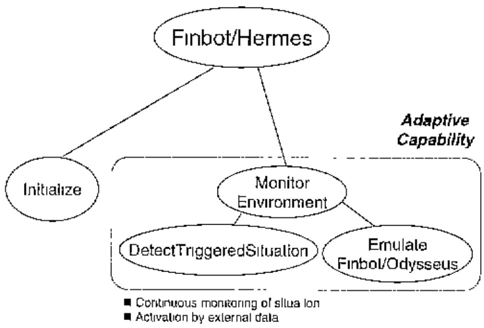

(Figure 3) Application server agent: Finbot/Zeus.



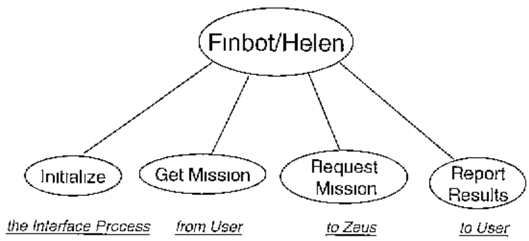

(Figure 4) Database server agent: Finbot/Athena.



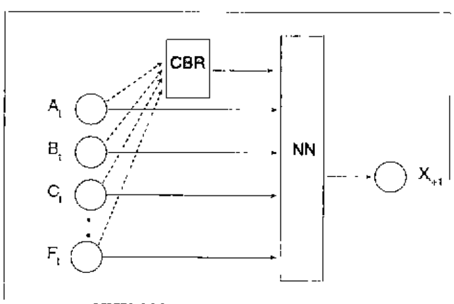

(Figure 5) Ad hoc mission agent: Finbot/Odysseus.




(Figure 6) Residential (permanent task oriented) agent: Finbot/Hermes.




(Figure 7) Client environment agent: Finbot/Helen.




(Figure 8) Architecture for the integrated methodology.