경영정보학연구
제7권 1호
1997년 6월

Extended Forecasts of a Stock Index using Learning Techniques: A Study of Predictive Granularity and Input Diversity

Steven H. Kim* and Dong-yun Lee**

The utility of learning techniques in investment analysis has been demonstrated in many areas, ranging from forecasting individual stocks to entire market indexes. To date, however, the application of artificial intelligence to financial forecasting has focused largely on short predictive horizons. Usually the forecast window is a single period ahead: if the input data involve daily observations, the forecast is for one day ahead; if monthly observations, then a month ahead; and so on. Thus far little work has been conducted on the efficacy of long-term prediction involving multiperiod forecasting.

This paper examines the impact of alternative procedures for extended prediction using knowledge discovery techniques. One dimension in the study involves temporal granularity: a single jump from the present period to the end of the forecast window versus a web of short-term forecasts involving a sequence of single-period predictions. Another parameter relates to the numerosity of input variables: a technical approach involving only lagged observations of the target variable versus a fundamental approach involving multiple variables. The dual possibilities along each of the granularity and numerosity dimensions entail a total of 4 models. These models are first evaluated using neural networks, then compared against a multi-input jump model using case based reasoning. The computational models are examined in the context of forecasting the S&P 500 index.

---

* KAIST 테크노경영대학원 부교수
** KAIST 테크노경영대학원 경영대학원 경영공학석사과정

이 논문은 1997년 3월 14일 접수하여 1997년 7월 3일 게재확정되었습니다.

MOTIVATION

Experience with artificial intelligence applications, especially since the early 1980s, has demonstrated the utility of learning systems for financial prediction and investment analysis. More specifically, knowledge-based systems may be employed to automate many routine decision making tasks, to serve as a substrate to combine a multiplicity of methodologies, and to improve system performance by learning to identify the utility of different combinations of techniques.

The factors behind investment performance include both macroeconomic and microeconomic variables. A systematic approach to knowledge discovery for investment analysis must therefore be able to accommodate these different types of information. To this end, the techniques of artificial intelligence and statistics may be combined to yield a synergistic methodology.

To date, however, the application of artificial intelligence to financial forecasting has focused largely on short predictive horizons. Usually the forecast window has been a single period ahead: if the input data involve daily observations, the forecast is for one day ahead; if monthly observations, then a month ahead. The efficacy of long-term financial forecasts involving multiperiod prediction is largely an unexplored field.

This paper examines the impact of alternative procedures for extended prediction using knowledge discovery techniques. One dimension in the study involves temporal granularity: a single jump from the present to the end of the forecast window versus a web of short-term forecasts involving a sequence of single-period predictions. Another parameter relates to the numerosity of input variables: a technical approach involving only lagged observations of the target variable versus a fundamental approach involving multiple variables. The dual possibilities along each of the granularity and numerosity dimensions entail a total of 4 models. These models are first evaluated using neural networks. Next, the 4 neural models are compared against a multi-input jump model using case based reasoning.

BACKGROUND

The advantages of combining multiple techniques to yield synergism for discovery and prediction have been recognized in the past [Kaufman et al., 1991; Kim, 1994a, 1994b; etc.]. An example lies in the call for a juxtaposition of spectral analysis and temporal regression for studies in the social sciences [Gottman, 1981].

A versatile approach to self-organization lies in neural networks [Anderson and Rosenfeld, 1988; Grossberg, 1974, 1976; Haken, 1988; Hebb, 1949; Hopfield, 1982; Kohonen, 1984; Rosenblatt, 1962; Rumelhart et al., 1986]. Neural nets are characterized by learning capability, the ability to improve performance over time. A closely related feature is that of generalization, relating to the recognition of new objects which are similar but not identical to previous ones. An additional characteristic relates to graceful degradation: the network fails gradually rather than catastrophically when it suffers partial damage.

To date, however, artificial networks have been burdened with a major limitation: protracted training periods. Hundreds or thousands of trials are usually required for satisfactory performance in various tasks. The time and effort required for training have hindered their widespread application to practical domains [Kim, 1994a, 1994b; Shibazaki and Kim, 1991]. To fully exploit the promise of neural nets by emulating the real-time responsiveness of biological systems, training time must be reduced dramatically.

One way to reduce training time and also enhance predictive power is to preprocess the data. In this study, input data streams are transformed into stationary variables and also culled for their predictive power.

METHODOLOGY

The learning techniques employed in this study relate to neural nets and case based reasoning. As indicated in the previous section, neural networks have been used extensively over the past decade for predicting financial markets. The application of case reasoning to forecasting, however, is an area with little prior history [Kim, 1996].

Prior Knowledge through Case Reasoning. A learning system should make increasingly useful decisions as it accumulates experience. This is the express goal of the work in case-based reasoning (CBR).

Perhaps the most important advantage of CBR is the affinity to human learning. People take account of observations and utilize them for future decision making. Often the extrapolation to new situations is ad hoc, as in modifying a set of evaluation criteria for the silicon-based computing industry into a similar one for the emerging vendors of photonic hardware. In other cases, the extrapolation is more formal and takes the form of inductive propositions such as formulas, principles, laws, or rules of thumb.

Related to the affinity of CBR to human learning is the ease of enhancing system performance. More generally, the knowledge in a particular domain can be stored in formats which are conventional for that domain. For instance, a knowledge base for balancing stocks, bonds, and other instruments in an investment portfolio can store the information about previous financial strategies in the cognitive format used by human analysts.

This is in contrast to other knowledge-level representations such as production rules, in which the system developer is required to extricate the pertinent decision rules used by a human. In general, the problem of knowledge extraction is further compounded by the fact that people often perform admirably in various domains without using - or even being aware of the existence of - any such decision rules.

The CBR methodology can be effective even if the knowledge base is imperfect. Certain techniques of automated learning, such as explanation-based learning, work well only if a strong domain theory exists. In contrast, case reasoning can use many examples to overcome the gaps in a weak domain theory while still taking advantage of the domain theory [Porter et al., 1990]. CBR can also be used when the descriptions of the cases, as well as the domain theory, are incomplete. A further advantage of CBR is the relative ease of combining techniques with other approaches such as production rules [Golding and Rosenbloom, 1991]. An example of such compatibility is a system which uses case reasoning to solve problems whenever possible; otherwise it resorts to heuristics to decompose a problem into a simple one.

Retrieving Precedents. Case reasoning requires the retrieval of past experience in the form of cases. In this task, two types of difficulties can arise. The *matching* problem refers to the task of associating a new problem to pertinent prior cases. A key issue lies in retrieving prior cases which are similar to the new problem in substantive rather than superficial ways. This relates in part to the issue of *indexing*, which deals with the organization of the case base.

Matching Problem. Problem solving in any arena is dedicated to the attainment of a goal. To this end, the decision maker must find prior cases which resolve the specified or comparable objectives, rather than those that match only surface features having little impact on the effectiveness of the solution. For instance, two portfolios may be of similar dollar amount and contain a number of shares in common; but one is directed toward high income while the other seeks stable growth. Consequently, a CBR system must search through the base of previous cases by first attempting to find solutions that meet the primary design goals, and subsequently examine them against secondary objectives.

The matching problem can be addressed in a number of ways. The default scheme is to perform an exhaustive search through the case base each time a new problem arises. However, system performance can be degraded by such a tedious approach.

A more systematic way is for a human to identify the relevant prior cases. Unfortunately, this technique requires continuing human intervention if a system is to improve its performance over time.

To automate the task of matching in CBR, previous cases can be organized in some fashion to enable the rapid identification of potentially relevant cases. To this end, previous problems and solutions can be indexed by their key attributes and the features which distinguish them from other cases.

Indexing Problem. The indexing problem refers to the task of storing cases for effective and efficient retrieval. In terms of efficacy, the subissues are *accuracy* - finding only relevant cases - and *completeness* - identifying all relevant cases.

In general the prior cases retrieved by case reasoning will match the required solution only imperfectly. In particular, the source cases may fail to fulfill some of the requisite objectives. At this point, an analogy can be formed between the functionality of the precedent solutions and the goals of the current problem. The prior solutions may then be modified to eliminate or circumvent the limitations. Then a process of iterative refinement can be employed to adapt an old solution to the new problem context [Kim, 1990]. Whether or not analogy is used, an organization may be imposed on the case base through the use of clustering techniques [Kim and Novick, 1993]. In this way, a target case may be readily accommodated into an existing case base.

Prediction through Regression. To date, work on econometric analysis has often relied on regression models. This approach is illustrated by the widespread practice of forecasting macroeconomic performance through multivariate regression. More recently, knowledge processing techniques such as neural nets have been declared as a generalization of classical regression to the nonlinear arena.

Expert system techniques have been coupled with statistical software to produce intelligent packages, including those for regression, clustering, and multivariate analysis of variance. A case in point is REX, a package to assist individuals with limited experience in the use of regression [Gale, 1986]. The system checks assumptions underlying the statistical models and alerts the user to violations, while suggesting remedial measures when it can.

Multiperiod Prediction. Prediction over an extended horizon may be classified into two types. The *jump* approach involves a matching of the current situation with one or more previous contexts, then a determination of the consequence after a comparable interval in the precedent situation(s).

More specifically let $\mathbf{x}(t)$ be a vector of values which defines the situation at the present time t. Further, a forecast is required for k periods ahead. To this end, the vector $\mathbf{x}(t)$ is matched against a case $\mathbf{x}(t')$ at time t' in the past. Then the forecast $\mathbf{f}(t + k)$ for k periods in the future is the k^{th} successor to $\mathbf{x}(t')$; in other words, $\mathbf{f}(t + k) \equiv \mathbf{x}(t' + k)$.

This matching process can be generalized by finding the N cases from the past which most resemble the present situation [Farmer and Sidorowich, 1987]. Then the forecast $\mathbf{f}(t + k)$ can be obtained as a weighted sum or some other combination of the historical evolutions; namely, $\mathbf{x}(t_1 + k)$, ..., $\mathbf{x}(t_N + k)$.

A second approach to extended prediction is through a series of iterated forecasts. The current state $\mathbf{x}(t)$ is used to predict the subsequent period, yielding a forecast $\mathbf{f}(t + 1)$. This forecast is added to the database and regarded as if it were an actual observation. Then the same procedure is applied while regarding the state $\mathbf{x}(t + 1)$ as the current state. The procedure is iterated k times if a k-period forecast is required.

When the state vector $\mathbf{x}(t)$ involves more than a single time series, the forecast involves multiple variables. The sequence of multiple variables over the course of k iterations can be regarded as a web of forecasts. Hence we refer to a multivariate, multiperiod forecast as a *web* prediction.

A similar strategy can be followed when the input vector is a sequence of lagged observations of a *single* variable. In that case, the current state can be defined as a sequence of consecutive data values. Suppose that the state vector has n components. Given a sequence of observations $u_1, u_2, \ldots, u_t$ the state vector at time t can be defined as $x(t) \equiv (u_{t-n+1}, \ldots, u_{t-1}, u_t)^T$. Figure 1 illustrates the approach when the state vector has $n = 3$ components.

An example of a predictive model to generate a web forecast is given in Figure 2. In particular, the diagram depicts a neural network using vectors as both inputs and outputs.

CASE STUDY

The case study involves the prediction of the S&P 500 index (SPX). For univariate inputs, only lagged values of SPX are utilized.

For multivariate inputs or outputs, five other variables play a role. These variables relate to federal funds (FF), money stock (M2), housing starts (HS), industrial production (IP), and the consumer price index (CPI). Further details concerning the variables are given in Table 1.

A plot of the target variable SPX over the entire period is given in Figure 3. All variables were partially straightened out through a log transform (L), then rendered dimensionless through a standardization (Z). The resulting variables are shown in Figure 4.

Selection of Variables. The autocorrelation of ZLSPX as a function of lag is given in Figure 5(a). The diagonal linkage, or lagged cross-correlation, of the other variables with ZLSPX are portrayed on the other charts in Figure 5.

Since housing starts, ZLHS, had only weak correlation with ZLSPX at all leads from -12 to +12, it was excluded from further consideration. For every other variable, only the lag exhibiting the highest correlation with ZLSPX was included in the multivariate models.

To explain the selected lag of each variable exhibiting the highest correlation with ZLSPX, a change in viewpoint is helpful. For a web prediction, each iteration involves a forecast window of length $k = 1$. Then only lead values of $L \leq -1$ may be considered; that is, all data up to the immediate past.

To illustrate, consider federal funds. For a web prediction, the candidate leads are FF(-12) through FF(-1). Among these, FF(-1) shows the highest correlation and is therefore selected.

To produce a forecast of $k = 1$ period ahead, however, the "current" time t has to be shifted forward by $k = 1$ period. In that case, the current value of federal funds, FF(0), is the appropriate value to use in generating a forecast for the next period, $t + 1$.

Consider now a jump prediction with forecast window size $k = 6$. Then any time series with lead $L \leq -6$ is a candidate input variable. For federal funds, the highest correlation among the data series FF(-12) through FF(-6) is exhibited by FF(-12). During the predictive procedure, the appropriate series to employ is FF(-6), since that represents a lag of 12 from the forecast destination at period $t + 6$. This reasoning explains the apparent discrepancy in lag values for the variables in Figure 5 versus those in Table 2.

Identification of Models. As explained in an earlier section, the dimension of temporal granularity involves a jump or a web prediction. A second dimension of the study relates to the number of variables: a univariate versus a multivariate model. The resulting 4 combinations have all been implemented on neural networks. In addition, a multivariate jump model using case base reasoning was tested. The 5 models and their performance are highlighted in Table 2.

The performance data for the neural models are presented in an array for ready interpretation in Table 3. The results indicate that jump prediction supersedes a web forecast. Moreover, multivariate inputs yield superior performance over univariate inputs. These conclusions are significant at $p < 0.03$, as shown in Table 4.

Finally, a multivariate jump model using case reasoning (MuJCBR) performs better than the best neural model. This result is mildly significant at level $p < 0.08$, as shown in Table 5.

DISCUSSION

It comes as no surprise that a fundamental analysis incorporating multiple variables performs better than a purely technical approach involving a single variable. A stock index is a composite measure of stock prices which in turn reflect the aggregate view of investors concerning corporate performance. Corporate profitability in turn depends in part on general economic conditions. Under the assumption of bounded rationality, there is no reason to believe in a perfectly efficient market. More specifically, stock prices need not embody all pertinent information.

Another anticipated result is the mild superiority of case reasoning over neural nets. The temporal CBR methodology has often dominated neural techniques, although its performance does depend on the particular application [Kim and Oh, 1996].

The major surprise relates to the superiority of a single jump over a web projection. Conventional econometric models tend to rely on a web approach to forecasting the economic variables rather than utilizing several jump models which are distinguished by the size of their respective forecast windows.

An analogous development has occurred in the field of engineering. Finite element analysis has at times been declared as the greatest triumph of computational engineering. This method relies on a web or mesh of locations superimposed on a physical object to calculate such parameters as temperature or stress as a function of location. The popularity of web techniques in both the socioeconomic and technological spheres would suggest its utility in the financial arena as well.

It may be argued that a series of iterated forecasts will produce a large cumulative error, thereby limiting the accuracy of the web approach. On the other hand, a jump approach will magnify any imperfection latent in the model at the start of the forecasting exercise.

These arguments can be made more precise as follows. At the current time t, the existing situation $\mathbf{x}(t)$ is matched against a set of precedent cases or neighbors $\mathbf{P}(t) \equiv \{\mathbf{x}(t_i)\}$. When case reasoning is used, the set $\mathbf{P}(t)$ is identified explicitly at time t. For a neural approach, the appropriate set of precedents is embodied implicitly in the arc weights and other parameters of the network.

Next, the kth successors to the neighbors, which we denote as $\mathbf{P}(t + k) \equiv \{\mathbf{x}(t_i + k)\}$ are aggregated through a weighted sum to yield the forecast $\mathbf{f}(t + k)$ for k periods hence.

Let ε_1 denotes the error introduced in the first iteration, $k = 1$. This error arises from several sources:

- In general, none of the neighbors $\{\mathbf{x}(t_i)\}$ will match the current situation $\mathbf{x}(t)$ exactly. Hence each successor $\mathbf{x}(t_i + k)$ should resemble the forecast $\mathbf{f}(t_i + \mathrm{k})$ only imperfectly.
- Suppose a perfect match exists between the neighbors and the current vector according to the accuracy available in all the data points. However, the data are only imperfect observations of the real world; for instance, GDP is a metric of economic activity, but it is only one imperfect measure of the state of the economy. For these reasons, two points $\mathbf{x}(t)$ and $\mathbf{x}(T)$ in the state space

may be identical numerically but actually represent the tops of very different icebergs. In this situation, **x**(t) and **x**(T) may well move forward in divergent directions.

Under a linear model of error propagation, the error ε_1 associated with the first step will be magnified k times for a jump forecast of length k:

$$\varepsilon^* = k\varepsilon_1$$

Consider now a web forecast. Let ε_i be the error associated with iteration i. Under an additive model of error accumulation, the total error after k steps is

$$\varepsilon_a = \sum_{i=1}^{k} \varepsilon_i$$

If the cumulative error grows geometrically rather than additively, the error due to a web forecast is

$$\varepsilon_g = \prod_{i=1}^{k} \varepsilon_i$$

There is no reason to suppose that ε^* should always be smaller or greater than either ε_a or ε_g

The preceding argument has relied on a linear versus an additive or multiplicative model of error propagation. However, a nonlinear assumption for error accumulation is almost certainly more appropriate for the chaotic variables arising in financial economics. With the nonlinear assumption, there is even less reason to expect a consistent relationship between the error ε^*due to a jump and the errors ε_a or ε_g due to a web prediction. A promising direction for the future is to investigate these arguments in greater detail and for additional domains.

CLOSURE

This study has explored a number of alternative ways to deploy learning techniques to generate long-term forecasts. As expected, a fundamental approach using multiple variables yielded better performance than a purely technical approach involving a single input variable.

Contrary to expectations, however, the use of jump prediction outperformed a web prediction. This result may be due to the accumulation of errors during the iterative process of providing single-step forecasts. However, it is not clear that the outcome should hold for all application domains.

An obvious direction for future work is to replicate the experiments for other target variables and for differing perio- city of the data series (e.g. daily or weekly data). These and similar investigations will provide a better understanding of the nature of financial markets as well as practical tools for risk management.

〈REFERENCES〉

Anderson, J.A. and Rosenfeld, E., eds., *Neurocomputing,* Cambridge, MA: MIT Press, 1988.

Farmer, J.D. and J.J. Sidorowich,

"Predicting Chaotic Time Series," *Phys. Rev. Lett.*, v. 59, 1987, pp. 845-848.

Gale, W.A., "REX Review," In W.A. Gale, ed., *Artificial Intelligence and Statistics*, Reading, MA: Addison-Wesley, 1986, pp. 173-227.

Golding, A.R. and P.S. Rosenbloom, "Improving Rule-Based Systems through Case-Based Reasoning," *Proc. AAAI-91*, v. 1, 1991, pp. 22-7.

Gottman, J.M., *Time-Series Analysis*, New York: Cambridge Univ. Press, 1981.

Grossberg, S., "Adaptive Pattern Classification and Universal Recoding: Part I. Parallel Development and Coding of Neural Feature Detectors," *Biological Cybernetics*, v. 23(3), 1976, pp. 121-134.

Grossberg, S., "Classical and Instrumental Learning by Neural Networks," *Progress in Theoretical Biology*, v. 3. NY: Academic Press, 1974, pp. 51-141.

Haken, H., ed., *Neural and Synergetic Computers*, New York: Springer-Verlag, 1988.

Hebb, D. O., *The Organization of Behavior*, NY: Wiley, 1949.

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

Kim, S. H., *Essence of Creativity*, NY: Oxford U. Press, 1990.

Kim, S. H., *Learning and Coordination*, Dordrecht, Netherlands: Kluwer, 1994a.

Kim, S.H., "Accelerating Neural Networks through Declarative Knowledge," *Proc. '94 Japan/Korea Joint Conference on Expert Systems*, Tokyo, Mar. 1994b, pp. 197-202.

Kim, S.H., "Integrated Knowledge Discovery for Investment Prediction: Critical Issues and System Design Implications," Work Paper, Graduate School of Management, KAIST, 1996.

Kim, S.H. and M.B. Novick, "Using Clustering Techniques to Support Case Reasoning," *International J. of Computer Applications in Technology*, v. 6(2/3), 1993, pp. 57-73.

Kim, S.H and Oh, H.S., "A Multistrategy Learning System to Support Predictive Decision Making," *Proc. Fall Conf. of Korean Financial Management Assoc.*, Seoul, Oct., 1996.

Kohonen, T., *Self-Organization and Associative Memory*, NY: Springer-Verlag, 1984.

Porter, B.W., R. Bareiss, and R.C. Holte, "Concept Learning and Heuristic Classification in Weak-Theory Domains," *Artificial Intelligence*, v. 45(1-2), 1990, pp. 229-263.

Rosenblatt, F., *Principles of Neurodynamics*, NY: Spartan, 1962.

Rumelhart, D.E., G.E. Hinton, R.J. Williams, "Learning Internal Representations by Back Propagation," In D.E. Rumelhart, J.L. McClelland and the PDP Research Group, *Parallel Distributed Processing,* v.1: Foundations, Cambridge, MA: M.I.T. Press, 1986, pp. 318-362.

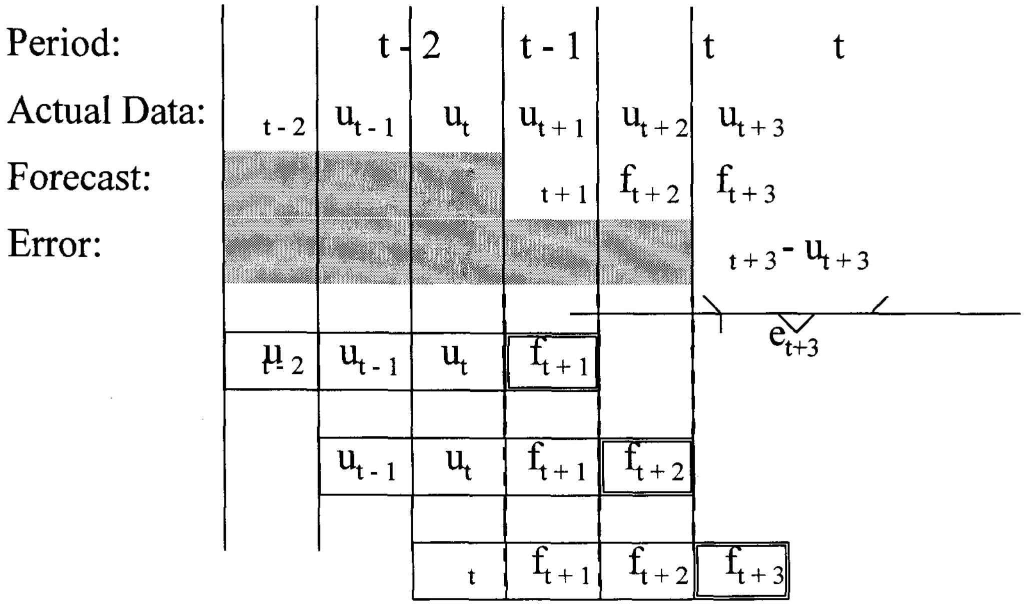


Figure 1. Strategy for univariate, multiperiod web prediction for the case of 1 variable and k = 3 periods ahead. In this illustration, the state vector $\mathbf{x}(t)$ at time t involves 3 consecutive values. For instance, the state at time (t+1) is given by $\mathbf{x}(t+1) = (u_{t-1}, u_t, f_{t+1})$. The first error to be considered is e_{t+3} at period (t + 3).

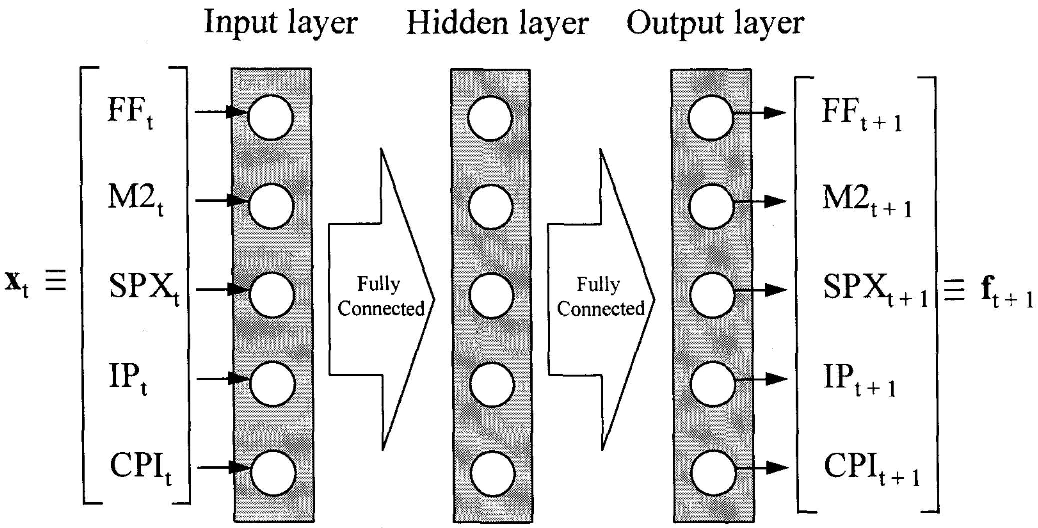


Figure 2. Strategy for multivariate prediction. The neural network architecture uses vectors as both inputs and outputs. Each iteration involves a forecast **f** for k = 1 period ahead. In web prediction, the vector of forecasts for the next period is added to the database along with existing data; for a forecast horizon of k periods, the procedure is iterated k times.

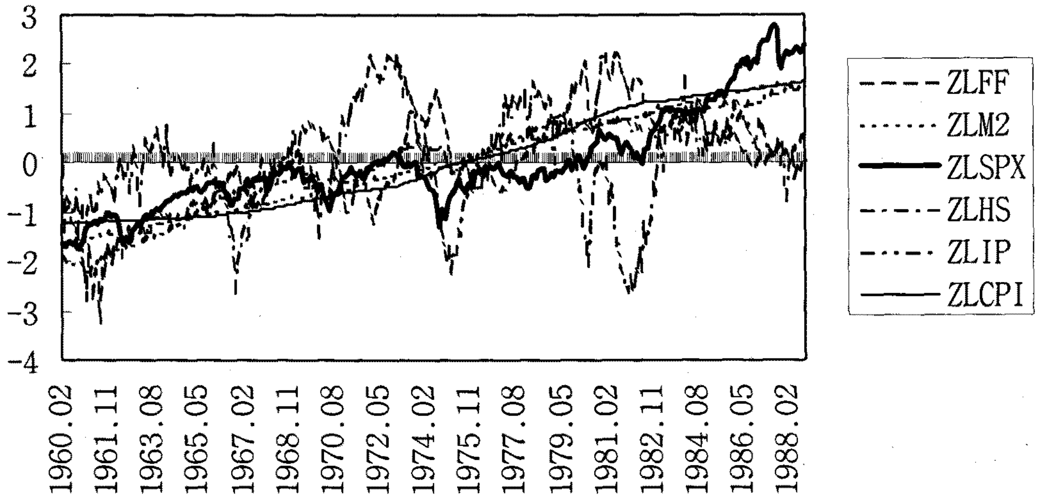


Figure 3. Time series of S&P 500 index: 1960.2 - 1992.12.

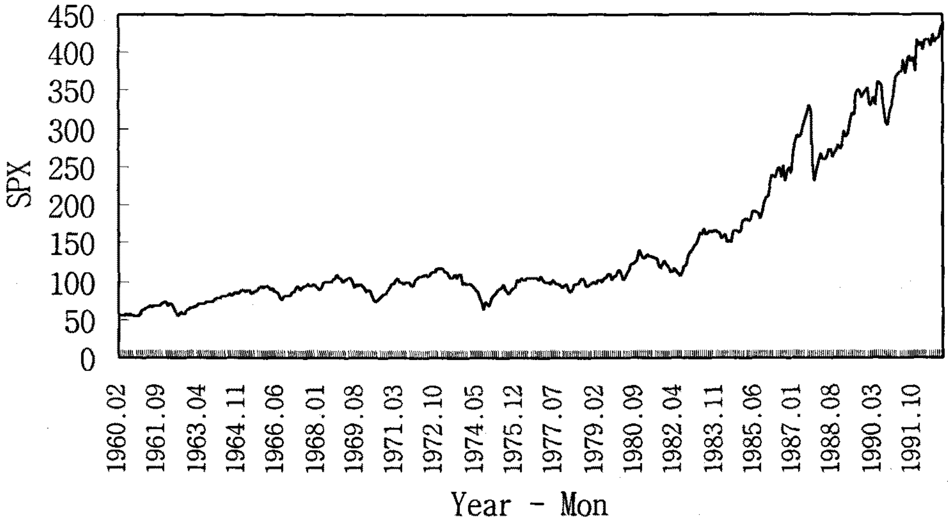
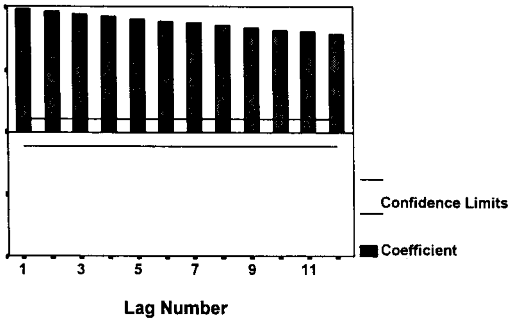
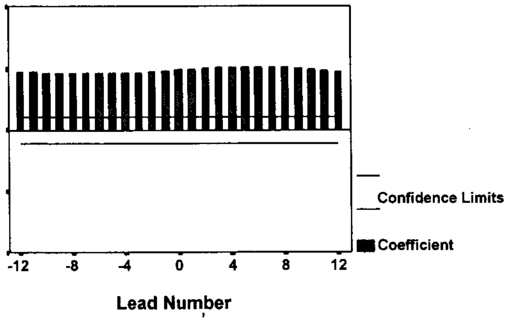


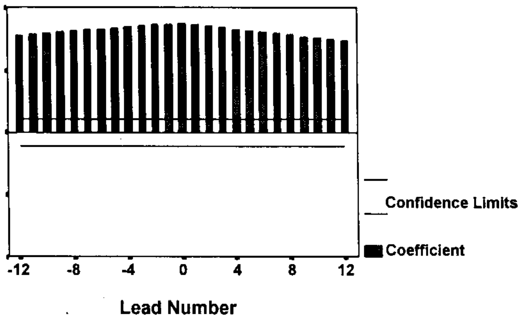
Figure 4. Time series plot of transformed variables. For instance, ZLFF denotes the standardized form of the logarithm of federal funds (FF).



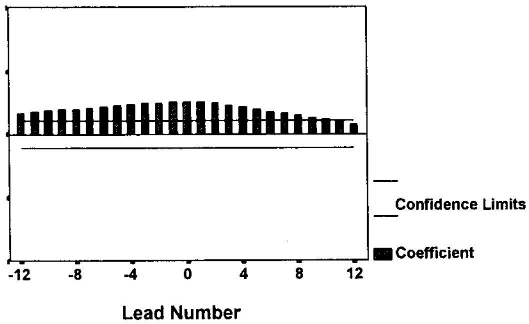
(a) Autocorrelation of ZLSPX



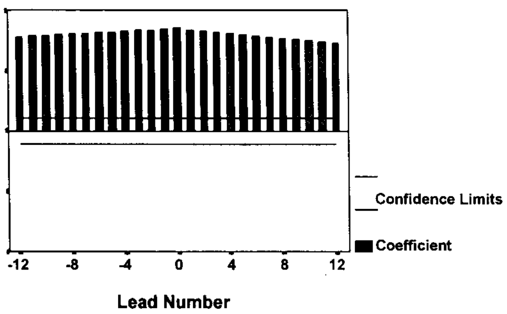
(b) Diagonal correlation of SPX with federal funds FF: ZLSPX vs. ZLFF.



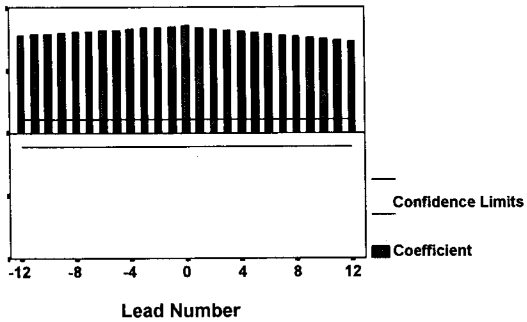
(c) Diagonal correlation of SPX with M2: ZLSPX vs. ZLM2



(d) Diagonal correlation of SPX with .HS: ZLSPX vs. ZLHS.



(e) Diagonal correlation of SPX with IP: ZLSPX vs. ZLIP



(f) Diagonal correlation of SPX with CPI: ZLSPX vs. ZLCPI.

Figure 5. Correlation diagrams between SPX and assorted variables: variation of autocorrelation function (ACF) or cross-correlation function (CCF) with the number of leads. Due to low correlations, ZLHS is dropped from further consideration. For every other variables, the lag (i.e. lead L 0) exhibiting the highest correlation with ZLSPX is selected. The appropriate value of L depends on whether a jump or web model is used, as explained in the text.

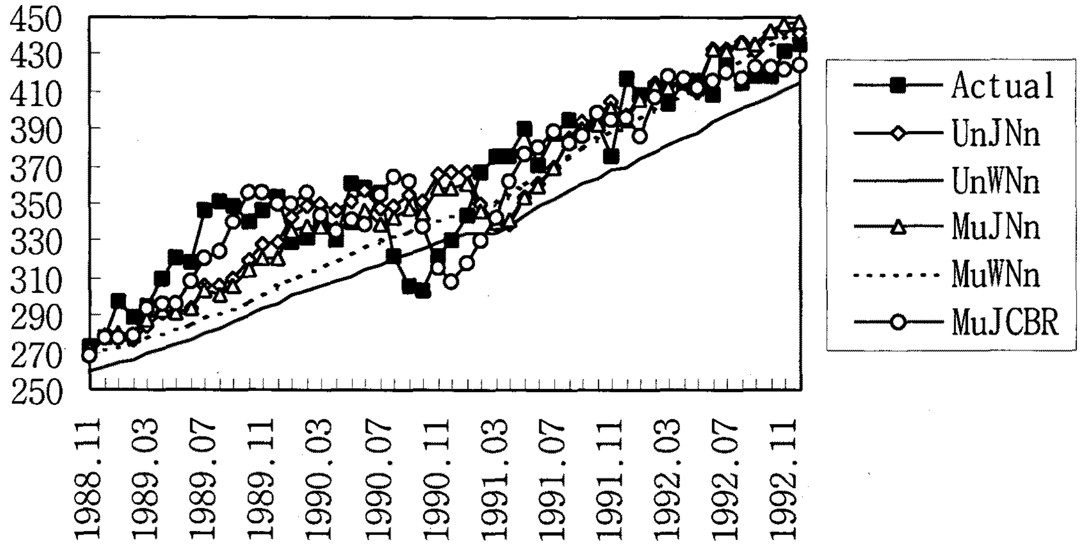


Figure 6. Predicted and actual values over the test period.

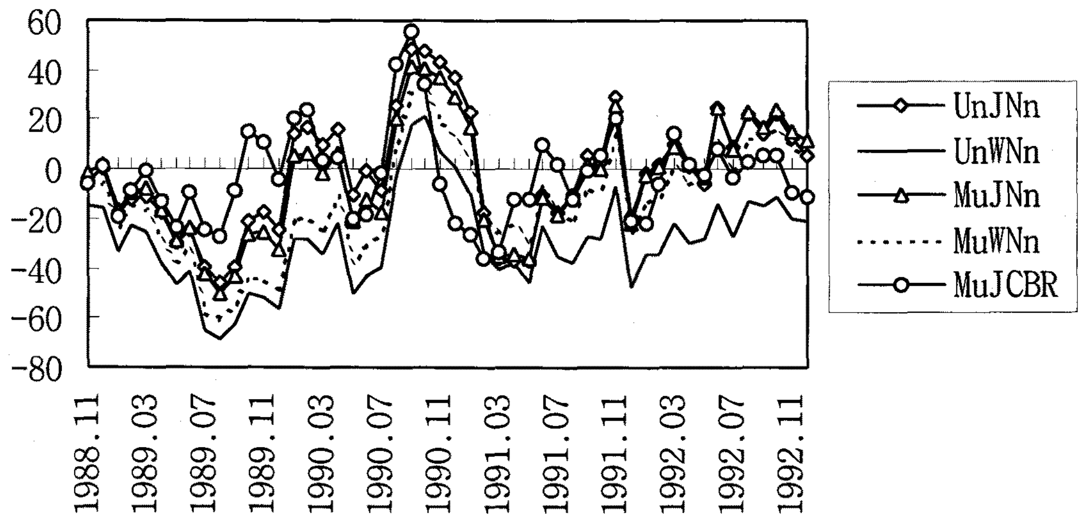


Figure 7. Residuals of the forecast values from Figure 6.

Table 1. *Description of original variables. The entire dataset consisted of 395 monthly observations* from Feb. 1960 to Dec. 1992. The training data were monthly observations from Feb. 1960 to Oct. 1988; and the test set from Nov. 1988 to Dec. 1992.

Label	Name	Description
SPX	Standard & Poors 500	Daily closing price on the last day of month.
FF	Federal Funds	Average of daily rates during the month.
M2	Money Stock	Seasonally adjusted. Billions of dollars.
HS	Housing Starts	New starts privately owned housing units. Seasonally adjusted.
IP	Industrial Production	Seasonally adjusted index. 1987 = 100.
CPI	Consumer Price index	All urban consumers. Not seasonally adjusted.

Table 2. List of models, their associated variables, and performance results. The following abbreviations are used: the number of distinct input variables is denoted Un for Univariate or Mu for Multivariate; the granularity of prediction is J for a single jump or W for a web of estimates; the learning methods are Nn for neural network or CBR for case based reasoning. The error criterion is the mean absolute percent error (MAPE).

Model	Input Variables	Output Variables	Target Variable	MAPE
UnJNn	SPX[0], SPX[-1], SPX[-2], SPX[-3], SPX[-4]	SPX[6]	SPX[6]	5.52
UnWNn	SPX[0], SPX[-1], SPX[-2], PX[-3], SPX[-4]	SPX[6]	SPX[6]	8.56
MuJNn	FF[-6], M2[0], SPX[0], IP[0], CPI[0]	SPX[6]	SPX[6]	5.31
MuWNn	FF[0], M2[0], SPX[0], IP[0], CPI[0]	FF[6], M2[6], SPX[6], IP[6], CPI[6]	SPX[6]	6.06
MuJCBR	FF[0], M2[0], SPX[0], IP[0], CPI[0]	SPX[6]	SPX[6]	4.09

Table 3. A partial table of results for neural network models from Table 2.

Projection Input Vector	Jump	Web
Univariate	5.52	8.56
Multivariate	5.31	6.06

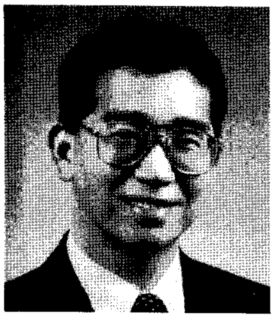
Table 4. Two-way ANOVA for the data in Table 3.

Source of Variation	Sum of Squares	DF	Mean Square	F	Sig of F
Main Effects	270.596	2	135.298	7.385	.001
Input_Vector	90.896	1	90.896	4.961	.027
Projection	179.700	1	179.700	9.809	.002
2-Way Interactions	65.530	1	65.530	3.577	.060
Input_vectorProjection	65.530	1	65.530	3.577	.060
Explained	336.126	3	112.042	6.116	.001
Residual	3590.769	196	18.320		
Total	3926.895	199	19.733		

Table 5. Pairwise t-tests for the difference of means, based on absolute values of residuals. The CBR methodology is tested against the best neural model, MuJNn; the case reasoning approach appears to be mildly superior at $p < 0.08$.

Pair	t-value	Significance
UnJNn vs. UnWNn	-3.82	0.000
MuJNn vs. MuWNn	-0.85	0.400
UnJNn vs. MuJNn	0.19	0.849
MuJNn vs. MuJCBR	1.79	0.077

◆ 저자소개 ◆



Steven H. Kim

Columbia University에서 1977년에 기계공학 학사, MIT에서 1978년에 기계공학 석사, Columbia University에서 1980년에 Operations Research 석사, MIT에서 1983년에 경영학 석사, MIT에서 1985년에 경영학 및 기계공학 박사를 취득하였다. 1985년 7월부터 1991년 6월까지 MIT 기계공학과에서 조교수로 근무하였고, 1995년 8월부터 현재까지 KAIST 테크노경영대학원 부교수로 재직하고 있다. 주요 연구분야는 Knowledge Discovery, Predictive Systems, Artificial Intelligence, Learning Systems 등이다.

이동윤

서울대학교에서 1994년에 조선해양공학 학사를 취득하고 현재 KAIST 테크노경영대학원 경영공학 석사과정에 재학중이다. 주요 연구분야는 전문가시스템, 자동학습 등이다.