A Two Tiered Cognitive Model for the Forecasting of Time Series Data

Andrew J. Novobilski
The University of Texas at Arlington
c/o
P.O. Box 4328
Gulf Shores, AL 36547
(334) 948-8373
andyn@novotech.com

Farhad A. Kamangar, Ph.D.
The University of Texas at Arlington
Computer Science & Engineering Dept.
Arlington, TX 76019
(817) 272-3617
kamangar@cse.uta.edu

This paper describes two mutually enhancing technologies that will be used to evolve Bayesian network based forecasting models;. human/artificial cognition and Bayesian networks. A two tiered representation is introduced which mimics the way the human brain is thought to organize itself. This representation can be manipulated using genetic programming techniques to extract both attributes and organization of a Bayesian Network that models the underlying stochastic process for time series data. Experimental results are presented that demonstrate the effectiveness of the method in forecasting daily prices of stock issues.

1. Introduction

Time series forecasting is a well established area of study supported by various methods for predicting a future value from historical information. Traditionally the domain of mathematicians, research is now being carried out by practitioners in other disciplines looking fore innovative forecasting methods. This interest in alternative methods is being led by efforts to utilize new information presented in areas of cognitive research and artificial intelligence.

To be successful in model prediction, it is necessary to select a model that provides both a comprehensive structure capable of expressing the domain being explored, while still having a computationally tractable representation. Network structured models based on concepts found in cognitive research such as neural networks and belief networks are being actively looked at by the artificial intelligence community as the underpinning for new applications in data mining.

Neural Networks have been used extensively to forecast the performance of products traded in the stock market[1]. For example, Refenes, et. al[2] describe a set of Neural Net based applications used to model the behavior of financial systems that incorporate both objective (time series) data and subjective (expected human trader response) information. Although effective, these frameworks rely upon a-priori domain knowledge from an expert to constrain the maximum number and type of exogenous variables used to represent the non-time series portion of the model. It is expected that a domain expert will select which non-time series variables to exclude based primarily upon their beliefs of the behavioral aspects of the traders working in the market. In essence, the authors made assumptions concerning the rational actions of other agents in the domain in which they were modeling that were neither based on observation nor could be updated during the execution of their model.

Another way to approach the problem of understanding the solution model is to use a Bayesian network structure that offers an intuitive explanation based on its contents and organization. A Bayesian network, also known as a belief network, causal network, and influence diagram, is a graphical modeling language for representing uncertain relationships[3]. A Bayesian network is a directed acyclic graph with nodes representing the attributes of the model and directed links representing a causal relationship between parent and child. Together, this information represents the dependence between variables and gives a concise specification of the joint probability of the model[4]. Unlike neural networks, it provides a white box approach to representing relationships that exist within the domain being modeled and can handle inferencing in the absence of complete information.

Several different forecasting frameworks have been constructed around Bayesian networks. For example, Abramson's ARCO1[5] was used to forecast pricing in the oil market using a belief network representing a one year world oil price model. ARCO1 performed inferencing using Monte Carlo simulation and was able to model both quantitative knowledge of economic variables and qualitative knowledge if subjective policies that affected the production and consumption of oil.

A second framework is presented by Dagum et. al. uses a synthesis of Bayesian belief network models and classical time-series analysis[6]. Their framework, known as the Dynamic Network Model provides the ability to consider complex non-linear relationships between variables without an intractable increase in computational complexity. They use an additive model approach to reduce the computation complexity by separating the conditional links between nodes based on slices of time. The authors present experimental results collected from using a Dynamic Network Model to model a multi-variate diagnostic process used to monitor patients in critical care situations.

Accepting the use of a network based model within a forecasting framework, one area of particular interest is the way in which both the structure and values of network representations can be learned from the data alone. To this end, a significant amount of work has been done on learning the network structure given an identified set of attributes of the data to be modeled. However, in the case of forecasting a time series, each past value becomes a potential attribute for use in the network model describing the process that produced the time series. This increase in computational complexity, plus the ability to enhance the forecast model using data sources external to the time series being forecast, underscores the need for an autonomous method for identifying the forecast model.

2. - Human and Artificial Cognition

There is escalating interest in the AI community in the use of hybrid techniques that combine classical knowledge representation with neurological based enhancements. For example, Uhr and Honavar[7] detail a system for performing constructive learning utilizing Knowledge based networks. In order to appreciate the impact of their work, it is necessary to consider several factors related to the understanding of computational technique to be applied to solving a problem in learning. Specifically, what are the differences between the two above mentioned techniques and in what framework can they be combined?

In an earlier work[8], Honavar states that "the dichotomy between SAI [Symbolic Artificial Intelligence] and NANN [Numeric Artificial Neural Networks] is more perceived than real." He goes on to strengthen his position by stating that "the fundamental working hypothesis that has guided most of the research in artificial intelligence as well as the information processing school of psychology is rather simply stated: "*Cognition, or thought processes can, at some level, be modeled by computation.*" He further points out that SAI and NANN rely on equivalent models of computation; i.e. the implementation of underlying models where proven to be exactly equivalent to the Turing Machine[9].

If these statements hold true, then it would be reasonable to expect that any problem capable of being solved using a SAI approach would also be solvable using a NANN approach. Of course, all things being equal does not necessarily make them equivalent in utility. For example, although both SAI and NANN use state space search, they do so at different times. A system tasked with classifying an item as being a member of a finite set that will not be changed might most efficiently be implemented using a NANN which trades the training process and inflexibility for linear runtime performance. Conversely, a SAI implementation would be more effective in a system that was required to show a chain of evidence as to why a certain action was selected. Although little training would be required of this system, its high degree of flexibility would come at the cost of the search though state space each time it was required to perform.

So, the question then becomes one of finding a way to combine techniques from both SAI and NANN methods to enhance both knowledge representation and access. It is interesting to point out that Pearl makes a strong argument for such a system existing in nature[10]. He contends that the human brain "assembles" belief networks (an SAI representation) from blocks of several Neural Networks in order to support higher cognitive functions. In other words, the brain collects, filters, and transmits events that are limited in scope to a higher level process that is responsible for interpreting and acting upon the information that is furnished. This belief is also shared by current research into understanding how the human brain learns[11][12].

3. – The Two Tiered Model

This section presents a tree based representation for Bayesian Network that is capable of being used as a member in a population manipulated by genetic programming[13] techniques. It implements the two tier approach by facilitating the selection of low level attributes that quantitize prior values of the time series and then combines this sensorial information into a reasoning layer implemented as a Bayesian Network. The genetic program then applies the principals of natural selection to a population of two tiered reasoning networks to produce a single individual Bayesian Network for predicting the underlying process that produced the time series data.

The reasoning layer used to model the time series Z in the two tier approach is defined:

$$Z_{t+1} = \delta, \max_{\min(Z) \le \delta \le \max(Z)} P(\delta \wedge Z_t \wedge Z_{t-1} \wedge \ldots) \quad [1]$$

where min(Z) and max(Z) are the minimum and maximum possible values of Z and δ is a continuous variable representing the possible values for Z at time t+1.

Although possible to work with predictors that use the continuous value δ, the model discussed in this paper relies on the set of discrete values D, containing elements $\{d^1, d^2, \ldots\}$. The values of D, which represent the sensory tier, are defined for the predictive model by dividing up the range (min(Z),max(Z)) into N discrete ranges R with boundaries λ,such that:

$$\min(Z) < R_1 \le \lambda_1 \quad [2]$$

$$\lambda_{i-1} < R_i \le \lambda_i \quad \text{for i=2,…,N-1} \quad [3]$$

$$\lambda_{N-1} < R_N < \max(Z) \quad [4]$$

with d^i defined as:

$$d^1 = \lambda_1 \quad [5]$$

$$d^i = \frac{\lambda_{i-1} + \lambda_i}{2} \quad \text{for i=2,…,N-1} \quad [6]$$

$$d^N = \lambda_N \quad [7]$$

Given the values of D, the predictive model is defined as a Bayesian Classifier. The classifier is trained using a set of attributes that relate values of d^i with prior values of Z and returns the predicted value for d_t by utilizing the maximum a posteriori hypothesis to create:

$$\hat{Z}_{t+1} = \hat{d}^i, \max_{d^i \in D} P(d^i \wedge Z_t \wedge Z_{t-1} \wedge \ldots) \quad [8]$$

Initially, the simplifying assumption is made that the prior values of Z are conditionally independent given the value of Z at time t. Based on this assumption, Eq 8 can be rewritten as the Naive Bayes Classifier:

$$\hat{Z}_{t+1} = \hat{d}^i, \max_{d^i \in D} P(d^i) \prod_{j=1}^{t} P(Z_{t-j} \mid d^i) \quad [9]$$

The computational impact of using the Naive Bayes classifier is important. It greatly reduces the amount of storage and number of calculations required to represent the Bayesian network. This is significant in an natural selection environment where all members of the population must be created, trained, and then tested for fitness during each generation. However, the simplifying assumption of conditional independence between prior values of ∇Z means that the MAP hypothesis will only be returned when this assumption is true. This is not the case when a process has an autoregressive component to it. Therefore, it is necessary to support the case where the value of Z_t is conditionally dependent on a subset of the values of Z at prior times.

The framework supports the possibility of autoregressive components by moving from a Naive Bayes classifier to a simple Bayes classifier, defined as:

$$\hat{Z}_{t+1} = \hat{d}^i, \max_{d^i \in D} P(d^i \mid Z_{t-l_1}, Z_{t-l_2}, \ldots) \prod_{\substack{j=1 \\ j \notin L}}^{t} P(Z_{t-j} \mid d^i) \quad [10]$$

where L is the subset of lags $\{l_1, l_2, l_3, \ldots\}$from K, the set all lags {0,1,…,t}.

The last improvement the framework makes to the basic classifier is the application of Occam's Razor to the number of terms used within the classifier. This subset of K, K' is then used giving:

$$\hat{Z}_{t+1} = \hat{d}^i, \max_{d_i \in D} P(d^i \mid Z_{t-l_1}, Z_{t-l_2}, \ldots) \prod_{\substack{j \in K' \\ j \notin L}} P(Z_{t-j} \mid d^i) \quad [11]$$

Figure 1 shows a sample classifier, in network form, for Z_{t+1} that relies on the conditionally dependent terms $\{Z_{t-2}, Z_{t-3}\}$ and the conditionally independent terms $\{Z_{t-1}, Z_{t-5}\}$.

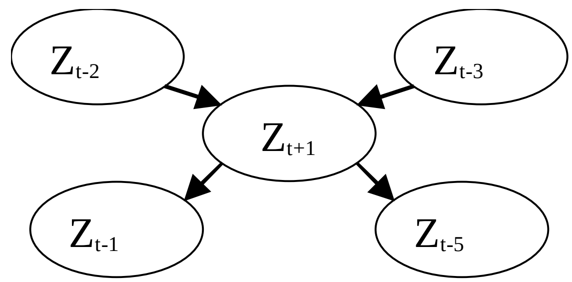

Figure 1 – A Simplified Bayesian Network for Selecting d_i.

It should be noted, that although the given predictive model contains a single query node, the presence of multiple query nodes is can be supported in the two tier approach. Second, because the network transforms the continuos model into a discrete form, it would be computationally straight forward to make qualitative evidence available to the decision layer as well. Both of these topics are mentioned here for completeness, but are beyond the scope of the work being discussed.

4. The Tree Representation

Given the desired two tier approach to the Bayesian Network model, it is necessary to specify a tree representation that will satisfy the requirements of both the natural selection process and the cognitive model previously discussed. These requirements are:

1. The tree structure must allow for the possible selection of any ∇Z_{t-i} into either the conditionally independent or conditionally dependent section of the network.
2. The tree must allow for the dynamic specification of range mapping (positioning of λ's) during the quantitization process of individuate states for each variable node.

3. The tree must not inhibit the genetic operators of reproduction, crossover, and mutation used in the natural selection process.

The tree used to represent the two tiered Bayesian Networks is best described as a set of trees within a tree structure. The root tree, shown in Figure 2, is used to describe the member variables contained within the network and the conditional relationships between them. Each node contains a single integer attribute that will be used to determine some aspect of the network based on the node's position relative to the root. The first child node always represents d, while each of the remaining children represent the remainder of the ∇Z_{t-i}'s used in defining the predicting model. Selection of the ∇Z_{t-i} is done according to the algorithm shown in Table 1.

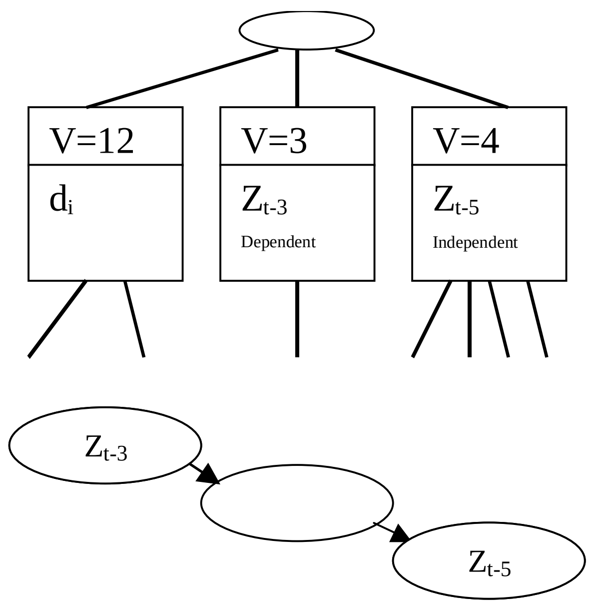

Figure 2 – A Sample Root Tree and the Bayesian Network it represents.

Each node contained within the root tree contains a subtree that encodes the λs used to partition the interval (-∞,+∞) into discrete categories. Figure 3 shows a sample tree that uses the algorithm in Table 3 to produce the categories (-∞,5], (5,10], (10,12], (12,30] and (30,+∞) for a series with observed minimum value 5 and maximum value 30. In the case of the query node, the values of d_i are determined according to Eq. 5, 6, and 7. For example, if the subtree in Figure 3 represented the query node, then the values of d_i would be 5, 7.5, 11, 16, and 30 respectively.

5. –Experimental Results

This section discusses the results of using the two tiered Bayesian Network model as a predictor for the future values of a time series from within a natural selection framework. This section presents the results of evolving two tiered Bayesian networks to forecast future values for the price of GE stock issues using the MAP Hypothesis method. This method utilized multi-lag uni-variate and multi-lag multi-variate data to produce Bayesian Networks that modeled the underlying process that generated the time series data. Each experimental forecast was made using a population of size 25, evolved through 15 generations, with the natural selection process repeated five times per forecast.

Table 1 – The Root Tree Node Attribute Assignment Algorithm.

```
Initialize LagList to contain
   available Attributes
Remove First Child from List of
   Children
For each remaining child of root, do
   Assign Attribute Lag Variable based
      on lag at list position V
   Remove Lag Variable from List
   If V & 0x01 then
      make d conditionally dependent
         on Attribute
   Else
      make Attribute conditionally
         dependent on d
   End If
End For
```

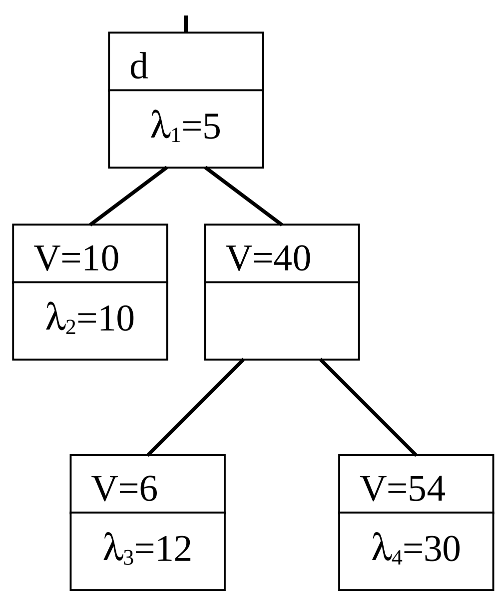

Figure 3 – A Sample Subtree.

Table 2 – The Subtree Node λ Assignment Algorithm.

```
Set max to Maximum Observed Value
   of Data
Set min to Minimum Observed Value
   of Data
Set Array of Lambdas to empty
Call ComputeLambda
   With min, max, subtree, lambdas
Exit

Function ComputeLambda
   ( min, max, subtree, lambdas )
   If subtree is empty, do
      Return
   End If
   Set total to sum of child values
   Set Range to max - min
   Set lower to min
   For each child in subtree, do
      If child is leaf node, do
         Add min to lambdas
      Else
         Set upper to lower
            + (value / total)*Range
         Call ComputeLambda With
            lower, upper, child,
            lambdas
         Set lower to upper
      End If
   End For
End Function
```

Figures 4 and 5 show the best Bayesian Network and resulting forecast for the daily price of GE stock while figures 6 and 7 show the best Bayesian Network and resulting forecast for the daily price of GE.

Overall, the forecasting results for the best naturally selected Bayesian Network were within 2.5% for first 15 values and 7% for remaining values over the data held back for evaluation of the selected time series. Also, in both series, the first forecast value was within 0.5% APE of the actual value. It is interesting to note that the selected forecast model holds within the noted error tolerance over an extended period of time without requiring retraining of the network.

It is also interesting to observe the makeup of the naturally selected Bayesian Network itself. In both cases, the best network evolved into a relatively highly quantitized representation of d_i as compared to the quantitization of the conditionally dependent and independent variables selected by the framework. Also, each of the evolved networks relied mostly on conditionally independent lag variables given the query node, with a single attribute that the query node is conditionally dependent on. This is indicative of the possibility that the use of a Naïve Bayes classifier might be just as effective in some applications requiring less accuracy, but quicker search times.

6. - Conclusions

The method discussed in this paper combines natural selection with a Bayesian Model based on current research in cognitive science. By utilizing a two tiered model similar to that found in the processing centers of the human brain, the model was able to be used in a natural selection framework to separate the classification of data points into relevant attributes from the task of using the discovered attributed in the high level forecasting model. Experimental results showed that this approach resulted in two tiered models capable of predicting future values from the underlying time series.

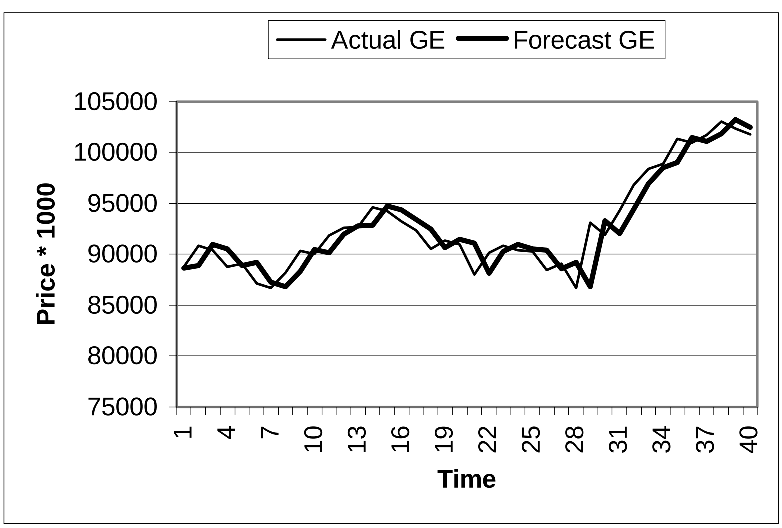

Figure 4 - The Forecasted Price of GE Stock..

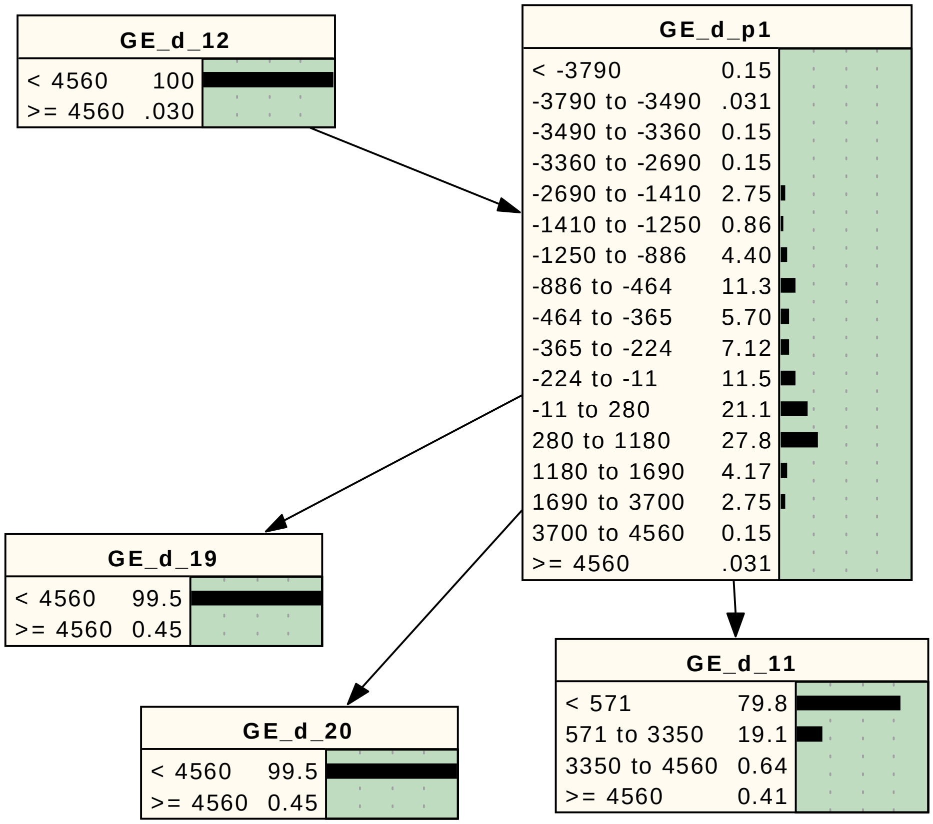

Figure 5 - The EBN Model of the Price of GE Stock.

References

[1] Greshenfeld, Neil, and Andreas Weigend. 1993. *Time Series Prediction: Forecasting the Future and Understanding the Past.*, Edited by A. S. Weigend and N. A. Gershenfeld. *The Future of Time Series*. Reading: Addison-Wesley.

[2] Refenes, A., A. Burgess, and Y. Bentz. 1997. Neural Networks in Financial Engineering: A Study in Methodology. *IEEE Transactions on Neural Networks*.

[3] Heckerman, D., A. Mamdani, and M. Wellman. 1995. Real World Applications of Bayesian Networks. *Communications of the ACM* 38, no. 5: 25-30.

[4] Russell, Stuart, and Peter Norvig. 1995. *Artificial Intelligence: A Modern Approach*. Englewood Cliffs: Prentice Hall.

[5] Abamson, B. 1991. ARCO1: An Application of Belief Networks to the Oil Market. In *Proceedings of The Uncertainty in Artificial Intelligence Conference*: 1-8.

[6] Dagum, P., A. Galper, E. Horvitz, and A. Seiver. 1995. Uncertain Reasoning and Forecasting. *International Journal of Forecasting* 11, no. 1: 73-87.

[7] Honavar, V. and Uhr, L. 1989. Brain-Structured Connectionist Networks that Perceive and Learn. *Connection Science* 1: 139-160.

[8] Honavar, V. 1994. *Computational Architectures Integrating Symbolic and Neural Processes*. Edited by R. Sun, and L. Bookman. *Symbolic Artificial Intelligence and Numeric Artificial Neural Networks: Toward a Resolution of the Dichotomy*. New York: Kluwer.

[9] Sudkamp, Thomas. 1994. *Languages and Machines*. Reading: Addison-Wesley..

[10] Pearl, J. 1997. Bayesian Networks. To appear in *MIT Encyclopedia of the Cognitive Sciences.* Cambridge: MIT Press.

[11] Sylwester, Robert. 1995. *A Celebration of Neurons: An Educator's Guide to the Human Brain.* Alexandria: Association for Supervision and Curriculum Development.

[12] Jensen, Eric. 1998. *Teaching with the Brain In Mind.* Alexandria: Association for Supervision and Curriculum Development.

[13] Koza, John. 1996. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* Cambridge: MIT Press.