An Information Theoretic Approach To Market Index Prediction

Michiko Kosaka[1]

[1]Department of Computer Science
Monmouth University
West Long Branch, NJ 07764

Abstract

This paper compares information theoretic approaches to building Bayesian belief networks to perform market index prediction. We show that the automatic model building can be done efficiently using the χ^2 criteria rather than mutual information alone. We suggest that when the number of variables are small, and instantiations are small, this criteria is a straightforward way of determining conditional independence. Both approaches use identical data to predict the stock market returns as a function of macroeconomic variables and the results are comparable (62% vs. 63% accuracy). We discuss the relative advantages of belief networks.

Keywords: Probabilistic reasoning, reasoning under uncertainty, information theory.

1. Introduction

The purpose of this paper is to compare two approaches to modeling probabilistic inferencing: belief networks and neural networks and to show that a belief network with an information theoretic approach is sufficient to rival the performance of neural networks. We use public domain market index data. The work reported here is based on a belief network. We compare our results with the results based on a neural network, as presented in [1]. The latter work provides the domain for experimentation, namely stock market returns as a function of macroeconomic variables.

In the paper by Leung, et al., [1], they report that a classification approach to solving the problem was consistently more successful than a curve-fitting approach, given the small data set of the experiment. In this case, a neural network was used to determine the classification. The neural network was constructed using all the available data points, though all but a few of the variables were eliminated in the final network structure. The disadvantages of the neural network approach are the high time complexity, the ambiguity in the threshold values for convergence (termination of construction of the neural network), and the lack of meaningfulness of the structure itself. There have been a number of solutions proposed to overcome these inherent neural nets disadvantages, e.g. neuro fuzzy models with comparable successes.

We experimented with another knowledge representation model, a belief network, which attempts to model actual cause-and-effect relationships between the variables, giving the network structure inherent meaning. Prior knowledge and training data are then used to determine the probabilistic values assigned to those relationships. Not all inputs (variables) need to be instantiated (assigned states or values), because probability can be calculated as a product of the probabilities of all the possible states. Furthermore, when instantiated inputs are reached during the search process, the search does not have to continue beyond those instantiated nodes, since the causal relationship between an instantiated node and its parents (causes) is irrelevant to the probability of its children (effects). We only concern ourselves with direct causal relationships between nodes (head and tail relationships of a digraph). We use the Bayes' theorem to compute the joint probability of the network.

The strengths of a belief network, as contrasted with a neural network, are: meaningfulness of the structure of the network with respect to the concepts the network models; the ability to prune the search space to the minimum size, especially important when dealing with complex structures; and the ability to incorporate prior knowledge on demand into the network without having to regenerate the entire network.

2. Bayesian Belief Networks

Applying Bayes' theorem to a causal network yields the joint probability of the entire network:

$$P(x_1,...,x_n) = \prod_i P(x_i \mid \pi_i)$$

In this equation, n is the number of nodes, x_i is node i, and π_i is the set of parent nodes of node i. Thus the probability of the entire node-set having a particular instantiation is equal to the product of the conditional probability of each node with respect to its parent nodes. A root node does not have parents, so therefore its conditional probability is equal to its prior probability. There are some observations which can be made using the example belief network in Figure 1:

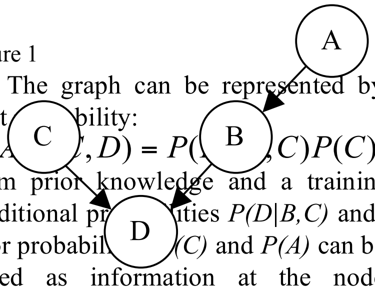

Figure 1

The graph can be represented by the following joint probability:

$$P(A,B,C,D) = P(D \mid B,C)P(C)P(B \mid A)P(A)$$

From prior knowledge and a training data set, the conditional probabilities *P(D|B,C)* and *P(B|A)* and the prior probabilities *P(C)* and *P(A)* can be calculated and stored as information at the nodes (conditional probability tables) and then used to determine the joint probability when queried. Suppose the query is to determine the probability of a particular state of the variable *D*. If the current states of the variables *B* and *C* are known, then the probability of *D* can be calculated directly using the stored conditional probability table at *D*, representing the conditional probability *P(D|B,C)*. The state of variable *A* is not required to calculate a deterministic value in this case, because its effects are already determined in the current state of *B*.

An example conditional probability table (or a contingency table) follows in Table 1:

TABLE I.
EXAMPLE CONDITIONAL PROBABILITY TABLE

		A	
		a1	a2
B	b1	25	75
	b2	75	25

The numbers in this table represent the probability of the co-occurrences of the values of the variables *A* and *B*. For example, the value of *B* will be b_1 when the value of *A* is a_1, 25% of the time. Suppose the query is to determine the probability of a particular state of *B* without knowing the current state of *A*. In this case, the probability is calculated as follows:

$$P(b) = \sum_i P(b \mid a_i)P(a_i)$$

Here, lowercase letters represent the instantiations of the variables. $P(b|a_i)$ is calculated at *B* using its conditional probability table, while $P(a_i)$ is calculated at *A*. This equation can be generalized to the situation where a variable has multiple parents:

$$P(x) = \sum P(x \mid \pi_1,...,\pi_n)P(\pi_1)...P(\pi_n)$$

This amounts to the summation over all possible combinations of instantiations of the parents of x of the product of the probability of x given the instantiation of its parents and the probabilities of each parent variable having that instantiation. When the state of a parent is known, the probability of the parent having that state becomes 1 and the probabilities of all the other states of that parent become 0 (the conditional probability table of that parent, and hence the structure of the network above that parent, do not have to be consulted). In this way, the query can be answered even though not all the relevant variables are instantiated.

3. Belief Network Stock Market Forecasting

The data set of [1] consists of monthly data points of five variables, specifically the short-term (3-month U.S. Treasury Bill) interest rate *ST*, the long-term (10-year U.S. Treasure Bill) interest rate *LT*, the Consumer Price Index *CPI*, the Industrial Production indicator *IP*, and the stock market rate of return *R*. The rate of return, *R*, is based on the Standard & Poor's 500 Index, and is defined as the natural log of the ratio of the current S&P 500 index value to the value from the previous month, and can be understood as the effective interest earned on overall stock market investment. The variable being predicted, *C*, is the probability of realizing a positive return in the following month. A positive return is when the stock market return is greater than the short-term interest rate.

The original data set included the years 1967 through 1995, split into a subset and a testing subset. Here, the experiment is performed twice; once with the original data set, and again with the data set augmented with data up to and including the year 2001.

The probabilistic neural network of [1] was based on the work of [6], and was developed by calculating the first-difference values (that is, the change over each one-month period, *t* vs. *t-1*) for the variables. The process of creating the neural network eliminated all the variables except for two, *ST* and *R*, and resulted in the network represented by the following equation:

$$C_t = F(R_{t-1}, ST_{t-1})$$

In this equation, *F* represents the function determined automatically by the neural network. This equation demonstrates the opacity of a neural network, in that it does not model the relationships between the variables in a visibly meaningful way, and there is no easy way to incorporate new knowledge into the model without regenerating the model from the beginning.

The following sections describe the process used to perform our experiment using a Bayesian belief network. First, the selection of variables is discussed. Next, the construction of the model is explained, then the results of the experiment are examined, and finally our conclusions are discussed.

4. Variables

The probabilistic neural network in [1] was generated from the entire training data set, with variables eliminated when their coefficients dropped below a threshold value. In developing our belief network, we chose to select our variable set by measuring the probabilistic dependence between each data variable (*ST*, *LT*, *CPI*, *IP*, *R*) and the prediction variable (*C*). Rather than limit the model to using differences in variables' values from month-to-month, we chose to automatically discover what time range would have the most significant influence. Each variable was examined individually with respect to *C*, across a time range of 1 to 10 months. This maximum range was selected because each instance of *C* would then be modeled using a maximum of 12 months of historic data. To accommodate this, the data set was augmented with data from 1966, the year prior to the start of the experimental period. In addition, all values of variables were examined according to their class, rather than their numerical values. Since the experiment is interested in an 'up' or 'down' movement of the stock market, variables are classified into positive and negative classes of movement.

A commonly used measure of probabilistic dependency between variables is mutual information, which is defined as the reduction in entropy of a variable when the value of another variable is known: $I(X;Y) = H(X) - H(X \mid Y)$, which is equivalent to:

$$I(X;Y) = \sum_{x,y} p(x,y) \log_e \frac{p(x,y)}{p(x)p(y)}$$

In this equation, the summation is over all possible combinations of the instantiations of variables X and Y. In theory, or in situations where the probabilities are well-understood, the mutual information between independent variables is zero, since knowing the value of one variable does not change the probability of a particular value of the other variable. When deriving probabilities from an empirical data set, however, there is enough noise in the data to prevent mutual information from equaling zero, even for independent variables; so determining which values of mutual information are significant becomes problematic. Most people sets an arbitrary threshold for the given situation.

We chose to augment it with a statistical dependency test which does clearly define a threshold value, namely the χ^2 test. In order to calculate χ^2 for two variables *X* and *Y*, a contingency table, such as that in Table II, is created, crossing all values of *X* with all values of *Y*, and the co-occurrences of each combination being counted:

TABLE II

EXAMPLE χ^2 CONTINGENCY TABLE

	y_1	y_2	Totals
x_1	$c(x_1,y_1)$	$c(x_1,y_2)$	$c(x_1)$
x_2	$c(x_2,y_1)$	$c(c_2,y_2)$	$c(x_2)$
Totals	$c(y_1)$	$c(y_2)$	N

In Table II, *c(...)* represents the count of the specified combination of values of the variables, and *N* is the total number of instances. The independence hypothesis specifies that the probability of the co-occurrence of the variables is equal to the product of the probabilities of each variable, as in:

$$p(x_i, y_i) = p(x_i)p(y_i)$$

From this it follows that the expected value for a particular variable combination can be determined by:

$$E_{i,j} = \frac{c(x_i) \times c(x_j)}{N}$$

The χ^2 value can be understood as a measure of how much the actual counts of the variable combinations differ from the expected counts (assuming the hypothesis is true). Therefore it is calculated as follows:

$$\chi^2 = \sum_{i,j} \frac{(c_{i,j} - E_{i,j})^2}{E_{i,j}}$$

Where $c_{i,j}$ is the measured count of the co-occurrences of x_i and y_j. The square in the equation prevents deviations from canceling each other out in the summation. As described before, the calculated value for χ^2 is compared against a critical value for significance. This critical factor is determined by the structure of the contingency table, and a selected error factor, α. Choosing α to be 0.05 and calculating $m = (rows - 1) \; x \; (columns - 1) = 1$, then the value of $\chi^2_{m,\alpha}$ = 3.8415 (from statistics tables). The hypothesis of independence is rejected if the value of χ^2 is greater than this critical value.

In this experiment, time ranges for the variables were chosen by selecting the time range for which From this training data, the model represented by the conditional probability $P(C|ST,LT,CPI,IP,R)$ was selected the largest χ^2 value greater than the critical value was calculated. All variables exhibited at least one time range for which χ^2 was greater than the critical value.

5. Model Building

The second phase of the experiment was to create the structure of the belief network. Many model-building strategies were examined, but they all have a level of complexity that was determined to be unnecessary for this experiment. Given the small number of variables in this experiment (6 in total), it is not prohibitive to generate an exhaustive list of possible network structures. Furthermore, since the values of all variables are known in each test data point, only the neighborhood of the network around the variable *C* (the one to be predicted) needs to be evaluated. Thus, the partial belief networks terminating in the leaf node *C* can sufficiently represent the sets of full networks that they are part of. There are precisely 31 such partial networks, from the smallest (such as *C* with parent *R* or parent *ST*), to the largest (*C* with all other variables as parents). The goal of the experiment is to develop one model which performs best at predicting the class of variable *C*. Therefore, each model was used to predict the class of *C* for each data point within the training set, and each model's performance was evaluated. Some of the models were very successful overall, but performed poorly in predicting one particular class of *C*. For this reason, *precision* and *recall* were employed in evaluating the models. Upon visual inspection of the performance of the models, recall tended to be stronger than precision. In fact, at least one model had perfect recall and very poor precision. Therefore, in order to maximize overall performance, precision and recall were combined using the *F-score*, with precision given more weight than recall. wotj a weight factor α (here chosen to be 0.75).

For the first trial run, using the same training data set as in [1], the top four models ranked by F-score were given. In this table, the models are represented by the conditional probability equation rooted at variable *C*, with the probability of *C* conditional on the values of its parents in the belief network.

6. Results

The experiment was performed twice, using the process described above, once on the original data set [1], and again on a larger data set augmented with more recent data up to and including for the year 2001.

For the first data set, the training data was taken from the years 1967 through 1990 inclusive, augmented with data from 1966. This model was tested against the test data (years 1991 through 1995 inclusive, and with data from 1990), it successfully predicted the class of variable *C* in 62% of the instances, which compares to the success rate of 63% for the probabilistic neural network of [1].

The results from performing the experiment on the second data set were consistent with the previous results. The same model was selected again, and its success rate in predicting the test data instances (in this case the data for the years 1997 through 2001, inclusive) was 62%.

7. References

[1] M. T. Leung, H. Daouk, and A. Chen, "Forecasting stock indices: a comparison of classification and level estimation models," in International Journal of Forecasting 16, pp. 173 – 190, 2000.

[2] D. Specht, "Probabilistic neural networks for classification, mapping, or associative memory," in IEEE International Conference on Neural Networks, 1988.