Application of an Instance Based Learning Algorithm for Predicting Stock Market Index

Ruppa K. Thulasiram* and Adenike Y. Bamgbade
Department of Computer Science
University of Manitoba, Winnipeg, MB, Canada

Abstract

This paper presents application of an instance based learning (IBL) algorithm for predicting stock index price changes. The objective is to determine the feasibility of stock price prediction using the IB3 variant of the IBL algorithms. Various testing proportions and normalization methods were experimented to obtain good predictions. The results obtained are promising.

1 Introduction

Financial data are usually represented as time series. These time series display certain characteristics that make it difficult to derive relationships from them for forecasting future values of the time series [2, 9, 8]. These characteristics are low noise-to-data ratio, non-linearity, and non-Gaussian noise distribution.

Stock index prices are represented in financial time series. Stock index data show independent price variations from one step to another in the long run. However, some form of regularity exists in the short run price variations of the stock market [6]. Data mining learning paradigms can be used to detect and learn from these short run regularities and predict the future behavior of stock index prices in the market. Instance based learning (IBL) is a class of data mining learning paradigm that applies specific cases or experiences to new situations by matching known cases and experiences with new cases [6]. The IB3 variant of the instance based leaning algorithms is optimized for reduced storage and works well in noisy domains [1].

This paper presents an application of the IBL algorithm for predicting daily stock index price changes of the SP 500 stock index between October 1995 and September 2000, given the daily changes in the exchange rate of the Canadian Dollar, the Pound Sterling, French Francs, Deutsch Marks and Yen, the monthly changes in the consumer price index, the GDP and the change in monthly rates of certificates of deposits. The IB3 variant of IBL algorithm is used to predict an increase, decrease or no change in the SP500 stock index between a business day and its previous.

2 Application of IBL Algorithm

IBL is a supervised data-mining learning paradigm that classifies unknown cases by matching them with known cases. IBL algorithms are an extension of the nearest neighbor pattern o classifier [3]. Aha et al. [1] present three variants of the instance based learning algorithms: IBl which is the simplest extension to the k-Nearest neighbor classifier; IB2, which is an improvement over IB1 in terms of storage reduction, and IB3, which is an improvement over IB2 in terms of noise tolerance. The main output of an IBL algorithm is a Concept Descriptor (CD). A CD includes a set of stored instances and in some cases, the classification and classification performance history of the instances. IBL algorithms do not construct extensional CDs, hence the set of instances in the CD may change after each training run. There are three basic components of the IBL algorithms: similarity function — This function computes the numerical value that shows the similarity between a query instance and the saved examples in the CD; classification function — this function uses the results from the similarity function and the classification performances of the saved instances in the CD to determine the classification of a query instance; the CD updater — maintains the instance in the CD. It updates their classification performances and determines which

*Author for Correspondence: tulsi@cs.umanitoba.ca

instances are acceptable, noisy or mediocre. IBL algorithms classify new instances based on the CD, the similarity function and classification function.

The algorithm is explained through the solution strategy in 3 phases: the data pre-processing phase, the training phase and the testing phase.

2.1 Data Pre-processing Phase

The SP 500 stock index daily price time series between October 1995 and September 2000 were collected. Seasonally adjusted *GDP* figures for the months of this period and the seasonally adjusted, monthly consumer price index(*CPI*) for all urban consumers on all items were also collected. The average daily figures of the *foreign exchange rates* of the US dollar to the Deutsch Mark, Yen, French Francs, and Canadian Dollar were collected. The exchange rate values were based on the noon buying rates in New York City for cable transfers payable in foreign currencies. The daily rates on nationally traded *certificates of deposit* were also collected. These rates are determined each business day, with the exception of the GDP,CPI and Certificates of deposit that are determined monthly. The SP 500 stock index data was obtained from the Yahoo! finance [5], while the CPI, GDP, foreign exchange rates and interest rates on certificates of deposit were obtained from the records of the United States federal reserves statistical release [7]. The data, was merged, normalized to the required format and stored in a Microsoft Access table. Each record in the table is an object instance. Each object instance is described by the changes in the closing stock price for the day, the changes in monthly GDP value, the monthly CPI, the changes in foreign exchange rates for the Deutsch Mark, Yen, French Francs and Canadian Dollar, the changes in the daily rates on certificates of deposits and the classification of the index change. We defined 3 disjoint index change classifications — increase, decrease, no change.

The important characteristics of the data are the period over which the data was generated and the significance of the factors used for the interpretation. Various factors such as economic growth, political environment, bank interest rates, inflation, expectations of the future earnings of a corporation, trade with foreign countries and the exchange rate of foreign currencies affect stock price changes. Only the quantitative factors can be computed. Hence data chosen for this work is based on the following assumptions: (1) Between October 1995 and September 2000, the United states of America enjoyed a stable political climate and economic growth. (2) The effect of trade with foreign countries and the currency rates was represented with the exchange rate changes of the Canadian Dollar, Deutsch Mark, Yen and Pound Sterling. (3) The effect of inflation was represented by the changes in the consumer price index on all commodities.

2.2 Training Phase

The goal of the training phase is to generate a non-extensional CD. At the end of the training phase, the CD should contain a set of good examples for classification in the testing phase.

A portion of the data in the pre-processed, stored data is used for training while the rest is used for testing. The portion of data used for the training (the training set) is user defined and varied for each testing run. The training starts with an initially empty CD and iterates over the steps described below for each instance in the training set: [Step 1]: The Euclidean distances between each instance from the training set and the instances currently in the CD are computed. [Step 2]: An "acceptable" instance in the CD with the shortest Euclidean distance from the training instance is assigned to the variable $ymax$. If none of the instances in the CD are acceptable then a random number i is generated between the value 1 and the current length of the CD descriptor. The i^{th} nearest neighbor of the training instance in the CD is assigned to the variable $ymax$. [Step 3]: If the classification of the training instance is the same as the instance $ymax$ in the CD, then the classification is correct; otherwise, the classification is wrong and the training instance is added to the CD. [Step 4]: The classification records of the instances in the CD are updated at this step. Instances with significantly poor records are removed from the CD. At the end of the training phase the instances saved in the CD are the example instance that will be used for classifying the testing phase instances. Instance acceptability in the testing phase is based on a confidence interval of proportions test [4] as used by [1]. The confidence interval test is also used to determine if an instance is mediocre or noisy. The confidence intervals are constructed around the current CD instances, current classification accuracy, and the observed relative frequency of its classification category.

2.3 Testing Phase

The testing phase uses the examples saved in the CD after the training phase, the similarity function and the classification function of the IB3 algorithm to determine the classification of the test instances. The testing phase is a step operation, iterated over for each training instance. The Euclidean distance between the training instance and each instance in the CD is computed. The classification function uses the nearest neighbor method to assign the classification of the nearest CD instance to the training instance. The nearest instance is the CD with the shortest Euclidean distance from the training instance. The proportion of the training dataset and the normalization method were varied for various training runs.

3 Results

During experimentation, the normalization method and training dataset proportions were varied for each test and train run. For each run, the level of significance for dropping an instance from the concept descriptor was set at 75% while a significance level of 90% was set for accepting an instance into the concept descriptor.

The object instances were normalized linearly by their range or by their standard deviation during the testing and training runs. The training data set proportions were varied from 5% to 95% in steps of 5%. Testing was carried out on the instances that made up the proportion of the dataset that was not used for training.

For each dataset proportion, normalization option at the set significance levels, 50 training and testing trials were carried out. The trials' best classification accuracy values were used for the results analysis.

The average classification accuracy on all the training and testing runs in the experiment is 51.8%. The average classification accuracy of the best accuracy values obtained from training and testing without normalization is 51.46%. Normalizing linearly (normalizing using the range) the average of the obtained best classification accuracies is 51.66%. For normalizations using the standard deviation, the average of the best classifications came to 52.25%.

In the figures below, the *x-axis* values are the test database proportions in percentages (0-100% increasing in steps of 10%), while *y-axis* values are the classification accuracies in percentages (47-55% in fig.1; 47-56% in fig. 2; and 44-55% in fig.3; all increasing in steps of 1%)

Figure (1) presents a graph of the best classification accuracies recorded from the testing and training trials normalized by standard deviations. The classification accuracies range between 50.5% and 54.25% for training properties less than 40% of the dataset. The classification accuracies of the training and testing runs carried out on more than 70% of the dataset ranged between 53.75% and 51.75%.

Figure (2) presents the best classification accuracies for the test and train trials using the linear normalization method. The classification accuracies increased steadily as the test dataset proportion was increased from 10% to 20%. Large proportions of about 65% and greater produced lower classification accuracies.

The best classification accuracies values of test and train trials that did not involve any form of normalization are shown in Figure (3). We observe that classification accuracies improved rapidly as the test proportion increased to 10% of the dataset size. The classification accuracies however reduced gradually as the the training size increased beyond 10% of the database.

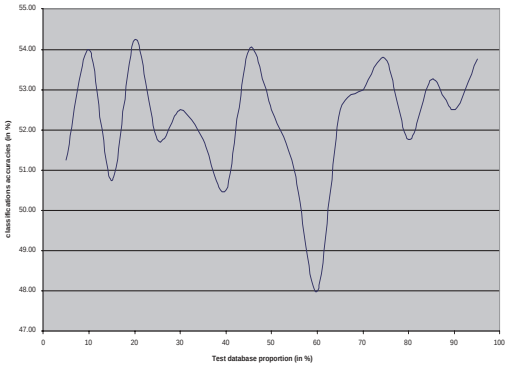

Figure 1. Best Classification Accuracies for Trail Normalized Linearly

4 Conclusions

Compared with the 46% average classification accuracy value obtained from the use of IBL classifications in other domains [1] , we can conclude that IBL can be successfully used for financial forecasting, hence stock price prediction. The results obtained show that higher classification accuracies can be obtained by normalizing the instance objects

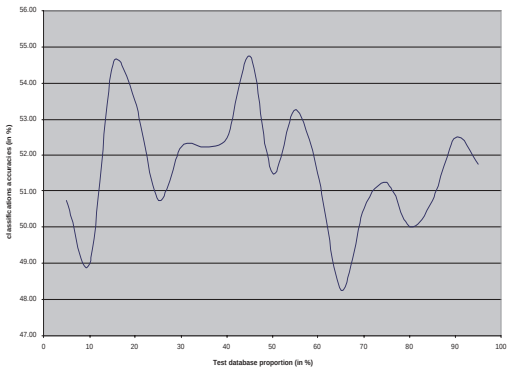

Figure 2. Best Classification Accuracies for Trail Normalized with Standard Deviations

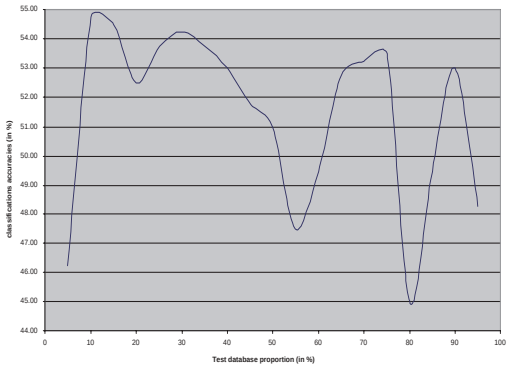

Figure 3. Best Classification Accuracies for Trial with no Normalization

during the training and testing phase. For test and training runs in which the instance objects are normalized using their standard deviations, 40% of the dataset or less will produce good classification results. Good classification results can be obtained from training 10% of the database if the instances are normalized linearly using their ranges. Test and train runs which do not employ any form of normalization will yield good classification results on 10% of the instance database. Further research can be carried out in identifying length of time that can be considered as an appropriate short run period for use with the IBL algorithm for higher accuracy.

Acknowledgement

The first author acknowledges the partial financial support from the Natural Sciences and Engineering Research Council of Canada and the University of Manitoba Research Grant Program.

References

[1] D. Aha, D. Kibler, and M.K.Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[2] Apostolos-P N. Refenes (Ed.). *Neural Networks in the Capital Markets*. John Wiley & Sons, Chichester, England, 1995.

[3] T. cover and P. Hart. Nearest neighbor pattern classification. *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, 13:21–27, 1967.

[4] R. Hogg and E. Tanis. *Probability and Statistical Inference*. Prentice-Hall, Inc., New Saddle river, NJ, 6th edition, 2001.

[5] Y. Incorporated. Yahoo finance - historical prices. http://table.finance.yahoo.com.

[6] B. Kovalerchuk and E. Vityaev. *Data Mining in Finance: Advances in Relational and Hybrid Methods*. Kluwer Academic Publishers, Norwell, MA, 2000.

[7] B. of Governors of the Federal reserve system. Federal reserve statistical release. http://www.federalreserve.gov/releases/H10/hist/.

[8] S. Oliker. A distributed genetic algorithm for designing and training modular neural networks in financial prediction. In *Nonlinear Financial Forecasting Proceedings of the first International Nonlinear Financial Forecasting Conference*, pages 183–190. Finance and Technology Publishing, 1997.

[9] A. Weigend, Y. Abu-Mostafa, and A.-P. Refenes. *Decision Technologies for Financial Engineering*. World Scientific, NewYork, NY, 1997.