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

Ruppa K. Thulasiram[1] and Adenike Y. Bamgbade[2]

[1] Department of Computer Science, University of Manitoba, MB, Canada
tulsi@cs.umanitoba.ca
[2] Department of Computer Science, University of Manitoba, MB, Canada
adenike@cs.umanitoba.ca

Instance based learning is a class of data mining learning paradigms that applies specific cases or experiences to new situations by matching known cases and experiences with new cases. This paper presents an application of the instance-based learning algorithm for predicting daily stock index price changes of the S&P 500 stock index between October 1995 and September 2000, given the daily changes in the exchange rate of the Canadian Dollar, the Pound Sterling, the French Franc, the Deutsche Mark and the Yen, the monthly changes in the consumer price index, GDP, and the changes in the monthly rates of certificates of deposit. The algorithm is used to predict an increase, decrease or no change in the S&P 500 stock index between a business day and the previous business day. The predictions are carried out using the IB3 variant of the IBL algorithms. 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 are experimented with to obtain good predictions.

Key words: Stock Market, Financial Forecasting, Computational Intelligence, Instance Based Learning, Stock Price Index

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 ([22, 18, 24]). These characteristics are a high noise-to-data ratio, non-linearity, and a non-Gaussian noise distribution.

Stock index prices are represented in financial time series. The stock index data exhibit 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 ([11]). 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. There are several computational intelligence techniques reported in the literature for various applications including those in economics and finance as briefly presented in the next section. In this study we have chosen an Instance Based Learning (IBL) algorithm and apply it to stock index prediction. To our knowledge IBL has not been used in financial applications before.

Instance based learning consists of a class of data mining learning paradigms that applies specific cases or experiences to new situations by matching known cases and experiences with new cases ([11]). The IB3 variant of the instance based learning 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 in the S&P 500 stock index between October 1995 and September 2000, given the daily changes in the exchange rate of the Canadian Dollar, the Pound Sterling, the French Franc, the Deutsche Mark and the Yen, the monthly changes in the consumer price index, GDP, and the changes in the monthly rates of certificates of deposit. The IB3 variant of the IBL algorithm is used to predict an increase, decrease or no change in the S&P 500 stock index between a business day and the previous business day. 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 are experimented with to obtain good predictions.

2 Background and Related Work

Financial forecasting is a time series prediction problem ([22, 18]). This prediction problem is a result of some properties of financial time series, such as poor signal-to-noise ratios, a non-Gaussian noise distribution and limited training data. Stock prices and indices are examples of financial time series.

According to the Efficient Markets Hypothesis (EMH) ([23]) and the Random Walk Hypothesis ([5]), forecasting from historical stock market time series is practically impossible. This difficulty arises as a result of the stochastic behavior of the price variations from one step to the next over the long run. This theory recognizes the possible existence of hidden short term regularities. However, these regularities do not take long to disappear. For a detailed study of these issues and further questions, we refer the readers to an interesting review [15] and the many references therein.

Computational intelligence techniques have been employed to exploit the above mentioned short term regularities in financial forecasting. Since our focus in this paper is on the proper application of computational intelligence techniques to finance, (a) we will concentrate only on the most relevant published work in this area; (b) the literature in finance theory is very vast and we will not do it justice by just selecting a few papers from general finance for the current purpose. However, to provide a motivation for the current research topic in the field of general finance, we refer the readers to [6, 17].

2.1 Time Series Forecasting

Reference [14] present the application of EpNet to the Hang Seng stock index forecasting problem. EpNet is a system which evolves generalized feed forward neural networks for predicting stock prices. The experimental results show that EpNet was able to evolve neural networks that were well generalized to the independent testing data. The results further showed that, compared with the actual stock index values, the evolved neural network captured almost all the changes in the time series. Reference [12] developed a feed forward neural network for predicting stock prices for companies. The results he obtained showed that the stock price movements, which he considered to be of more importance than the precise value of the stock market, were captured by the neural network predictions. Yao, et al's research into the prediction of the Kuala Lumpur composite index of Malaysian stocks shows that neural networks successfully predict the stock price changes ([26]). In addition, [26] were able to show that neural networks gave better results compared to the conventional ARIMA models. However, a general impediment of the neural network algorithm is the training time of the network, which is generally large. A recent study ([21, 20]) expedites the training by designing parallel algorithms for the neural network architecture. This study produces accurate results, that are more accurate than traditional regression models such as ARMA ([16]). The results from a comparative test of statistical and logical learning algorithms show that the Nearest Neighbor (NeNe) algorithm outperforms the feed-forward neural networks in terms of prediction accuracy ([9]). Soft computing using fuzzy and rough sets has been one of the tools of analysis in finance (see for example [7, 2]). In [7] the authors try to forecast future prices from past data using a new mathematical theory of Rough Sets ([19]) and have applied it to a data set from the New Zealand stock exchange. The authors claim that a reasonable prediction could be achieved with rough rules obtained from training data. In [2] fuzzy mathematics has been applied to the study of the option pricing problem in finance. Datamining, especially machine learning algorithms, is instanced based acquiring prominence in finance. In such problems as the detection of outliers in financial time series, the distance based mining algorithm (based on [10]) and statistics based algorithm ([13]) have recently been reported to predict outliers. A hybrid algorithm ([13]) has been shown to outperform these two algorithms by capturing almost all kinds of outliers. These studies have shown that data mining algorithms are simpler and easier to use than neural networks. Instance based algorithms are an extension of the Nearest Neighbor algorithm.

2.2 The 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 classifier ([4]). Reference [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 instances in the CD. It updates their classification performances and determines which instances are acceptable, noisy or mediocre.

IBL algorithms classify new instances based on the CD, the similarity function and the classification function.

The algorithm is explained through the solution strategy in 3 phases: (1) the data pre-processing phase, (2) the training phase, and (3) the testing phase.

3 Predicting Stock Price Index Variation

The focus of this work is to predict the daily stock index price changes of the S&P 500 stock index using the IB3 algorithm. The IB3 algorithm is described as a pseudo code in this section. The S&P 500 stock index price will be predicted using its historical time series, the gross domestic product, the consumer price index, the exchange rates of the Canadian Dollar, the Deutsche Mark, the French Francs and the Japanese Yen and the monthly interest rates on bank certificates of deposit. These data values between October 1995 and September 2000 were used in this study. The exchange rates and the interest rates are examples of quantitative factors other than time series that affect stock price movement.

```
==================================================
CD <= 0
For each x in training set do
for each y in CD do
sim[y] <= similarity(x,y)
if there exists {y in CD | acceptable(y)}
then ymax <= some acceptable y in CD with maximal sim[y]
else
i <= a randomly selected value in {1|CD|}
ymax <= some y in CD that is the i-th most similar instance to x
for each y in CD do
if sim[y] >= sim(ymax)
then
update y’s classification record
if y’s record is significantly poor
then CD <= C - {y}

===================================================
The IB3 Algorithm
===================================================
```

Let D represent the instance space of the short run stock index times series with d instance objects, where d represents a record in the time series

$$D = \{d_i, ...\}, d_i = (d_{i1}, d_{i2}, ...d_{in}), i = 1, m$$

Each object d in the instance space is represented by an ordered set of n attributes. The first $n - 1$ attributes represent the predictor values while the n^{th} attribute represents the classification of object d.

We need to select a set of objects from the instance space D such that given a target object d_t with $d_{tn} = \emptyset$, the value of d_{tn} can be predicted using the Euclidean distance between the known $n - 1$ attributes of d_t and the $n - 1$ attributes of all the selected instances from the instance space D. The Euclidean distance between d_t and each of the selected objects d_i in the selected set can be computed by the function $\vartheta(d_i, d_t)$

$$\vartheta(d_i, d_t) = \sqrt{\sum_{j=1}^{n-1} (d_{ij} - d_{tj})^2}$$

4 Experimental Framework

4.1 Data Pre-processing Phase

The S&P 500 stock index daily price time series data covering the period between October 1995 and September 2000 were collected. Seasonally adjusted *GDP* figures for the months during this period and the seasonally adjusted, monthly consumer price index(*CPI*) for all urban consumers for all items were also collected. The average daily figures of the *foreign exchange rates* of the US dollar to the Deutsche Mark, Yen, French Franc, and the 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 S&P 500 stock index data were obtained from the Yahoo! finance ([25]), 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 ([3]). The data were 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 Deutsche Mark, Yen, French Franc and Canadian Dollar, the changes in the daily rates on certificates of deposit 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 were generated and the significance of the factors used for the interpretation. Various factors such as economic growth, the political environment, bank interest rates, inflation, expectations regarding 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 the data chosen for this work are based on the following assumptions: (1) Between October 1995 and September 2000, the U.S. enjoyed a stable political climate and economic growth. (2) The effect of trade with foreign countries and the currency rates were represented by the exchange rate changes involving the Canadian Dollar, Deutsche Mark, Yen and Pound Sterling. (3) The effect of inflation was represented by the changes in the consumer price index for all commodities.

4.2 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 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 instances 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 ([8]) 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.

4.3 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 testing instance. The nearest instance is the CD with the shortest Euclidean distance from the training instance. The proportion of the training data set and the normalization method were varied for various training runs.

5 Results

During experimentation, the normalization method and training data set proportions were varied for each test and training 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 deviations 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 data set that was not used for training.

For each data set proportion, and 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 of all the training and testing runs in the experiment was 51.8%. The average classification accuracy of the best accuracy values obtained from training and testing without normalization was 51.46%. Normalizing linearly (normalizing using the range) the average of the obtained best classification accuracies was 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 the *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 test properties that are less than 40% of the dataset. The classification accuracies of the training and testing runs carried out on more than 70% of the data set ranged between 53.75% and 51.75%.

Figure 2 presents the best classification accuracies for the testing and training trials using the linear normalization method. The classification accuracies increased steadily as the test data set 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 Fig. (3). We observe that classification accuracies improved rapidly as the test proportion increased to 10% of the dataset size. The classification accuracies, however, declined gradually as the the test size increased beyond 10% of the database.

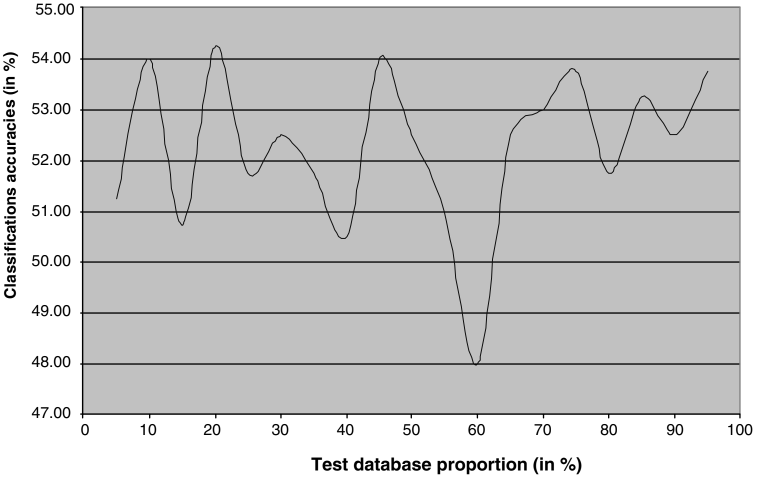

Fig. 1. Best Classification Accuracies for Trial Normalized by Standard Deviations

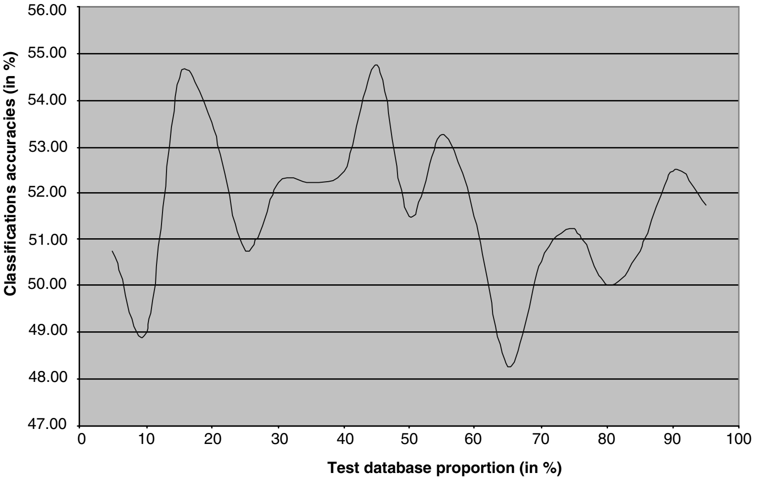

Fig. 2. Best Classification Accuracies for Trial Normalized Linearly

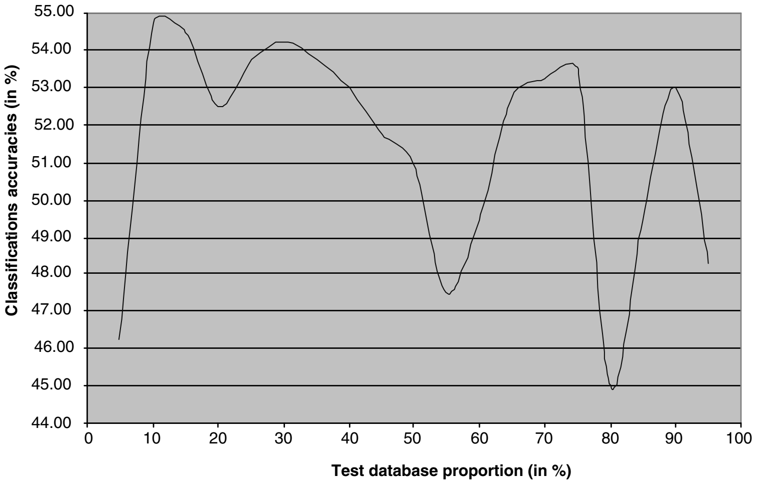

Fig. 3. Best Classification Accuracies for Trial with no Normalization

6 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, and hence stock price prediction. The results obtained show that higher classification accuracies can be obtained by normalizing the instance objects 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. Testing and training runs which do not employ any form of normalization will yield good classification results for 10% of the instance database. Further research can be carried out in identifying the length of time that can be considered to be an appropriate short run period for use with the IBL algorithm to achieve higher accuracy.

Acknowledgement

The authors are grateful for the constructive comments from anonymous reviewers. 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. Aha DW, Kibler D, Albert MK (1991) Instance-based learning algorithms. Machine Learning 6:37–66
2. Appadoo SS, Thulasiram RK, Bector CR (2004) Fuzzy algebraic approach to option pricing - a fundamental investigation. In: Proceedings (CD-RoM) of the Administrative Sciences Association of Canada (ASAC) conference, Quebec City, Canada
3. Board of governors of the Federal reserve system Federal reserve statistical release. http://www.federalreserve.gov/releases/H10/hist/
4. Cover T, Hart P (1967) Nearest neighbor pattern classification. Institute of Electrical and Electronics Engineers Transactions on Information Theory 13:21–27
5. Cowles A, Jones H (1937) Some a posteriori probabilities in stock market action. Econometrica 5:280–294
6. Dixit AK, Pyndick RS (1994) Investment under uncertainty. Princeton University Press
7. Herbert J, Yao J (2005) Time-series data analysis with rough sets. In: Proceedings (CD-RoM) of the computational intelligence in economics and finance, Salt Lake City, UT
8. Hogg RV, Tanis EA (2001) Probability and statistical inference. Prentice-Hall, Inc., New Saddle river, NJ, 6th edition
9. King R, Feng C, Sutherland A (1995) Statlog: comparison of classification algorithms on large real-world problems. Applied Artificial Intelligence 9(3):289–333
10. Knorr EM, Ng RT (1998) Algorithms for mining distance-based outliers in large datasets. In: Proceedings of the VLDB:392–403
11. Kovalerchuk B, Vityaev E (2000) Data mining in finance: advances in relational and hybrid methods. Kluwer Academic Publishers, Norwell, MA
12. Landt FWO (1997) Stock price prediction using neural networks. Master's thesis, Leiden University
13. Leung CK, Thulasiram RK, Bondarenko D (2005) The use of data mining techniques in detecting noise and pre-processing financial time series. In: Proceedings (CD-RoM) of the computational intelligence in economics and finance (CIEF), Salt Lake City, UT
14. Liu Y, Yao X (2001) Evolving neural networks for Hang Seng stock index forecast. In: Proceedings of the 2001 congress on evolutionary computation. IEEE Press:256–260
15. Lo A (2000) Finance: a selective survey. Journal of the American Statistical Association 95(450):629–635
16. Makridakis S, Wheelright S (1978) Forecasting methods and applications. John Wiley & Sons, New York, USA
17. Mendelbrot B (2004) The misbehavior of markets. Basic Books
18. Oliker S (1997) 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. Finance and Technology Publishing:183–190
19. Pawlak Z (1991) Rough sets-theoretical aspects of reasoning about data. Kluwer Academic
20. Rahman MR (2002) Distributed and multithreaded neural network algorithms for stock price learning. MSc Thesis, Department of Computer Science, University of Manitoba, Winnipeg, MB, Canada
21. Rahman MR, Thulasiram RK, Thulasiraman P (2002) Forecasting stock prices using neural networks on a Beowulf cluster. In: Akl SG, Gonzalez TF (eds) Proceedings of the fourteenth IASTED international conference on parallel and distributed computing and systems. IASTED Press:465–470

22. Refenes APN (ed) (1995) Neural networks in the capital markets. John Wiley & Sons, Chichester, England
23. Samuelson P (1965) Proof that properly anticipated prices fluctuate randomly. Industrial Management Review 6:41–49
24. Weigend AS, Abu-Mostafa YS, Refenes APN (1997) Decision technologies for financial engineering. World Scientific, New York, NY
25. Yahoo! Incorporated. Yahoo finance - historical prices. http://table.finance.yahoo.com
26. Yao J, Tan C, Pohyao H (1999) Neural networks for technical analysis: a study on klci. International Journal of Theoretical and Applied Finance 2(2):221–241