Combining Ordinal Financial Predictions with Genetic Programming

Edward P.K. TSANG and Jin LI

Department of Computer Science, University of Essex,
Wivenhoe Park, Colchester, Essex CO4 3SQ, United Kingdom
Email: {edward, jli}@essex.ac.uk

Abstract. Ordinal data play an important part in financial forecasting. For example, advice from expert sources may take the form of “bullish”, “bearish” or “sluggish”, or "buy" or "do not buy". This paper describes an application of using Genetic Programming (GP) to combine investment opinions. The aim is to combine ordinal forecast from different opinion sources in order to make better predictions. We tested our implementation, FGP (Financial Genetic Programming), on two data sets. In both cases, FGP generated more accurate rules than the individual input rules.

1 Introduction

Ordinal data could be useful in financial forecasting, as Fan et. al. [6] quite rightly pointed out. For example, forecast by experts may predict that a market is “bullish”, “bearish” or “sluggish”. A company’s books may show “deficit” or “surplus”. A share’s price today may have “risen”, “fallen” or “remained unchanged” from yesterday’s. The question is how to make use of such data.

Let Y be a series, gathered at regular intervals of time (such as daily stock market closing data or weekly closing price). Let Y_t denote the value of Y at time t. Forecasting at time t with a horizon h means predicting the value of Y_{t+h} based on some information set I_t of other explanatory variables available at time t. The conditional mean

$$F_{t,h} = E[Y_{t+h} \mid I_t]$$

represents the best forecast of the most likely Y_{t+h} value [8]. In terms of properties of value Y, forecast could be classified into point forecast, where Y_t is a real value, or *ordinal forecasts*, where Y_t is an interval estimate. In terms of the property of I_t, forecast could be classified into *time-series forecast*, where I_t consists of nothing but Y_{t-i} where $i \geq 0$, or *combining forecast*, where I_t only includes a finite direct forecast results from different sources.

In recent years, there has been growing interest in combining forecasts; for example, see [17, 13] for combining point forecasts and [6, 3] for combining ordinal forecasts. The methodologies adopted in these researches are mainly statistical methods and operation research methods. The full potential of AI forecasting techniques such as genetic algorithms [9] has yet to be realized.

K.S. Leung, L.-W. Chan, and H. Meng (Eds.): IDEAL 2000, LNCS 1983, pp. 532−537, 2000.

In this paper, we follow the study of Fan and his colleagues and focus on combining ordinal forecasts. We demonstrate the potential of Genetic Programming (GP) [11] in combining and improving individual predictions in two different data sets:

(i) a small data set involving the Hong Kong Heng Seng index as reported by Fan and his colleagues [6]; and

(ii) a larger data set involving S&P 500 index from 2 April 1963 to 25 January 1974 (2,700 trading days).

2 FGP for Combining Ordinal Forecasts

2.1 Background: Genetic Programming and Its Application to Finance

Genetic algorithm (GA) is class of optimization technique inspired by the principle of natural selection in evolution. *Genetic Programming* is a promising variant of genetic algorithms that evolves tree representations instead of strings. The basic algorithm is as follows. Candidate solutions are referred to as chromosomes and the program maintains a set of chromosomes, which is referred to as a population. Each chromosome is evaluated for its fitness according to the function that is to be optimized. Fitter strings are given more chance to be picked to become parents, which will be used to generate offspring. Offspring copy their material from both parents using various mechanisms under the name of crossover. Offspring are sometimes given a chance to make minor random changes, which are referred to as mutations. Offspring may replace existing members of the population. The hope (supported by theoretical analysis, see for example [7]) is that after enough number of iterations, better candidate solutions can be generated. GPs have been successful in many applications, including financial applications, e.g. see [1, 12, 14, 4].

FGP (Financial Genetic Programming) is a genetic programming implementation specialized for financial forecasting. It is built as a forecasting tool under the EDDIE project [16]. In this paper, we shall focus on its application in combining individual expert predictions in order to generate better predictions.

2.2 Candidate Solutions Representation

In the Hong Kong stock market example in the next section, the set of possible categories is {bullish, bearish, sluggish, uncertain}. In the S&P 500 index example in the subsequent section, the set of categories is {buy, not-buy}.

FGP searches in the space of decision trees whose nodes are functions, variables, and constants. Variables and constants take no arguments and they form the leaf nodes of the decision trees. In the applications described in this paper, both the variables (input) and the predictions (output, constants) are ordinal categories. The grammar determines the expressiveness of the rules and the size of the rule space to be searched. Functions take arguments and they form subtrees. In this paper, we take {if-then-else, and, or, not, >, <, =} as functions.

2.3 Experimental Details

Being a genetic programming system, FGP needs a suitable fitness function that measures the predictability of each decision tree. One fundamental measure of predictability is the rate of correctness (RC) – the proportion of correct predictions out of all predictions:

$$\text{RC} \equiv \text{number of correct predictions} \div \text{total number of predictions}$$

In the experiments described below, crossover rate is 90% and mutation rate is 1%. Elitism is employed by randomly picking 1% of the population, biased towards the fitter individuals, and putting them directly into the next generation. Among existed selection methods in GP, we used tournament selection with tournament size set to 4. Population size is set to 1,200. The termination condition is 40 generations or two hours, whichever reached first. Initial GDTs are limited to a depth of 5. The maximum depth of any tree is set to 17. FGP-1 was implemented in Borland C++ (version 4.5). All experiments described in this paper were run in a Pentium PC (200MHz) running Windows 95 with 64 MB RAM.

3 Application of FGP to the Hong Kong Stock Market

FGP was applied to the prediction of changes in the Heng Seng Index in the Hong Kong Stock Market. We used the data set given in the appendix of [6], which comprises 103 data cases, each of which comprises nine expert predictions for the following week and the actual market changes. Predictions by each of the 9 experts fall into four categories, which Fan et al. labeled as:

1. bullish, defined as "the index rises by over 1.3% in the next week";
2. bearish, defined as "the index falls by over 1.3% in the next week";
3. sluggish, defined as "the index is neither bullish nor bearish"; and
4. uncertain, which means the expert did not make a prediction.

The period under this study was from 25 May 1991 to 16 October 1993.

Fan et al [6] used the "leave-one-out cross-validation strategy" to assess the forecasting accuracy. This means to generate a forecasting for time *t*, all but the experts' predictions at time *t* were used to generate a combined prediction. Predictions generated this way were evaluated. For simplicity without lost of generality, we used 3-fold cross-validation to estimate FGP's forecasting performance: we partitioned the data set into three mutually exclusive subsets (the folds):

D1: 34 data cases from 25 May 1991 to 11 January 92;
D2: 35 data cases from 18 January 1992 to 5 December 1992;
D3: 34 data cases from 12 December 1992 to 16 October 1993

Each of these data sets was used as the testing data set once, whilst the remaining two sets were employed as the training data set. The mean forecasting accuracy was the overall number of correct forecasts divided by number of cases in the whole data set [10]. For each of D1, D2, D3, we ran FGP 10 times, so a total of 30 runs were used in our experiments.

FGP-1 achieved an average RC of 60.88%, 45.14% and 45.29% over D1, D2, D3, respectively. The mean RC of FGP method was 50.39%, which is comparable with (if slightly better than) the Multinomial Logic Method (MNL, 50.16%) and the Linear Programming Method (LP, 45.63%) presented in [6]. The best expert prediction input (Expert 7) achieved an RC of 43.69%. It was encouraging to see that MNL, LP and FGP can all improve the accuracy of the best expert's forecast. However, this example involves relatively small data cases and therefore one should not generalize the results without further experimentation.

4 Application of FGP to the S&P 500 Index

Encouraged by FGP's promising forecasting performance on the Heng Seng Index, we tested FGP on the S&P-500 daily index. Available to us were data from 2 April 1963 to 25 January 1974 (2700 data cases). Our goal is to see whether FGP could improve forecasting accuracy on textbook-type predictions.

Six technical rules (three different types) derived from the financial literature [2, 5, 15] are used as input to FGP-1. They were used to predict whether the following goal is achievable at any given day:

G: the index will rise by 4% or more within 63 trading days (3 months).

The six technical rules we used were as follows:

- Two Moving Average Rules (MV):
  The L-days simple moving average at time t, SMV(L, t), is defined as the average price of the last L days from time t. The rule is "if today's index price is greater than SMV(L, t), then buy; else do not buy." L = 12 and L = 50 were used.
- Two Trading Range Break Rules (TRB):
  The rule is: "buy if today's price is greater than the maximum of the prices in the previous L days; else do not buy". L = 5 and L = 50 were used.
- Two Filter Rules:
  This rule is "buy when the price rises by y percent above its minimum of the prices in the previous L days; else do not buy." Two rules, with y = 1(%) and L = 5 and L = 10 were used.

Our sole concern is whether FGP can combine technical rules in order to generate more accurate forecasting. Therefore, the quality of the individual rules is not crucial to our study.

The FGP algorithm is the same as that in the first example. In addition to the rate of correctness (RC), we added two factors to the fitness function: the rate of missing chance (RMC) and the rate of failure (RF). RMC and RF are defined as follows:

$$\text{RMC} \equiv \text{\# of erroneous not-buy signals} \div \text{total number of opportunities}$$
$$\text{RF} \equiv \text{\# of erroneous buy signals} \div \text{total number of buy signals}$$

Weights were given to RC, RMC and RF in the fitness function. By adjusting these weights, we can reflect the preference of investors. For example, a conservative investor may want to avoid failure and consequently put more weight on RF.

Table 1. Performance comparisons between individual rules and FGP rules

Individual rule performances			FGP rule performances		
Rules	Accuracy (RC)	ARR	Runs	Accuracy (RC)	ARR
MV (L=12)	0.4956	0.3020	FGP Rule 1	0.5400	0.3952
MV (L=50)	0.5189	0.2666	FGP Rule 2	0.5389	0.3945
TRB (L=5)	0.4733	0.3319	FGP Rule 3	0.5400	0.3952
TRB (L=50)	0.4756	0.2102	FGP Rule 4	0.5522	0.3911
Filter (L=5)	0.4944	0.3746	FGP Rule 5	0.5444	0.3964
Filter(L=10)	0.4889	0.3346	FGP Rule 6	0.5367	0.3935
			FGP Rule 7	0.5389	0.3945
			FGP Rule 8	0.5356	0.3928
			FGP Rule 9	0.5433	0.3960
			FGP Rule10	0.5300	0.4187
			Mean	0.5400	0.3968

In our experiments, RC, RMC and RF were given weights of 1, 0.2 and 0.3 respectively. 1,800 cases (02/04/1963 -- 02/07/1970) were used as training data. 900 cases (06/07/1970 -- 25/01/1974) were used as test data. We ran FGP 10 times. For each run, the best rule evolved in training was applied to the testing data. The results of FGP rules on testing data and the six individual rules were recorded in Table 1. Among the six technical rules, the MV(L=50) rule was the best individual rule for this set of data. It achieved an accuracy of 51.89%. In contrast, even the poorest FGP rule (FGP rule 10) achieved an accuracy of 53.00%. The average accuracy of FGP rules was 54.00%. So although only 10 decision trees were generated, the results were conclusive: FGP produced better forecasting consistently by combining individual decisions.

For reference, we measured the *annualised rate of return* (ARR) by the rules above using the following hypothetical trading behaviour with simplifying assumptions:

> ***Hypothetical trading behaviour****: whenever a buy signal is generated, one unit of money is invested in a portfolio reflecting the S&P-500 index. If the index rises by 4% or more within the next 63 days, then the portfolio is sold at the index price of day t; else sell the portfolio on the 63rd day, regardless of the price.*

We ignored transaction costs and the bid-ask spread. Results in Table 1 show that rules generated by FGP achieved an ARR of 39.68% in average. In comparison, the best of the input rules (Filter rule, with L=5) achieved an ARR of 37.46%, which is lower than the poorest ARR generated by FGP in the ten runs (39.11% by rule 4).

Acknowledgements

This work is partly supported in part by the University of Essex Research Promotion Fund. Jin Li is supported by *Overseas Research Students Awards (ORS)* and University of Essex Studentship. James Butler and Tung Leng Lau contributed to earlier parts of this research.

References

1. Angeline, P. & Kinnear, K. E., (ed.), *Advances in genetic programming II*, MIT Press. 1996. Blume, L., Easley, D. & O'Hara, M., Market statistics and technical analysis: the role of volume, *Journal of finance*, 49, (1994), 153-181.
2. Brock, W., Lakonishok, J. & LeBaron, B., Simple technical trading rules and the stochastic properties of stock returns, *Journal of Finance*, 47, (1992), 1731-1764.
3. Cesa-Bianchi, N., Freund, Y., Haussler, D., Helmbold, D.P., Schapire, R.E. & Warmuth M.K., How to use expert advice, *Journal of the ACM*, Vol. 44, No. 3., (1997), 427-485.
4. Chen, S-H. & Yeh, C-H. , Speculative trades and financial regulations: simulations based on genetic programming, *Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFEr)*, New York City, (1997), 123-129.
5. Fama, E.F. & Blume, M.E., filter rules and stock-market trading, *Journal of Business* 39(1), (1966), 226-241.
6. Fan, D.K., Lau, K-N. & Leung, P-L., Combining ordinal forecasting with an application in a financial market, *Journal of Forecasting*, Vol. 15, No.1, Wiley, January, (1996), 37-48.
7. Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.
8. Granger, C.W.J., Forecasting, in Newman, P. , Milgate, M. & Eatwell, J. (ed.), *New palgrave dictionary of money and finance*, Macmillan, London, (1992), 142-143.
9. Holland, J. H., *Adaptation in natural and artificial system*, University of Michigan Press, 1975.
10. Kohavi, R., A study of cross-validation and bootstrap for accuracy estimation and model selection, *Proceedings of International Joint Conference on Artificial Intelligence* (IJCAI), (1995), 1137-1143.
11. Koza, J.R., *Genetic Programming: on the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press, 1992.
12. Koza, J., Goldberg, D., Fogel, D. & Riolo, R. (ed.), *Proceedings of the First Annual Conference on Genetic programming*, MIT Press, 1996.
13. Lobo, G., Alternative methods of combining security analysts' and statistical forecasts of annual corporate earnings, *International Journal of Forecasting*, (1991), 57-63.
14. Neely, C., Weller, P. & Ditmar, R., Is technical analysis in the foreign exchange market profitable? a genetic programming approach, in Dunis, C. & Rustem, B.(ed.), *Proceedings, Forecasting Financial Markets: Advances for Exchange Rates, Interest Rates and Asset Management*, London. 1997.
15. Sweeney, R. J., Some new filter rule test: Methods and results, *Journal of Financial and Quantitative Analysis*, 23, (1988), 285-300.
16. Tsang, E.P.K, Butler, J. M. & Li, J., Eddie beats the bookies, *Journal of Software, Practice and Experience*, Wiley, Vol.28 (10), August 1998, 1033-1043.
17. Wall, K. & Correia, C., A preference-based method for forecast combination, *Journal of Forecasting*, (1989), 269-192.