Stock Trading with Genetic Algorithm---Switching from One Stock to Another

Tomio Kurokawa, Member, IEEE-CIS

Abstract—It has been a stock trader’s dream to do stock trading by switching from one stock to another while buying at bottoms and selling at tops. This paper investigates the effectiveness and the practicability of such a trading model. For a learning system, nothing is more influential than the data to learn. Any data, whether it is in-sample or out-of-sample, is intrinsically particular. It is typically true if the training data are small. Take some past one year stock data of a single stock name for instance. It is only a small segment of data patterns if compared with the unseen out-of-sample data of potentially numerous patterns. Naturally any technical system trained with some limited in-sample data are more or less overfitted. Therefore data of large size are desired for training a trading system and concurrently monitoring stocks of many names in practical trading. Trying to take advantages of large size data, this study examines the trading system trained with such data and with the model of switching one stock to another using genetic algorithm.

Index Terms—Genetic algorithm, In-sample data, Overfitting, Switching one stock to another.

I. INTRODUCTION

Learning has been applied to technical rules for stock market decision. It has been the problem that what are learned by using some past data is not necessarily effective for the actual problem. This problem frequently happens to be overfitting. It has been the major problem of learning of technical market decision. In some past studies, solutions were tried to be sought.

First is about in-sample data selection. Wang et al. [1] employed, in GA (genetic algorithm), the methods with which newer in-sample data were used for training by sliding the in-sample data window as the trading proceeds. Lam et al. [2] employed sliding (incremental and dynamic) in-sample data approach for training the system with GA and fuzzy mechanism. These schemes were based on the presumption that newer data might be better representing coming unseen out-of-sample data. Neely et al. [3] used, in GP (genetic programming), a technique regarded as validation procedure, in which a selection period is placed after the training period in order to select one good program for next generation.

Second is related with learning itself. It is to improve the learning process so that the trading rule should have good performance and avoid overfitting. Obtaining a good performance rule is concerned with many things such as learning process itself, elements of computational structure including indicators, fitness strategies, maintaining simplicity, generality and so on.

In the past, Becker et al. [4]-[5] used, in GP, reduced set of operators and set of increased indicators (elements of computational structure), complexity penalizing strategy (fitness strategies) in the training process and maintained the simplicity of the tree structure of GP-generated rule by limiting the number of nodes and the depth of the tree (simplicity). Simplicity is very important for avoidance of overfitting because complexity is more likely to bring about overfitting [4]. Lin et al. [6] set sub-ranges for parameters of technical trading rules (fitness strategies) by GA and obtained robust results.

Third is to devise effective technical indicators. Technical indicators play important roles in rule making with learning, especially in evolutionary process. In the process, they are usually pre-given as essential components. For GA, indicators with parameters are given and are optimized. For GP, though it has the ability to find good computational structure of technical rules, indicator functions such as moving averages, etc. must be given as in component set at the beginning. Eventually, some of the components play important roles in the generated rules. Such indicators were also used in GP process of the studies by Becker et al. [4] and they showed positive results. Potvin et al. [7] applied GP to Canadian individual stocks and reported that it did not necessarily outperform buy and hold (B&H) approach. Pavlidis et al. [8] compared moving average based rules and GP-generated rules on money exchange rates and obtained the results that both are profitable but moving average based rule is more robust than the GP-generated one. Mabu et al. [9] included several conventional indicators in GNP (genetic network programming) and showed positive results. Kurokawa [10]-[12] tried to seek better technical indicators used in GA process and also examined the method with GA using a large number of stock names concurrently processed.

II. WHERE POTENTIAL SOLUTIONS ARE

Looking over a stock market, the data are very vast. Even data of a single stock name are vast when considering unseen out-of sample data of potentially numerous patterns. Hence it is not reasonable to try to cope with it by learning only a small portion of past data. One or two year market data of a stock name are very small compared to unseen potentially vast out-of-sample data. Learning process is to make the target system particular. It is to adjust the system to the in-sample data. Accordingly, the target system hardly becomes general.

In order to solve the problem, some generality introducing mechanism outside of learning process is necessary. Possible solutions may exist in 1) in-sample data of large size and concurrent processing of stocks of multiple names in trading, 2) sophisticated learning process which can make use of large stock data of multiple names and avoid...
particularity of in-sample data or overfitting and 3) devising effective technical indicators.

As for 1), most studies in literature about market timing by learning have been done with in-sample data of small size, that is, one or two years for a single name or an index or of several years. Small size data usually cannot have generality. Naturally it is hardly possible for any process to extract generality from data of small size. Hence in-sample data of large size are needed. Larger size data are more likely to have more generality. However, it is time consuming for processing. There are two ways of expanding in-sample data. One is with data of long period. There are, however, some limitations about the size. The other is with stocks of many names. In this study, the latter case was examined with GA procedure using data of hundreds of stock names.

As for 2), this is the area where many studies have been done in literature. In order to handle data of large size, effective learning mechanism with fast processing speed and ability to avoid overfitting becomes important. Both of sophistication and speed are necessary at the same time. The learning should have the ability to handle the data of large size and to extract the generality.

As for 3), technical indicator is directly related with data computation. It is the device by which trading signals are directly detected. Since many market timing systems by learning employ technical indicators, developing effective indicators is essential. What’s more, indicators are independent of learning process in the sense that they are made before learning process operates. Hence, they could bring about generality if not totally influenced by learning. Simplicity and comprehensibility could be given by human heuristics. Hence, effective indicators are of great utility. Some studies were made in this area [10].

III. TRADING EXPERIMENTS WITH THE MODEL OF SWITCHING ONE STOCK TO ANOTHER WITH DATA OF DIFFERENT SIZES

It has been pointed out that large size data have more generality and therefore possibly more effectiveness than a small size data to learn from them. More precisely, that is, there are more generality in training and more opportunities in practical trading. However, large size data are not easy to handle [11]. The immediate difficulty is how to avoid the time consuming process. Based on the results of the theory and the demonstrated effectiveness by the experiments with large size data [12], simulations were organized to examine the trading model of switching from one stock to another in this study. In order to see the essence, the model was designed as simple as possible.

Two kinds of experiments were done with different stock data of different sizes for comparison. The experimental procedure is shown by Fig1. They were the following two:

1) Experiment 1 with each single stock name, and
2) Experiment 2 with 10 stock names for concurrent monitoring.

But, the data periods were the same for the two. The data from 2001/1/1 to 2002/12/31 were used for training and those from 2003/1/1 to 2004/12/31 were for testing. All data are from daily data of Tokyo Stock Market Division I.

The following are how the experimental trading was done. At the start, some amount of cash, supposedly very large, was provided. For the experiment with a single stock name, whenever buy signal appeared, the stocks of the name were bought as many as possible with the available cash and when the sell condition appeared (when the profit rate became more than predetermined level or stock holding length got beyond the predefined number of days), all the stocks were sold. It is a very simple trading with buy first then sell.

The trading system employs the technical timing rule called SP-method [13] with the related parameters. Chromosome was organized to have a set of parameters, which specify how to run the trading system. They were optimized in the GA procedure. Call them as genes of Gene0, Gene1, Gene2 and Gene3, Gene4, and Gene5. They were

1) Gene0 (SP-wave rate: SP%),
2) Gene1 (maximum SP-minus change rate: SP%-),
3) Gene2 (maximum number of days to hold bought stocks),
4) Gene3 (minimum up rate to sell bought stocks),
5) Gene4 (minimum down shooting speed), and
6) Gene5 (maximum price level).

For the model with concurrently monitoring multiple stock names, more than one name could show buy timing at a time. In the case, the stock name with the smallest SP-minus change rate (explained later) was chosen.

IV. COMPUTED INDEXES, FITNESS, AND GA PARAMETERS

In stock trading, an ordinary criterion for fitness is simple. It is to maximize the profit or the return. In the experiments, the total return (TRN), the amount of cash at the end of simulation divided by the initial cash at the end of simulation, was used as fitness. Some other indexes were also computed to see how the experiments were done. They are the number of trades (N: total counts of trades), win counts (WCT: counts of profitable trades), win rate (WRT: WCT divided by N), average return (ARN: total of individual return divided by the number of trades). “Trade” may be used as a pair of buy and sell.

The following equations are given for the specific definitions:

$$RTN(i) = \frac{sell\_price(i)}{buy\_price(i)},$$

$$ARN = \frac{1}{N} \sum_{i=1}^{N} RTN(i),$$

$$WRT = \frac{WCT}{N},$$

where $i$ is the identifier for each trade, $buy\_price(i)$ is the price at the buy of trade $i$, $sell\_price(i)$ is the price at the sell of trade $i$ and $N$ is the total counts of trades, $WCT$ is the number of trades with plus profit and $TRN$ is the total return, that is, the final amount of cash divided by the initial amount.
of cash. The actual computation was done as the equations (1) and (4) for simplicity.

\[ TRN = \prod_{i=1}^{N} RTN(i) \]  

(4)

As seen by the equations (1) and (4), transaction costs were ignored.

As for the optimization process, an ordinary GA was employed. The GA parameters used were crossover rate: 0.7; mutation rate: 0.1; population size 20; generation length 200 and elitisms. The population size was rather set to the small number for fast processing of large size data but within a generally allowable range. Those specific parameters were arbitrarily chosen and there is not a particular reason for them. Since the parameters of population size 20 (rather small) and generation length 200 (rather short) were also arbitrarily chosen, hitting global optimal point might not necessarily possible. Optimization to some extent was considered acceptable for this study.

V. SWITCHING STRATEGY

In the trade model of switching from one stock to another, the selection policy for next stock to buy becomes very important. There might possibly be many ways to select next stock. What was most effective was yet to study for the future including this study. However, a very simple method was used in this study. That is, the stock with the smallest SP-minus wave rate was selected from among the stocks showing buy timing. The very small values (minus) of SP-minus rate were supposed to show a bottom of stock price like moving average method but somewhat differently.

VI. EXPERIMENTS

The details of the two experiments were as follows.

A. Experiment 1

In the experiments, no switching strategy could be employed, because only one stock name was used for each simulation. Ten trading simulations were done independently, one for each of ten names of stock. The stock names are 1) Shimizu (a construction company), 2) Itoham Foods, 3) Oji Paper, 4) Sumitomo Chemical, 5) Asahi Glass, 6) Sumitomo Heavy Industry, 7) Toshiba, 8) Matsushita Electric works, 9) Sumitomo (Commerce), and 10) Yusen (a see transformation company).

In each simulation, training was first done by using in-sample data of each stock name, and then test was made using the out-of-sample data of the same name. The results are shown in Table 1. N-No. shows stock name number as given previously. As seen, the values of WCT and N in testing are much lower than those in training. In the lines with Stock name No. of 6 and 8, there are symbols of “NA” which mean there were no opportunities for trading in the test period. These results seem showing typical overfitting. The average counts of successful trade were 5.7 in training and those in testing were only 1.7. However, the most important result, the return, TRN for the out-of-sample period was 1.207 as the average. It is not a bad result. Additionally, the win rates (WRT) also were surprisingly good, i.e., 0.921 for in-sample period and 0.811 for out-of-sample period.

At the right end of the table, start and end prices of each stock are shown as well as each of the returns by buy and hold strategy (BHRTN). Start-P is price at the start of the period and End-P is the price at the end of out-of-sample period. On the bottom of BHRTN is the average of BHRTNs. It is 1.79. The average return of Experiment 1 is 1.207. This value 1.207 seemed good until the average return of the ten stocks turned out to be 1.79. Nikkei average during 2003 to 2004 increased about 30%. Looking the start and end prices of No. 6 stock, the return of buy and hold is 5.69, extremely high value. The price of the stock rose up so steep in the period. Without the stock, the average return of the remaining nine is about 1.37. It is still far better than the average return of this experiment.

B. Experiment 2

In this experiment, the same stock data of 10 stock names of Experiment 1 were used. By concurrently monitoring the stocks of 10 names [12] for the in-sample period, just one trading rule was generated instead of one rule for each name (Generating a technical rule for each stock name and concurrently monitoring each stock with each corresponding rule in testing is possibly better, though. However, it is time consuming). Of course, the specified fitness was the same as that of Experiment 1: TRN. The results are shown in Table 2. The same experimental procedure was executed ten times. At the bottom of the table, the averages are shown. “NA” in the table suggests that the system looked for a trading signal but in vain. It is possibly showing overfitting. In Experiment 2, the trading opportunities (WCT and N) were increased in both of the training and testing. Both of WRT and ARN of testing show still good numbers, 0.837 and 1.068. The average of the total returns (TRN) was also increased from 1.207 to 1.488 in Experiment 2. The average total return of 1.488 seems to be very good compared to that of Nikkei Average (Japanese representative stock index) that increased about 30% during the years 2003 to 2004. However, the average of buy and hold return of the ten stocks were about 1.79, far better the 1.488. So the result of the switching model is not surprisingly good. However, as mentioned previously, the return of 1.488 is better than the average return 1.37 which is without No. 6 stock (Sumitomo Heavy Industry) of which BHRTN is 5.69. This company did not give any opportunity of trading in the out-of-sample period of Experiment 1. It is probably because of too steep gain of the price. It is not clear, though, that same situation occurred in Experiment 2.

As seen in the table, when poor performances were seen for WCT in training (Exp. No. 1, 3, and 6), there were poor performances for TRN in testing, that is, TRNs were all 1.00. Those simulations of Exp. No.1, 3 and 6 in Table 2, were with WCTs of all 2, very small in training. It seems that in these simulations, the GA processes had fallen in hitting local optimal points. This might be possibly suggesting that more optimization in training with more generations or larger population might bring better performance for TRN in testing.
VII. EVALUATION OF STOCK SWITCHING

It is very difficult to evaluate the results of Experiment 2, the trading model of switching one stock to another. The average value of \( TRN \) should be considered very good, but not good compared with the average of buy and hold returns 1.79. However, it is true that the switching strategy brought about great improvement. That is, average return of Experiment 1 was 1.207, but that of Experiment 2 was 1.488, which is a great improvement. This was brought about by concurrent processing of 10 stock names. Experiments with more stock names are strongly desired. It is also desired to examine how the model behaves when the market is in a down trend.

VIII. CONCLUDING REMARKS

The model with switching from one stock to another showed the big performance improvement in experimental results of total return as the data size increased despite the short generation and small size population. This seems to imply that the model has a great hope.

REFERENCES


Fig. 1. Simulation procedure using stock switching
Table 1. Performance results for Experiment 1.

<table>
<thead>
<tr>
<th></th>
<th>Category 1</th>
<th>Category 2</th>
<th>Category 3</th>
<th>Category 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter 1</td>
<td>Value 1</td>
<td>Value 2</td>
<td>Value 3</td>
<td>Value 4</td>
</tr>
<tr>
<td>Parameter 2</td>
<td>Value 1</td>
<td>Value 2</td>
<td>Value 3</td>
<td>Value 4</td>
</tr>
<tr>
<td>Parameter 3</td>
<td>Value 1</td>
<td>Value 2</td>
<td>Value 3</td>
<td>Value 4</td>
</tr>
<tr>
<td>Parameter 4</td>
<td>Value 1</td>
<td>Value 2</td>
<td>Value 3</td>
<td>Value 4</td>
</tr>
</tbody>
</table>

Table 2. Performance results for Experiment 2.

<table>
<thead>
<tr>
<th></th>
<th>Category 1</th>
<th>Category 2</th>
<th>Category 3</th>
<th>Category 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter 1</td>
<td>Value 1</td>
<td>Value 2</td>
<td>Value 3</td>
<td>Value 4</td>
</tr>
<tr>
<td>Parameter 2</td>
<td>Value 1</td>
<td>Value 2</td>
<td>Value 3</td>
<td>Value 4</td>
</tr>
<tr>
<td>Parameter 3</td>
<td>Value 1</td>
<td>Value 2</td>
<td>Value 3</td>
<td>Value 4</td>
</tr>
<tr>
<td>Parameter 4</td>
<td>Value 1</td>
<td>Value 2</td>
<td>Value 3</td>
<td>Value 4</td>
</tr>
</tbody>
</table>