Strategy Tree Construction and Optimization with Genetic Programming

Chi Xu[1], Jianxiong Qiao[1] and Na Jia[2]
[1]*Department of Computer Science, North China University of Technology, 5 Jinyuanzhuang Street, Beijing, China*
[2]*Institute of Science, North China University of Technology, 5 Jinyuanzhuang Street, Beijing, China*

Keywords: Artificial Intelligence, Evolutionary Algorithm, Machine Learning, Regressive Decision Rule.

Abstract: We applied genetic programming (GP) to search for a strategy in a technical analysis (TA) indicator candidate pool for stock market trading and optimized it through historical data. The method provides decision rule optimization scheme to deal with problems in the real trading in financial market, and it optimizes strategies in relatively complicated contents. GP is used to construct the condition in decision rule with different logical operations. The method has been applied to the optimization of investment strategies with good return results in simulation experiments.

1 INTRODUCTION

1.1 Background

A reasonable strategy for trading in financial markets is one of the most important topics for researchers to work with, but it is always difficult to implement the essential characteristics or complicated contents in a strategy. In the research area of machine learning, decision tree is used to take in the condition input and feed to the system for decision making. The complexity of any regressor depends on the number of inputs. Although a decision rule has the advantages, like:

- being simple to understand and interpret,
- requiring little data preparation,
- handling both numerical and categorical data,
- explaining well problems through boolean logic, and also
- being robust to validate models using statistical tests while handling large data with little time consumption.

it still has limitations on problems for optimization which has to be NP-complete under several aspect of optimality and even for simple concepts (Hyafil and Rivest, 1976; Murthy, 1998). Hence, the optimization of a strategy in a practical problem is not extremely easy to achieve. The complexity of any regressor depends on the number of inputs, and it determines both the time and space complexity and the necessary number of training examples to train such a regressor (Alpaydin, 2010).

In this paper, we applied genetic programming to optimize the regressive decision rule like strategies to deal with real trading problems. In the proposed method, each ”individual” of an evolving population encodes a candidate strategy to the given problem, and the individual is evaluated by a problem/application-oriented fitness function based on natural selection of survival and reproduction of the fittest individuals. GP forms a tree structure of the strategy with boolean logic operators on the inner nodes.

1.2 Decision Rule Optimization

The optimization of decision rules are difficult, since over-complex trees may not generalize the data properly, which is called overfitting. In addition, a decision rule cannot express well some conceptual information. Many approaches have been adopted to improve the tree structure and performances. (Blockeel and Struyf, 2002) proposed an efficient cross-validation algorithm to reduce the overhead in the induction process of a logic programming tree, and conducted various experiments on different data sets to evaluate the optimization performance. (Bennett, 1994) proposed a non-greedy decision rule algorithm to construct decision rules and to update existing decision rules. A global tree optimization is used to explicitly consider all decisions in the tree concurrently. (Suarez and Lutsko, 1999) proposed a fuzzy

Xu C., Qiao J. and Jia N..
Strategy Tree Construction and Optimization with Genetic Programming.
DOI: 10.5220/0004201104250428
In *Proceedings of the 5th International Conference on Agents and Artificial Intelligence* (ICAART-2013), pages 425-428
ISBN: 978-989-8565-39-6

decision rule which transforms the tree into a powerful functional approximation while remaining easily interpretable, and a global optimization algorithm fixes the parameters of the fuzzy splits. (Mookerjee and Mannino, 1997) introduced a sequential decision model to optimize an expert system when the cost or time to collect inputs is significant and inputs are not known until the system operates. (Liang et al., 2010) used a decision rule to handle uncertain concepts, so the dynamic data stream with uncertain numerical attributes can be classified efficiently.

The paper is organized as follow. We provide a survey on related works in Section II. In Section III, our proposed a model is discussed. In Section IV, we provide two practical scenarios, investing in financial market and gaming in chess competition, as the test beds for the performance evaluation of the proposed method. In Section V, we draw a conclusion and make some suggestions on our future work.

2 METHODOLOGY

In stock market, people usually use technical analysis (TA) indicators to analyze the trend of market.In this paper, we mainly use indicators including MA, CCI, RSI, KDJ and MACD to build strategy trees, and to evaluate fitness values of each strategy tree.

2.1 System Architecture

Our system consists of a GP Engine, a control core and a market data input module. The system architecture is given in figure 1.

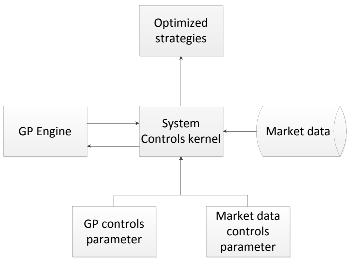

Figure 1: System architecture.

In our system, the function of control kernel is to control whole system's working, the parameters such as population size, crossover rate, mutation rate and period are defined by users before the system gets launched. After users submitted control parameters, the control kernel transmits control parameters to GP engine for operation. Besides, it receives market data from market data module, and users can choose whether to train several strategy trees using sample data, or use some specific strategy trees to perform out-of-sample testing using unseen data or live data. In training, the control kernel calculates fitness value of each individual in population. After calculated fitness, it determines whether to do more operation or to terminate the system. In testing, the control kernel also calculates the fitness values, then it summarizes the results of calculation and draws some diagram to show strategy trees' performance.

In the system, the outcomes of each generation of population, and the program running condition are recoded in the 'System log', for the information of every aspects about system running. People can use it to do more analyze or collect good individuals.

2.2 Strategy Trees

In our system, strategy trees should be constructed by allowing the GP engine to combine several technical indicator-based rules (see the appendix) with boolean operators, AND, OR and XOR. According to the corresponding rule, each indicator can be evaluated as two values as TRUE and FALSE. Once a strategy tree has been constructed, it can represent a trading rule, and the tree also can be evaluated as TRUE and otherwise FALSE. For example, a strategy tree is given in figure 2. In this tree, it represents the rule of the form:

- IF RULE RSI IS *TRUE* OR RULE CCI IS *FALSE* XOR (RULE MACD IS *TRUE* AND RULE KDJ IS *TRUE*) THEN **OPERATION**

Where RULE RSI is TRUE could be, for example, the value of RSI is above 70, then it stand for overbought, RULE RSI would. If the value of RSI is below 30, then it stand for oversold, therefore RULE RSI would be FALSE. Similarly, RULE CCI and RULE MACD should meet their conditions using similar judgement methods.

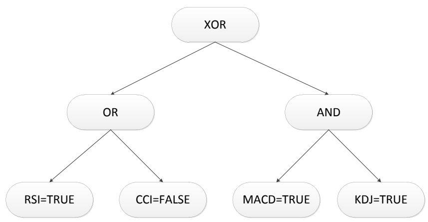

Figure 2: Strategy tree.

Each individual in population consists of two strategy trees, one is buy tree, while another one is sell tree. Evaluating the tree's value, when buy tree's value is true, it emits buy signal, the system will perform buy operation. While sell tree's value is true, it emits sell signal, the system will perform sell operation. a individual is given in figure 3

In the individual, the buy strategy can be expressed as table 1, and the sell strategy can be expressed as table 2. From the table 1 we can see that the buy rule is ”BUY IF CCI=FALSE OR MACD=TRUE AND RSI=TRUE”. While from the table 2, we can see that the sell rule is ”SELL IF RSI=TRUE OR CCI=FALSE AND KDJ=TRUE”. According to these two rules, the system can perform it on real data sets and then calculate the fitness value and the stability of the individual.

Table 1: Buy tree expression.

CCI	CONNECTOR	MACD	CONNECTOR	RSI
TRUE	OR	TRUE	AND	FALSE

Table 2: Sell tree expression.

RSI	CONNECTOR	CCI	CONNECTOR	KDJ
TRUE	OR	FALSE	XOR	TRUE

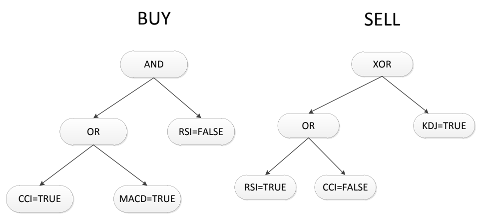

Figure 3: Individual form.

3 RESULTS

In this section, we choose 5 stocks randomly, and use 50% of data to select a strategy, while the left 50% data to test strategies' performance. By applying this process a strategy is tested whether it fits for the market sets in other periods.

The primary parameters as table 3:

Table 3: GP Parameters.

parameter	value
population	100
max iteration	15
max tree depth	5
regeneration rate	0.05
crossover rate	0.75
initial fund	\$10000

In the first sample, the market data is from Patterson-UTI Energy Inc. We use the data from *1993-11-02* to *2003-01-07*, to train strategies. The training result is shown in figure 4.

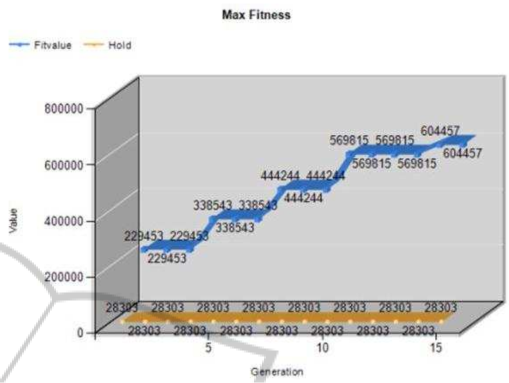

Figure 4: Training result.

According to the figure 4, during the process of training, the optimal strategies' fitness value is increased with iterations' increasing. In other words, with the increasing of iterations, the optimal strategies can perform better and get more returns. This result shows that the system is effective and more useful strategies have been found.

After training, we use the data from *2003-01-07* to *2012-04-17* to test strategies which was found in the training process. The testing result is shown in figure 5.

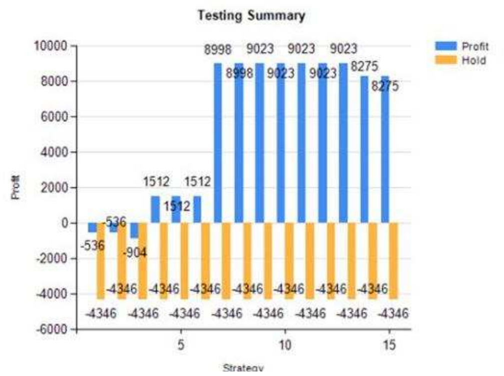

Figure 5: Testing result.

According to the figure 5, we can conclude that, Every generation of the best strategies' performance is different, but comparing with the BUY-and-HOLD strategy's -43.36% of yield, the best strategy in the system can reach 90.34% . This evidence shows that in this stock data sets, the strategies trained in the system have better performance than those of BUY-and-HOLD strategy, it can work at same stock but different period of data sets.

In the below, we choose more stocks to testing the performance of system, the result is given as table 4.

Table 4: Testing of one stock but different period.

Stock name	Training	Testing	Best return	Buy-and-Hold
AMD	1983-03-21~ 2000-03-01	2000-03-01~ 2012-05-06	147.53%	-84.22%
Dell Inc.	1988-08-17~ 2000-01-20	2000-01-20~ 2012-05-16	-28.36%	-65.66%
HP	1987-11-05~ 2000-11-22	2000-11-22~ 2012-05-16	37.71%	35.21%
FORD	1977-01-03~ 1990-08-14	1990-08-14~ 2012-05-25	215.18%	-71.92%
INTC	1986-07-09~ 2000-04-12	2000-04-12~ 2012-05-16	-46.62%	-78.25%

The table 4 shows that, in this section of testing, in most cases, the strategies which generated by the optimization system have better performance than those of buy-and-hold strategy. Besides, the performance of the strategies can keep at a stable level relatively.

4 CONCLUSIONS

GP is applied to automatically produce various trading decisions composing of logic operations for TA indicators, and historical data is used to optimize the strategy return performance.

Simulation experiments leads us to the conclusion that GP is effective in searching for strategies with high return performances. With the genetic operations in GP, good performance strategy with complicated contents can be generated.

The applications of GP to investment problems lead us say that the such a system could be adopted into solving different targeted problems with the change of various conditions. Its problem solving ability is satisfactory for our future researches.

ACKNOWLEDGEMENTS

This paper is partially supported by the National Natural Science Foundation of China under Grant #61111130121/F020202.

REFERENCES

Alpaydin, E. (2010). *Introduction to Machine Learning*. MIT Press, Cambridge, Mass.

Bennett, K. (1994). Global tree optimization: A non-greedy decision tree algorithm. In *Computing Science and Statistics*, pages 156–160.

Blockeel, H. and Struyf, J. (2002). Efficient algorithms for decision tree cross-validation. *Journal of Machines Learning Research*, 3:621–650.

Hyafil, L. and Rivest, R. (1976). Constructing optimal binary decision trees is np-complete. *Information Processing Letters*, 5(1:15 17).

Liang, C., Zhang, Y., and Song, Q. (2010). Decision tree for dynamic and uncertain data streams. In *JMLR 2nd Asian Conference on Machine Learning (ACML2010)*, pages 209–224.

Mookerjee, V. and Mannino, M. (1997). Sequential decision models for expert system optimization. *Knowledge and Data Engineering, IEEE Transactions on*, 9(5):675 –687.

Murthy, S. (1998). Automatic construction of decision trees from data: A multidisciplinary survey. In *Data Mining and Knowledge Discovery*.

Suarez, A. and Lutsko, J. (1999). Globally optimal fuzzy decision trees for classification and regression. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(12):1297 –1311.