Applied Soft Computing 36 (2015) 357–367

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Ensemble learning of rule-based evolutionary algorithm using multi-layer perceptron for supporting decisions in stock trading problems

Shingo Mabu*, Masanao Obayashi, Takashi Kuremoto

Graduate School of Science and Engineering, Yamaguchi University, Tokiwadai 2-16-1, Ube, Yamaguchi 755-8611, Japan

ARTICLE INFO

Article history:
Received 18 July 2014
Received in revised form 30 June 2015
Accepted 7 July 2015
Available online 30 July 2015

Keywords:
Ensemble learning
Evolutionary computation
Rule-based algorithm
Multi-layer perceptron
Trend judgment
Stock trading

ABSTRACT

Classification is a major research field in pattern recognition and many methods have been proposed to enhance the generalization ability of classification. Ensemble learning is one of the methods which enhance the classification ability by creating several classifiers and making decisions by combining their classification results. On the other hand, when we consider stock trading problems, trends of the markets are very important to decide to buy and sell stocks. In this case, the combinations of trading rules that can adapt to various kinds of trends are effective to judge the good timing of buying and selling. Therefore, in this paper, to enhance the performance of the stock trading system, ensemble learning mechanism of rule-based evolutionary algorithm using multi-layer perceptron (MLP) is proposed, where several rule pools for stock trading are created by rule-based evolutionary algorithm, and effective rule pools are adaptively selected by MLP and the selected rule pools cooperatively make decisions of stock trading. In the simulations, it is clarified that the proposed method shows higher profits or lower losses than the method without ensemble learning and buy&hold.

1. Introduction

Stock markets are quite complicated real world environments where a huge number of factors influences the movement of the markets. Basically, there are two methods to predict the trends of markets and decide the timing of buying and selling stocks, i.e., fundamental analysis and technical analysis. Fundamental analysis is based on the financial statement reported by companies, economic trends of domestic and international environments, international relationships, and so on. Technical analysis is based on numerical analysis using the past movement of the stock prices, where many technical indexes such as moving average, golden cross, dead cross, etc. are used for stock trading decisions. Stock trading models and stock price prediction based on computational intelligence basically belong to technical analysis, and many methods using softcomputing and machine learning algorithms such as neural networks [1], genetic algorithms [2], genetic programming [3], support vector machines [4] have been proposed. The proposed method in this paper also belongs to technical analysis.

Due to the high complexity of stock markets, it is very difficult for single centralized system to represent and predict the movements of the markets because a huge number of rules covering the possible situations must be contained. Divide and conquer is a famous concept to solve a complicated problem by dividing it to several small problems [5]. Several expert programs for each small problem cooperatively solve their target problems, which contributes to making the whole system be compact. In the research of pattern recognition, many methods based on divide and conquer have been proposed to enhance the generalization ability of classification. One of the famous mechanisms is ensemble learning, where several classifiers are created and the classification is made by combining the results generated by the classifiers. Bootstrap aggregating (Bagging) [6] is one of the ensemble learning methods which creates several classifiers using bootstrapping samples [7], and makes classification by a majority vote. Adaptive boosting (AdaBoost) [8] creates several classifiers one by one considering the learning results of the previous classifiers. AdaBoost gives large weights on misclassified data and small weights on correctly classified data, therefore, the classifiers created later phase will execute learning concentrating on the misclassified data. Random forests [9] enhances Bagging algorithm and creates several decision trees by randomly selecting the fixed number of attributes for learning. In this paper, to apply the concept of ensemble learning to a stock

* Corresponding author. Tel.: +81 836 85 9519; fax: +81 836 85 9519.
E-mail address: mabu@yamaguchi-u.ac.jp (S. Mabu).

http://dx.doi.org/10.1016/j.asoc.2015.07.020
1568-4946/© 2015 Elsevier B.V. All rights reserved.

trading model, an algorithm that selects good combinations of classifiers by multi-layer perceptron (MLP) considering the trends of stock markets is proposed.

In the proposed method, a large number of stock trading rules combining technical indexes are extracted by a graph based evolutionary algorithm named genetic network programming (GNP) [10,11]. GNP has been proposed to create programs mainly for dynamic environments using directed graph structures. Directed graph structures have some advantages comparing to tree structures used in genetic programming (GP) [12], for example, compact structures, repetitive process generation, and adaptability to non-Markovian process. GNP has three basic components: start node, judgment node and processing node, and combines them to make many if-then rules. There have been studies on knowledge extraction from data and representing the extracted knowledge as if–then rules. For example, in [13], a knowledge extraction method using probabilistic and fuzzy logical reasoning is proposed to generate if–then rules, and in [14], if–then rule extraction method from noisy time series data is proposed. The unique points of the proposed method in this paper are the if–then rule extraction utilizing the distinguished representation abilities of graph structures and its extension using MLP for realizing effective ensemble learning.

GNP has been applied to many applications such as elevator group supervisory control systems [15], data mining [16,17], network intrusion detection systems [18], stock trading models [19]. The algorithm used in this paper is a rule-based GNP [20] which extracts a large number of rules through the evolutionary process. General evolutionary computation basically aims to evolve individuals, and the best individuals obtained in the last generation become the solutions to the problems. However, rule-based GNP extracts rules from the elite individuals every generation, therefore, the rule extraction is carried out by all the elite individuals throughout the generations. This is the different point from the general framework of evolutionary computation.

In the rule extraction using GNP, the training period is divided into several sub-periods, and the expert rules for each sub-period are generated separately, which results in making several rule pools, i.e., classifiers, for ensemble learning. Then, MLP is used to calculate the similarity degrees of the stock data in the testing phase to the sub-periods in the training phase. In summary, MLP effectively selects good rule pools for stock trading in the current situation according to the similarity/matching calculation between training and testing data. The combination of rule-based decision making model, where a large number of rules are contained for adapting to various kinds of situations, and the automatic rule pool (classifier) selection by MLP can effectively make trading decisions.

There have been many studies on financial trading systems using artificial neural networks, evolutionary computation, optimization algorithms and so on. In [21], particle swarm optimization (PSO) is applied to the decision making in stock trading problems. PSO optimizes the weights on several technical indexes using multi-objective evaluation functions. In [22], an artificial neural network (ANN) is applied to a model that predicts future trading signals. In addition, a dynamic threshold method is proposed to judge whether the trading signals generated by ANN satisfy the conditions of buying or selling decisions. In [23], gene expression programming (GEP) is applied to the trading on mutual funds. GEP is an evolutionary algorithm which extends GP by using linear gene structures, and can explicitly represent trading rules by evolved tree structures. Comparing to the above methods, there are different features in the proposed method. First, the proposed method extracts a large number of association rules and stores them in the rule pools, therefore, various kinds of knowledge on the stock market movements can be saved for decision making. Both the proposed method and GEP [23] can explicitly represent trading rules, however, the proposed method is a rule-based system, while GEP is a tree structure-based system. ANN model [22] shows distinguished generalization ability to predict trading signals, but it is a black box model, thus it is difficult to explicitly show the rules of buying and selling decisions. Second, the proposed method uses only the important rules for the decision in the current situation by introducing an ensemble learning with MLP. In [21], PSO determines the importance of each technical index by adjusting weights, but it does not adapt to the changes of the trends in stock markets.

This paper is organized as follows. Section 2 reviews the rule extraction algorithm using rule-based GNP. Section 3 explains the rule pool generation and how to make MLP for selecting good rule pools depending on the stock market conditions. Section 4 introduces the simulation environments of the stock trading and analyzes the simulation results, and finally Section 5 is devoted to conclusions.

2. Review of rule-based genetic network programming

GNP has been proposed as an extension of GP in terms of the gene structure [11]. The gene structure of GNP is represented by a directed graph, which shows some advantages over tree structures of GP such as reusability of nodes, compact structure and decision making considering past events. In this section, the representation of the programs, how to execute and evolve the programs, and how to make rule pools are explained. More detailed explanation is described in [19].

2.1. Basic structure

The basic program structure is shown in Fig. 1, where there are three major components: start node, judgment node and processing node. Each judgment node has a if–then branch decision function and each processing node has an action function. In the stock trading problem, for example, the judgment nodes in Fig. 1 examine technical indexes generally used in the technical analysis and select one branch to the next node according to the judgment result such as {A = high, B = low}. The processing nodes determine buying and selling actions. The role of the start node is to determine the first node to be executed. Therefore, the node transition starts from the start node, and judgment and processing nodes are sequentially executed according to the connections between nodes. If the connections are optimized by the evolutionary algorithm, good action rules can be obtained.

In this paper, an extended structure of GNP is used to make programs, that is, GNP with reinforcement learning (GNP-RL) [11]. The feature of GNP-RL is that there are several subnodes in each judgment and processing node (Fig. 2). Each subnode has its own function and *Q*-value [24], and in the node transition, reinforcement

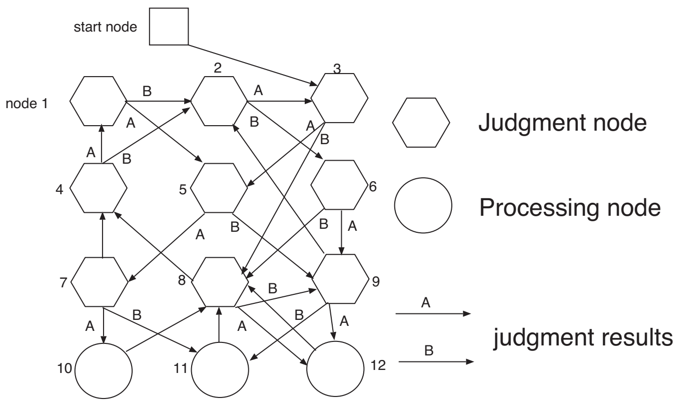

Fig. 1. Basic structure of GNP.