Neurocomputing 122 (2013) 470–479

Contents lists available at ScienceDirect

Neurocomputing

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

A cellular learning automata model of investment behavior in the stock market

Milad Mozafari *, Rahim Alizadeh

Department of Mathematics, Shahed University, P.O. Box 18151-159, Tehran, Iran

ARTICLE INFO

Article history:
Received 14 March 2013
Received in revised form
11 June 2013
Accepted 14 June 2013
Communicated by W.S. Hong
Available online 30 June 2013

Keywords:
Cellular automata
Learning automata
Stock market
Investment behavior

ABSTRACT

In this paper, we present a cellular learning automata model for the investment behavior in the stock market. In this model, investors decide to hold, buy, or sell the stocks based on the evolution rules and learn how much they can trust on other investors based on the learning rules. We analyze the effects of imitation, reliability and macrofactors on the stock market and compare the obtained results with the previous approach that is based on cellular automata.

1. Introduction

Stock markets are complex systems, which are composed of a large number of nonlinear-coupled subsystems. Many scholars have been trying to discover their mechanism and explain the principle of these markets behaviors. In these systems, investors use their own analysis and formulas for investing and decide according to their own perception of the events. In all of this analysis, they consider behaviors of other investors and events about the stock. Besides physical parameters, psychological factors affect the macroeconomic behavior and cause crisis in finance markets. These various factors make impossible the accurate prediction in a stock market. Until now, different approaches for modeling different parts of stock markets have been studied. One of these approaches is Genetic Programming (GP). Genetic Programming is a generalization of genetic algorithms and tries to evolve a population of programs to the new generation of programs in order to get better results. Genetic operators such as crossover and mutation are also available here for evolving a population and a fitness function (measure) is needed to select individuals for the next generation. The general idea behind genetic programming is to find a combination of functions that will give the best results. For example, the authors in [14] used two different methods of GP: Multi-Expression Programming (MEP) and Linear Genetic Programming (LGP) for the prediction of stock index (index of market prices of a particular group of stocks). MEP uses a linear representation of chromosomes and it has the ability of storing multiple solutions for a problem in one chromosome, and LGP acts on linear genomes that consist of expressions of an imperative language. They also used the ensemble of these methods to improve the accuracy of results in their paper; every chromosome represents one trading rule and each gene stands for a parameter value of a trading rule. Another useful tool for modeling complex systems is Artificial Neural Network (ANN). ANNs are inspired from human neural system and try to simulate brain abilities. In these networks, nodes are considered as neurons and arcs between them, as input or output channels for neuron signals; these channels are weighted so that the impact of signals on each neuron can be controlled. ANNs can handle lots of parameters with complex dependencies and their accuracy can be improved by learning (updating and tuning weights) from training samples. There are several types of ANNs such as feed forward networks, radial basis function (RBF) networks, multi-layer perception (MLP), etc. that can be used in different types of problems. In [9], performance and accuracy of multi-layer feed forward, generalized, and probabilistic networks in predicting stock returns have been compared against classical linear regression methods and the predictive relationships between different economic variables have been evaluated by information gain techniques in machine learning for data mining. You can find other works using ANNs in [8,22,32,33]. Decision trees are other kinds of tools that can model dynamic and complex decision processes. Decision trees are trees in which nodes are decision

* Corresponding author. Tel.: +98 919 2182516.
E-mail addresses: milad.mozafari69@outlook.com, milad.mozafari69@gmail.com (M. Mozafari), alizadeh@shahed.ac.ir (R. Alizadeh).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.06.002