Expert Systems With Applications 46 (2016) 307–315

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Robust technical trading strategies using GP for algorithmic portfolio selection

José Manuel Berutich[a,*], Francisco López[a], Francisco Luna[a], David Quintana[b]

[a] Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, Málaga, Spain
[b] Departamento de Informática, Universidad Carlos III de Madrid, Leganés, Madrid, Spain

ARTICLE INFO

Keywords:
Genetic programming
Algorithmic trading
Portfolio management
Trading rule
Finance

ABSTRACT

This paper presents a Robust Genetic Programming approach for discovering profitable trading rules which are used to manage a portfolio of stocks from the Spanish market. The investigated method is used to determine potential buy and sell conditions for stocks, aiming to yield robust solutions able to withstand extreme market conditions, while producing high returns at a minimal risk. One of the biggest challenges GP evolved solutions face is over-fitting. GP trading rules need to have similar performance when tested with new data in order to be deployed in a real situation. We explore a random sampling method (RSFGP) which instead of calculating the fitness over the whole dataset, calculates it on randomly selected segments. This method shows improved robustness and out-of-sample results compared to standard genetic programming (SGP) and a volatility adjusted fitness (VAFGP). Trading strategies (TS) are evolved using financial metrics like the volatility, CAPM alpha and beta, and the Sharpe ratio alongside other Technical Indicators (TI) to find the best investment strategy. These strategies are evaluated using 21 of the most liquid stocks of the Spanish market. The achieved results clearly outperform Buy&Hold, SGP and VAFGP. Additionally, the solutions obtained with the training data during the experiments clearly show during testing robustness to step market declines as seen during the European sovereign debt crisis experienced recently in Spain. In this paper the solutions learned were able to operate for prolonged periods, which demonstrated the validity and robustness of the rules learned, which are able to operate continuously and with minimal human intervention. To sum up, the developed method is able to evolve TSs suitable for all market conditions with promising results, which suggests great potential in the method generalization capabilities. The use of financial metrics alongside popular TI enables the system to increase the stock return while proving resilient through time. The RSFGP system is able to cope with different types of markets achieving a portfolio return of 31.81% for the testing period 2009–2013 in the Spanish market, having the IBEX35 index returned 2.67%.

1. Introduction

Algorithmic trading using evolutionary computation has been a hot topic of research in the recent years for academics from both finance and soft-computing domains with a large number of published research articles (Aguilar-Rivera, Valenzuela-Rendón, & Rodríguez-Ortiz, 2015; Hu et al., 2015). Normally, it is very hard for a simple investor to optimize his investments without requiring the skills of financial advisers. The main goal of this work is to provide an application which helps investors achieve a significant profit on buying and selling financial securities in an automatic way without requiring the help of portfolio managers.

Selecting the most promising securities is a very difficult problem for Genetic Programming (GP) due to the dynamic and stochastic nature of the markets and the vast quantities of data that needs to be analyzed.

In this paper we are interested in developing robust technical trading rules using GP which can replace the intervention of human money managers, and be applied systematically to manage a portfolio of stocks. One of the most important benefits of systematic trading is that it helps to remove emotional decision making from the investment process, as emotions can easily overwhelm rational decision making. This can be lessened to a large extent by having a system that automatically makes the decisions for you.

Another important benefit of systematic strategies is that they can be tested on historical data. This ability to simulate a strategy is one of the biggest benefits of systematic trading. Back-testing tells you how well the strategy would have done in the past. While back-tested performance does not guarantee future results, it can be very helpful

* Corresponding author. Tel.: +34619266536.
E-mail addresses: jmberutich@gmail.com (J.M. Berutich), valverde@lcc.uma.es (F. López), flv@lcc.uma.es (F. Luna), dquintan@inf.uc3m.es (D. Quintana).

http://dx.doi.org/10.1016/j.eswa.2015.10.040
0957-4174/

when evaluating potential strategies. Back-tested results can be used to filter strategies that either do not suit the required investment style or are not likely to meet risk/return performance goals.

One of the biggest difficulties GP evolved strategies face is over-fitting. While solutions perform well in the training dataset, once they are tested out-of-sample with new data, their performance is seriously degraded. We will explore a method for reducing over-fitting of GP solutions. Robust GP solutions should display similar behaviour during out-of-sample testing as during training. Moreover, GP investment strategies need to be robust in order to be deployable in a real portfolio management situation.

1.1. Robustness

The term "*robust*" has many definitions depending on the author. It can be broadly defined as the ability of a system to preserve its functionality despite internal *(genotypic robustness)* or external perturbations *(phenotypic robustness)* (Branke, 1998; Soule, 2003).

1.1.1. Genotypic robustness

Genotypic robustness aims at achieving insensitivity of fitness to perturbations from genetic operators. Soule (2003) finds that the *code bloat* phenomenon in GP, where an increase of the size of the trees does not result in fitness improvement, is a redundancy mechanism. Trees grow *introns* which safeguard valuable code and protect it against loss during crossover or mutation. Even though this approach favors broad plateaus instead of peaks of high fitness, it is of negligible use when the surface of the search space changes.

1.1.2. Phenotypic robustness

Phenotypic robustness deals with resilience to external changes and can be categorized into:

- *Generalization Robustness (GR):* Robustness as the generalization ability of evolved solutions. From a machine learning standpoint, it is the predictive accuracy of a learner for new unseen cases. The objective here is reducing over-fitting and producing solutions whose performance is similar for both in-sample and out-of-sample datasets (Kushchu, 2002).
- *Environmental Robustness (ER):* Robustness to external environmental perturbations. Financial markets suffer abrupt structural changes which tend to persist in time. (Granger & Hyung, 2004). Robust GP solutions should withstand periods of extreme volatility and trend change. Yan and Clack (2010) propose training on three distinct scenarios; a *bull* or rising market, a *bear* or falling market and a volatile *sideways* market.
- *Robustness to Noise (RN):* Robustness to noise inherent in the data or the readings produced by the system (Kitano, 2004).
- *Self Repair (SRS):* Robustness as the ability to self-repair after severe phonotypic damage (Bowers, 2006).

In this paper we are concerned with the generalization and environmental robustness of evolved solutions. We explore how we can design solutions that display similar performance for both in-sample and out-of-sample data, as well as solutions that can resist abrupt trend changes and extreme volatility periods.

Our approach is substantially different to previous work and is centered around how we calculate the fitness function. Our main contribution lies in evaluating the fitness using a random sampling method which will explain later in Section 3.4.

The main contributions of this study can be summarized as follows: (1) The use of a random sampling mechanism to divide the time series of a basket of stocks into segments without requiring user intervention, or any unsupervised machine learning method to group segments into the distinct market conditions (bull, bear and sideways markets). (2) The use of a robust fitness function that uses all sampled segments to calculate an overall fitness score across random market conditions reducing over-fitting of solutions. (3) The use of different financial metrics never used together with technical indicators in previous research such as the returns, the moving average of returns, the Capital Asset Pricing Method (CAPM) alpha and beta, the Sharpe ratio and the volatility of the stocks calculated over different period lengths. (See Table 2).

Consequently, in this study a robust GP evolutionary approach will be presented to automate buying and selling decisions in order to maximize the Sterling ratio (total return divided by maximum drawdown). The proposed method will be tested on a basket of 21 stocks from the Spanish market using 13 years of daily price data and compared to the IBEX35 market index, the results will be analyzed and some possible conclusions will be discussed.

The remainder of the paper will be organized continuing in the next section with the most relevant previous work followed by the algorithmic approach, experiments performed and discussion of the results obtained. We finalize with our conclusions and future work.

2. Related work

GP was first employed by Allen and Karjalainen (1999) for technical trading rule discovery. The dataset used in their experiments was the S&P 500 index using daily prices from 1928 to 1995. Their results demonstrated that although GP could find profitable trading rules, it failed to produce excess-returns over the passive strategy of "*Buy & Hold*" (B&H), which consists in buying on the first evaluation day and selling on the last.

Neely (2003) extends the previous work by Allen and Karjalainen (1999) using a risk adjustment selection criteria to generate rules with the hope of improving performance. However, the results show no evidence that the rules significantly outperform B&H on a risk-adjusted basis.

Becker and Seshadri (2003) present results of GP-evolved technical trading rules, which outperform a buy-and-hold strategy on the S&P 500 after taking into account transaction costs. They introduce several changes to the original work of Allen and Karjalainen (1999), which include a complexity-penalizing factor, a fitness function that considers consistency of performance, and co-evolution of separate buy and sell rules. Monthly data is used instead of daily.

Lohpetch and Corne (2009) replicate the work of Becker and Seshadri (2003) and the authors find that the results are sensitive to the data periods chosen for the experiments. Their results are improved by using a validation set, used for choosing the best rule found during training.

Mallick and Lee (2008) used GP to find trading rules on the thirty component stocks of the Dow Jones Industrial Average index. The authors find Statistical evidence of outperforming B&H in falling markets, and confirm that GP based trading rules generate a positive return under bull (rising) and bear (falling) markets.

Yan and Clack (2010) use GP for building a symbolic regression expression that measures the attractiveness of each stock; Each month a portfolio is constructed with the most attractive stocks according to the GP model. The portfolio is a market neutral long/short portfolio of Malaysian equities. The authors propose two approaches for evolving robust trading rules. First by splitting the training dataset into three extreme environment periods: up, down and sideways volatile. Secondly instead of using just one solution, a voting comity is used, formed by the three best solutions trained on each of the extreme environments. The authors show results that considerably beat the benchmark index, but the results have a significant caveat, i.e. they used a small out-of-sample period (July 1997–December 1998), which is before the training period (January 1999–December 2004). Monthly data was used to simulate portfolio, meaning at the beginning of a month the stocks which the system recommends are bought, and at the end of the month the position is reassessed.