MULTIOBJECTIVE GENETIC PROGRAMMING PERFORMANCE COMPARISON IN FINANCIAL INVESTING

Martin Jakubéci – Michal Greguš

Abstract

Multiobjective Genetic programming (MOGP) offers a way to find Pareto fronts for diverse problems. The algorithm uses operations inspired by the evolution theory to search for solutions, which have a tree representation. Many applications in the area of financial investing were presented, although the performance is mostly compared only with the basic buy and hold strategy. This strategy is very simple, the market is entered in the beginning of the period and left at the end. This is not sufficient and does not proof the efficiency of MOGP. We propose a set of different strategies to compare our implementation of MOGP with. It includes random strategy, strategies based on both technical and fundamental approaches and also buy and hold strategy. Such comparison could be done with all MOGP implementations in financial investing. Our implementation of MOGP is generating stock evaluation rules. Those rules are used in an investment strategy, which creates stock portfolios. Not only historical prices and indicators, but also different internet activity data are used as data source for training and evaluation. It's shown that MOGP is able to compete with the selected strategies and outperforms them in most of the cases.

Key words: multiobjective genetic programming, portfolio, stock

JEL Code: C52, G11

Introduction

Investors are using different investment strategies, to find stocks for their portfolio and minimize risk (Bohdalová&Šlahor, 2007). They are mostly based on historical price data, however data from the activity of millions on users on the internet can also forecast future moves on the financial markets. Recent studies show this capabilities, for example:

- Google Trends – they contain the search popularity of different terms in the Google search engine. Preis, Moat and Stanley (2013) used popularity of 98 search terms to build in-vestment strategies.

- Wikipedia page views - Moat et al. (2013) found out, that market falls are preceded by increased number of page views of financial terms and companies.
- Twitter posts – Ruiz, Hristidis, Castillo, Gionis and Jaimes (2012) found correlation of posts about companies with trade volume and also smaller correlation with stock price, which was used in a trading strategy.

We are using Google Trends and Wikipedia page views of the traded companies to rank their stock. This rank is then used to select stocks in an investment strategy. We use multiobjective genetic programming to find the model. In our previous papers, we described our multiobjective algorithm (Jakubéci, 2015a) and compared it with a set of different investment strategies (Jakubéci, 2015b). We continue our research by comparing our implementation with selected strategies in more time periods, including the crisis period and introduce the validation period, as described by Lohpetch&Corne (2011).

The problem with most of the publications dealing with evolutionary algorithms in financial investing is that they compare it only with the most basic buy and hold strategy (Potvin, Soriano &Vallée, 2004; Chen, Huang & Hong, 2014). Chen &Navet (2007) criticize the research in the area of genetic programming usage in investment strategies and suggest more pretesting. They compare strategies with random strategies and lottery training without getting good results. Exceptions are only in a few papers, Briza& Naval (2008) compare their algorithm not only with market, but also with 5 technical indicators and Lohpetch&Corne (2011) compare their algorithm with market and bonds. Overview of different evolutionary algorithms in the area of financial investing was done by Tapia &Coello (2007).

Genetic programming is an evolutionary optimization algorithm, which is searching for problem solutions. Solution is a program represented by a tree structure. The tree based solutions are formed from 2 different sets of vertices. The first group are terminal symbols, for example inputs, constants or any method calls, which do not accept any parameters. Those are leafs of the tree structure. The second set are non-terminals, or functions, that accept parameters. For example arithmetic operators, logical operators, conditions etc. They are expected to be type and run safe, so that the solutions can be executed to transform inputs to outputs. The first vertex in the tree is called root and the depth of every vertex is defined as the distance from the root. First generation of solutions is created randomly. Every next generation is created by stochastic transformation of the previous generation. Transformation is done by applying operators, which are inspired by the evolution theory. These operators are mostly selection, mutation and crossover (Potvin, Soriano &Vallée, 2004). Every next generation is expected to be better.

The quality of the solutions is evaluated by the fitness function. When dealing with multiobjective optimization, there are multiple fitness functions required, one for every objective. There are many algorithms to handle multiple objectives in evolutionary algorithms, overview was done by Ghosh &Dehuri (2005).

1 Goal

Our goal is to evaluate our implementation of genetic programming by comparing its result with a set of investment strategies in different time periods. To prove its usability, it should outperform most of the strategies in the experiments. Our hypothesis is that our algorithm will have the highest rate of return in most of the cases. We are also interested, how it will perform in the financial crisis period.

2 Methods

Genetic programming is used to generate stock evaluation rules using historical stock prices and normalized internet popularity data from Google and Wikipedia (of the company names as terms). The rule is then used for daily investing in the 30 Dow Jones Industrial Index companies. These functions were used:

- arithmetic operations: addition, subtraction, multiplication, division, negation and exponentiation,
- logical operations: conjunction, disjunction, negation,
- equality: higher, lower, equal, or any combination,
- trigonometric operations: sine, cosine,
- condition,
- list operations: lag, moving average.

We use two fitness methods:

- Rate of return (RoR) $r = \frac{R_t - R_0}{R_0}$, where R_0 is the initial portfolio value and R_0 is the final portfolio value. This is a measure of revenue.
- Standard deviation (StdDev) of average portfolio value $\sigma_i = \sqrt{\frac{1}{T}(\sum_{i=1}^{T}(r_i - \mu)^2}$, where T is the number of periods, r_i is the portfolio value at time i and μ is the average value. This is a measure of risk.

Transaction fees are ignored. Portfolio is updated daily, 10 companies with highest rank are bought and 10 companies with lowest rank are sold. SPEA2 algorithm was used to handle multiple objective, because it overcomes some issues in other algorithms. It's based on elitism, Pareto dominant solutions are kept in a separate archive with fixed size. The algorithm works this way (Zitzler, Laumanns& Thiele, 2001):

1. Initialization - Generate an initial population and create the empty archive (external set). Set t = 0.
2. Fitness assignment - Calculate fitness values of individuals in population and archive.
3. Environmental selection - Copy all nondominated individuals in population and archive to the new archive. If size of the new archive exceeds M then reduce new archive by means of the truncation operator, otherwise if size of new archive is less than N then fill new archive with dominated individuals in population and archive.
4. Termination: If t >= T or another stopping criterion is satisfied then set A to the set of decision vectors represented by the nondominated individuals in the archive. Stop.
5. Mating selection: Perform binary tournament selection with replacement on the new archive in order to fill the mating pool.
6. Variation: Apply recombination and mutation operators to the mating pool and set new population to the resulting population. Increment generation counter (t = t + 1) and go to Step 2.

Implementation was done in the C# language, which has a high performance but is still easy to use. The language integrated many features from dynamic programming, for example the expression trees, which allow working with an algorithm as a data structure. This is important for the genetic programming algorithm, because it allows modifications in the solutions and application of the evolutionary operators. The Metalinq library was used, to simplify these modifications (at http://metalinq.codeplex.com/). We used 3 data sources

- Historical prices were downloaded from Yahoo Finance, at http://finance.yahoo.com/
- Google term popularity was downloaded from Google Trends service, at http://www.google.com/trends/,

- Wikipedia article popularity was downloaded from Wikipedia article traffic statistics, at http://stats.grok.se/.

Strategies created by the genetic programming implementation were compared with a number of strategies:

- Lottery trading is doing decisions randomly. That means, that it always gives a random evaluation of a stock.
- Risk free investment is represented by 3 year US treasury bonds.
- Buy and hold strategy means that the asset is bought on the beginning of the period and sold at the end. It is the most basic strategy and it was applied to the DJI index.
- Dogs of the Dow strategy is investing to 10 companies from the DJI index with the highest dividend yield.
- Simple moving averages (SMA) is calculated as an average of previous days, when the price rises above the moving average, stock should be bought, when it falls under the moving average, it should be bought (Kirkpatrick &Dahlquist, 2010).
- Exponential moving averages (EMA) is similar to the SMA, but with decreasing effect of the older days in the calculation (Kirkpatrick &Dahlquist, 2010).
- Moving average convergence divergence (MACD) is calculated as a difference between 26-period EMA and 12-period EMA, when it crosses the signal line (EMA of MACD) from below, it is a buy signal (Kirkpatrick &Dahlquist, 2010).

3 Results

We split the data in 3 periods. First one is used for training, second one, called validating, is used for selecting best solutions from the Pareto front and the third one is used for evaluation. We used two different settings:

- Training on period 2010-2012, validation on period 2013 and two evaluations – first half of 2014 and the crisis period 2008-2009.
- Training on period 2008-2009, validation on 2010 and two evaluations – period 2011-2012 and 2013.

Pareto fronts for training and validation can be seen on fig. 1 and 2. It is obvious, that the solutions receive smaller revenue, but also risk. There are also less solutions in Pareto front of the validation period.

Fig. 1: Training Pareto front for the first setting

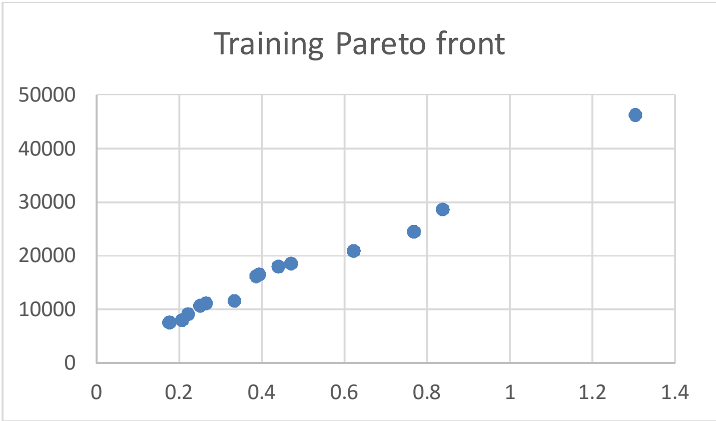
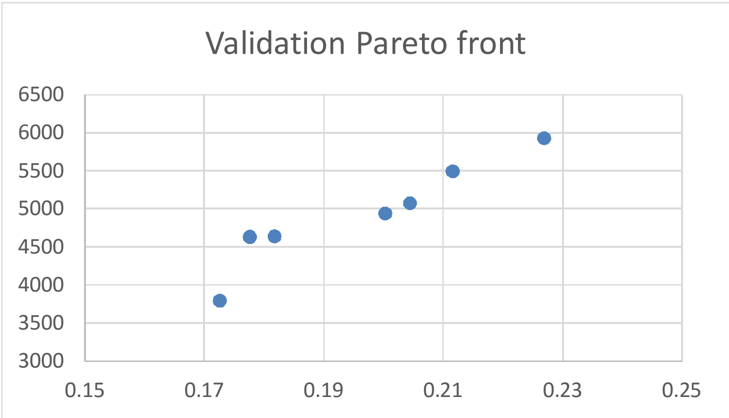


Fig. 2: Validation Pareto front for first setting



Tab. 1: First setting results in testing period

DJI	DOG	Bonds	Random	ma10	ema	macd	GP
0,29	0,52	0,07	0,13	0,12	0,05	0,10	0,55
10140	16668	447	33900	6433	8520	23398	17073

Tab. 2: First setting results in validation period

DJI	DOG	Bonds	Random	ma10	ema	macd	GP
0,23	0,21	0,08	-0,27	0,19	0,15	0,17	0,23

6090	6632	88	9097	6081	4820	5122	4899

Tab. 3: First setting results in evaluation period 1

DJI	DOG	Bonds	Random	ma10	ema	macd	GP
0,03	0,06	0,08	-0,22	-0,01	-0,01	0,02	0,06
3135	3031	80	6603	1442	1631	1692	2904

Tab. 4: First setting results in evaluation period 2

DJI	DOG	Bonds	Random	ma10	ema	macd	GP
-0,39	-0,39	0,05	-0,56	-0,26	-0,27	-0,08	-0,03
19632	17949	644	24589	14993	16806	8639	13396

Tab. 5: Second setting results in testing period

DJI	DOG	Bonds	Random	ma10	ema	macd	GP
-0,39	-0,39	0,05	-0,76	-0,26	-0,26	-0,04	0,52
19632	17949	644	28740	14993	16806	8639	16095

Tab. 6: Second setting results in validation period

DJI	DOG	Bonds	Random	ma10	ema	macd	GP
0,06	0,13	0,06	-0,32	0,02	0,12	0,14	0,07
3580	4576	207	10313	3682	5947	6467	4361

Tab. 7: Second setting results in evaluation period 1

DJI	DOG	Bonds	Random	ma10	ema	macd	GP
0,18	0,28	0,07	-0,84	0,01	0,02	0,20	0,12
6488	9496	204	22962	6797	6890	15062	5158

Tab. 8: Second setting results in evaluation period 2

DJI	DOG	Bonds	Random	ma10	ema	macd	GP
0,22	0,21	0,08	-0,22	0,19	0,15	0,17	0,23
6090	6632	88	8867	6081	4820	5122	5194

Tab. 1 to 8 shows the return of investment and standard deviation of portfolio value. It is clear, that strategies created by genetic programming perform well and are very competitive and outperforms the other strategies in 5 out of 8 cases. Strategies based on MACD, DOG and the market index perform very well too.

Interesting are the results during the financial crisis (tab. 4 and 5), where it highly outperforms the other strategies, even if trained on a rising market (tab. 4). In this case, it is outperformed only by bonds, which is understandable. Bonds give a stable low profit.

Conclusion

Investing strategies created by genetic programming proved to be competitive with other strategies. They outperformed the other strategies in most of the cases and give promising results also in the period of the financial crisis of 2008-2009. Such strategies could be used to secure the revenue during a market fall. We think, that such evaluation, by comparing with diverse investment strategies, should be done with all implementations in the area of financial investing.

Acknowledgment

This research has been supported by a VUB grant no. 2015-3-02/5

References

Bohdalová, M., &Šlahor, Ľ. (2007). Modeling of the risk factors in correlated markets using a multi-variate t-distributions. *Applied natural sciences* 2007, 162-172.

Briza, A. C., & Naval, P. C. (2008). Design of stock trading system for historical market data using multiobjective particle swarm optimization of technical indicators. *Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary* computation, 1871–1878. http://dx.doi.org/10.1145/1388969.1388992

Ghosh, A., &Dehuri, S. (2005). Evolutionary Algorithms for Multicriterion Optimization: A Survey. *Internation Journal of Computing & Information Sciences*, 2(1), 38-57.

Chen, S. H., &Navet, N. (2007). Failure of Genetic-Programming Induced Trading Strategies: Distinguishing between Efficient Markets and Inefficient Algorithms. *Computational Intelligence in Economics and Finance*, 2, 169-182.

Chen, S. S., Huang, CH. F., & Hong, T. P. (2014). An Improved Multi-Objective Genetic Model for Stock Selection with Domain Knowledge. *Technologies and Applications of Artificial Intelligence, Lecture Notes in Computer Science*, 8916, 66-73. http://dx.doi.org/10.1007/978-3-319-13987-6_7

Jakubéci, M. (2015a). Generating investment strategies using multiobjective genetic programming and internet term popularity data. *Scientific Annals of the "AlexandruIoan*Cuza" University of Iasi, 62(1). http://dx.doi.org/10.2478/v10316-012-0034-5

Jakubéci, M. (2015b). Evaluation of Investment Strategies Created by Multiobjective Genetic Programming. *Proceedings of the 7th International Scientific Conference Finance and Performance of Firms in Science, Education and Practice*, 498-509.

Kirkpatrick, CH., &Dahlquist, J. (2010). *Technical analysis*. New Jersey: FT Press.

Lohpetch, D., &Corne, D. (2011). Multiobjective algorithms for financial trading: Multiobjective out-trades single-objective. *IEEE Congress on Evolutionary Computation*, 192–199. http://dx.doi.org/10.1109/CEC.2011.5949618

Moat, H. S., Curme, CH., Avakian, A., Kenett, D. Y., Stanley, H. E., &Preis, T. 2013. Quantifying Wikipedia Usage Patterns Before Stock Market Moves. *Scientific Rep*, 3, 1801. http://dx.doi.org/10.1038/srep01801

Potvin, J. Y., Soriano, P., &Vallée, M. (2004). Generating trading rules on the stock markets with genetic programming. *Computers & Operations Research,* 31(7). http://dx.doi.org/10.1016/S0305-0548(03)00063-7

Preis, T., Moat, S. H., & Stanley, H. E. (2013). Quantifying Trading Behavior in Financial Markets Using Google Trends. *Sci Rep*, 3, 1684. http://dx.doi.org/10.1038/srep01684

Ruiz, J. E., Hristidis, V., Castillo, C., Gionis, A., &Jaimes, A. (2012). Correlating Financial Time Series with Micro-Blogging Activity. *Proceedings of the fifth ACM international conference on Web search and data mining*, 513-522. http://dx.doi.org/10.1145/2124295.2124358

Tapia, G. C., &Coello, C. A. (2007). Applications of Multi-Objective Evolutionary Algorithms in Economics and Finance: A Survey. *IEEE Congress on Evolutionary Computation*, 532-539.

Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Evolutionary Methods for Design, Optimization, and Control. *CIMNE*, 95–100.

Contact

Martin Jakubéci
Dept. of Information Systems, Faculty of Management, Comenius University in Bratislava
Faculty of Management of CU, Odbojarov 10, P.O.BOX 95, 820 05 Bratislava 25, Slovakia
Martin.jakubeci@fm.uniba.sk

Michal Greguš
Dept. of Information Systems, Faculty of Management, Comenius University in Bratislava
Faculty of Management of CU, Odbojarov 10, P.O.BOX 95, 820 05 Bratislava 25, Slovakia
Michal.gregus@fm.uniba.sk