Stock Portfolio Evaluation: An Application of Genetic-Programming-Based Technical Analysis

Liad Wagman
Computer Science Department
Stanford University
Stanford, California 94305
wagman@cs.stanford.edu

ABSTRACT

Recent studies in financial economics suggest that technical analysis may have merit to predictability of stock. When attempting to create an efficient portfolio of stocks, there are numerous factors to consider. The problem is that the evaluation involves many qualitative factors, which causes most approximations to go off track. This paper presents a genetic programming approach to portfolio evaluation. By using a set of fitness heuristics over a population of stock portfolios, the goal is to find a portfolio that has a high expected return over investment.

1. Introduction

Technical analysis tools are based on the belief that historical stock statistics exhibit regularities. According to the Efficient Market Hypothesis (EMH) (Fama 1970; Malkiel 1992), since historical statistics data is already reflected in the present price, technical analysis tools are irrelevant for predicting future price movements. This hypothesis has been challenged by several studies in recent years, which supply evidence of the predictability of investment risk from historical price patterns. The purpose of this paper is not to provide justification for technical analysis tools, but to show how genetic programming (GP) (Koza 1992), a decision-tree-based method to evolutionary computation, can be used to predict an efficient stock portfolio.

A portfolio in the context of this paper is a selection of stocks with different weights assigned to each stock. The sum of all the weights is 1, i.e., they are normalized. The return over investment (ROI) rate of a portfolio is measured over a period of time and is compared to the prevailing interest rate in the market. If the annual ROI rate of a portfolio is 4.5%, then for every $100 invested in the portfolio for a period of one year, one would generate $4.5 at the end of the year.

In this paper, we incorporate six technical indicators derived from rules in the finance literature into a fitness function (Alexander 1964; Brock et al. 1992; Fama & Blume 1966; Tsang 1999; Sweeney 1988). That fitness function is then applied to the following specific problem: finding a portfolio of stocks that is expected to have a high (greater than the prevailing interest rate in the market) ROI rate.

2. Background

Evolutionary computation is a standard term that encompasses a class of search, adaptation, and optimization techniques based on the principles of natural selection and evolution. Genetic programming (GP) is a promising variant of genetic algorithms that uses tree representations instead of strings. In evolutionary computation, a population of candidate solutions is maintained. A heuristic function is needed to evaluate the fitness of each candidate solution with regards to the task to be performed. Candidate solutions are selected for involvement in the next generation based on their fitness.

GP is attractive for financial applications because it manipulates decision trees as opposed to strings in genetic algorithms. This allows one to handle rule sets of variable size (Tsang 1999). Evolutionary computation has been applied to a broad range of problems with some success from traditional optimization in engineering and operational research (Bäck 1997) to non-traditional areas such as data mining, composition of music and financial prediction (Kinnear 1994; Angeline & Kinnear 1996; Koza 1996).

The specific GP structures used in this paper are described in following sections.

3. Preparatory Steps

3.1. Gathering Data

The data used for testing purposes in this paper is an artificial stock market representation over a period of 12 months, which is a conservative (low price fluctuations) image of the Dow Jones Industrial Average Index during 1979-1980. The data contains the daily price fluctuation records in percentages of 300 stocks during active trading days, and is organized into a 365 by 300 matrix (day vs. stock). A non-active day is marked as -1 (interpreted as -100%) and is ignored in computations. The estimated interest rate prevailing in this artificial market is 4.5%.

3.2. Representing Individuals

Each portfolio individual has between 1 and 10 stocks. The 10 stocks limit was chosen in order to reduce the complexity space and in order to differentiate portfolios by stocks as well as by weights. In theory, however, there is no limit to the amount of member stocks in a portfolio. The representation scheme is an array of length 10, where each cell is a record structure that contains 3 fields (a 10 by 3 matrix):

Table 1: Sample representation of an individual portfolio

Stock Number (implied by array index)	1	2	3	4	5	6	7	8	9	10
Stock Identification	Indexed from 1 to 300									
Normalized Percentage	Stock percentage of entire portfolio, from 0 to 1									
Value Added by Indicators	Normalized value from 0 to 1.									

3.3. Genetic Operators

General mechanisms of reproduction, crossover and mutation are used to combine or change selected candidate portfolios in order to generate offspring, which will form the population in the next generation. For this context, they are defined as follows:

<u>Reproduction:</u> a portfolio is chosen to be present in the next generation based on its fitness with respect to the average fitness of the population

<u>Crossover:</u> a random number between 1 and 9 is chosen to be the cutoff point where two portfolios are cut and combined. The traditional crossover takes place, where the right part of each cut portfolio is replaced with the one of the other, thus creating two new portfolios. Special normalization is then applied on each created portfolio in order to maintain the shares amount and normalized percentage constraints. The special normalization ensures that a stock does not appear twice in a portfolio by combining multiple appearances and placing 0 in the normalized percentage field of vacant cells.

<u>Mutation:</u> a random stock member out of the 10 is chosen and replaced with another random stock from the stock population. The amount of its shares is also randomly chosen, and is then normalized with the rest of the portfolio's stock percentages.

A new operation named Shuffle is also used:

<u>Shuffle:</u> select two random stocks from a portfolio and randomly modify their shares percentage, then normalize the weights of all member stocks.

The probability of crossover is 0.4 (40% of offspring portfolios are chosen for crossover), that of mutation is 0.2, and that of shuffle is 0.3.

3.4. Generating Initial Population

The initial population is composed of 1000 randomly generated portfolios. Each portfolio contains 10 randomly chosen stocks with randomly chosen percentages, which are then normalized. Since there are 300 stocks, there are (300 choose 10) = 300*299*298*…*291/10! = $1.4*10^{18}$ possible combination of 10 stocks (assuming that if less than 10 stocks are chosen, the percentage of the vacant slots is 0). Note that a portfolio represents a certain combination of stocks with specific stock weights, thus there is an infinite number of possible portfolios.

The amount of 1000 portfolios is chosen since among them, they are likely to contain all 300 stocks. Due to the high rate of the crossover, shuffle and mutation operators, the population is expected to evolve rapidly, whereas the weaker portfolios (with lower expected ROI) die out. Hence, even though the population is relatively small to the entire search space, the high evolution rate makes up for that ratio.

Table 2: Tableau of Portfolio Evaluation

Objective:	To obtain the portfolio with highest ROI evolved over G generations.
Terminal Set:	The portfolio 10 by 3 matrix.
Function Set:	Genetic operators: reproduction, crossover, mutation, and shuffle, as defined in (3.3).
Fitness Cases:	$1.4*10^{18}$ possible portfolios, with infinite possible individual, normalized stock weights.
Raw Fitness:	The normalized evaluated fitness of the portfolio based on the satisfaction of technical rules by its stocks over a 6 months period.
Standardized Fitness:	Same as raw fitness.
Hits:	The numbers of rules satisfied by the portfolio's member stocks over a period of 6 months.
Wrapper:	None.
Parameters:	M = 1000, G = 50, S = 300 (300 stocks)
Success Predicate:	A portfolio scores the maximum fitness (i.e., 1), or in other words, its stocks score the maximum number of hits.

4. Evaluation of Fitness

Six technical indicators derived from rules in the finance literature are incorporated into a fitness function (Alexander 1964; Brock et al. 1992; Fama & Blume 1966; Tsang 1999; Sweeney 1988).
There are four indicators for risk measurement and two indicators for return measurement as follows:

Risk:

(1.) Today's price – the average price of the previous 12 trading days
(2.) Today's price – the average price of the previous 50 trading days
(3.) Today's price – the maximum price of the previous 5 trading days
(4.) Today's price – the maximum price of the previous 50 trading days

Return:

(5.) Today's price – the minimum price of the previous 5 trading days
(6.) Today's price – the minimum price of the previous 63 trading days

We apply six technical rules using the above six indicators to generate "buy" or "not-buy" signals:
The *moving average rules* (1) and (2) generate "buy" signals if today's price is greater than the average price of the preceding *n* days (*n = 12* and *50* respectively). The *trading range breakout rules* (3) and (4) generate "buy" signals if today's price is greater than the maximum price of previous *n* days (*n* = 5 and 50 respectively). The *filter rules* (5) and (6) generate "buy" signals if today's price has risen *1%* than minimum of previous *n* days (*n* = 5 and 63 respectively). Here 1% is a threshold which an investor can choose (Tsang 1999).

The signals are then incorporated into a percentage system:
Rules 1-4 have weights 15% each, whereas rules 5-6 have weights 20% each. Again, this can be modified by an investor, but in this case we are assuming 60% risk hedging versus 40% return, which is a safe assumption (Luenberger, 1998). We would like to maximize the amount of rules satisfied by each member stock over a 6 months period in order to obtain a better portfolio.

Each portfolio contains a maximum of ten stocks. Each stock is evaluated monthly over a period of six months against the above rules. Each rule satisfied adds the appropriate weight (e.g. if for stock #6 only rule (3) is satisfied, and it is satisfied only in months 3, 4, 5, then the stock adds (normalized weight of the stock) * (0.15+ 0.15+ 0.15)/6 to the overall fitness of its portfolio).

Note the division by 6, which keeps the weights normalized, while each stock in a portfolio already has a field with its normalized percentage. Basically, the portfolio fitness is evaluated by the weighted average of the value added by each stock with respect to the stock's percentage of the entire portfolio. Because all values used are normalized, the weighted average resulting in the fitness value of the portfolio is between 0 and 1 as well.
Specific examples are provided in the following section.

5. Illustrative Examples

Below are two sample portfolios:

Table 3: Portfolio example

Stock Number	1	2	3	4	5	6	7	8	9	10
Stock Identification	12	35	28	36	45	108	270	225	123	3
Normalized Percentage	0.1	0.2	0.05	0.05	0.15	0.05	0.1	0.1	0.1	0.1
Value Added by Indicators	0.2	0.4	0.5	0.6	0.6	0.7	0.3	0.1	0.5	0.5
Portfolio Fitness	**0.420 (weighted average of value added with the percentages as weights)**									

Table 4: Portfolio example:

Stock Number	1	2	3	4	5	6	7	8	9	10
Stock Identification	54	34	100	196	57	30	22	205	233	105
Normalized Percentage	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.3	0.3	0.1
Value Added by Indicators	Not evaluated						0.6	0.56	0.3	0.7
Portfolio Fitness	**0.508**									

6. Complexity of Evaluation

We have a population of 1000 portfolios, and each portfolio contains 10 stocks. Each stock is evaluated monthly over a 6 months period and each evaluation requires 63 table scans. The matrix is small enough to be stored in main memory, hence table scans and computational variable access are treated equally (ignoring cache memory).
The amount of table scans is then as follows:

10 stocks per portfolio
6 monthly checks
6 technical rules to evaluate
<u>Each monthly check requires 63 table scans</u>

This results in 69 scans and variable uses per month
<u>Hence, 6*69 = 414 checks per stock</u>

4140 checks per portfolio
<u>Population of 1000 portfolios</u>

This results in a total of 4,140,000 checks per generation, which takes an average of 10 seconds on a Pentium IV processor, hence a total of 500 seconds for a run with 50 generations.

A clear optimization would be to preprocess the entire stock data with regards to the technical indicators so that the value added of stocks is handy. This will allow us to use a larger population of portfolios, since the complexity of evaluation would be much lower – no repeated evaluation of stocks. In future work, it will be done.

7. Results

Over 15 trial runs, the best-of-all-generations portfolios evolved by GP yielded an average annual interest rate of 7.5% (2.5% ROI over 4 months), which is 3% greater than the estimated prevailing interest rate of 4.5%. The ROI was computed by considering the daily price fluctuations of the member stocks of the candidate portfolio over the remaining 4 months of data.

The graph below is a compilation of several of the best-of-generation individuals of one trial run. The graph represents the best-of-generation individuals versus their ROI rate:

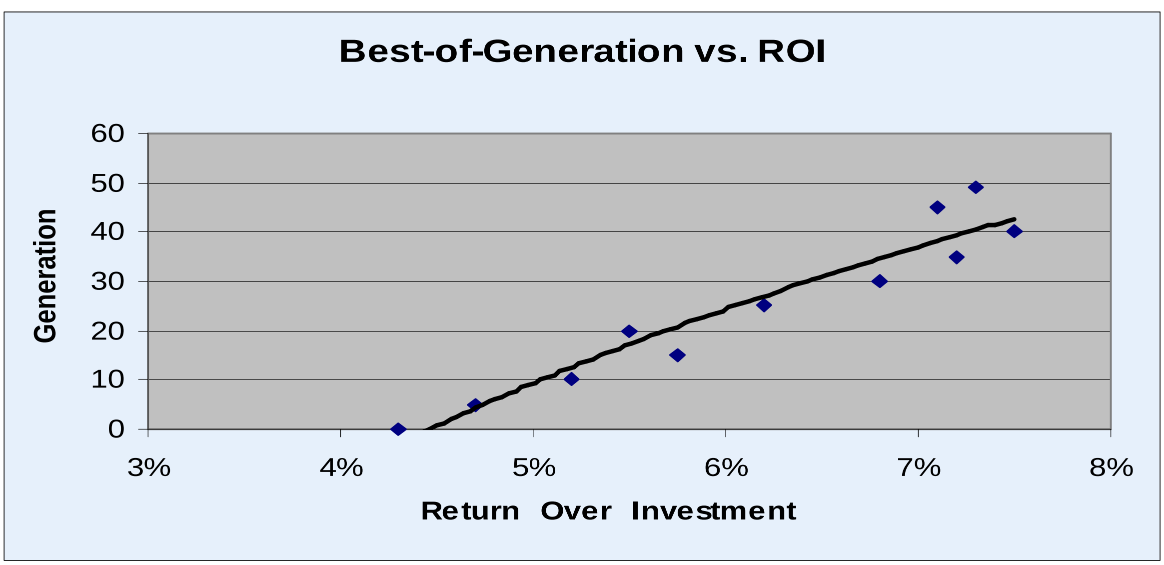

Figure 1: Best-of-Generation individuals ROI

As can be seen by the logarithmic trendline, the graph is converging towards the area of 7.5% with the ROI fluctuating within the [7, 7.5] range in later generations. It is important to note that the artificial stock population consisted of conservative data, i.e. small fluctuations in stock prices. Theoretically, the use of this data was realistic, since the rare occurrence of arbitrage disappears soon after its appearance (Luenberger, 1998), and newly offered stocks usually have higher investment risk, whereas this example was based on 60% risk hedging.

One of the best-of-all-generations portfolio was the following:

Table 5: Best-of-all-generations portfolio, trial run #5

Stock Number	1	2	3	4	5	6	7	8	9	10
Stock Identification	5	7	30	101	170	35	22	88	201	67
Normalized Percentage	0.25	0.15	0.05	0.05	0.15	0.05	0.0	0.07	0.13	0.0
Value Added by Indicators	0.85	0.94	0.73	0.85	0.94	0.73	-	0.94	0.5	-
Portfolio Fitness	**0.7408**									

Clearly, the fitness of the portfolio is not perfect. However, that can be expected from a conservative stock population. It can also be seen that the weights of the stocks are spread roughly evenly with an average weight of 0.09 (note that the percentages are rounded to the second decimal).

8. Future Work

EDDIE (Evolutionary Dynamic Data Investment Evaluator), developed by the University of Essex, U.K., is a forecasting system that helps investors to make the best use of the information available to them (Tsang et al. 1998). Such information may include technical rule indicators, individual company's performance indicators, expert predictions, etc. EDDIE allows a potential investor to make hypothesis about the factors that are relevant to a forecast. It then tests those hypotheses using historical data and evolves, by way of natural selection, decision trees which aim to provide a good return over investment (Tsang 1999).

The next step would be to use EDDIE to evolve portfolio evaluation rules using GP. This would make it possible to incorporate the entire evaluation process in the GP context, which would make solutions more credible, since the evaluation would be entirely based on the data provided. Also, the rules are bound to be more productive, since they will be evolved over the same population of stocks. We will apply evaluation preprocessing on the population of stocks with the rules evolved by EDDIE.

It is important to note that the application presented is not complete, since important issues such as transaction costs, risk of investment (based on company profile), and capital adequacy were ignored. In future work, these factors will be considered. Also, as mentioned before, the data used is a conservative image of the Dow Jones Industrial Average (DJIA) Index during 1979-1980. The use of conservative data was necessary because the technical indicators used for evaluating fitness were set ahead of time. In future work, since we will be using the EDDIE system to generate evaluation rules, it will be possible to consider the DJIA Index itself (Tsang 1999), and thus apply the results to the real market.

Another step would be to incorporate constraint satisfaction techniques, which have been demonstrated to be useful in genetic algorithms (Lau & Tsang 1997; Tsang 1993).

9. Related Work

Although still at its infancy, papers on the application of evolutionary computation in finance have been published. Bauer (1994) reported his genetic-algorithm-based intelligent systems which aim at finding tactical market timing strategies. Mahfoud and Mani (1996) presented a new genetic-algorithm-based system and applied it to the task of predicting the future performance of individual stocks. Neely *et al.* (1997) applied genetic programming to foreign exchange forecasting. Jin Li and Edward Tsang (1999) reported success with an application of EDDIE to technical rules generation used to evaluate stocks.

It seems that in the near future, one will be able to efficiently incorporate both qualitative and quantitative factors into evaluation rules which will be used to evaluate the performance of stock portfolios with relatively high success rate.

10. Conclusions

While technical analysis is widely used as an investment approach among practitioners, it is rarely accepted by academics. The purpose of this paper is not to defend technical analysis, although our results show that there is some predictability in a conservative image of the DJIA index based on historical data alone. The main objective of this paper is to illustrate that genetic programming can be used in portfolio evaluation to pursue the task of constructing a portfolio that provides a high ROI rate (with high probability).

Our experiment shows that in the context of our conservative market, the generated portfolios can achieve better ROI rates than the prevailing interest rate in that market. It is evident that although the reachable portfolio population is theoretically infinite, using a relatively small population (1000 portfolios) with high probabilities of the genetic operators crossover, mutation and shuffle, one could obtain effective results. In specific, the portfolios evolved by GP yielded an ROI rate 3% greater than the prevailing interest rate in the market.

Acknowledgments

The artificial stock data used for testing purposes was generously provided by Yossef Siegel of the University of Haifa, Israel. This paper is the result of a research project in CS426/BMI226 – Genetic Algorithms and Genetic Programming, under Professor John R. Koza at Stanford University. I thank Professor Koza for his teaching and guidance in the field of genetic programming.

References

Alexander, S.S., *Price movement in speculative markets: Trend or random walks, No. 2*, in Cootner, P.(ed.), the random character of stock market prices, MIT Press, Cambridge, MA, 1964, 338-372.

Angeline, P. & Kinnear, K.E., (ed.), *Advances in genetic programming II*, MIT Press. 1996.

Bäck, T., *Evolutionary algorithms in theory and practice*, New York: Oxford University Press, 1996.

Bäck, T., (ed.), *Proceedings of the seventh international conference on genetic algorithms*, San Francisco, California: Morgan Kaufmann Publishers, Inc., 1997.

Bauer, R. J. Jr. , *Genetic Algorithms and Investment Strategies.* New York, John Wiley & Sons, Inc., 1994.

Brock, W., Lakonishok, J. & LeBaron, B., Simple technical trading rules and the stochastic properties of stock returns, *Journal of Finance*, 47, (1992), 1731-1764.

Fama, E.F. & Blume, M.E., filter rules and stock-market trading, *Journal of Business* 39(1), (1966), 226-241.

Fama, E.F., Efficient capital markets: A review of theory and empirical work, *Journal of Finance* 23, (1970), 383-417.

Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

Goonatilake, S. & Treleaven, P. (ed.), *Intelligent systems for finance and business*, Wiley, New York, 1995.

Holland, J.H., *Adaptation in natural and artificial system*, University of Michigan Press, 1975.

Jegadeesh, N. & Titman, S., Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency, *Journal of Finance*, 48, No.1, (1993), 65-91.

Kinnear, K.E. (ed.), *Advances in genetic programming*, MIT Press, 1994.

Koza, J.R., *Genetic Programming: on the programming of computers by means of natural selection.* MIT Press, 1992.

Koza, J., Goldberg, D., Fogel, D. & Riolo, R. (ed.), *Procedings of the First Annual Confrence on Genetic programming*, MIT Press, 1996.

Lau, T.L. & Tsang, E.P.K., Solving the processor configuration problem with a mutation-based genetic algorithm, *International Journal on Artificial Intelligence Tools* (IJAIT), World Scientific, Vol.6, No.4, (1997), 567-585.

Luenberger, D.G., *Investment Science*, Oxford University Press, 1998.

Mahfoud, S. & Mani, G., Financial Forecasting Using Genetic Algorithms, *Journal of Applied Artificial Intelligence* Vol.10, Num 6, (1996), 543-565.

Malkiel, B., Efficient market Hypothesis, in Newman,P., Milgate, M. and Eatwell, J.(eds.), *New Palgrave Dictionary of Money and Finance*, Macmillan, London, (1992), pp739.

Neely, C., Weller, P. & Ditmar, R., Is technical analysis in the foreign exchange market profitable? A genetic programming approach, in Dunis, C. & Rustem, B.(ed.), *Proceedings, Forecasting Financial Markets: Advances for Exchange Rates, Interest Rates and Asset Management*, London. 1997.
Neftci, S.N., Naïve trading rules in financial markets and Wiener-Kolmogorov prediction theory: A study of 'technical analysis' , *Journal of Business*, 64, (1991), 549-571.
Sweeney, R. J., Some new filter rule test: Methods and results, *Journal of Financial and Quantitative Analysis*, 23, (1988), 285-300.
Tsang, E.P.K., Li, J. & Butler, J.M., EDDIE beats the bookies, *International Journal of Software, Practice & Experience*, Wiley, Vol.28 (10), 1033-1043, August 1998.
Tsang, E.P.K. Li, J., *Improving Technical Analysis Predictions: An Application of Genetic Programming*: *Proceedings of the Florida Artificial Intelligence Research Symposium*, USA, 1999
Tsang, E.P.K., *Foundations of constraint satisfaction*, Academic Press, London, 1993.